MU887000A
TRX Test Module
Operation Manual

18th Edition

- **For safety and warning information, please read this manual before attempting to use the equipment.**
- **Additional safety and warning information is provided in the MT8870A Universal Wireless Test Set Operation Manual. Please refer to this document before using the equipment.**
- **Keep this manual with the equipment.**

ANRITSU CORPORATION

Document No.: M-W3606AE-18.0

Safety Symbols

To prevent the risk of personal injury or loss related to equipment malfunction, Anritsu Corporation uses the following safety symbols to indicate safety-related information. Ensure that you clearly understand the meanings of the symbols BEFORE using the equipment. Some or all of the following symbols may be used on all Anritsu equipment. In addition, there may be other labels attached to products that are not shown in the diagrams in this manual.

Symbols used in manual

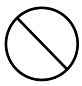
⚠ DANGER This indicates a very dangerous procedure that could result in serious injury or death if not performed properly.

⚠ WARNING This indicates a hazardous procedure that could result in serious injury or death if not performed properly.

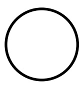
⚠ CAUTION This indicates a hazardous procedure or danger that could result in light-to-severe injury, or loss related to equipment malfunction, if proper precautions are not taken.

Safety Symbols Used on Equipment and in Manual

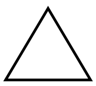
The following safety symbols are used inside or on the equipment near operation locations to provide information about safety items and operation precautions. Ensure that you clearly understand the meanings of the symbols and take the necessary precautions BEFORE using the equipment.



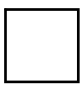
This indicates a prohibited operation. The prohibited operation is indicated symbolically in or near the barred circle.



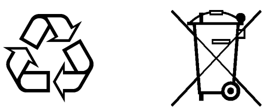
This indicates an obligatory safety precaution. The obligatory operation is indicated symbolically in or near the circle.



This indicates a warning or caution. The contents are indicated symbolically in or near the triangle.



This indicates a note. The contents are described in the box.



These indicate that the marked part should be recycled.

MU887000A
TRX Test Module
Operation Manual

20 August 2012 (First Edition)
19 January 2018 (18th Edition)

For Safety

⚠ WARNING

- ALWAYS refer to the operation manual when working near locations at which the alert mark shown on the left is attached. If the advice in the operation manual is not followed, there is a risk of personal injury or reduced equipment performance. The alert mark shown on the left may also be used with other marks and descriptions to indicate other dangers.

Calibration

- The performance-guarantee seal verifies the integrity of the equipment. To ensure the continued integrity of the equipment, only Anritsu service personnel, or service personnel of an Anritsu sales representative, should break this seal to repair or calibrate the equipment. Be careful not to break the seal by opening the equipment or unit covers. If the performance-guarantee seal is broken by you or a third party, the performance of the equipment cannot be guaranteed.

Use in a Residential Environment

- This equipment is designed for an industrial environment.
  In a residential environment, this equipment may cause radio interference in which case the user may be required to take adequate measures.

Equipment Certificate

Anritsu Corporation certifies that this equipment was tested before shipment using calibrated measuring instruments with direct traceability to public testing organizations recognized by national research laboratories, including the National Institute of Advanced Industrial Science and Technology, and the National Institute of Information and Communications Technology, and was found to meet the published specifications.

Anritsu Warranty

Anritsu Corporation will repair this equipment free-of-charge if a malfunction occurs within one year after shipment due to a manufacturing fault. However, software fixes will be made in accordance with the separate Software End-User License Agreement. Moreover, Anritsu Corporation will deem this warranty void when:

- The fault is outside the scope of the warranty conditions separately described in the operation manual.
- The fault is due to mishandling, misuse, or unauthorized modification or repair of the equipment by the customer.
- The fault is due to severe usage clearly exceeding normal usage.
- The fault is due to improper or insufficient maintenance by the customer.
- The fault is due to natural disaster, including fire, wind, flooding, earthquake, lightning strike, or volcanic ash, etc.
- The fault is due to damage caused by acts of destruction, including civil disturbance, riot, or war, etc.
- The fault is due to explosion, accident, or breakdown of any other machinery, facility, or plant, etc.
- The fault is due to use of non-specified peripheral or applied equipment or parts, or consumables, etc.
- The fault is due to use of a non-specified power supply or in a non-specified installation location.
- The fault is due to use in unusual environments(Note).
- The fault is due to activities or ingress of living organisms, such as insects, spiders, fungus, pollen, or seeds.

In addition, this warranty is valid only for the original equipment purchaser. It is not transferable if the equipment is resold.

Anritsu Corporation shall assume no liability for injury or financial loss of the customer due to the use of or a failure to be able to use this equipment.

Note:
For the purpose of this Warranty, "unusual environment" means use:

- In places of direct sunlight
- In dusty places
- Outdoors
- In liquids, such as water, oil, or organic solvents, and medical fluids, or places where these liquids may adhere
- In salty air or in place chemically active gases (sulfur dioxide, hydrogen sulfide, chlorine, ammonia, nitrogen dioxide, or hydrogen chloride etc.) are present
- In places where high-intensity static electric charges or electromagnetic fields are present
- In places where abnormal power voltages (high or low) or instantaneous power failures occur
- In places where condensation occurs
- In the presence of lubricating oil mists
- In places at an altitude of more than 2,000 m
- In the presence of frequent vibration or mechanical shock, such as in cars, ships, or airplanes

Anritsu Corporation Contact

In the event of this equipment malfunctions, contact an Anritsu Service and Sales office. Contact information can be found on the last page of the printed version of this manual, and is available in a separate file on the PDF version.

Notes On Export Management

This product and its manuals may require an Export License/Approval by the Government of the product's country of origin for re-export from your country.

Before re-exporting the product or manuals, please contact us to confirm whether they are export-controlled items or not.

When you dispose of export-controlled items, the products/manuals need to be broken/shredded so as not to be unlawfully used for military purpose.

Trademark and Registered Trademark

CDMA2000® is a registered trademark of the Telecommunications Industry Association (TIA-USA) in the United States and other countries.

Reuse Parts

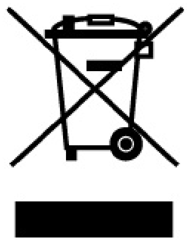
Anritsu group promotes recycling activities in order to reuse available resources and save energy. This product may use recycled parts (mechanical components) that conform to Anritsu’s quality standards.

Lifetime of Parts

The life span of certain parts used in this instrument is determined by the operating time or the power-on time. Due consideration should be given to the life spans of these parts when performing continuous operation over an extended period. These parts must be replaced at the customer's expense even if within the guaranteed period described in Warranty at the beginning of this manual. For details on life span, refer to the corresponding section in this manual.

Crossed-out Wheeled Bin Symbol

Equipment marked with the Crossed-out Wheeled Bin Symbol complies with council directive 2012/19/EU (the “WEEE Directive”) in European Union.



For Products placed on the EU market after August 13, 2005, please contact your local Anritsu representative at the end of the product's useful life to arrange disposal in accordance with your initial contract and the local law.

Software End-User License Agreement (EULA)

Please read this Software End-User License Agreement (hereafter this EULA) carefully before using (includes executing, copying, registering, etc.) this software (includes programs, databases, scenarios, etc., used to operate, set, etc., Anritsu electronic equipment). By reading this EULA and using this software, you are agreeing to be bound by the terms of its contents and Anritsu Corporation (hereafter Anritsu) hereby grants you the right to use this Software with the Anritsu-specified equipment (hereafter Equipment) for the purposes set out in this EULA.

1. Grant of License and Limitations

1. Regardless of whether this Software was purchased from or provided free-of-charge by Anritsu, you agree not to rent, lease, lend, or otherwise distribute this Software to third parties and further agree not to disassemble, recompile, reverse engineer, modify, or create derivative works of this Software.
2. You may make one copy of this Software for backup purposes only.
3. You are not permitted to reverse engineer this software.
4. This EULA allows you to install one copy of this Software on one piece of Equipment.

2. Disclaimers

To the extent not prohibited by law, in no event shall Anritsu be liable for personal injury, or any incidental, special, indirect or consequential damages whatsoever, including, without limitation, damages for loss of profits, loss of data, business interruption or any other commercial damages or losses, arising out of or related to your use or inability to use this Software.

3. Limitation of Liability

a. If a fault (bug) is discovered in this Software, preventing operation as described in the operation manual or specifications whether or not the customer uses this software as described in the manual, Anritsu shall at its own discretion, fix the bug, or exchange the software, or suggest a workaround, free-of-charge. However, notwithstanding the above, the following items shall be excluded from repair and warranty.
   - i) If this Software is deemed to be used for purposes not described in the operation manual or specifications.
   - ii) If this Software is used in conjunction with other non-Anritsu-approved software.
   - iii) Recovery of lost or damaged data.
   - iv) If this Software or the Equipment has been modified, repaired, or otherwise altered without Anritsu's prior approval.
   - v) For any other reasons out of Anritsu's direct control and responsibility, such as but not limited to, natural disasters, software virus infections, etc.

b. Expenses incurred for transport, hotel, daily allowance, etc., for on-site repairs by Anritsu engineers necessitated by the above faults shall be borne by you.

c. The warranty period for faults listed in article 3a above covered by this EULA shall be either 6 months from the date of purchase of this Software or 30 days after the date of repair, whichever is longer.

4. **Export Restrictions**

You may not use or otherwise export or re-export directly or indirectly this Software except as authorized by Japanese and United States law. In particular, this software may not be exported or re-exported (a) into any Japanese or US embargoed countries or (b) to anyone on the Japanese or US Treasury Department's list of Specially Designated Nationals or the US Department of Commerce Denied Persons List or Entity List. By using this Software, you warrant that you are not located in any such country or on any such list. You also agree that you will not use this Software for any purposes prohibited by Japanese and US law, including, without limitation, the development, design and manufacture or production of missiles or nuclear, chemical or biological weapons of mass destruction.

5. **Termination**

Anritsu shall deem this EULA terminated if you violate any conditions described herein. This EULA shall also be terminated if the conditions herein cannot be continued for any good reason, such as violation of copyrights, patents, or other laws and ordinances.

6. **Reparations**

If Anritsu suffers any loss, financial or otherwise, due to your violation of the terms of this EULA, Anritsu shall have the right to seek proportional damages from you.

7. **Responsibility after Termination**

Upon termination of this EULA in accordance with item 5, you shall cease all use of this Software immediately and shall as directed by Anritsu either destroy or return this Software and any backup copies, full or partial, to Anritsu.

8. **Dispute Resolution**

If matters of dispute or items not covered by this EULA arise, they shall be resolved by negotiations in good faith between you and Anritsu.

9. **Court of Jurisdiction**

This EULA shall be interpreted in accordance with Japanese law and any disputes that cannot be resolved by negotiation described in Article 8 shall be settled by the Japanese courts.

Notice

The following actions are strictly prohibited for all of the software installed in this product or otherwise provided by Anritsu:

1. Copying, except for archival purposes.
2. Transferring to a third party separately from this product.
3. Analyzing the incorporated software including but not limited to modifying, decompiling, disassembling, and reverse engineering.

Cautions against computer virus infection

- Copying files and data

  Only files that have been provided directly from Anritsu or generated using Anritsu equipment should be copied to the instrument.

  All other required files should be transferred by means of USB or CompactFlash media after undergoing a thorough virus check.
- Adding software

  Do not download or install software that has not been specifically recommended or licensed by Anritsu.
- Network connections

  Ensure that the network has sufficient anti-virus security protection in place.

Protection Against Computer Virus Infections

Prior to the software installation

Before installing this software or any other software recommended or approved by Anritsu, run a virus scan on your computer, including removable media (e.g. USB memory stick and CF memory card) you want to connect to your computer.

When using this software and connecting with the measuring instrument

- Copying files and data

  On your computer, do not save any copies other than the following:

  ・Files and data provided by Anritsu

  ・Files created by this software

  ・Files specified in this document

  Before copying these files and/or data, run a virus scan, including removable media (e.g. USB memory stick and CF memory card).
- Connecting to network

  Connect your computer to the network that provides adequate protection against computer viruses.

Cautions on Proper Operation of Software

This software may not operate normally if any of the following operations are performed on your computer:

- Simultaneously running any software other than that recommended or approved by Anritsu
- Closing the lid (Laptop computer)
- Turning on the screen saver function
- Turning on the battery-power saving function (Laptop computer)

For how to turn off the functions, refer to the operation manual that came with your computer.

CE Conformity Marking

Anritsu affixes the CE conformity marking on the following product(s) in accordance with the Decision 768/2008/EC to indicate that they conform to the EMC, LVD and RoHS directive of the European Union (EU).

CE marking

1. Product Model

Plug-in Units: MU887000A TRX Test Module

2. Applied Directive and Standards

When the MU887000A TRX Test Module is installed in the MT8870A, the applied directive and standards of this unit conform to those of the MT8870A main frame.

PS: About main frame

Please contact Anritsu for the latest information on the main frame types that MU887000A can be used with.

RCM Conformity Marking

Anritsu affixes the RCM mark on the following product(s) in accordance with the regulation to indicate that they conform to the EMC framework of Australia/New Zealand.

RCM marking

1. Product Model

Plug-in Units: MU887000A TRX Test Module

2. Applied Directive and Standards

When the MU887000A TRX Test Module is installed in the MT8870A, the applied directive and standards of this unit conform to those of the MT8870A main frame.

PS: About main frame

Please contact Anritsu for the latest information on the main frame types that MU887000A can be used with.

About This Manual

This manual mainly describes the use, panels, and specifications of the MU887000A TRX Test module.

Products related to the MT8870A Universal Wireless Test Set include:

- MT8870A Universal Wireless Test Set (main unit)
- Modules installed in the MT8870A
- Application software installed in the modules
- Control software installed in a PC controller

These products are referred to as the “Universal Wireless Test Set Series”. The operation manuals of the Universal Wireless Test Set Series consist of separate documents for the main unit, module(s), application software, and control software, as shown below.

this manual

Main Unit Operation Manual

Describes precautions for safe use, how to connect power and interfaces, and operation and maintenance of the unit.

Module Operation Manual

Describes specifications and operation of the modules.
Each module has its own operation manual.

MU887000A TRX Testing Module Operation Manual

Application Software Operation Manual

Describes how to use the application software installed in the modules and remote commands.
Each application software has its own operation manual.

Control Software Operation Manual

Describes how to install and operate the software for controlling the MT8870A system via a PC.
Each control software has its own operation manual.

For safety precautions and maintenance, refer to *the MT8870A Universal Wireless Test Set Operation Manual*.

For the performance test, refer to the operation manual for each application software.

Table of Contents

Chapter 1 Outline

This chapter reviews the MU887000A TRX Test Module functions and product configurations. For performance specifications and functions, refer to Appendix A "Specifications".

1.1 Overview

The MU887000A TRX Test Module (module hereafter) is installed in the MT8870A Universal Wireless Testing System (MT8870A hereafter).

The module sends and receives signals in the frequency range of 10 MHz to 3.8 GHz. This range can be extended up to 6 GHz by installing an option.

The module functions are:

- Signal generation
  Generates high-frequency signals at the programmed frequency and amplitude.
  Supports signal modulation based on IQ modulation data in file to allow application of additive white Gaussian noise (AWGN).
- Sequencing
  Generates signals at signal frequency, level, modulation signal, timing, and other conditions specified in list.
  Up to 16 condition lists can be saved.
  Various aspects of list processing (number of cycles, next list, trigger conditions, etc.,) can be edited.
- System setting
  The system settings manage date and time, network address settings, license key, and module software version.
  These settings can be configured from a computer using the included MX887900A MT8870A Utility Tool.
- Status register
  The module can detect events and errors using a status register as specified by IEEE 488.2 or the Standard Commands for Programmable Instruments (SCPI).

Users can install optional measurement software to send or receive signals based on the following communication specifications:

- W-CDMA/HSPA
- LTE FDD, LTE TDD
- 1xEV-DO
- IEEE 802.11a/b/g/n/ac/p/ax
- IEEE 802.15.4
- NB-IoT
- GSM/EDGE
- CDMA2000
- TD-SCDMA
- Bluetooth
- Z-Wave
- Category M

The module is controlled by commands sent via Ethernet or GPIB.

It supports the following command formats:

- SCPI Commands in SCPI format
- Native Commands compatible with existing Anritsu devices

1.2 Features

The module features:

(1) High-speed measurement

(2) Software switching to support broad range of wireless specifications

Switching the installed software supports testing of W-CDMA, LTE, CDMA2000, etc., mobiles, as well as testing of data communications over wireless LAN.

1.3 Product Configuration

This section introduces the standard module configuration, options for extending the module functions, application software, application parts, and warranty services.

1.3.1 Standard configuration

The following table lists standard module configuration.

Table 1.3.1-1 Standard Configuration

Items	Model/Code	Name	Qty
Main Unit	MU887000A	TRX Test Module	1
Standard Accessories	W3606AE	MU887000A TRX Test Module Operation Manual	1

The operation manual is included on the supplied storage media (DVD, etc.).
The electronic files are stored in one or more storage media (DVD, etc.).

1.3.2 Options

The following table lists the options for extending the module functions. All options are sold separately. To order, specify the model/code, name and quantity.

Table 1.3.2-1 MU887000A Options

Model/Code	Name	Remarks
MU887000A-001	6 GHz Frequency Extension	
MU887000A-101	6 GHz Frequency Extension Retrofit	
MU887000A-002	Audio Measurement Hardware	
MU887000A-102	Audio Measurement Hardware Retrofit	

1.3.3 Application software

Application software and waveform files for the MU887000A are sold separately. For details about the functions and performance of the application software, contact the Anritsu Service and Sales offices or your sales representative.
The following table lists the application software and waveform files.

Table 1.3.3-1 Application Software

Model/Code	Name
MX887010A	Cellular Standards Sequence Measurement
MX887011A	W-CDMA/HSPA Uplink TX Measurement
MX887012A	GSM/EDGE Uplink TX Measurement
MX887013A	LTE FDD Uplink TX Measurement
MX887014A	LTE TDD Uplink TX Measurement
MX887015A	CDMA2000 Reverse Link TX Measurement
MX887016A	1xEV-DO Reverse Link TX Measurement
MX887017A	TD-SCDMA Uplink TX Measurement
MX887030A	WLAN 802.11b/g/a/n TX Measurement
MX887031A	WLAN 802.11ac TX Measurement
MX887032A	WLAN 802.11p TX Measurement
MX887033A	WLAN 802.11ax TX Measurement
MX887040A	Bluetooth TX Measurement
MX887050A	Short Range Wireless Average Power and Frequency Measurement
MX887060A	IEEE 802.15.4 TX Measurement
MX887061A	Z-Wave TX Measurement
MX887065A	Category M FDD Uplink TX Measurement
MX887067A	NB-IoT Uplink TX Measurement
MX887070A	FM/Audio TRX Measurement
MX887090A	Multi-DUT Measurement Scheduler

Table 1.3.3-2 Waveforms

Model/Code	Name
MV887011A	W-CDMA/HSPA Downlink Waveforms
MV887012A	GSM/EDGE Downlink Waveforms
MV887013A	LTE FDD Downlink Waveforms
MV887014A	LTE TDD Downlink Waveforms
MV887015A	CDMA2000 Forward Link Waveforms
MV887016A	1xEV-DO Forward Link Waveforms
MV887017A	TD-SCDMA Downlink Waveforms
MV887030A	WLAN 802.11b/g/a/n Waveforms
MV887031A	WLAN 802.11ac Waveforms
MV887032A	WLAN 802.11p Waveforms
MV887033A	WLAN 802.11ax Waveforms
MV887040A	Bluetooth Waveforms
MV887060A	IEEE 802.15.4 Waveforms
MV887061A	Z-Wave Waveforms
MV887065A	Category M FDD Downlink Waveforms
MV887067A	NB-IoT Downlink Waveforms
MV887070A	FM RDS Waveforms
MV887100A	GPS Waveforms
MV887102A	GLONASS Waveforms
MV887103A	BeiDou Waveforms
MV887110A	DVB-H Waveforms
MV887111A	ISDB-T Waveforms
MV887112A	ISDB-Tmm Waveforms

Table 1.3.3-3 Application Software Option

Model/Code	Name
MX887013A-001	LTE-Advanced FDD Uplink CA TX Measurement
MX887014A-001	LTE-Advanced TDD Uplink CA TX Measurement
MX887040A-001	DLE TX Measurement
MX887040A-002	2LE TX Measurement
MX887040A-003	BLR TX Measurement

Table 1.3.3-4 Waveforms Option

Model/Code	Name
MV887040A-001	DLE Waveforms
MV887040A-002	2LE Waveforms
MV887040A-003	BLR Waveforms

1.3.4 Application parts

Use application parts (accessories) as required. All application parts are ordered separately. To order, specify the model/code, name and quantity.

Table 1.3.4-1 Application Parts

Model/Code	Name	Remarks
W3606AE	MU887000A Operation Manual	English, storage media (DVD, etc.)
J0576B	Coaxial Cord, 1.0 m	N-P•5D-2W•N-P
J0576D	Coaxial Cord, 2.0 m	N-P•5D-2W•N-P
J0127A	Coaxial Cord, 1.0 m	BNC-P•RG58A/U•BNC-P
J0127C	Coaxial Cord, 0.5 m	BNC-P•RG58A/U•BNC-P
MN8116A	Multi-Port Switch	Number of TRX ports: 16 (32 when option 001 is installed)

1.3.5 Warranty service

Table 1.3.5-1 MU887000A Warranty Service

Model/Code	Name	Remarks
MU887000A-ES210	2 Years Extended Warranty Service	
MU887000A-ES310	3 Years Extended Warranty Service	
MU887000A-ES510	5 Years Extended Warranty Service	

1.4 Abbreviations

The abbreviations used in this manual and printed on the module panel are listed in Table 1.4-1.

Table 1.4-1 Abbreviations

Abbreviation	Name
ARB	Arbitrary Waveform Generator
AWGN	Additive White Gaussian Noise
C/N	Carrier to Noise Ratio
CW	Continuous Wave
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
GPIB	General Purpose Interface Bus
GPRF	General Purpose Radio Frequency
I^2S	Inter IC Sound
IP	Internet Protocol
L	Left
MAC	Media Access Control
Max.	Maximum
R	Right
rms	Root Mean Square
SA	Signal Analyzer
SG	Signal Generator
TRX	Transceiver
VISA	Virtual Interface Software Architecture
VSG	Vector Signal Generator
WLAN	Wireless Local Area Network

Chapter 2 Before Use

This chapter lists the names and functions of each part, and explains how to install the module and connect external devices.

2.1 Part Names and Functions

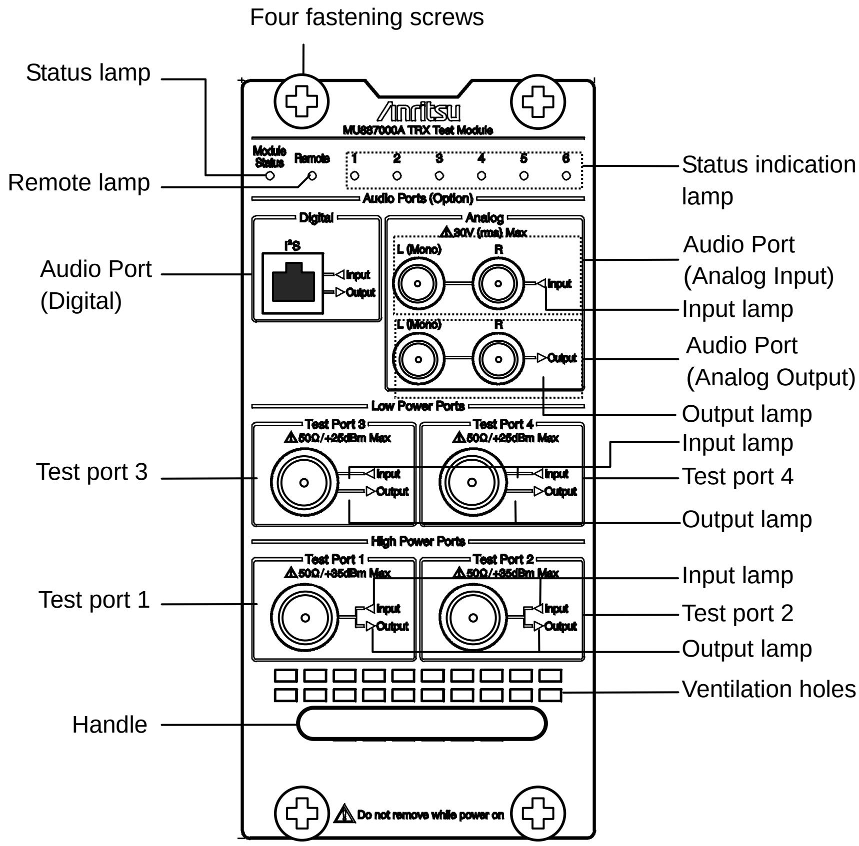


Figure 2.1-1 Front Panel Part Names

Table 2.1-1 Functions of Front-Panel Parts

Name	Function
Fastening screws	Secures module in MT8870A
Status lamp (Module status)*	Indicates the status of module. Refer to Appendix D "Status indication of lamps" for details.
Remote lamp (Remote)*	On when MT8870A remote-controlled Refer to Appendix D "Status indication of lamps" for details.
Status indication lamp (1 to 6)*	Indicates the status of module or application software. Refer to Appendix D "Status indication of lamps" for details.
Input lamp*	On at signal input via connector All input lamps are on during level calibration execution.
Output lamp*	On at signal output via connector All output lamps are on during level calibration execution.
Audio Port (Digital)	Connector for audio signals Available when the option 002/102 is added.
Audio Port (Analog Input)	Input connector for audio signals Use L for monaural. Available when the option 002/102 is added.
Audio Port (Analog Output)	Output connector for audio signals Use L for monaural. Available when the option 002/102 is added.
Test ports 1 and 2	Connectors for high-frequency signals Both simultaneous signal input and output supported
Test ports 3 and 4	Connectors for high-frequency signals Either simultaneous signal input or output supported
Ventilations	For cooling
Handle	Grip for installing or removing modules from MT8870A

*: When the power is supplied, all the lamps are lit orange.

⚠ CAUTION

Always tighten the fastening screws before using the MT8870A. The MT8870A might be dropped if used with loose fastening screw, resulting in injury.

2.2 Installing Module

CAUTION

Disconnect the power to the MT8870A before installing a module.

2.2.1 Installing module

A module can be installed to every position of a slot.

1. Loosen the four screws on the blank panel using a Phillips screwdriver #1.
2. Pull the screws to remove the blank panel.
3. Insert the module firmly into the slot.
4. Tighten the four screws on the blank panel using a Phillips screwdriver.

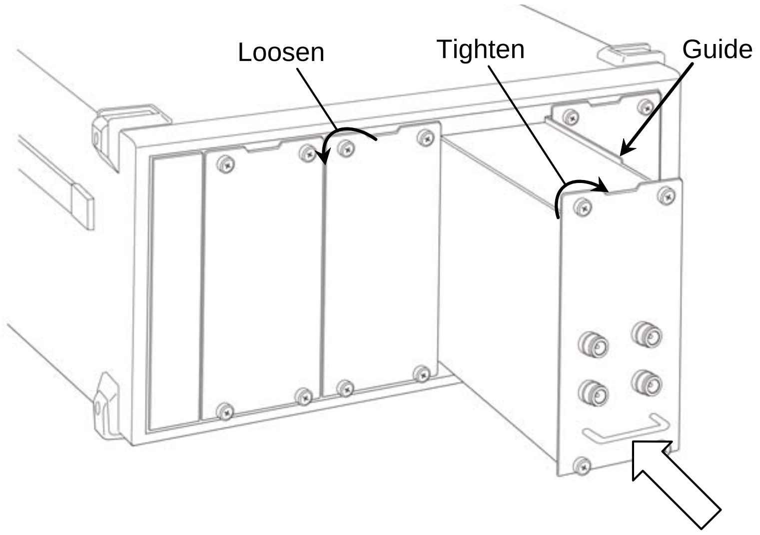


Figure 2.2.1-1 Installing Module

CAUTION

- When installing or removing a module, take anti-static precautions, such as wearing an electrostatic discharge wristband.
- When installing or removing a module, disconnect the coaxial cables from the module. Otherwise, a signal may be applied to the module accidentally, possibly damaging it.

⚠ CAUTION

When installing a module, do not block the panel vent holes. If the ventilation is blocked by a label, etc., the module will not be cooled and the performance may be degraded or unstable.

2.2.2 Removing module

1. Loosen the four screws on the panel using a Phillips screwdriver.
2. Grip the handle to pull the module out.
3. Replace the blank panel in the slot.
4. Tighten the four screws on the panel using a Phillips screwdriver.

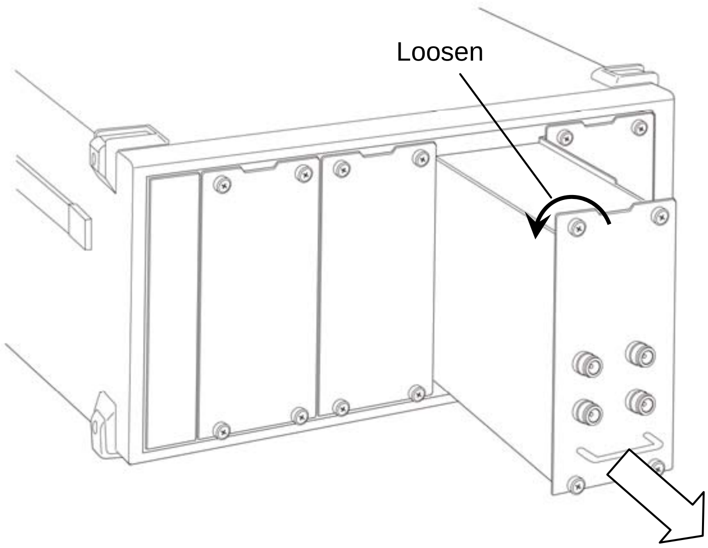


Figure 2.2.2-1 Removing Module

⚠ CAUTION

Unused slots must be covered by blank panels. Otherwise, the inside of the MT8870A may not be sufficiently cooled, resulting in degraded performance or unstable operation.

When removing a module, hold on both sides of the handle so as not to apply excessive force to one part or risk dropping the module.

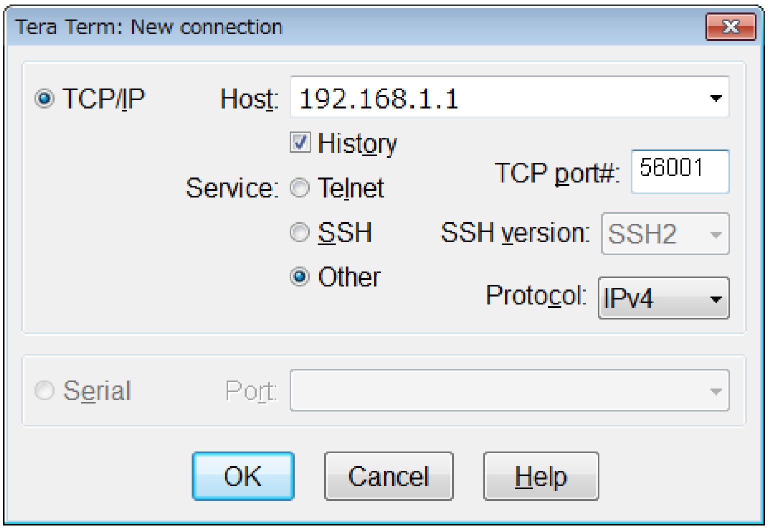
2.3 Confirming Connection

Confirm that the external PC controller recognizes the MT8870A via Ethernet.
This description assumes that the module is installed in slot 1 of the MT8870A and the interface settings are as follows:
IPv4 address: 192.168.1.1
Port number: 56001

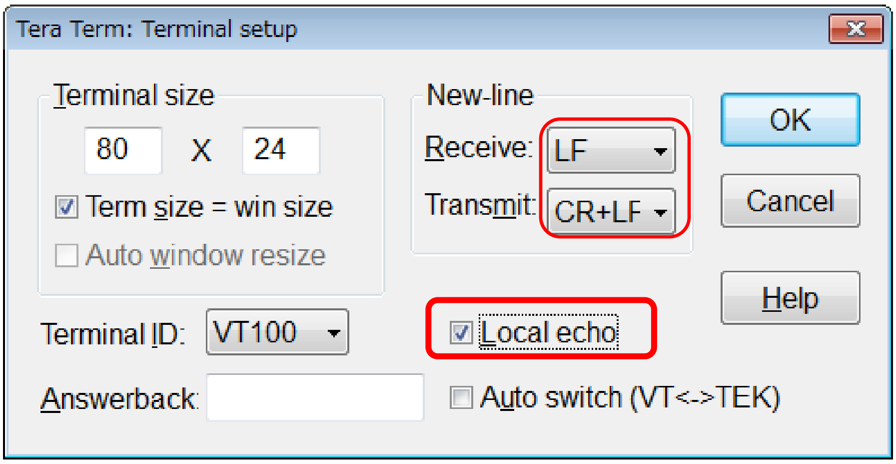
2.3.1 Ethernet

This section describes how to use the Tera Term freeware.

1. On starting Tera Term, the New connection dialog is displayed. Input the IP address of the MT8870A at Host and the port number at TCP port#.
   Specify Other for Service and **IPv4** for Protocol.
   Click the **OK** button.



2. The Communication window is displayed when Tera Term recognizes the MT8870A.
3. Click Setting(S) – Terminal (T)… at the menu of the Tera Term.
4. Set Receive to **LF**, and Transmit to **CR+LF**.
   Put a checkmark in the Local echo checkbox.
5. Click the **OK** button



6. Send the *IDN? remote control command.
   Confirm that the response from the MT8870A is displayed on window of the Tera Term.

2.3.2 GPIB

1. Install the software supplied with the GPIB interface to be connected to a control PC.
2. Run the software.
3. Confirm that the device with the GPIB address specified at the MT8870A is displayed.

2.4 Electrostatic Countermeasures

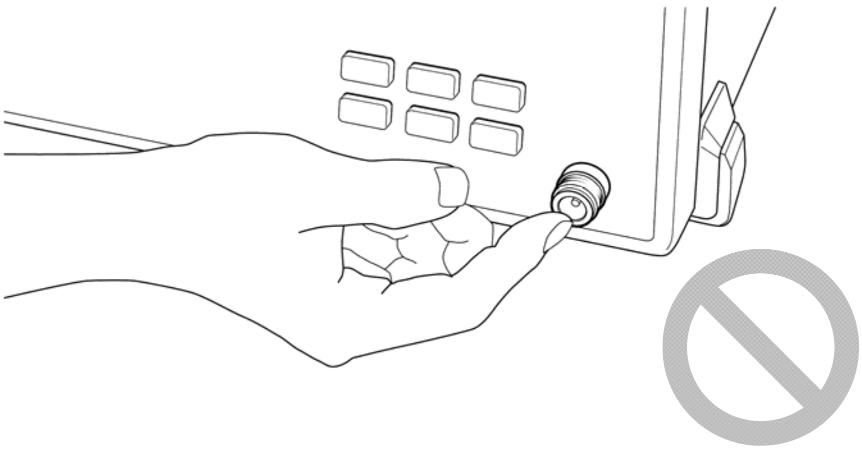
This section describes the general precautions against electrostatic discharges.

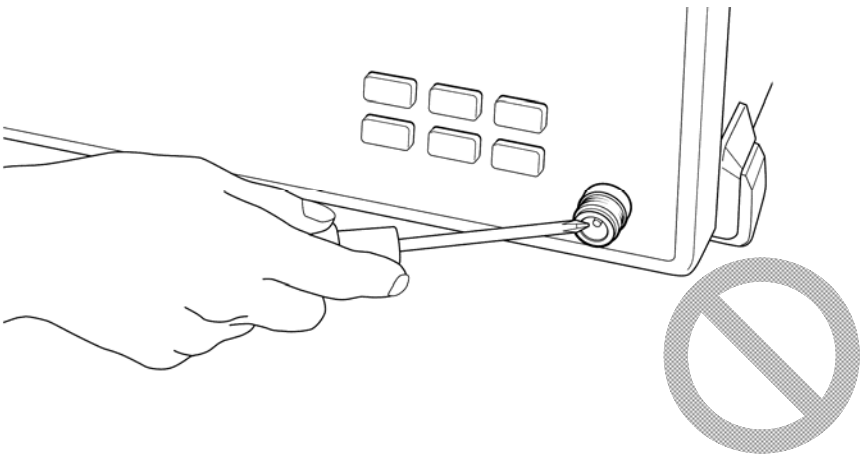
CAUTION

- Always ground both the MT8870A and DUT (including test circuit) using a 3-pin power cable. After confirming that both the MT8870A and DUT (including test circuit) are grounded, connect them using coaxial cables.

  NEVER connect the MT8870A and DUT without grounding, otherwise an electrostatic discharge may damage the MT8870A input circuits.

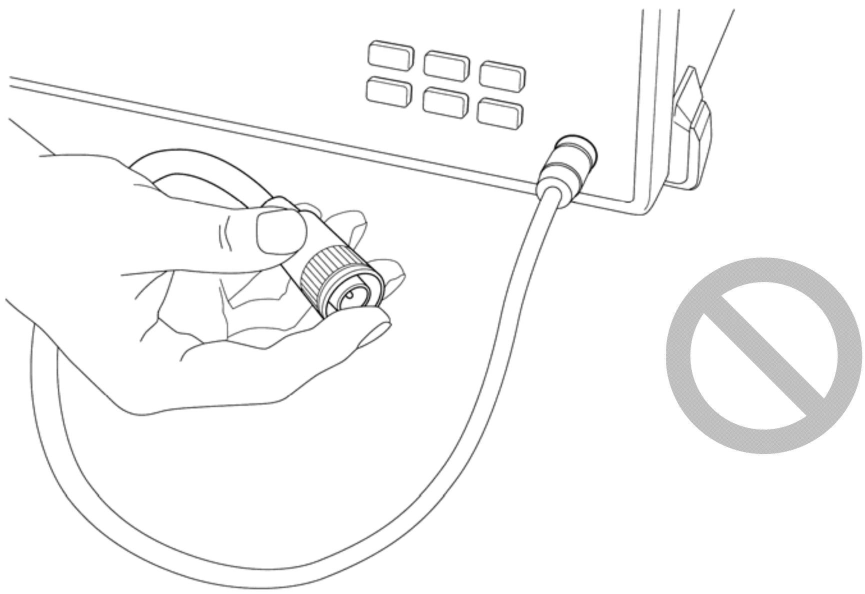
- Do not touch a metal conductor to the center pin of the connectors, and make sure metal parts are not touched, otherwise the MT8870A circuits may be damaged.





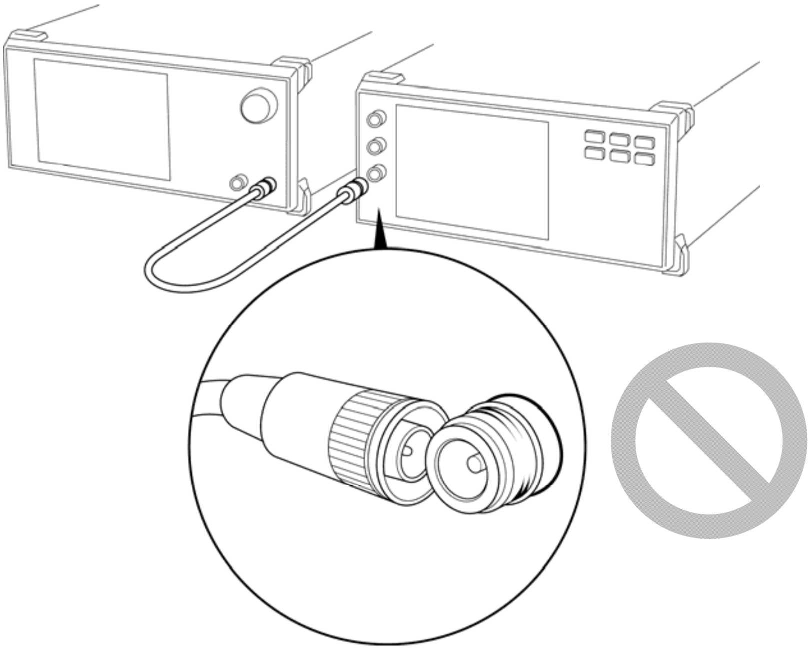
⚠ CAUTION

Do not touch a metal conductor to the center of the coaxial cable connected to the MT8870A input connector, and ensure metal parts are not touched. Otherwise, the MT8870A input circuits may be damaged.



⚠ CAUTION

When connecting a coaxial cable to a connector, ensure the center conductor does not touch other metal parts. Otherwise, the MT8870A input circuits may be damaged.



2.5 Connecting Mobile

The connection method between the module and mobile to be measured depend on RF connectors shape and assignment. The basic connection method is described below.

Note:

To measure RF signals, confirm that the output connector matches with connector specified in Section 3.5.4 “Configuring ports”.

To connect Test Port 1 to mobile:

Use an RF cable to connect Test Port 1 on the front panel and the RF signal I/O part of the mobile. Prepare an adapter if direct connection is not possible.

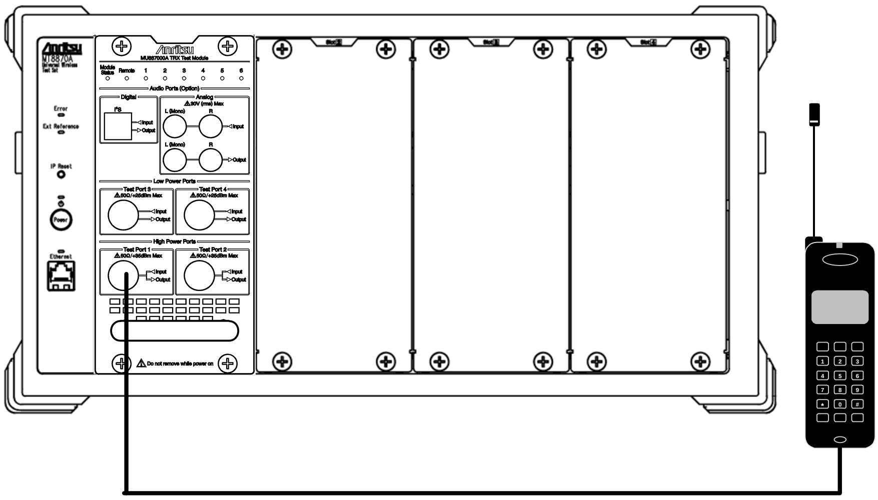


Figure 2.5-1 Mobile Connection Example

Chapter 3 Fundamental Operation

This chapter describes the basic operation of the module.
For details of commands used here, refer to Chapter 5 “SCPI Command Reference” and Chapter 6 “Native Command Reference”.

3.1 Command Language Modes

Commands for remote controlling of a module have two language modes: the SCPI command mode, and the Native command mode.

- SCPI command mode
  SCPI commands are written in the format defined by the SCPI standard (http://www.ivifoundation.org/docs/SCPI-99.PDF).
  SCPI commands are delimited by colons and designed for individual functionality in the layer structure. An example is shown below. Characters in lower case or enclosed by square brackets can be omitted.

  ```
  :SYSTem:ERRor:CODE[:NEXT]
  ```

  The SCPI command mode uses operation status registers and questionable registers to monitor the status of devices.

- Native command mode
  Native commands are not in the SCPI standard format. They can be written using fewer characters than SCPI commands. An example is shown below. Unlike SCPI commands, characters cannot be omitted in Native commands.

  ```
  SYSERRCODE
  ```

  The Native command mode uses event status registers to monitor the status of devices.

To specify the operation mode, use the following commands:
To switch from the Native command mode to the SCPI command mode:

```
SYST:LANG SCPI
```

To switch from the SCPI command mode to the Native command mode:

```
:SYSTem:LANGuage NAT
```

Both modes can use the following common commands defined by IEEE488.2:

```
*CLS, *ESE, *ESR, *IDN, *OPC, *RST, *SRE, *STB, *TRG, and
*TST, *WAI
```

3.2 Configuring Communications

Modules are controlled via Ethernet or GPIB.
One or both of the Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) can be used.

Note:

In Section 3.3 “Transferring Files”, files are transferred using the File Transfer Protocol (FTP) over Ethernet. Files cannot be transferred via GPIB.

Use Ethernet when using the software described in Chapter 8 “Utility Tool” to install applications or transfer waveform data. GPIB does not support these communication protocols.

3.2.1 Configuring Ethernet

Specify either IPv4 or IPv6, or both IPv4 and IPv6.

IPv4 Address settings

Select whether to obtain an IP address automatically using DHCP or to use a specified IP address.
When using specified IP address, set the IP address, subnet mask, and default gateway.
When obtaining an IP address using DHCP, confirm the assigned IP address, subnet mask, and default gateway.

Using DNS

When a module uses a host name to communicate with other devices, turn on the Domain Name System (DNS).
To use a DNS server, specify the primary address, secondary address, host name, and DNS domain name.
If the application software operation requires DNS, enable it.

IPv6 Address settings

Select whether to obtain an IPv6 address automatically using DHCP or to use a specified IPv6 address.
When using a specified IPv6 address, set the IP address, prefix length, and default router.
When obtaining an IPv6 address using DHCP, confirm the assigned IP address, prefix length, default gateway, and link local address.

Using DNS

When a module uses a host name to communicate with other devices, turn on the Domain Name System (DNS).

To use a DNS server, specify the primary address, secondary address, host name, and DNS domain name.
If the application software operation requires DNS, enable it.

Initialization

Ethernet settings can be initialized by the following procedure.

1. Turn off the MT8870A power supply.
2. Turn On the power supply pressing the IP Reset button of the MT8870A.
3. Keep pressing the IP Reset button until the error lamp of the MT8870A starts blinking in red.
4. Release the IP Reset button when the error lamp starts blinking.

The default settings after initialization are shown in the tables below.

Table 3.2.1-1 IP Address Default Settings

MT8870A Slot #	IPv4 Address	IPv6 Address/Prefix Length
1	192.168.1.1	0/0
2	192.168.1.2	0/0
3	192.168.1.3	0/0
4	192.168.1.4	0/0

Table 3.2.1-2 Ethernet Default Settings

	Item	Value
IPv4	Subnet mask	255.255.255.0
	Default gateway	(N/A)
	DNS settings	Not used
	DNS primary address	(N/A)
	DNS secondary address	(N/A)
IPv6	Link local address/ prefix length	Assigned automatically using MAC address
	Default router	(N/A)
	DNS settings	Not used
	DNS primary address	(N/A)
	DNS secondary address	(N/A)
Common	Host name	HOST-n*
	Domain name	DOMAIN

*: n is the number of the slot (1 to 4).
Example: If the MU887000A is installed in slot 3 of the MT8870A, the host name is HOST-3.

⚠ CAUTION

When terminating the control via Ethernet, close the current session, wait until the Remote lamp on the panel is off, and exit the computer control software.

When connected with MU887000A via Ethernet, MU887000A may become uncontrollable if you exit the control software with the Remote lamp lit. In this case, turn off the MT8870A power, and then turn it back on again.

Use the following commands to configure and query Ethernet settings.

Common to IPv4 and IPv6

- Whether using an IPv4 or IPv6 address
  `IPVER`
  `:SYSTem:COMMunicate:NET:IPVersion`
- Domain name
  `DOMAINNAME`
  `:SYSTem:COMMunicate:NET:DNS:DOMain`
- Host name
  `HOSTNAME`
  `:SYSTem:COMMunicate:NET:DNS:HOST`

IPv4

- Type of IPv4 address
  `IPV4_TYPE`
  `:SYSTem:COMMunicate:NET:IPVFour:ADDRess:TYPE`
- Fixed IPv4 address
  `IPV4_STATIC_ADDR_ALLMODULE`
  `:SYSTem:COMMunicate:NET:IPVFour:STATic:IPADdress`
- Fixed subnet mask
  `IPV4_STATIC_SUBNETMASK`
  `:SYSTem:COMMunicate:NET:IPVFour:STATic:SMASk`
- Fixed default gateway
  `IPV4_STATIC_DEFAULTGATEWAY`
  `:SYSTem:COMMunicate:NET:IPVFour:STATic:GIP`
- Query current IPv4 address
  `IPV4_CURRENT_ADDR`
  `:SYSTem:COMMunicate:NET:IPVFour:CURRent:IPADdress`
- Query current subnet mask
  `IPV4_CURRENT_SUBNETMASK`
  `:SYSTem:COMMunicate:NET:IPVFour:CURRent:SMASk`

- Query current default gateway
  IPV4_CURRENT_DEFAULTGATEWAY
  :SYSTem:COMMunicate:NET:IPVFour:CURRent:GIP
- DNS
  IPV4_DNS_AUTO_SW
  :SYSTem:COMMunicate:NET:IPVFour:DNS:AUTO:ENABle
- DNS Primary address
  IPV4_DNS_PRIMARY
  :SYSTem:COMMunicate:NET:IPVFour:DNS:PRIMary
- DNS Secondary address
  IPV4_DNS_SECONDARY
  :SYSTem:COMMunicate:NET:IPVFour:DNS:SECondary

IPv6

- Type of IPv6 address
  IPV6_TYPE
  :SYSTem:COMMunicate:NET:IPVSix:ADDRess:TYPE
- Fixed IPv6 address
  IPV6_STATIC_ADDR_ALLMODULE
  :SYSTem:COMMunicate:NET:IPVSix:STATic:IPADdress
- Fixed default router
  IPV6_STATIC_DEFAULTROUTER
  :SYSTem:COMMunicate:NET:IPVSix:STATic:DEFRouter
- Query current IPv6 address
  IPV6_CURRENT_ADDR
  :SYSTem:COMMunicate:NET:IPVSix:CURRent:IPADdress
- Query current link local address
  IPV6_CURRENT_LINK_LOCAL_ADDR
  :SYSTem:COMMunicate:NET:IPVSix:CURRent:LOCaladdress
- Query current default router
  IPV6_CURRENT_DEFAULTROUTER
  :SYSTem:COMMunicate:NET:IPVSix:CURRent:DEFRouter
- DNS
  IPV6_DNS_AUTO_SW
  :SYSTem:COMMunicate:NET:IPVSix:DNS:AUTO:ENABle
- DNS Primary address
  IPV6_DNS_PRIMARY
  :SYSTem:COMMunicate:NET:IPVSix:DNS:PRIMary
- DNS Secondary address
  IPV6_DNS_SECONDARY
  :SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary

3.2.2 Configuring GPIB

Specify the GPIB address for a specific slot in the MT8870A. When moving a module to a different slot, the GPIB address changes to the one specified for the slot.
The GPIB addresses at factory shipment are listed below. If the power is switched on while the MT8870A IP Reset button is pressed, the settings are initialized to the defaults described in section 3.2.1 Ethernet settings.

Table 3.2.2-1 GPIB Address Default Settings

MT8870A Slot #	GPIB Address
1	1
2	1
3	1
4	1

Use the following commands to configure and query GPIB settings.

- GPIB address
  `GPIBADDR`
  `:SYSTem:COMMunicate:GPIB:ADDR`
- GPIB Addresses of all modules
  `GPIBADDRALL`
  `:SYSTem:COMMunicate:GPIB:ADDR:ALL`

3.2.3 Line termination

Use the following command to specify the character string at the end of communication message sent by the module. Any of LF or CRLF, or no termination may be used.

```
DELM
:SYSTem:COMMunicate:GPIB[:SELF]:DELimiter
TRM
:SYSTem:COMMunicate:GPIB[:SELF]:TERMinator
```

3.2.4 Updating settings

When settings such as Ethernet, GPIB, and terminator, are updated, the new settings are enabled when the module in the MT8870A is restarted. At restarting, use the following commands for modules with updated Ethernet and GPIB settings, etc. (Other modules are not updated.)

```
NETRESTART
:SYSTem:COMMunicate:NET:RESTart
```

The communication settings are not changed by toggling the Power switch off and on or by using the IEEE488.2 common *RST command.

3.3 Transferring Files

Files for controlling modules can be transferred by FTP over Ethernet using the module internal storage media.
To do this, register the following at the module before starting the FTP file transfer.

- User name: MT8870A
- Password: anritsu

However, only one account can be used.

Use the following commands to control user names and passwords.

- To query user name:
  FTPUSER_LIST
  :SYSTem:COMMunicate:NET:FTP:USER:CATalog
- To change user name:
  FTPUSER_CHANGE
  :SYSTem:COMMunicate:NET:FTP:USER:CHANGe:NAME
- To change password:
  FTPUSER_PASSWD
  :SYSTem:COMMunicate:NET:FTP:USER:CHANGe:PASS

To transfer files, install FTP client software in the control PC.

Input the IP address, user name, and password in the FTP client software to enable FTP communication.

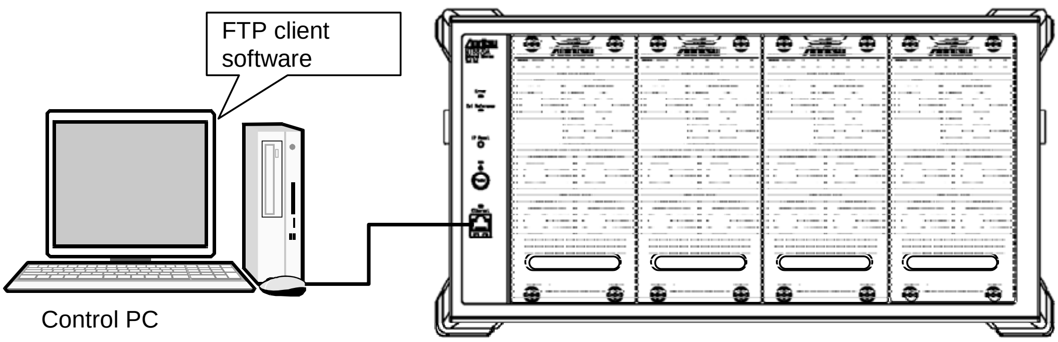


Figure 3.3-1 File Transfer Setup

If the password has been forgotten, press the **IP Reset** button on the MT8870A panel at power-on to reset the user name and password to the initial values.

3.4 System Settings

3.4.1 General settings

The following settings and queries are system wide:

- Querying results of self-diagnosis
- Date/Time
- Buzzer
- Language mode
- Switching application software
- Querying errors
- Retrieving log
- Updating firmware
- Installing license key
- Initializing parameters
- Restarting

Use the following commands to configure and query the general system settings.

- Querying results of self-diagnosis
  `*TST`
  `SYSTEST`
  `:SYSTem:SELF:TEST`
- Date/Time
  `DATETIME`
  `:SYSTem:DATE`
- Buzzer
  `BEEP`
  `:SYSTem:BEEPer`
- Language mode
  `SYST:LANG`
  `:SYSTem:LANGuage`
- Querying error
  `SYSERR`
  `:SYSTem:ERRor`
- Querying error count
  `SYSERRCNT`
  `:SYSTem:ERRor:COUNt`
- Querying error code
  `SYSERRCODE`
  `:SYSTem:ERRor:CODE[:NEXT]`

- Querying all errors
  SYSERRALL
  :SYSTem:ERRor:ALL
- Querying all error codes
  SYSERRCODEALL
  :SYSTem:ERRor:CODE:ALL
- Deleting log
  LOGCLEAR
  :MMEMory:LOG:CLEar
- Retrieving a log
  LOGREAD
  :MMEMory:LOG:LOAD
- Querying log size
  LOGSIZE
  :MMEMory:LOG:SIZe
- Querying firmware information
  FIRMCHECK
  :SYSTem:BASE:UPDate:FIRMware:CHECK
- Updating firmware
  FIRMUPDATE
  :SYSTem:BASE:UPDate:FIRMware
- Querying license key
  KEYCHECK
  :SYSTem:BASE:LICense:CHECk
- Installing license key
  KEYINST
  :SYSTem:BASE:LICense:INSTall
- Initializing parameters
  *RST
- Restart
  REBOOT
  :SYSTem:REBoot
- Shutdown
  SHUTDOWN
  :SYSTem:SHUTdown

3.4.2 Viewing product information

The following product information can be viewed:

- Information on MT8870A Universal Wireless Testing System
  Model, serial number, hardware options, FPGA version, software license, software options, internal temperature, power-on count, and run time
- Information on MU887000A TRX Test module
  Model, serial number, hardware options, FPGA version
  power-on count, run time, insertion/removal count, MAC address
- Numbers of MT8870A slots with installed modules
- Package version
- Software license expiry date

Use the following commands to view the MT8870A information:

- Model, serial number, and manufacturer
  `*IDN`
- Model, product name, hardware options, FPGA version, software license, software options
  `MFINFO`
  `:SYSTem:INFormation:MAINframe`
- Serial number
  `MFSERIAL`
  `:SYSTem:INFormation:MAINframe:DEVice:ID`
- Internal temperature
  `MFTEMP`
  `:SYSTem:INFormation:MAINframe:TEMPerature`
- Power-on count
  `MFPOWERONCNT`
  `:SYSTem:INFormation:MAINframe:POWeron:COUNt`
- Run time
  `MFTMCNT`
  `:SYSTem:INFormation:MAINframe:RTIMe`
- Firmware version
  `MCFV`
  `:SYSTem:INFormation:MAINframe:PACKage:VERSion`
- Software model
  `MCSOPT`
  `:SYSTem:INFormation:MAINframe:SOFTware`
- Software option number
  `MCOPT`
  `:SYSTem:INFormation:MAINframe:SOFTware:OPTion`
- Waveform license
  `MCWOPT`

:SYSTem:INFormation:MAINframe:WAVeform

Use the following commands to view information on the MU887000A TRX Test module:

- Model, product name, hardware options, and FPGA version
  SYSINFO
  :SYSTem:INFormation
- Serial number
  SYSSERIAL
  :SYSTem:INFormation:DEVice:ID
- Internal temperature
  SYSTEMP
  :SYSTem:INFormation:TEMPerature
- Power-on count
  SYSPOWERONCNT
  :SYSTem:INFormation:POWeron:COUNt
- Run time
  SYSTMCNT
  :SYSTem:INFormation:RTIMe
- Insertion/removal count
  SYSPLUGCNT
  :SYSTem:INFormation:PLUG:COUNt
- MAC Address
  MACADDR
  :SYSTem:COMMunicate:NET:HWADdress

Use the following command to view the numbers of the MT8870A slots with installed modules:

- Slot number
  SYSSLOT
  :SYSTem:INFormation:SLOT

The package version is a control number for the combination of the following software:

- Built-in program of MT8870A
- Built-in program of MU887000A
- Application software installed on MU887000A

Use the following commands to view the information about the MU887000A TRX Test module:

SYSVER
:SYSTem:VERSion

The application software installed on the MT8870A is licensed. The software license may have an expiry date. Use the following command to view the expiry date of the software license:

- Query license key information
  `KEYINFO`
  `:SYSTem:BASE:LICense:INFormation`

3.5 Preparing for Measurement

3.5.1 Specifying frequencies

Choose one of the following frequency reference signals. The reference signal settings are common to all modules installed in the MT8870A.

- MT8870A internal signal (internal reference signals)
- Signal input to MT8870A Ref Input connector (external reference signal)

Use the following command to specify a frequency reference signal:

```
FREQREF
:SYSTem:BASE:REFerence:FREQuency
REF
:SYSTem:BASE:REFerence:FREQuency:CONFigure
```

Use the following commands to view the type of the reference signal and whether or not the internal circuit of the MT8870A is phase-synchronized to the frequency reference signal.
This is also indicated by the MT8870A Ext. Reference lamp.
If the phase is not synchronized, the frequency accuracy might be out of specification.

```
FREQREFSOURCE
SYSTem:BASE:REFerence:FREQuency:SOURce
EXTREF
:SYSTem:BASE:REFerence:FREQuency:CONFigure:SOURce
```

Use the following command to correct the frequency when the internal reference signal is selected:
Use this command to change the module frequency.

```
ADJREFCLK
:SYSTem:BASE:REFerence:FREQuency:ADJust
```

Use the following command to view the frequency when resetting the frequency correction to the factory default.
Executing this command does not reset the frequency correction value to the factory default. To reset the value to the factory default, use the command above to specify the queried value again.

```
ADJREFCLKDEF
:SYSTem:BASE:REFerence:FREQuency:ADJust:DEFault
```

3.5.2 Level calibration

Calibrating modules adjusts for differences in level accuracies caused by internal temperature changes by flattening the frequency characteristics of level accuracies for the input and output levels.
Calibration should be executed after:

- Turning on power and warming up
- Changing test mobile
- If ambient temperature temperatures changes significantly

The calibration explained here is executed by the module itself. There are two types of calibration: full calibration, and band calibration. Band calibration is executed in the communication system bandwidth supported by the application software. Full calibration is executed in all the RF I/O bandwidths of the module. Band calibration is faster than full calibration.

Note:

Wait at least 60 minutes after power-on before executing full calibration. After full calibration, band calibration should be executed at a specific instance, such as when changing the mobile.

Use the following commands to execute full calibration and retrieve information.

To execute full calibration:
```
FULLCAL
:CALCulate:CALibration:FULL:STARt
```
To retrieve time when full calibration executed:
```
FULLCAL_TM
:CALCulate:CALibration:FULL:LASTtime
```
To retrieve time when full calibration failed:
```
FULLCALERR_TM
:CALCulate:CALibration:FULL:ERRor:LASTtime
```
To retrieve count when full calibration failed:
```
FULLCALERRCNT
:CALCulate:CALibration:FULL:ERRor:COUNt
```
To retrieve result of full calibration:
```
FULLCALRESULT
:CALCulate:CALibration:FULL:RESult
```

To retrieve elapsed time since executing full calibration:

```
CALTIME
:CALCulate:CALibration:ELAPsed:TIME
```

To retrieve temperature when full calibration executed:

```
CALTEMP
:CALCulate:CALibration:TEMPerature
```

Use the following commands to execute band calibration and retrieve the information.

To execute band calibration:

```
BANDCAL
:CALCulate:CALibration:BAND:STARt
```

To execute band calibration when internal temperature changed:

```
BANDCAL_TEMP
:CALCulate:CALibration:BAND:STARt:TEMPerature
```

To retrieve time when band calibration executed:

```
BANDCAL_TM
:CALCulate:CALibration:BAND:LASTtime
```

To retrieve time when band calibration failed:

```
BANDCALERR_TM
:CALCulate:CALibration:BAND:ERRor:LASTtime
```

To retrieve band calibration failure count:

```
BANDCALERRCNT
:CALCulate:CALibration:BAND:ERRor:COUNt
```

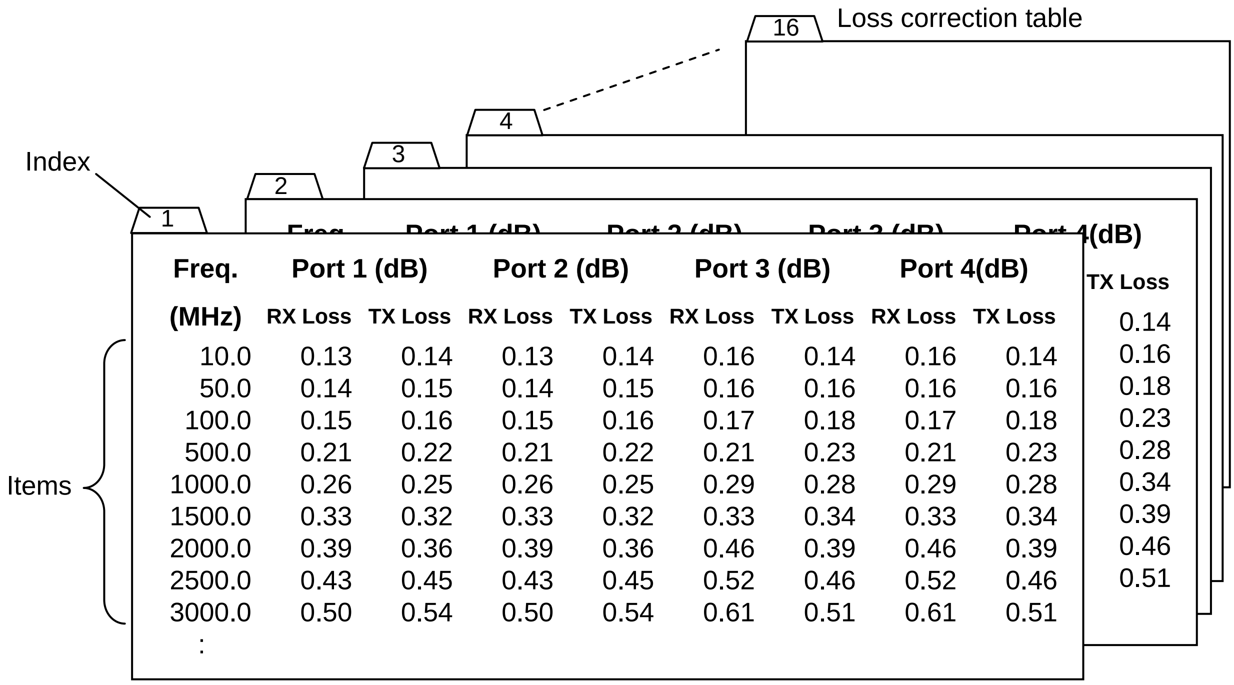
To retrieve band calibration result:

```
BANDCALRESULT
:CALCulate:CALibration:BAND:RESult
```

3.5.3 Correcting cable losses

The MU887000A can store 16 sets of loss correction data for coaxial cables connected to module connectors.

The correction data are saved in tables showing loss for each port by frequency.



Freq.	Port 1 (dB)		Port 2 (dB)		Port 3 (dB)		Port 4(dB)	
(MHz)	RX Loss	TX Loss	RX Loss	TX Loss	RX Loss	TX Loss	RX Loss	TX Loss
10.0	0.13	0.14	0.13	0.14	0.16	0.14	0.16	0.14
50.0	0.14	0.15	0.14	0.15	0.16	0.16	0.16	0.16
100.0	0.15	0.16	0.15	0.16	0.17	0.18	0.17	0.18
500.0	0.21	0.22	0.21	0.22	0.21	0.23	0.21	0.23
1000.0	0.26	0.25	0.26	0.25	0.29	0.28	0.29	0.28
1500.0	0.33	0.32	0.33	0.32	0.33	0.34	0.33	0.34
2000.0	0.39	0.36	0.39	0.36	0.46	0.39	0.46	0.39
2500.0	0.43	0.45	0.43	0.45	0.52	0.46	0.52	0.46
3000.0	0.50	0.54	0.50	0.54	0.61	0.51	0.61	0.51
:								

Figure 3.5.3-1 Loss Correction Data Layout

There are two types of loss correction data for each Port:

- For signals received by MU887000A (uplink)
- For signals sent by MU887000A (downlink)

When losses of a measurement system part are affected by direction, such as a directional coupler, specify correction factors by signal direction.

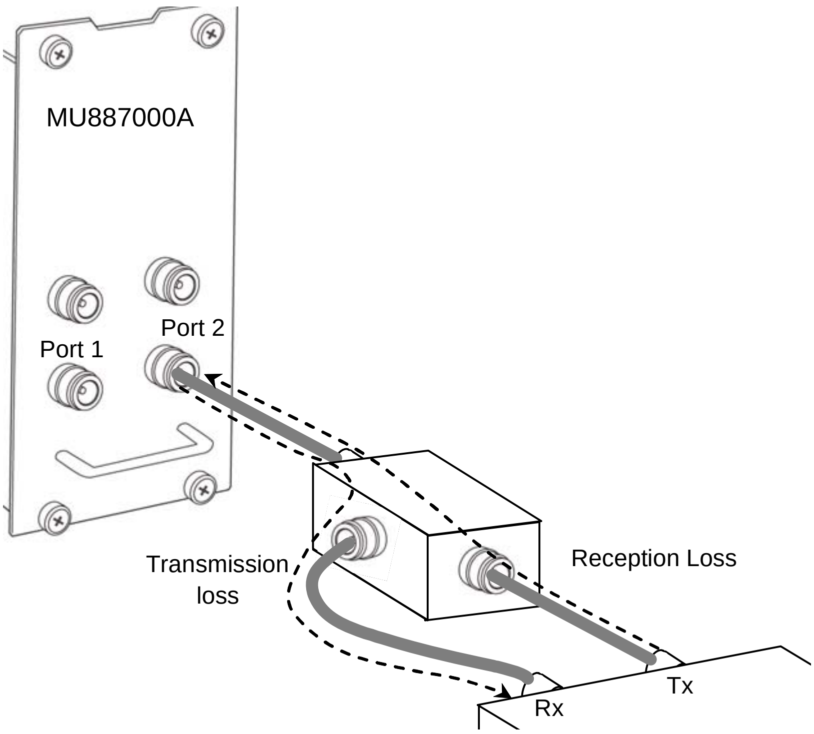


Figure 3.5.3-2 Loss Correction Data Structure

Use the following commands to edit the contents of loss correction data tables:

```
LOSSTBLVAL
:CALCulate:EXTLoss:TABLe:VALue
LOSSTBLVALALL
:CALCulate:EXTLoss:TABLe:VALue:ALL
DELLOSSTBL
:CALCulate:EXTLoss:TABLe:DELete
LOSSTBLCNT
:CALCulate:EXTLoss:TABLe:COUNt
```

Use the following command to select the index from the 16 tables to apply changes:

```
LOSSTBL
:CALCulate:EXTLoss:TABLe:SETTing
```

Use the following command to set loss correction ON/OFF for all ports

```
EXTLOSSW
[:ROUTe]:EXTLoss:TABLe:SWITch
```

3.5.4 Configuring ports

Use the following command to specify a test port on the front panel for signal I/O:

```
PORT
:ROUTe:PORT:CONNect:DIRection
```

Simultaneous signal input and output can be specified for test ports 1 and 2.
Signal input or output can be specified for test ports 3 and 4.

3.5.5 Configuring trigger connector

When using the rear-panel Trigger connector during measurement, use the following commands to set connector input or output and the trigger signal type.

Trigger connector input/output

```
MFTRIGPORT
:ROUTe:TRIGger:PORT
```

Types of trigger input signals

```
SYSTRIGINSRC
:ROUTe:TRIGger:INPut:SOURce
```

Types of trigger output signals

```
SYSTRIGOUTSRC
:ROUTe:TRIGger:OUTPut:SOURce
```

Specify the Trigger connector input/output for each module.

3.6 Status Registers

The module includes registers to show the status of devices, such as errors or command status. This section describes the registers.

3.6.1 Configuring registers

Figures 3.6.1-1 and 3.6.1-2 show the register configurations.

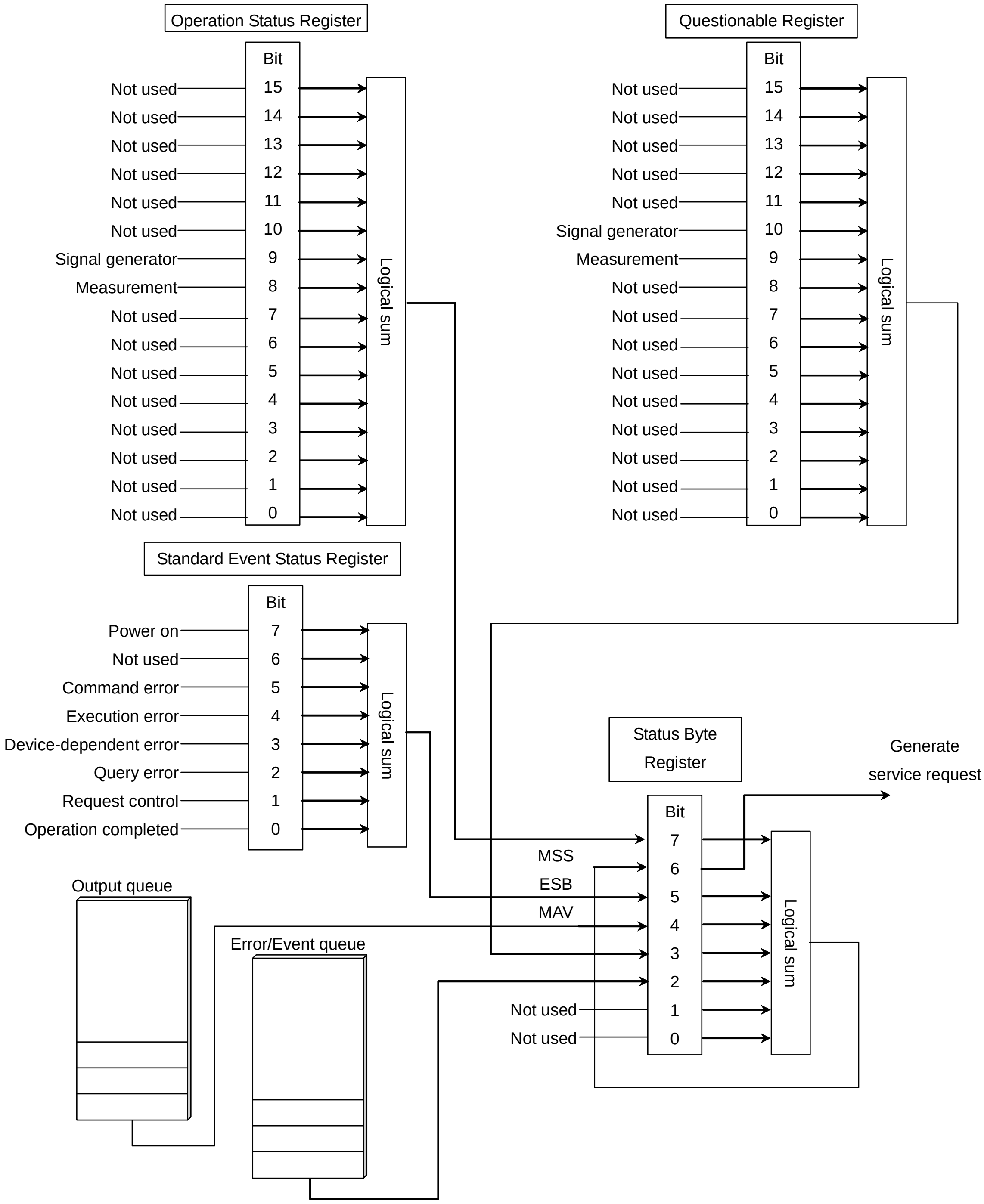


Figure 3.6.1-1 Register Configurations (SCPI Command Mode)

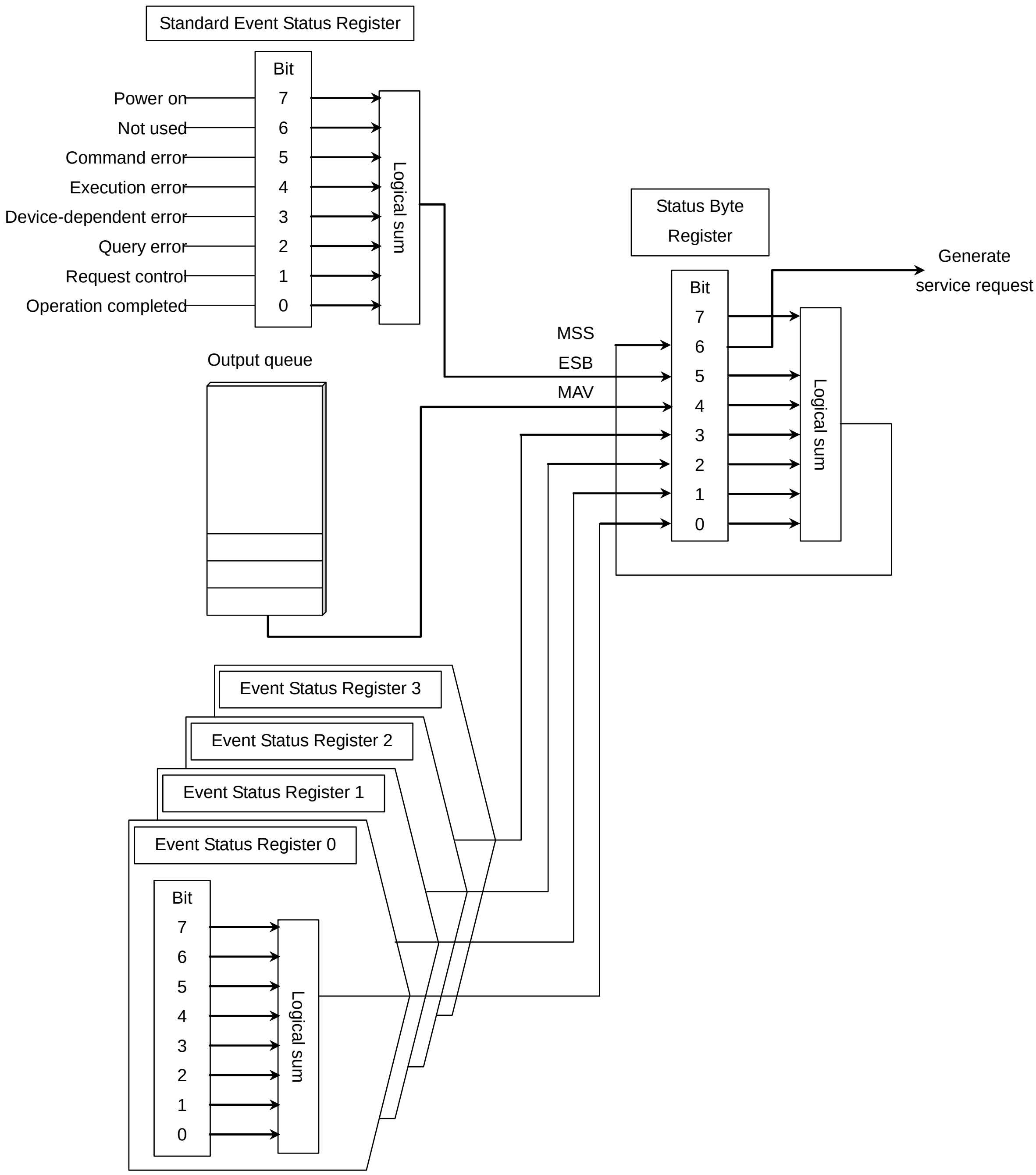


Figure 3.6.1-2 Register Configurations (Native Command Mode)

Each register has 8- or 16-bit data. The register output values are the totals of the decimal values for each bit shown in Table 3.6.1-1.

Table 3.6.1-1 Converted Decimal Values corresponding to Register Bits

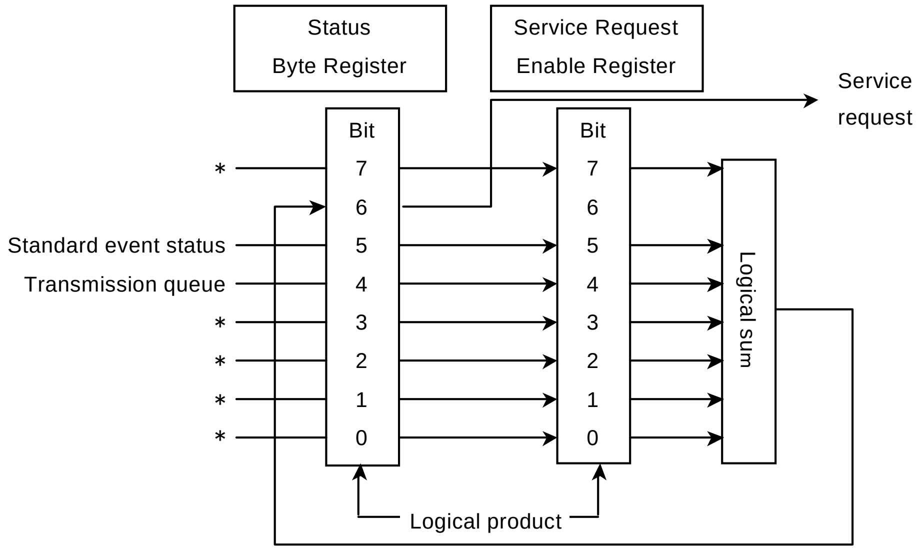
Bit	Decimal Value	Bit	Decimal Value
0	1	8	256
1	2	9	512
2	4	10	1024
3	8	11	2048
4	16	12	4096
5	32	13	8192
6	64	14	16384
7	128	15	32768

Each register includes a pair of bit-enable registers.

3.6.2 Status byte register

The status byte register (STB) shows the status of GPIB standard devices. A change in the device status changes a value in the status byte register. When this occurs, an interrupt can be generated for the PC controller. The interrupt is called a service request.

The status byte register is paired with the service request enable register (SRER). The service request enable register selects a bit of the status byte to generate a service request.



*Signal bits 7 and 3 to 0 depend on command modes.

Figure 3.6.2-1 Configuration of Status Byte Register and Service Request Enable Register

Note:

A service request uses the GPIB interface.

To read the status byte register:

- Use the common command *STB?
- Use GPIB serial polling (when option 001/101 installed).
  With serial polling, even if bit 6 is 1, it changes to 0 after reading once.

To configure and read the service request enable register, use the common commands *SRE and *SRE?. Set the appropriate bit of the service request enable register to 1 to output data from the status byte register.

The following table shows the status byte register bit definitions.

Table 3.6.2-1 Bit Definitions of Status Byte Register

Bit	Description
7	SCPI command mode: operation status register Native command mode: not used and always set to 0
6	Master Summary Register (MSS) Logical sum of bits 7 and 5 to 0 of logical product of status byte register and service request enable register
5	Standard event register
4	Message Available summary (MAV) Changed to 1 if response message exists in output queue of measuring instrument
3	SCPI command mode: questionable register Native command mode: event register 3
2	SCPI command mode: changed to 1 if data exists in error event queue of measuring instrument Native command mode: event register 2
1	SCPI command mode: not used and always set to 0 Native command mode: event register 1
0	SCPI command mode: not used and always set to 0 Native command mode: event register 0

Bit 7 of the status byte register indicates information on the operation status register.
For details, refer to Section 3.6.5 “Operation status register”.

Bit 6 of the status byte register is known as the Master Summary Status (MSS). When MSS is 1, there is information from the module to the PC controller. A service request is generated when MSS is changed from 0 to 1.

Bit 5 of the status byte register indicates the standard event register information.
For details, refer to Section 3.6.3 “Standard event status register”.

The common command *CLS changes bits 7 and 5 of the status byte register to 0.
If *CLS is sent after a command or *CLS is sent before a query, the transmission queue is cleared and bit 4 is changed to 0.

The *CLS command does not set the service request enable register to 0. Use the *SRE command.

3.6.3 Standard event status register

The standard event status register (ESR) is paired with the standard event status enable register (ESE). The logical product of these two registers is calculated and the logical sum of the results of each bit is output to bit 5 of the status byte register.

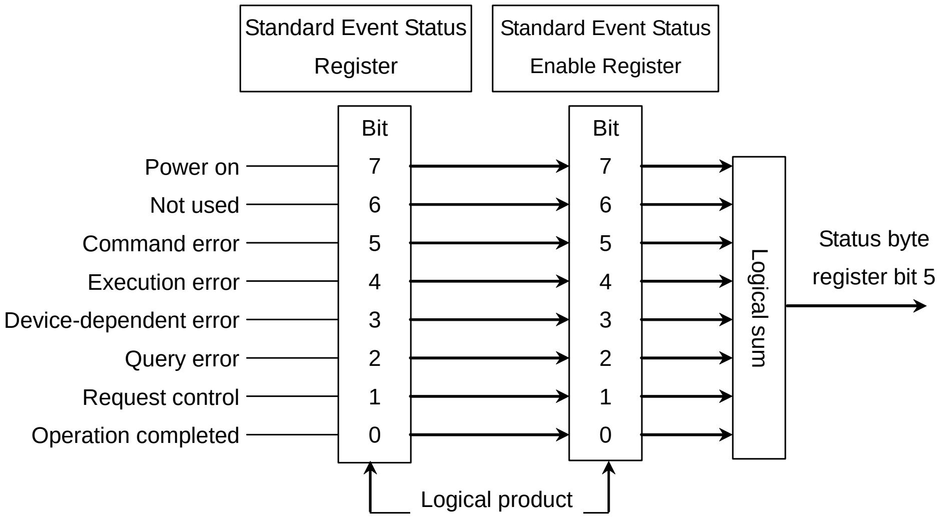


Figure 3.6.3-1 Standard Event Register Configuration

The bit definitions of the standard event status register are as follows:

Table 3.6.3-1 Standard Event Register Bit Definitions

Bit	Description
7	Powered on Changed to 1 at power-on and changes to 0 after reading once
6	Not used and always set to 0
5	Command error Changed to 1 when received program message undefined, does not meet syntax, or has spelling error
4	Execution error Changed to 1 when received program message not command error and cannot be executed
3	Device-dependant error Changed to 1 at error other than command error, execution error, or query error
2	Query error Changed to 1 if reading process executed when no response data, or response data overflows from queue and lost
1	Request control Changed to 1 when module asks to be controller
0	Operation completed Changed to 0 when program message received Changed to 1 when command operation completed or query response data sent

The common command `*OPC` reads bit 0.
The common command `*CLS` changes bits of the standard register to 0.

3.6.4 Event register

Like the standard event status register, the event status register is paired with the enable register.
The module event status registers 0 to 3 have the following functions:

0: End event status register (signal generator)
1: Error event status register (signal generator)
2: End event status register (measurement)
3: Error event status register (measurement)

Table 3.6.4-1 Bit Definitions of End Event Status Register (Signal Generator)

Bit	Description
7 to 1	Not used and always set to 0
0	Changed to 1 while reading file

Table 3.6.4-2 Bit Definitions of Error Event Status Register (Signal Generator)

Bit	Description
7 to 1	Not used and always set to 0
0	Changed to 1 at error in read file

Table 3.6.4-3 Bit Definitions of End Event Status Register (Measurement)

Bit	Description
7 to 1	Not used and always set to 0
0	Changed to 1 during measurement

Table 3.6.4-4 Bit Definitions of Error Event Status Register (Measurement)

Bit	Description
7 to 3	Not used and always set to 0
2	Changed to 1 when measurement timed out
1	Changed to 1 when measurement result level under
0	Changed to 1 when measurement result level over

Use the following commands to specify or query each register:

End event status register (signal generator): `ESE0` and `ESR0`

Error event status register (signal generator): `ESE1` and `ESR1`

End event status register (measurement): `ESE2` and `ESR2`

Error event status register (measurement): `ESE3` and `ESR3`

3.6.5 Operation status register

The operation status register (OSR) consists of the following registers:

- Operation status condition register
- Transition filter
- Operation status event register
- Operation status enable register (OSER)

The operation status condition register shows status changes. When the status changes, the value of this register also changes.

The operation status event register records changes of values of the operation status condition register. The transition filter, which defines conditions to be written, is before the operation status event register. The transition filter changes the bit of the operation event register to 1 when one of the following conditions is met:

- Bit changes from 0 to 1;
- Bit changes from 1 to 0; or
- Bit changes from 0 to 1, or bit changes from 1 to 0.

The operation status enable register specifies output for each bit of the operation status event register. The logical product of these two registers is calculated and the logical sum of the results of each bit is output to bit 7 of the status byte register.

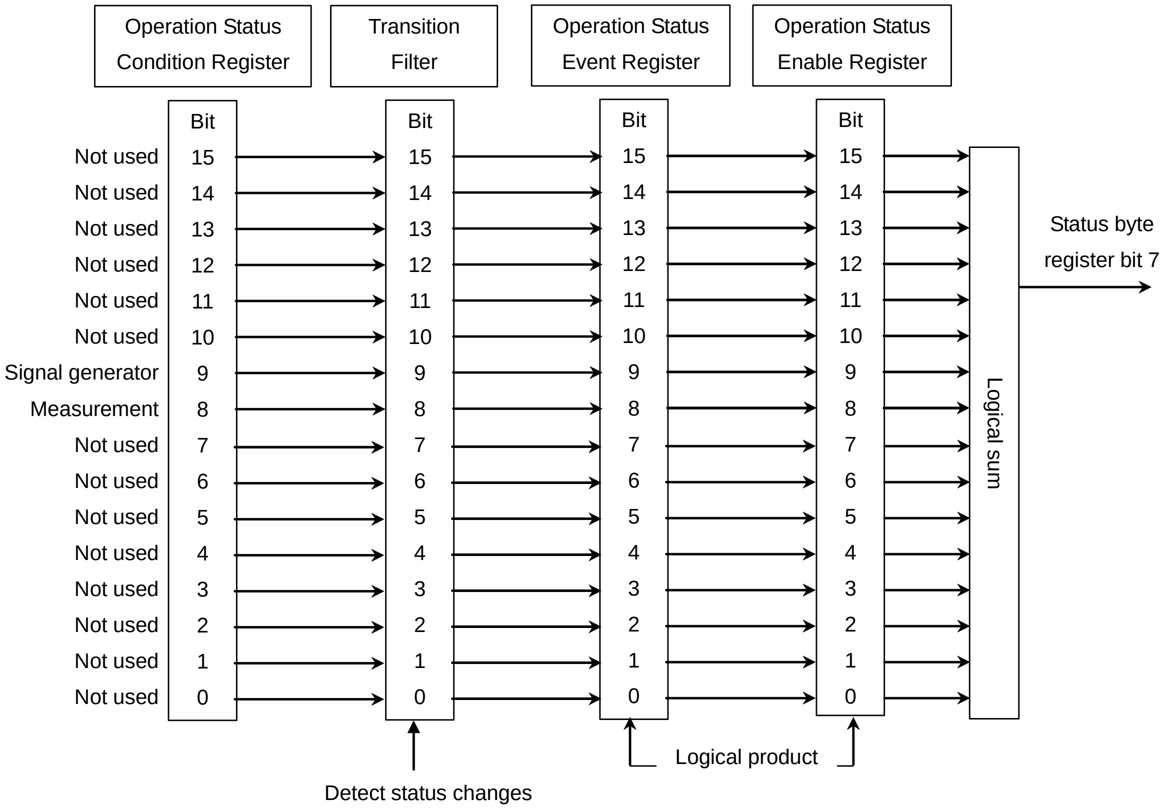


Figure 3.6.5-1 Operation Status Register Configurations

The bit definitions of the operation status condition register are as follows:

Table 3.6.5-1 Bit Definitions of Operation Status Condition Register

Bit	Description
15 to 10	Not used and always set to 0
9	Logical sum of signal generator status registers
8	Logical sum of measurement status registers
7 to 0	Not used and always set to 0

Bits 9 and 8 of the operation status condition register are set to the logical sum of the signal generator status register and the measurement status register with the same configuration as the operation status register.

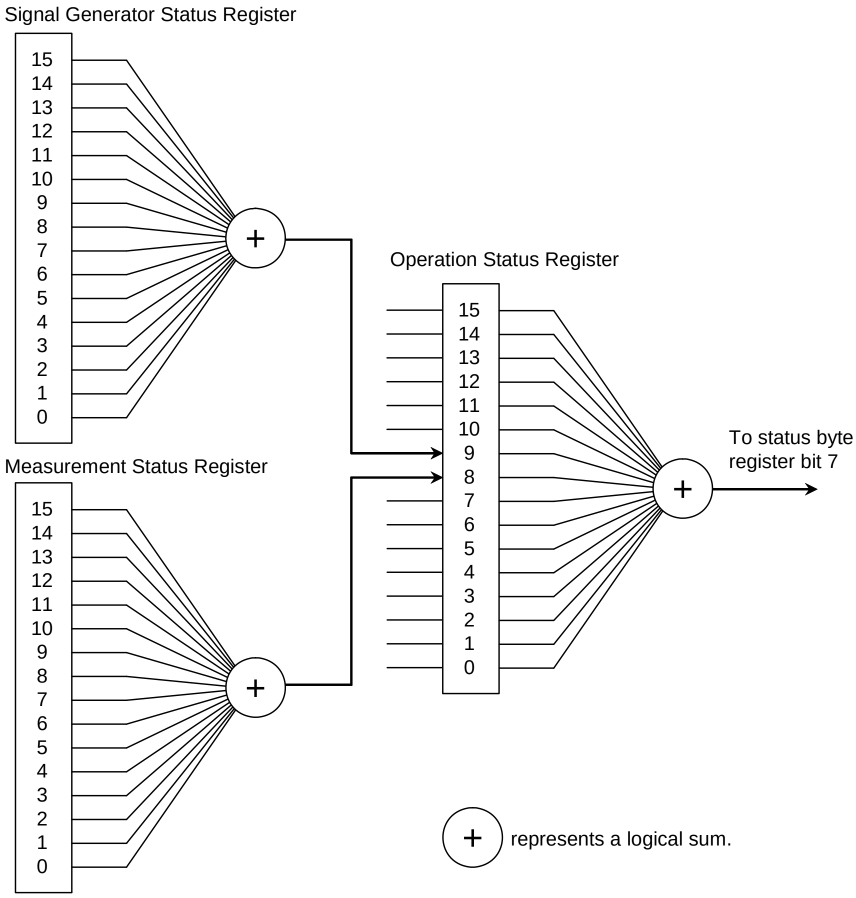


Figure 3.6.5-2 Operation Status Register Connections

The table below shows the bit definitions of the signal generator status register and the measurement status register.
The application software may assign functions to unused bits.

Table 3.6.5-2 Bit Definitions of Signal Generator Status Register

Bit	Description
15 to 3	Not used and always set to 0
2	Changed to 1 while preparing synchronized output of waveform pattern.
1	Changed to 1 while executing defragmentation.
0	Changed to 1 while reading file

Table 3.6.5-3 Bit Definitions of Measurement Status Register

Bit	Description
15 to 1	Not used and always set to 0
0	Changed to 1 during measurement

Use the following commands to specify or query the operation status register:

```
:STATus:OPERation[:EVENt]
:STATus:OPERation:CONDition
:STATus:OPERation:ENABle
:STATus:OPERation:NTRansition
:STATus:OPERation:PTRansition
```

Use the following commands to specify or query the signal generator status register:

```
:STATus:OPERation:GENerator[:EVENt]
:STATus:OPERation:GENerator:CONDition
:STATus:OPERation:GENerator:ENABle
:STATus:OPERation:GENerator:NTRansition
:STATus:OPERation:GENerator:PTRansition
```

Use the following commands to specify or query the measurement status register:

```
:STATus:OPERation:MEASure[:EVENt]
:STATus:OPERation:MEASure:CONDition
:STATus:OPERation:MEASure:ENABle
:STATus:OPERation:MEASure:NTRansition
:STATus:OPERation:MEASure:PTRansition
```

3.6.6 Questionable register

The questionable condition register changes when:

- Signals output from the device may not be set appropriately.
- The measurement result may be incorrect.

For example, a bit of the questionable register is set if the level or frequency of the actual output signal does not match the setting, or if the level is not measured because it is over- or under-range.

Like operation status register, the questionable register consists of the following registers:

- Questionable condition register
- Transition filter
- Questionable event register
- Questionable enable register (QER)

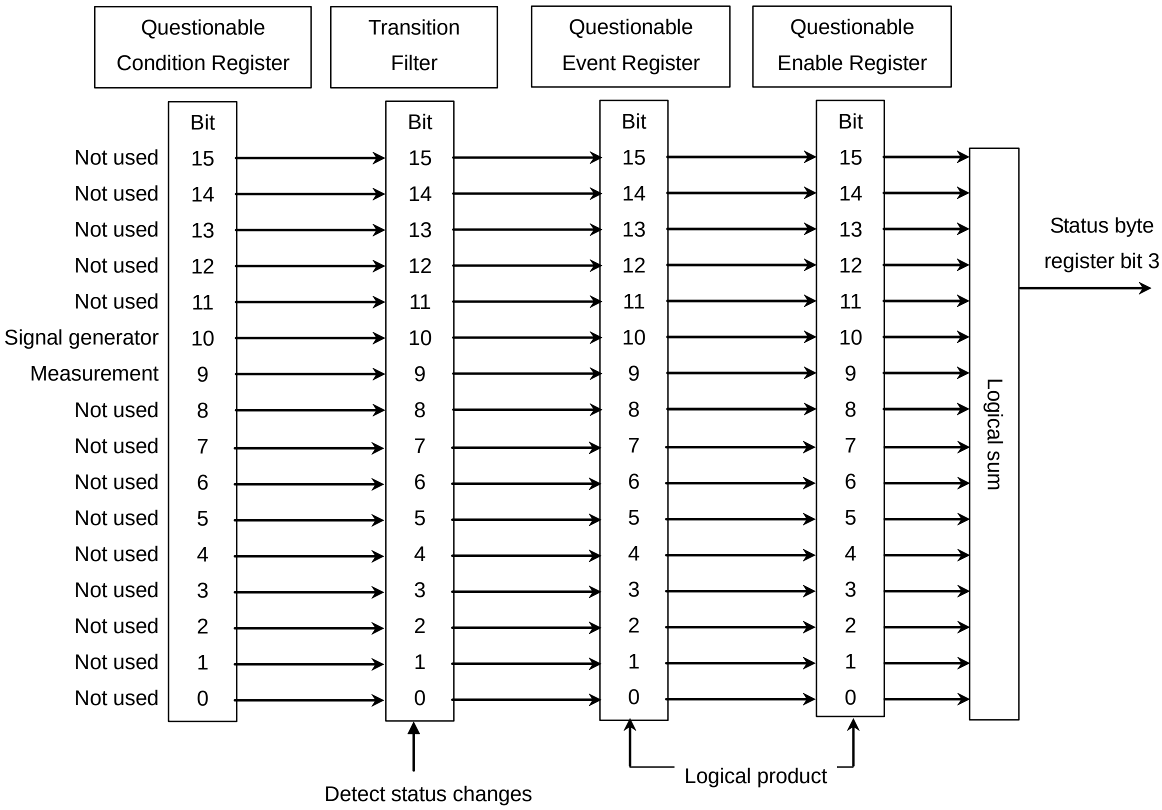


Figure 3.6.6-1 Questionable Register Configurations

The bit definitions of the questionable condition register are as follows:

Table 3.6.6-1 Bit Definitions of Questionable Condition Register

Bit	Description
15 to 11	Not used and always set to 0
10	Logical sum of signal generator questionable registers
9	Logical sum of logical sums of measurement questionable registers
8 to 0	Not used and always set to 0

Bits 10 and 9 of the questionable condition register are set to a logical sum of the signal generator questionable register and the measurement questionable register that have the same configuration as the questionable register.

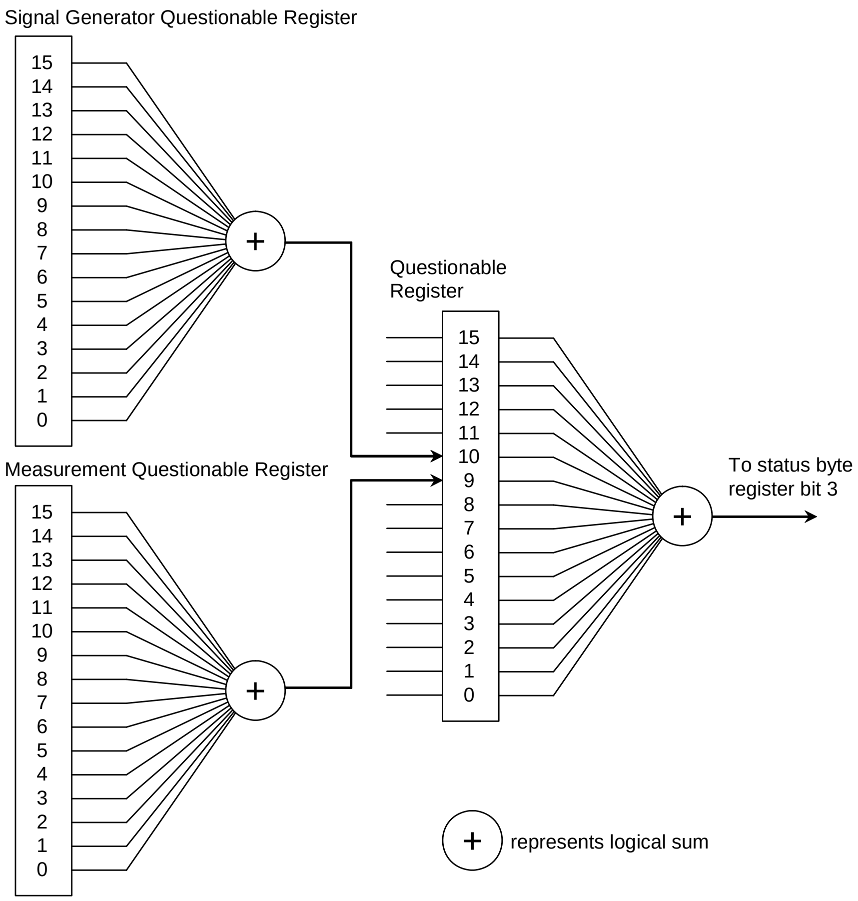


Figure 3.6.6-2 Questionable Register Connections

The table below shows the bit definitions of the signal generator questionable register and the measurement questionable register. The application software may assign functions to unused bits.

Table 3.6.6-2 Bit Definitions of Signal Generator Questionable Register

Bit	Description
15 to 1	Not used and always set to 0
0	Changed to 1 at error in read file

Table 3.6.6-3 Bit Definitions of Measurement Questionable Register

Bit	Description
15 to 3	Not used and always set to 0
2	Changed to 1 when measurement timed out
1	Changed to 1 when measurement result under level
0	Changed to 1 when measurement result over level

Use the following commands to specify or query the questionable register:

```
:STATus:QUEStionable[:EVENt]
:STATus:QUEStionable:CONDition
:STATus:QUEStionable:ENABle
:STATus:QUEStionable:NTRansition
:STATus:QUEStionable:PTRansition
```

Use the following commands to specify or query the signal generator questionable register:

```
:STATus:QUEStionable:GENerator[:EVENt]
:STATus:QUEStionable:GENerator:CONDition
:STATus:QUEStionable:GENerator:ENABle
:STATus:QUEStionable:GENerator:NTRansition
:STATus:QUEStionable:GENerator:PTRansition
```

Use the following commands to specify or query the measurement questionable register:

```
:STATus:QUEStionable:MEASure[:EVENt]
:STATus:QUEStionable:MEASure:CONDition
:STATus:QUEStionable:MEASure:ENABle
:STATus:QUEStionable:MEASure:NTRansition
:STATus:QUEStionable:MEASure:PTRansition
```

Chapter 4 Vector Signal Generator

This chapter describes how to use MU887000A TRX Test module as a vector signal generator.
For the command details, refer to Chapter 5 “SCPI Command Reference”.

4.1 Vector Signal Generator (VSG) Introduction

A signal generator (SG) generates test signals at the programmed frequency, level, and modulation. A signal generator that can modulate both phase and amplitude is called a vector signal generator (VSG).

The output voltage V of a signal generator is expressed by the following equation.

$$V = R\sin(2\pi ft + \varphi)$$

R: Amplitude
f: Frequency [Hz]
φ: Phase difference (rad.)

A VSG modulates phase by combining sine-wave signals with different phases by π/2 (rad.).

$$R\sin(2\pi ft + \varphi) = I\cos(2\pi ft) + Q\sin(2\pi ft)$$

$$I = R\cos(\varphi),\ \ Q = R\sin(\varphi)$$

The amplitude elements of phase modulation are called I (In phase) and Q (Quadrature phase) respectively. IQ amplitude signals for phase modulation are called baseband signals.

The MU887000A TRX Test module performs phase modulation by regenerating the baseband signal saved in a file.

Refer to the *Waveform File for Cellular Application Operation Manual* for the waveform file specifications.

4.2 Operation Mode

A VSG has two operation modes.

- Normal
  Used as signal generator with remote control
- Sequence
  Executes sequence function
  This mode has two statuses: executing sequence (Active), and stopped (Inactive).

The following figure shows the mode setting changes.

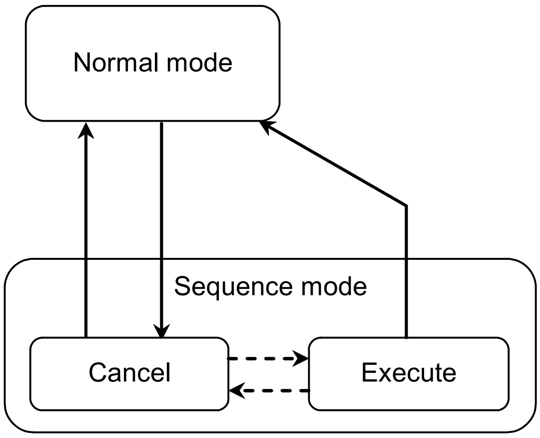


Figure 4.2-1 Operation Mode Setting Changes and Directions

Use the `:SOURce:GPRF:GENerator:MODE` command to set the operation mode.
Use the following commands to execute/cancel the Sequence mode.

`:SOURce:GPRF:GENerator:SEQuence:EXECute`
`:SOURce:GPRF:GENerator:SEQuence:CANCel`

However, when controlling the VSG in the Native command mode, use the SCPI command with short format. Do not use the SCPI command head colon.

SCPI command mode
`:SOURce:GPRF:GENerator:SEQuence:CANCel`
Native command mode
`SOUR:GPRF:GEN:SEQ:CANC`

4.3 Normal Mode

4.3.1 Fundamental setting

Send the following command to set the VSG to Normal mode.

`:SOURce:GPRF:GENerator:MODE NORMAL`

The VSG basic settings are as follows:

- Signal output port selection
- Signal output On/Off
- Frequency
- Level
- Modulation On/Off
- Spectrum reverse
- AWGN addition On/Of
- AWGN level
- Waveform pattern regeneration

Use the following commands to set non-modulation waveform data. For commands of modulation waveform data, refer to Section 4.3.2 “Modulation”.

- Output control
  Output On/Off
  `:SOURce:GPRF:GENerator:STATe`
  Port selection
  `:ROUTe:PORT:CONNect:DIRection`
  `PORT` (Native command)
- Frequency
  `:SOURce:GPRF:GENerator:RFSettings:FREQuency`
- Level
  Settings
  `:SOURce:GPRF:GENerator:RFSettings:LEVel`
  Output range confirmation
  `:SOURce:GPRF:GENerator:RFSettings:LEVel:SETTing`
- Modulation
  Modulation On/Off
  `:SOURce:GPRF:GENerator:BBMode`
  Spectrum reverse
  `:SOURce:GPRF:GENerator:RFSettings:DM:POLarity`
  AWGN Addition On/Off
  `:SOURce:GPRF:GENerator:ARB:NOISe:STATe`
  AWGN Level
  `:SOURce:GPRF:GENerator:ARB:NOISe:CN`

Waveform pattern play

`:SOURce:GPRF:GENerator:ARB:WAVeform:RESTart`

The procedure to output a continuous waveform (CW) with a frequency of 2000 MHz and a level of –10 dBm at Port 1 is described below as an example.

1. Set operation mode to Normal.

   `:SOURce:GPRF:GENerator:MODE NORMAL`
2. Set frequency to 2000 MHz.

   `:SOURce:GPRF:GENerator:RFSettings:FREQuency 2GHZ`
3. Set level to –10 dBm.

   `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`
4. Set AWGN level to 30 dB.

   `:SOURce:GPRF:GENerator:ARB:NOISe:CN 30`
5. Set modulation to None (CW).

   `:SOURce:GPRF:GENerator:BBMode CW`
6. Set Port 1 to Output.

   `:ROUTe:PORT:CONNect:DIRection PORT1, PORT1`
7. Output signal at port.

   `:SOURce:GPRF:GENerator:STATe ON`
8. Query whether output level within guaranteed level accuracy.

   `:SOURce:GPRF:GENerator:RFSettings:LEVel:SETTing?`
9. Add AWGN.

   `:SOURce:GPRF:GENerator:ARB:NOISe:STATe ON`

Note:

The level of signals output at the panel connector is the sum of the set value and cable loss.

Refer to section 3.5.3 “Correcting cable loss for the cable loss commands”.

4.3.2 Modulation

The MU887000A modulates signals using a waveform file describing IQ data.

The waveform file is saved in the MU887000A so the waveform memory must be queried to use it. The operation to read the waveform file into the waveform memory is called “loading.”

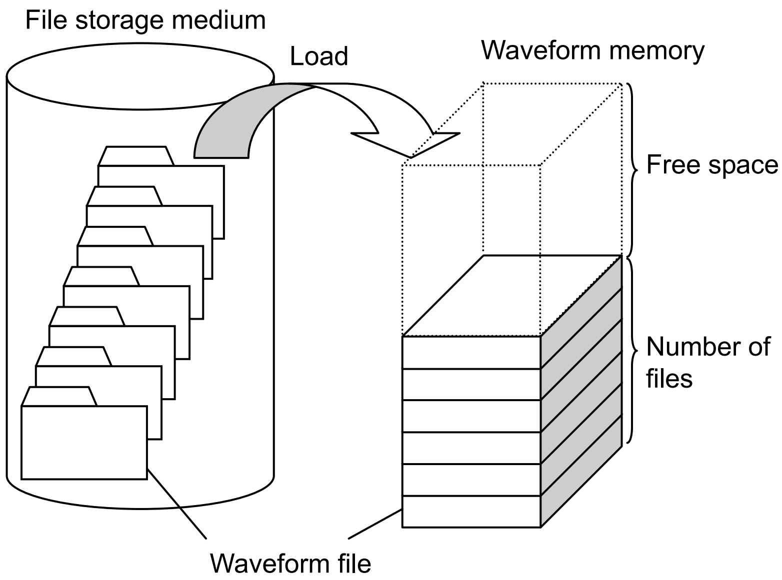


Figure 4.3.2-1 Loading Waveform File

To use the waveform file for modulation, execute the following settings.

- Specify the waveform file to be used for modulation from the waveform files loaded in waveform memory.
- Set whether IQ data is inverted.
- Set the modulation to ARB (arbitrary waveform).

Note:

When the waveform file for modulation is not specified, signals are not output from the test port on the panel even after setting the modulation to ARB.

Waveform file configuration

The waveform file includes one or more waveform patterns, and has an extension “.xml” after the file name. The waveform pattern is composed of pattern (IQ data), pattern name (title), group number, and note.
The order of the waveform patterns saved in the file is called group index.
In addition to this, a group number is assigned to each waveform pattern. The pattern to use is specified by group number.

Note:

Group numbers are equivalent to pattern numbers for the cellular application software.

Waveform file

Number of pattern

Group Index	Title	Pattern	Group Number	Note
1	FL Wave, RL Request : PCB 0dB		1	
2	FL Wave, RL Request : PCB +1dB		2	
3	FL Wave, RL Request : PCB –1dB		5	
4	FL Wave, RL Request : PCB 0dB		7	

Figure 4.3.2-2 Waveform file configuration

Gap

To modulate by waveform file, the pattern's play interval (gap) can be set.

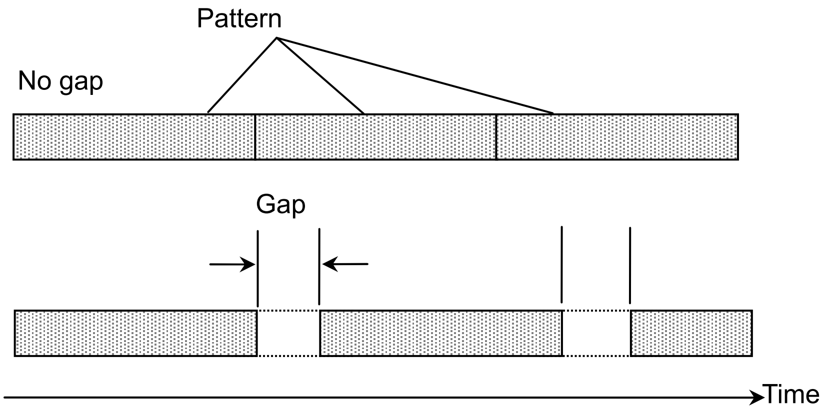


Figure 4.3.2-3 Pattern Gap

Use the following commands to operate the waveform file and memory. When the waveform file is being loaded, the status indication lamp 2 on the panel is flashing in green. For the description of the lamp, refer to Appendix D “Status indication of lamps”.

- Load waveform file
  `:SOURce:GPRF:GENerator:ARB:FILE:LOAD`
- Stop loading waveform file
  `:SOURce:GPRF:GENerator:ARB:FILE:LOAD:CANCel`

- Query waveform file read status
  :SOURce:GPRF:GENerator:ARB:FILE:LOAD:STATus
- Query group information number of waveform file in the file storage media.
  :SOURce:GPRF:GENerator:ARB:FILE:PATTern:COUNt
- Query group number of waveform file in the file storage media.
  :SOURce:GPRF:GENerator:ARB:FILE:PATTern
- Query group information pattern name of waveform file in the file storage media.
  :SOURce:GPRF:GENerator:ARB:FILE:PATTern:NAME
- Delete waveform data
  :SOURce:GPRF:GENerator:ARB:WAVeform:DELete
  :SOURce:GPRF:GENerator:ARB:WAVeform:DELete:ALL
- Generate zero pattern
  :SOURce:GPRF:GENerator:ARB:WAVeform:GENerate:ZVSPattern
- Query read waveform file count
  :SOURce:GPRF:GENerator:ARB:WAVeform:COUNt
- Optimize memory
  :SOURce:GPRF:GENerator:ARB:WAVeform:DEFRag
- Select waveform data to play
  :SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect
- Set sub-section selection method
  :SOURce:GPRF:GENerator:ARB:WAVeform:SSWitch
- Query file version
  :SOURce:GPRF:GENerator:ARB:FILE:VERSion
- Query memory free space
  :SOURce:GPRF:GENerator:ARB:WAVeform:FREE
- Query file count
  :SOURce:GPRF:GENerator:ARB:WAVeform:COUNt
- Query file name
  :SOURce:GPRF:GENerator:ARB:WAVeform:NAME
- Query group number
  :SOURce:GPRF:GENerator:ARB:WAVeform:PATTern
- Query number of group information units
  :SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:COUNt
- Query gap length setting.
  :SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:GAP
- Query pattern name
  :SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:NAME
- Query note character string
  :SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:NOTE
- Query sampling rate
  :SOURce:GPRF:GENerator:ARB:WAVeform:SCLock:RATE

The procedure to output the following signal at Port 2 is described below as an example.

Frequency: 1800 MHz
Level: –20 dBm
Waveform file: MV887011A_WCDMA_0002
Output port: 2
Spectrum reverse: None

1. Set operation mode to Normal.
   `:SOURce:GPRF:GENerator:MODE NORMAL`
2. Set modulation (any waveform).
   `:SOURce:GPRF:GENerator:BBMode ARB`
3. Set frequency to 1800 MHz.
   `:SOURce:GPRF:GENerator:RFSettings:FREQuency 1800MHZ`
4. Set level to –20 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel -20`
5. Query whether output level within assured level accuracy.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel:SETTing?`
6. Set spectrum reverse to No.
   `:SOURce:GPRF:GENerator:RFSettings:DM:POLarity NORMAL`
7. Set Port 2 to Output.
   `:ROUte:PORT:CONNect:DIRection PORT1, PORT2`
8. Delete all files in memory.
   `:SOURce:GPRF:GENerator:ARB:WAVeform:DELete:ALL`
9. Read files in memory.
   `:SOURce:GPRF:GENerator:ARB:FILE:LOAD "MV887011A_WCDMA_0002"`
10. Query file read status.
    `:SOURce:GPRF:GENerator:ARB:FILE:LOAD:STATus?`
11. Set pattern for group No. 1 from multiple files saved in memory.
    `:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect "MV887011A_WCDMA_0002",1,1`
12. Output signal to port.
    `:SOURce:GPRF:GENerator:STATe ON`
13. Stop signal output.
    `:SOURce:GPRF:GENerator:STATe OFF`

4.3.3 Waveform pattern synchronization among modules

The MU887000A can output waveform patterns of multiple modules in synchronization.
Two types of synchronization modes are available as described below.

- Synchronization with the external trigger timing.
  Waveform patterns are output at the timing when trigger is input to Trigger Connector 1 on the rear panel.

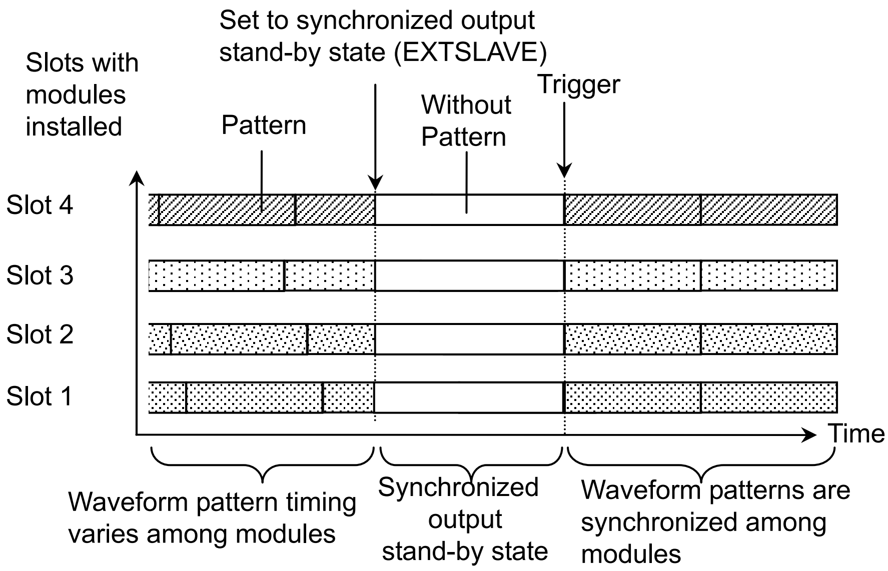


Figure 4.3.3-1 Synchronization with the Timing of External Trigger

- Synchronization with the waveform pattern of the Slot 1 module.
  The modules of Slot 2 to 4 output patterns at the timing when a command is sent to the Slot 1 module.

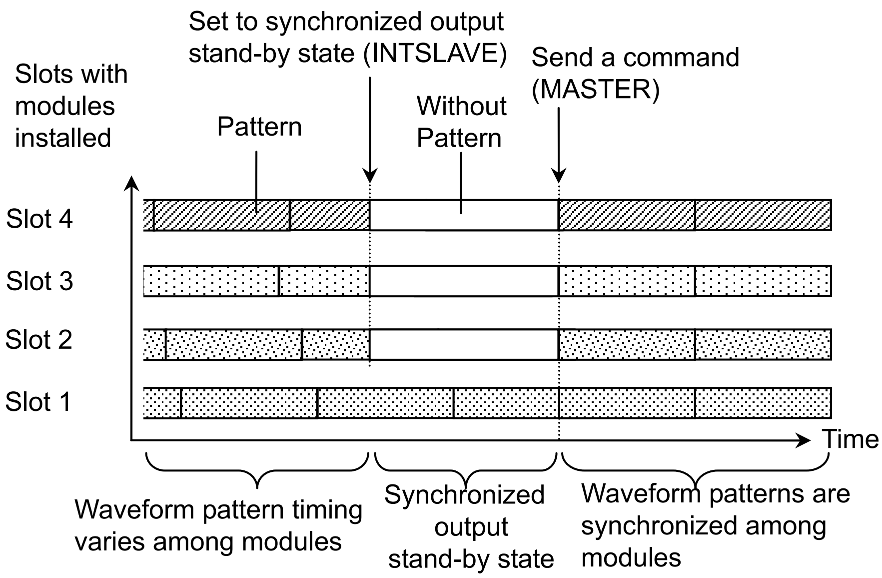


Figure 4.3.3-2 Synchronization with Waveform Pattern of Slot 1 Module

Use the following commands for setting waveform pattern synchronization among modules.

- Synchronization mode
  `:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect`
- Confirmation of synchronized output stand-by state
  `:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:STATe`
- Cancel of synchronized output stand-by state
  `:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:CANCel`

First, set synchronization mode for the module.

MASTER: Reference for waveform pattern timing of other modules

INTSLAVE: Synchronizes the waveform pattern timing with that of the Slot 1 module.

EXTSLAVE: Synchronizes the waveform pattern timing with external trigger.

Available synchronization modes differ among the slots.

Table 4.3.3-1 Available Synchronization Modes

Synchronization Mode \ Slot	Slot 1	Slot 2	Slot 3	Slot 4
MASTER	✓	–	–	–
INTSLAVE	–	✓	✓	✓
EXTSLAVE	✓	✓	✓	✓

For the following cases, the waveform patterns cannot be synchronized with that of the Slot 1 module.

- A module is not installed to Slot 1.
- The synchronization mode for Slot 1 is set to EXTSLAVE.

When set to INTSLAVE or EXTSLAVE, a module is set to synchronization output stand-by state. To check the status, use the following command.
:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:STATe

When the Slot 1 module is set to MASTER, the modules set to INTSLAVE synchronize with Slot 1 and output waveform patterns. The modules are released from synchronized output stand-by state.

When the module is set to EXTSLAVE, it outputs waveform pattern synchronized with the trigger input timing to Trigger Connector 1 on the rear panel of the MT8870A.
When the trigger is input to Trigger Connector 1 after the modules are released from synchronized output stand-by state, the waveform patterns are not synchronized.

To cancel synchronized output stand-by state, use either of the commands below.
:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:CANCel
*RST

Note:

The waveform pattern set by :SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect is changed to that set by :SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect.

When the synchronization mode is specified to a module, the trigger setting on the rear panel of the MT8870A is changed.

Table 4.3.3-2 Trigger Setting at Pattern Synchronization

Synchronization Mode / Trigger Setting Command	MASTER	INTSLAVE	EXTSLAVE
:ROUTe:TRIGger:PORT	OUTPUT	INPUT	INPUT*
:ROUTe:TRIGger:OUTPut:SOURce	SGSYNC	(No Change)	(No Change)
:ROUTe:TRIGger:INPut:SOURce	(No Change)	SLOT1	TRGIN1

*: The trigger settings of the slots set to EXTSLAVE and Slot 1 are changed to INPUT.

Set the two units of the MT8870A as the figure below. When setting the Slot 1 module of the MT8870A-1 to MASTER, the waveform patterns of the 8 modules synchronize.

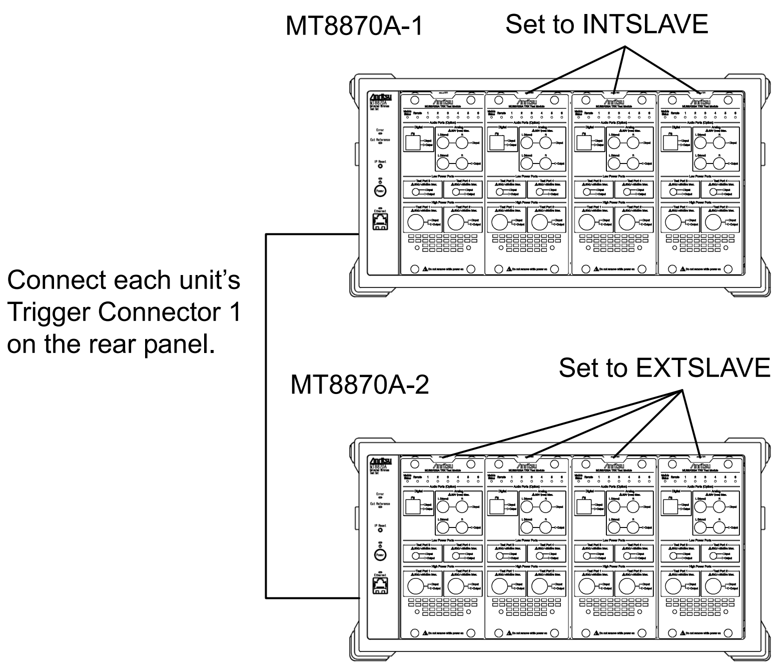


Figure 4.3.3-3 Waveform Pattern Synchronization Among 8 Modules

4

Vector Signal Generator

The steps to synchronize the patterns of the modules in the setting of Figure 4.3.3-3 are described below.

1. Set the operation mode of all the modules to Normal mode.
   :SOURce:GPRF:GENerator:MODE NORMAL
2. Set the frequency of all the modules to 1800 MHz.
   :SOURce:GPRF:GENerator:RFSettings:FREQuency 1800MHZ
3. Set the level of all the modules to –20 dBm.
   :SOURce:GPRF:GENerator:RFSettings:LEVel -20
4. Query each module if its output level is within the guaranteed accuracy.
   :SOURce:GPRF:GENerator:RFSettings:LEVel:SETTing?
5. Set the spectrum invert of all the modules to None.
   :SOURce:GPRF:GENerator:RFSettings:DM:POLarity NORMAL
6. Set Port 1 of all the modules to output.
   :ROUTe:PORT:CONNect:DIRection PORT1, PORT1
7. Delete all the files in the memory of all the modules.
   :SOURce:GPRF:GENerator:ARB:WAVeform:DELete:ALL
8. Load a file into the memory of all the modules.
   :SOURce:GPRF:GENerator:ARB:FILE:LOAD "MV887012A_GSM_0002"
9. Query the file loading status of all the modules.
   :SOURce:GPRF:GENerator:ARB:FILE:LOAD:STATus?
10. Set all the modules to ARB (with modulation).
    :SOURce:GPRF:GENerator:BBMode ARB
11. Output signal to the ports of all the modules.
    :SOURce:GPRF:GENerator:STATe ON
12. Set all the modules of the MT8870A-2 as below.

Synchronization mode:	Synchronizes with external trigger (EXTSLAVE)
Waveform pattern:	Pattern 1 of "MV887012A_GSM_0002"

    The waveform pattern output of the MT8870A-2 module stops at this time.
    :SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect EXTSLAVE,"MV887012A_GSM_0002",1,1
13. Set the modules from Slot 2 to 4 of the MT8870A-1 as below.

Synchronization mode:	Synchronizes with Slot 1 pattern (INTSLAVE)
Waveform pattern:	Pattern 1 of "MV887012A_GSM_0002"

The waveform pattern output to Slot 2 to 4 stops at this time.

:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect INTSLAVE,"MV887012A_GSM_0002",1,1

14. Query all the modules of the MT8870A-2 and the Slot 2 to 4 modules of the MT8870A-1 to check if they are in waveform pattern synchronized output stand-by state.

    :SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:STATe?

15. Set the Slot 1 module of the MT8870A-1 as below.

Synchronization mode:	Master of pattern (MASTER)
Waveform pattern:	Pattern 1 of "MV887012A_GSM_0002"

    When sending this command, the waveform patterns of all the modules are output in synchronization.

    :SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect MASTER,"MV887012A_GSM_0002",1,1

4.4 Sequence Mode

4.4.1 Operation details

The Sequence mode generates RF signals with frequency, level, and modulation waveform changing according to the order described in the sequence table.

Set the number of time steps to generate the RF signals. Standards such as W-CDMA, GSM, and EVDO are supported with changed time step.

The timing to switch RF signals is selected from Time or Trigger. Trigger can be selected from a signal input at the rear panel Trigger connector, a trigger described in the sequence table, the output signal level, or waveform group number.

Since the frequency and level sequence as well as the modulation waveform sequence can be set independently, the modulation waveform can be controlled asynchronously with respect to the frequency and level.

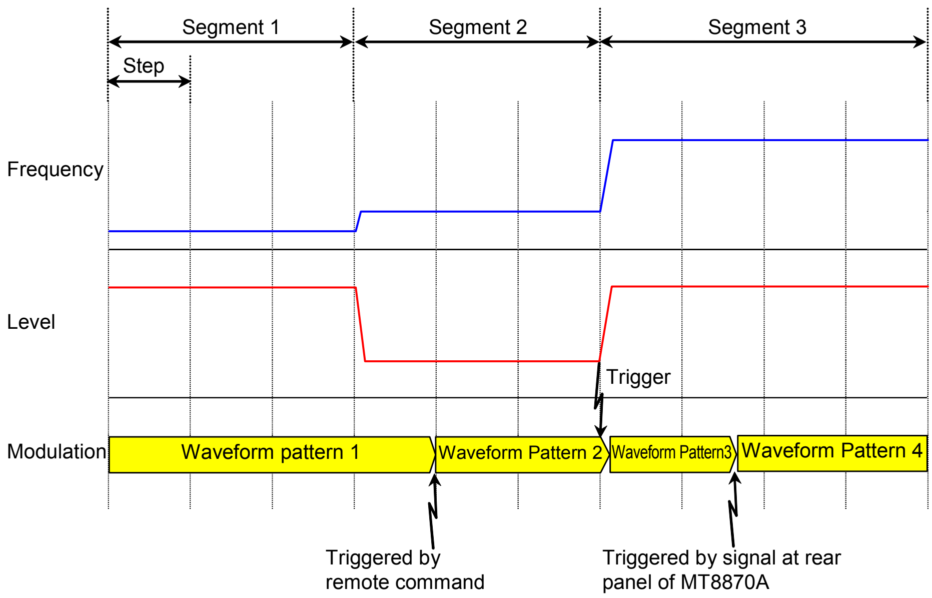


Figure 4.4.1-1 Sequence Mode Execution

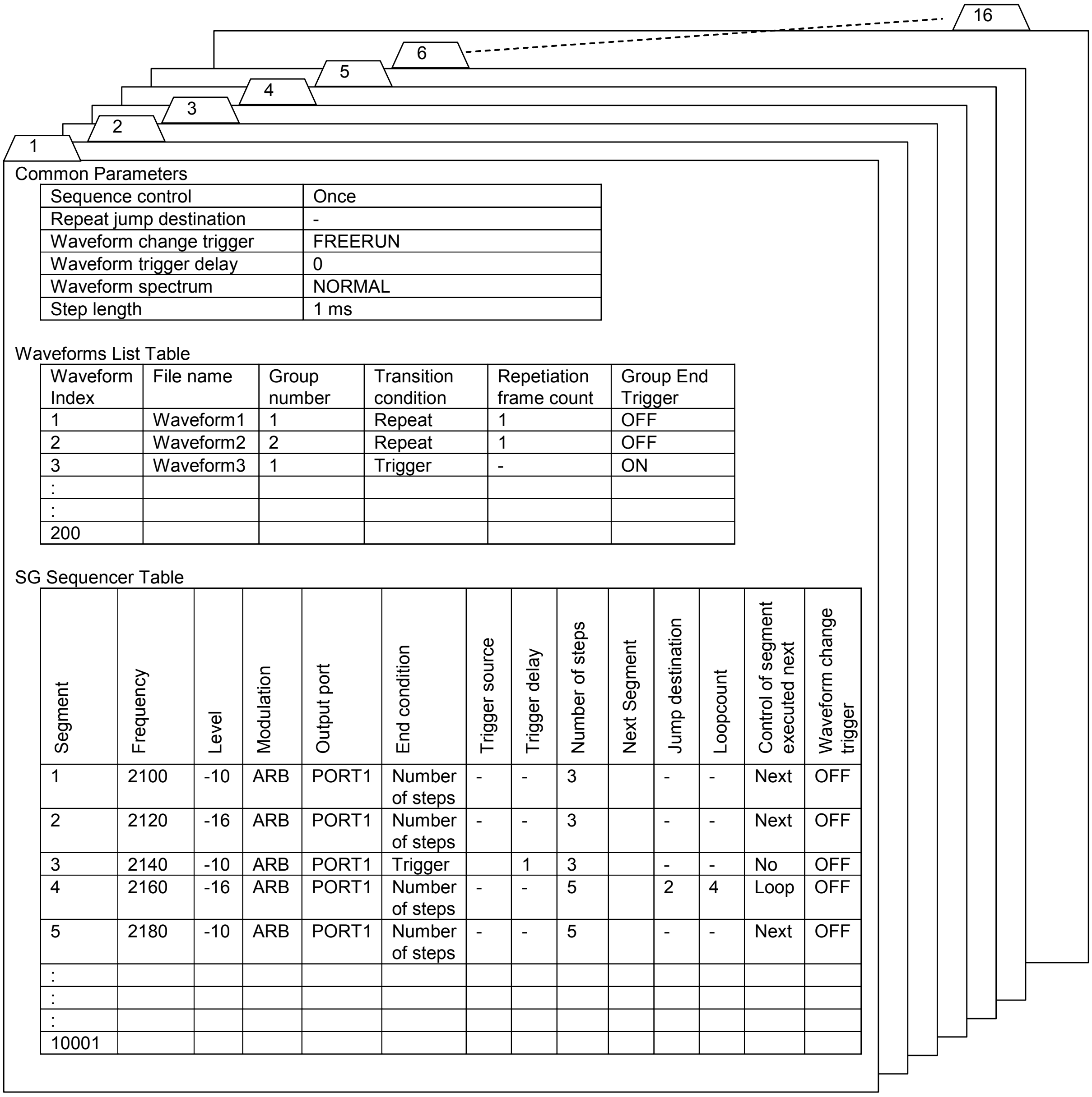
⚠ CAUTION

It takes about 500 μs for the signal to stabilize after RF switching. Start the DUT test after allowing 500 μs to elapse.

4.4.2 Sequence table configuration

The sequence mode generates RF signals according to the description in up to 16 sequence tables.

The sequence table is composed of common parameters, waveform list table, and SG (Signal Generator) sequencer table.



Common Parameters

Sequence control	Once
Repeat jump destination	-
Waveform change trigger	FREERUN
Waveform trigger delay	0
Waveform spectrum	NORMAL
Step length	1 ms

Waveforms List Table

Waveform Index	File name	Group number	Transition condition	Repetition frame count	Group End Trigger
1	Waveform1	1	Repeat	1	OFF
2	Waveform2	2	Repeat	1	OFF
3	Waveform3	1	Trigger	-	ON
:					
:					
200					

SG Sequencer Table

Segment	Frequency	Level	Modulation	Output port	End condition	Trigger source	Trigger delay	Number of steps	Next Segment	Jump destination	Loopcount	Control of segment executed next	Waveform change trigger
1	2100	-10	ARB	PORT1	Number of steps	-	-	3		-	-	Next	OFF
2	2120	-16	ARB	PORT1	Number of steps	-	-	3		-	-	Next	OFF
3	2140	-10	ARB	PORT1	Trigger		1	3		-	-	No	OFF
4	2160	-16	ARB	PORT1	Number of steps	-	-	5		2	4	Loop	OFF
5	2180	-10	ARB	PORT1	Number of steps	-	-	5		-	-	Next	OFF
:													
:													
:													
10001													

Figure 4.4.2-1 Sequence Table Configuration

4

Vector Signal Generator

Common parameters

Set the SG sequencer control method to Common parameters.

The Common parameters setting items are as follows:

- Sequence control
  Sequence handling method

Single:	Execute once
Continuous:	Execution repeated

- Repeat jump destination
  Start segment number for second and later execution for Continuous (repeated execution) sequence control
- Waveform change trigger
  Selection of trigger used in waveforms list table

External trigger:	Signal from Trigger connector on rear panel
Level:	Signal level
SG:	Trigger generated at SG sequencer processing
Manual:	Trigger by command

- Waveform trigger delay
  Time difference between trigger generation and modulation waveform change (ms)
- Waveform spectrum
  Baseband signal IQ inversion On/Off
- Step length
  Step length set with SG sequencer

Set time:	0.5 to 80 ms
GSM Slot time:	15/26 ms
GSM Frame time:	60/13 ms
WCDMA Slot time:	10/15 ms
EVDO Slot time:	80/48 ms
EVDO Frame time:	80/3 ms

- Registered number of waveforms
  Number of waveforms registered in waveforms list table (read only)

The following commands are used to set and query common parameters:

- Sequence control
  `:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:REPetition`
- Repeat jump destination
  `:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:GOTO`
- Waveform change trigger
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce`
- Waveform trigger delay
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:DELay`
- Waveform spectrum
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:DM:POLarity`
- Step length
  `:SOURce:GPRF:GENerator:SEQuence:GENeral:RX:GENeral:STEP:LENGth`
- Registered number of waveforms (read only)
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:PCOunt`

Waveforms list table

The waveforms list table settings are as follows:

- Waveform Index
  Number indicating waveform pattern storage location
- File name
  Waveform file name
- Group number
  Group numbers of waveform patterns in a waveform file.
- Transition condition
  Condition for switch to next waveform pattern

None:	No switching
Repeat:	At repetition frame count
Trigger:	At trigger

- Repetition frame count
  Repetition count for waveform pattern
- Group End Trigger
  On/Off of trigger generated when waveform file of same group number ends (Group End Trigger)

The following commands are used for setting and querying the waveforms list table:

- Selecting waveform pattern
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:SELect`
- Deleting waveform pattern
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:DELete`
- Transition condition
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:ENDCondition`
- Repetition count of waveform pattern
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:IREPetition`
- Group End Trigger
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:GETRigger`

SG sequencer table

The SG sequencer table settings are as follows:

- Segment
  Number of setting item
- Frequency
  Output frequency
- Level
  Output level (dBm)
- Modulation
  Modulation On/Off
- Output port
  RF signal output panel connector number (Port 1 to 4)
- End condition
  Condition for ending executing segment

Step:	At time corresponding to number of steps
Trigger:	At trigger

- Trigger source
  Selection of trigger used with SG sequencer

External trigger:	Signal at Trigger connector on rear panel
Level:	Signal level
Waveform group:	Trigger at waveform pattern change
Waveform marker:	Waveform Marker 1 to 3

- Trigger delay
  Time difference between trigger generation and segment processing end (ms)
- Number of steps
  Segment continuous time
- Next segment
  Control of segment executed next

Next segment number:	Segment number 1 larger than current segment number
Loop:	Segment number set at jump destination

- Jump destination
  Segment number executed when loop set at "Next segment"
- Loop count
  Number of jumps executed when loop set at "Next segment"
- Waveform change trigger
  On/Off of trigger generation to control waveforms list table execution

The SQ sequencer table setting commands are described below.

- Frequency
  :SOURce:GPRF:GENerator:SEQuence:RX:FREQuency
  :SOURce:GPRF:GENerator:SEQuence:RX:FREQuency:ALL
- Level
  :SOURce:GPRF:GENerator:SEQuence:RX:LEVel
  :SOURce:GPRF:GENerator:SEQuence:RX:LEVel:ALL
- Modulation
  :SOURce:GPRF:GENerator:SEQuence:RX:BBMode
  :SOURce:GPRF:GENerator:SEQuence:RX:BBMode:ALL
- Output port
  :SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe
  :SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe:ALL
- End condition
  :SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition
  :SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition:ALL
- Trigger source
  :SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce
  :SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce:ALL
- Trigger delay
  :SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay
  :SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay:ALL
- Step count
  :SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt
  :SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt:ALL
- Next segment
  :SOURce:GPRF:GENerator:SEQuence:RX:NSLControl
  :SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:ALL
- Jump destination
  :SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment
  :SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment
  :ALL
- Loop count
  :SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt
  :SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt:ALL
- Waveform pattern change trigger
  :SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger
  :SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger:ALL

4.4.3 File operations

The data of SG sequencer table (SG sequencer data) can save and load a file using the following commands.

- SG sequencer data save
  `:SOURce:GPRF:GENerator:SEQuence:STORe`
- SG sequencer data load
  `:SOURce:GPRF:GENerator:SEQuence:LOAD`

Waveform file information can be obtained using the following commands.

Waveform files are loaded from the storage media to the memory using the following commands.

- Waveform file loading
  `:SOURce:GPRF:GENerator:ARB:FILE:LOAD`
- Waveform file loading stop
  `:SOURce:GPRF:GENerator:ARB:FILE:LOAD:CANCel`
- Query waveform loading status
  `:SOURce:GPRF:GENerator:ARB:FILE:LOAD:STATus`

Use the following command to associate the waveform file with the waveforms list table.

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:SELect
```

When the memory free space is small, delete waveform files from memory using the following commands.

```
:SOURce:GPRF:GENerator:ARB:WAVeform:DELete
:SOURce:GPRF:GENerator:ARB:WAVeform:DELete:ALL
```

4.4.4 Setting item error check

Check whether settings have errors for the following sequence table items.

- Input level
- Output level
- Step count
- Capture memory amount

Query whether the sequence table has an error using the following command.

```
:SOURce:GPRF:GENerator:SEQuence:RX:ERRor
```

Response values and error contents are described in the following table. When there are multiple errors, the response is the sum of the error values.

Table 4.4.4-1 Sequence Measurement Conditions

Value	Error contents
1	Loop nesting occurs
2	Loop jump destination out of execution sequence range
4	Loop set with trigger waiting segment
8	Level out of setting range
16	Unusable waveform pattern set

The sequence cannot be executed if an error is found in the sequence table. However, if the error value is 8, the sequence can be executed by specifying the force mode :SOURce:GPRF:GENerator:SEQuence:EXECute. In this case, the signal is output at the highest level possible if the set value exceeds the actual upper limit and at the lowest level possible if the set value goes under the actual lower limit.

4.4.5 Sequence execution

The following commands are used to control sequence execution:

- Set number and order of executed sequences
  `:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern`
- Start sequence
  `:SOURce:GPRF:GENerator:SEQuence:EXECute`
  `:SOURce:GPRF:GENerator:SEQuence:RLISt`
- Query sequence status
  `:SOURce:GPRF:GENerator:SEQuence:STATus`
- Execute manual trigger
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:MTEXecute`
- Stop sequence
  `:SOURce:GPRF:GENerator:SEQuence:CANCel`

After the sequence table is edited, execute the sequence using the following procedure.

1. Set operation mode to Sequence Mode.
   :SOURce:GPRF:GENerator:MODE SEQUENCE
2. Set sequence tables to execute and their order.
   :SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern
3. Use following command when changing mode to Normal after sequence finished.
   :SOURce:GPRF:GENerator:SEQuence:REINitialization ON
4. Send following command to start sequence.
   :SOURce:GPRF:GENerator:SEQuence:EXECute
5. Use following command to check whether sequence executing.
   :SOURce:GPRF:GENerator:SEQuence:STATus
6. Send following command to generate trigger when waveform change trigger set to Manual.
   :SOURce:GPRF:GENerator:SEQuence:WAVeform:MTEXecute
7. Send following command to cancel sequence.
   :SOURce:GPRF:GENerator:SEQuence:CANCel
   The sequence can be cancelled even when the Normal Mode is set by sending :SOURce:GPRF:GENerator:MODE NORMAL.

4.4.6 Waveform pattern synchronization among modules

The MU887000A can output waveform patterns of multiple modules in synchronization when executing sequence.

The two types of synchronization modes are available in Sequence mode as in Normal mode.

- Synchronization with the timing of external trigger
- Synchronization with the waveform pattern of the Slot 1 module

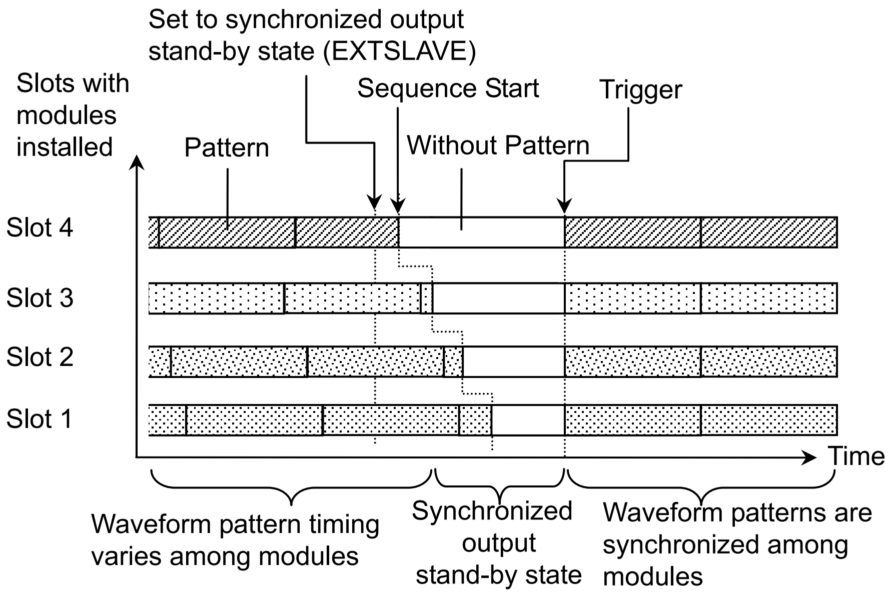


Figure 4.4.6-1 Synchronization with External Trigger Timing

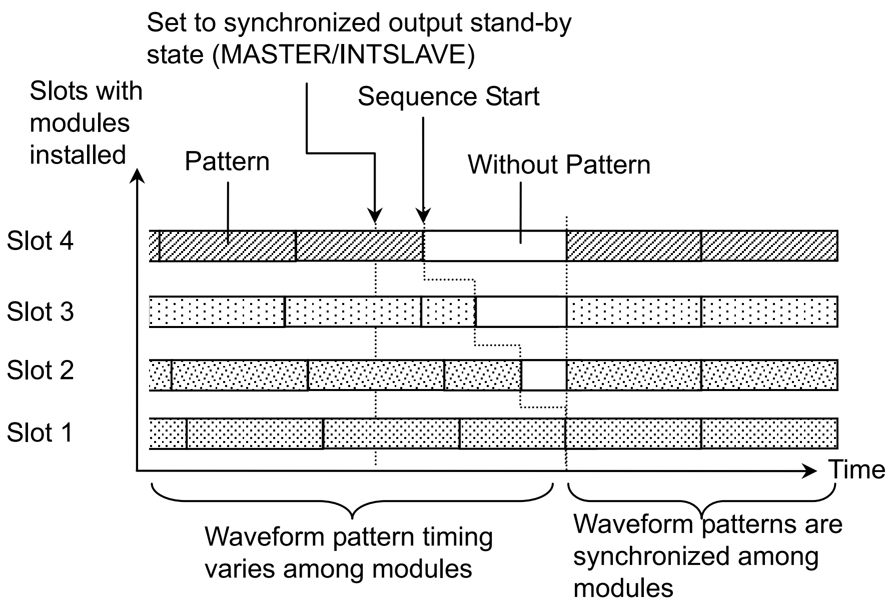


Figure 4.4.6-2 Synchronization with Waveform Pattern of Slot 1 Module

Use the following commands for setting waveform pattern synchronization among modules.

- Synchronization mode
  `:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize`
- Confirmation of synchronized output stand-by state
  `:SOURce:GPRF:GENerator:SEQuence:WAVeform:SYNChronize:STARt:STATe`
- Cancel of synchronized output stand-by state
  `:SOURce:GPRF:GENerator:SEQuence:CANCel`

First, set synchronization mode for the modules.

NONE:	No synchronization of waveform patterns among modules.
MASTER:	Reference for waveform pattern timing of other modules.
INTSLAVE:	Sets the waveform pattern timing to the Slot 1 module.
EXTSLAVE:	Sets the waveform pattern timing to external trigger.

Available synchronization modes differ among the slots.

Table 4.4.6-1 Available Synchronization Modes

Synchronization Mode \ Slot	Slot 1	Slot 2	Slot 3	Slot 4
NONE	✓	✓	✓	✓
MASTER	✓	–	–	–
INTSLAVE	–	✓	✓	✓
EXTSLAVE	✓	✓	✓	✓

For the following cases, the waveform patterns cannot be synchronized with that of the Slot 1 module.

- A module is not installed to Slot 1.
- The synchronization mode for Slot 1 is set to NONE or EXTSLAVE.

When set to INTSLAVE or EXTSLAVE, the module falls in synchronized output stand-by state. To confirm the state, use the command below.

:SOURce:GPRF:GENerator:SEQuence:WAVeform:SYNChronize:STARt:STATe

To start synchronization, send either of the following commands.
:SOURce:GPRF:GENerator:SEQuence:EXECute
:SOURce:GPRF:GENerator:SEQuence:RLISt

The module set to INTSLAVE outputs waveform pattern synchronized with the Slot 1 module.
The module set to EXTSLAVE outputs waveform pattern synchronized with the trigger input timing to Trigger Connector 1. The waveform patterns are synchronized only at the first trigger input. They are not synchronized at the second trigger and after.

To cancel synchronized output stand-by state, use either of the commands below.
:SOURce:GPRF:GENerator:SEQuence:CANCel
*RST

Note:

When using *RST, the operation mode is changed from Sequence to Normal.

When the synchronization mode is specified to a module, the trigger setting on the rear panel of the MT8870A is changed.

Table 4.4.6-2 Trigger Setting at Pattern Synchronization

Trigger Setting Command \ Synchronization Mode	NONE	MASTER	INTSLAVE	EXTSLAVE
:ROUTe:TRIGger:PORT	(No Change)	OUTPUT	INPUT	INPUT*
:ROUTe:TRIGger:OUTPut:SOURce	(No Change)	SGSYNC	(No Change)	(No Change)
:ROUTe:TRIGger:INPut:SOURce	(No Change)	(No Change)	SLOT1	TRGIN1

*: The trigger setting of the slots set to EXTSLAVE and Slot 1 are changed to INPUT.

The steps to synchronize the waveform patterns of all the modules with that of the Slot 1 module of the MT8870A are described below. Edit the sequence table of each module in advance.

1. Set the operation mode of all the modules to Sequence mode.

   :SOURce:GPRF:GENerator:MODE SEQUENCE
2. Set the sequence table and order of all the modules by the commands below.

   :SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern
3. Set the modules from Slot 2 to 4 as below.
   Synchronization mode: Synchronizes with Slot 1 pattern (INTSLAVE)

   :SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize INTSLAVE
4. Execute the sequence of the modules from Slot 2 to 4 as below. The waveform pattern is not output at this timing.

   :SOURce:GPRF:GENerator:SEQuence:EXECute
5. Query the modules from Slot 2 to 4 if they are in synchronized output stand-by state.

   :SOURce:GPRF:GENerator:SEQuence:WAVeform:SYNChronize:STARt:STATe?
6. Set the module of Slot 1 as below.
   Synchronization mode: Master of pattern (MASTER)

   The waveform pattern is not synchronized among modules at this timing.

   :SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize MASTER
7. Execute the sequence of the Slot 1 module as below. When the Slot 1 module starts sequence, the waveform patterns of all the modules are synchronized.

   :SOURce:GPRF:GENerator:SEQuence:EXECute

Chapter 5 SCPI Command Reference

This chapter describes the details of SCPI commands.

5.1 List of Commands

5.1.1 Fundamental operation commands

Calibration

Function	Command	Query	Response
Band Calibration Error Count	-----	:CALCulate:CALibration:BAND:ERRor:COUNt?	<count>
BAND Calibration Error Last Time	-----	:CALCulate:CALibration:BAND:ERRor:LASTtime?	<year>,<month>,<day>,<hour>,<min>
BAND Calibration Last Time	-----	:CALCulate:CALibration:BAND:LASTtime?	<year>,<month>,<day>,<hour>,<min>
Result of Band Calibration	-----	:CALCulate:CALibration:BAND:RESult?	<Result>
Execute BAND Calibration	:CALCulate:CALibration:BAND:STARt	-----	-----
BAND Calibration Temperature	:CALCulate:CALibration:BAND:STARt:TEMPerature <temperature>	-----	-----
Elapsed Time after Full Calibration	-----	:CALCulate:CALibration:ELAPsed:TIME?	<time>
Full Calibration Error Count	-----	:CALCulate:CALibration:FULL:ERRor:COUNt?	<count>
FULL Calibration Error Last Time	-----	:CALCulate:CALibration:FULL:ERRor:LASTtime?	<year>,<month>,<day>,<hour>,<min>
FULL Calibration Last Time	-----	:CALCulate:CALibration:FULL:LASTtime?	<year>,<month>,<day>,<hour>,<min>
Result of Full Calibration	-----	:CALCulate:CALibration:FULL:RESult?	<Result>

Calibration (Cont'd)

Function	Command	Query	Response
Execute FULL Calibration	:CALCulate:CALibration:FULL:STARt	-----	-----
Initial Calibration	:CALCulate:CALibration:INITial	-----	-----
Temperature at Full Calibration Execution	-----	:CALCulate:CALibration:TEMPerature?	<temp>

Communications

Function	Command	Query	Response
GPIB Address	:SYSTem:COMMunicate:GPIB:ADDR <gp_address>	:SYSTem:COMMunicate:GPIB:ADDR?	<gp_address>
Delimiter	:SYSTem:COMMunicate:GPIB[:SELF]:DELimiter <code>	:SYSTem:COMMunicate:GPIB[:SELF]:DELimiter?	<code>
Terminator of Message	:SYSTem:COMMunicate:GPIB[:SELF]:TERMinator <code>	:SYSTem:COMMunicate:GPIB[:SELF]:TERMinator?	<code>
Domain Name	:SYSTem:COMMunicate:NET:DNS:DOMain <name>	:SYSTem:COMMunicate:NET:DNS:DOMain?	<name>
Host Name	:SYSTem:COMMunicate:NET:DNS:HOST <host>	:SYSTem:COMMunicate:NET:DNS:HOST?	<host>
MAC Address	-----	:SYSTem:COMMunicate:NET:HWADdress?	<macaddr>
IP Version	:SYSTem:COMMunicate:NET:IPVersion <ipver>	:SYSTem:COMMunicate:NET:IPVersion?	<ipver>
IPv4 Type	:SYSTem:COMMunicate:NET:IPVFour:ADDRess:TYPE <ipv4type>	:SYSTem:COMMunicate:NET:IPVFour:ADDRess:TYPE?	<ipv4type>
Get Current Default Gateway	-----	:SYSTem:COMMunicate:NET:IPVFour:CURRent:GIP?	<gateway>

Communications (Cont'd)

Function	Command	Query	Response
Get Current IPv4 Address	-----	:SYSTem:COMMunicate:NET:IPVFour:CURRent:IPADdress?	<num>,<ipaddr>
Get Current Subnet Mask	-----	:SYSTem:COMMunicate:NET:IPVFour:CURRent:SMASk?	<mask>
DNS Server Auto	:SYSTem:COMMunicate:NET:IPVFour:DNS:AUTO:ENABle <switch>	:SYSTem:COMMunicate:NET:IPVFour:DNS:AUTO:ENABle?	<switch>
DNS Primary Address	:SYSTem:COMMunicate:NET:IPVFour:DNS:PRIMary <ip4addr>	:SYSTem:COMMunicate:NET:IPVFour:DNS:PRIMary?	<ip4addr>
DNS Secondary Address	:SYSTem:COMMunicate:NET:IPVFour:DNS:SECondary <ip4addr>	:SYSTem:COMMunicate:NET:IPVFour:DNS:SECondary?	<ip4addr>
Default Gateway	:SYSTem:COMMunicate:NET:IPVFour:STATic:GIP <gateway>	:SYSTem:COMMunicate:NET:IPVFour:STATic:GIP?	<gateway>
IPv4 Address	:SYSTem:COMMunicate:NET:IPVFour:STATic:IPADdress <ipv4_addr1>,<ipv4_addr2>,<ipv4_addr3>,<ipv4_addr4>	:SYSTem:COMMunicate:NET:IPVFour:STATic:IPADdress?	<ipv4_addr1>,<ipv4_addr2>,<ipv4_addr3>,<ipv4_addr4>
Subnet Mask	:SYSTem:COMMunicate:NET:IPVFour:STATic:SMASk <mask>	:SYSTem:COMMunicate:NET:IPVFour:STATic:SMASk?	<mask>
IPv6 Type	:SYSTem:COMMunicate:NET:IPVSix:ADDRess:TYPE <ipv6type>	:SYSTem:COMMunicate:NET:IPVSix:ADDRess:TYPE?	<ipv6type>
Get Default Router Address	-----	:SYSTem:COMMunicate:NET:IPVSix:CURRent:DEFRouter?	<ip6addr>
Get Current IPv6 Address and Prefix	-----	:SYSTem:COMMunicate:NET:IPVSix:CURRent:IPADdress?	<num>,<ipaddr>
Get Link Local Address	-----	:SYSTem:COMMunicate:NET:IPVSix:CURRent:LOCaladdress?	<ip6addr>
DNS Server Auto	:SYSTem:COMMunicate:NET:IPVSix:DNS:AUTO:ENABle <switch>	:SYSTem:COMMunicate:NET:IPVSix:DNS:AUTO:ENABle?	<switch>

Communications (Cont'd)

Function	Command	Query	Response
DNS Primary Address	:SYSTem:COMMunicate:NET:IPVSix:DNS:PRIMary <ip6addr>	:SYSTem:COMMunicate:NET:IPVSix:DNS:PRIMary?	<ip6addr>
DNS Secondary Address	:SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary <ip6addr>	:SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary?	<ip6addr>
Default Router	:SYSTem:COMMunicate:NET:IPVSix:STATic:DEFRouter <ip6addr>	:SYSTem:COMMunicate:NET:IPVSix:STATic:DEFRouter?	<ip6addr>
IPv6 Address	:SYSTem:COMMunicate:NET:IPVSix:STATic:IPADdress <ipv6_addr1>,<ipv6_addr2>,<ipv6_addr3>,<ipv6_addr4>	:SYSTem:COMMunicate:NET:IPVSix:STATic:IPADdress?	<ipv6_addr1>,<ipv6_addr2>,<ipv6_addr3>,<ipv6_addr4>
Network Restart	:SYSTem:COMMunicate:NET:RESTart	-----	-----
Raw Socket Port Number	:SYSTem:COMMunicate:NET:PORT:SETTing <port>	:SYSTem:COMMunicate:NET:PORT:SETTing?	<port>

FTP

Function	Command	Query	Response
FTP User List	-----	:SYSTem:COMMunicate:NET:FTP:USER:CATalog?	<name>
Change FTP User Name	:SYSTem:COMMunicate:NET:FTP:USER:CHANge:NAME <before>,<after>	-----	-----
Change FTP User Password	:SYSTem:COMMunicate:NET:FTP:USER:CHANge:PASS <name>,<before>,<after>	-----	-----

Information

Function	Command	Query	Response
Log Clear	:MMEMory:LOG:CLEar	-----	-----
Log Reading	-----	:MMEMory:LOG:LOAD? [<size>]	<log>
Log Size Reading	-----	:MMEMory:LOG:SIZE?	<log>
Module Model Number	-----	:MAINtenance:SYSTem:INFormation:DEVice:MODel?	<model>
Main Frame Model Number	-----	:MAINtenance:SYSTem:INFormation:MAINframe:MODel?	<model>
License Information	-----	:SYSTem:BASE:LICense:INFormation? <model>	<kind>,<days>
Module Information	-----	:SYSTem:INFormation? <index>	-----
Module Serial Number	-----	:SYSTem:INFormation:DEVice:ID?	<serial>
Module Power ON Count	-----	:SYSTem:INFormation:POWeron:COUNt?	<num>
Main Frame Information	-----	:SYSTem:INFormation:MAINframe? <index>	-----
Main Frame Serial Number	-----	:SYSTem:INFormation:MAINframe:DEVice:ID?	<serial>
Chassis Power ON Count	-----	:SYSTem:INFormation:MAINframe:POWeron:COUNt?	<num>
Main Frame Running Time Count Read	-----	:SYSTem:INFormation:MAINframe:RTIMe?	<time>
Firmware Version	-----	:SYSTem:INFormation:MAINframe:PACKage:VERSion?	<ver>
Type Name of Software	-----	:SYSTem:INFormation:MAINframe:SOFTware?	<n>,<model>
Option Number	-----	:SYSTem:INFormation:MAINframe:SOFTware:OPTion? <model>	<n>,<num>

Information (Cont'd)

Function	Command	Query	Response
Type Name of Waveform	-----	:SYSTem:INFormation:MAINframe:WAVeform?	<n>,<model>
Main Frame Temperature	-----	:SYSTem:INFormation:MAINframe:TEMPerature?	<temp>
Count of Module Insertion	-----	:SYSTem:INFormation:PLUG:COUNt?	<num>
Module Run Time Count Read	-----	:SYSTem:INFormation:RTIMe?	<time>
Module Slot Number	-----	:SYSTem:INFormation:SLOT?	<slot>
Module Temperature	-----	:SYSTem:INFormation:TEMPerature?	<temp>
Package Version	-----	:SYSTem:VERSion?	<ver>
Drive Size Reading	-----	:MMEMory:SIZE? [<target>]	<total>,<blank>

Operation Status Register

Function	Command	Query	Response
Operation Status Condition Register Query	-----	`:STATus:OPERation:CONDition?`	`<oscr>`
Operation Status Enable Register	`:STATus:OPERation:ENABle <oser>`	`:STATus:OPERation:ENABle?`	`<oser>`
Generator Operation Status Register Query	-----	`:STATus:OPERation:GENerator[:EVENt]?`	`<gosr>`
Generator Operation Status Condition Register Query	-----	`:STATus:OPERation:GENerator:CONDition?`	`<goscr>`
Generator Operation Status Enable Register	`:STATus:OPERation:GENerator:ENABle <goser>`	`:STATus:OPERation:GENerator:ENABle?`	`<goser>`
Generator Operation Status Negative Transition Filter	`:STATus:OPERation:GENerator:NTRansition <gosr_ntf>`	`:STATus:OPERation:GENerator:NTRansition?`	`<gosr_ntf>`
Generator Operation Status Positive Transition Filter	`:STATus:OPERation:GENerator:PTRansition <gosr_ptf>`	`:STATus:OPERation:GENerator:PTRansition?`	`<gosr_ptf>`
Measurement Operation Status Register Query	-----	`:STATus:OPERation:MEASure[:EVENt]?`	`<mosr>`
Measurement Operation Status Condition Register Query	-----	`:STATus:OPERation:MEASure:CONDition?`	`<moscr>`
Measurement Operation Status Enable Register	`:STATus:OPERation:MEASure:ENABle <moser>`	`:STATus:OPERation:MEASure:ENABle?`	`<moser>`
Measurement Operation Status Negative Transition Filter	`:STATus:OPERation:MEASure:NTRansition <mosr_ntr>`	`:STATus:OPERation:MEASure:NTRansition?`	`<mosr_ntr>`
Measurement Operation Status Positive Transition Filter	`:STATus:OPERation:MEASure:PTRansition <mosr_ptr>`	`:STATus:OPERation:MEASure:PTRansition?`	`<mosr_ptf>`
Operation Status Negative Transition Filter	`:STATus:OPERation:NTRansition <osr_ntf>`	`:STATus:OPERation:NTRansition?`	`<osr_ntf>`

Operation Status Register (Cont'd)

Function	Command	Query	Response
Operation Status Positive Transition Filter	:STATus:OPERation:PTRansition <osr_ptf>	:STATus:OPERation:PTRansition?	<osr_ptf>
Operation Status Register Query	-----	:STATus:OPERation[:EVENt]?	<osr>
Preset Status	:STATus:PRESet	-----	-----

Questionable Register

Function	Command	Query	Response
Questionable Status Condition Register Query	-----	:STATus:QUEStionable:CONDition?	<qscr>
Questionable Status Enable Register	:STATus:QUEStionable:ENABle <qser>	:STATus:QUEStionable:ENABle?	<qser>
Generator Questionable Status Condition Register Query	-----	:STATus:QUEStionable:GENerator:CONDition?	<gqscr>
Generator Questionable Status Enable Register	:STATus:QUEStionable:GENerator:ENABle <gqser>	:STATus:QUEStionable:GENerator:ENABle?	<gqser>
Generator Questionable Status Negative Transition Filter	:STATus:QUEStionable:GENerator:NTRansition <gqsr_ntf>	:STATus:QUEStionable:GENerator:NTRansition?	<gqsr_ntf>
Generator Questionable Status Positive Transition Filter	:STATus:QUEStionable:GENerator:PTRansition <gqsr_ptf>	:STATus:QUEStionable:GENerator:PTRansition?	<gqsr_ptf>
Generator Questionable Status Register Query	-----	:STATus:QUEStionable:GENerator[:EVENt]?	<gqsr>
Measurement Questionable Status Condition Register Query	-----	:STATus:QUEStionable:MEASure:CONDition?	<mqscr>
Measurement Questionable Status Enable Register	:STATus:QUEStionable:MEASure:ENABle <mqser>	:STATus:QUEStionable:MEASure:ENABle?	<mqser>
Measurement Questionable Status Negative Transition Filter	:STATus:QUEStionable:MEASure:NTRansition <mqsr_ntr>	:STATus:QUEStionable:MEASure:NTRansition?	<mqsr_ntf>

Questionable Register (Cont'd)

Function	Command	Query	Response
Measurement Questionable Status Positive Transition Filter	:STATus:QUEStionable:MEASure :PTRansition <mqsr_ptr>	:STATus:QUEStionable:MEASure :PTRansition?	<mqsr_ptf>
Measurement Questionable Status Register Query	-----	:STATus:QUEStionable:MEASure [:EVENt]?	<mqsr>
Questionable Status Negative Transition Filter	:STATus:QUEStionable:NTRansition <qsr_ntf>	:STATus:QUEStionable:NTRansition?	<qsr_ntf>
Questionable Status Positive Transition Filter	:STATus:QUEStionable:PTRansition <qsr_ptf>	:STATus:QUEStionable:PTRansition?	<qsr_ptf>
Questionable Status Register Query	-----	:STATus:QUEStionable[:EVENt]?	<qsr>

Reference Clock

Function	Command	Query	Response
Adjust Reference Clock Default	-----	:SYSTem:BASE:REFerence:FREQuency:ADJust:DEFault?	<val>
Frequency Reference	:SYSTem:BASE:REFerence:FREQuency <source>	:SYSTem:BASE:REFerence:FREQuency?	<source>
Frequency Reference	:SYSTem:BASE:REFerence:FREQuency:CONFigure <source>	:SYSTem:BASE:REFerence:FREQuency:CONFigure?	<source>
Adjust Reference Clock	:SYSTem:BASE:REFerence:FREQuency:ADJust <val>	:SYSTem:BASE:REFerence:FREQuency:ADJust?	<val>
Reference Signal Query	-----	:SYSTem:BASE:REFerence:FREQuency:SOURce?	<source>
Reference Signal Query	-----	:SYSTem:BASE:REFerence:FREQuency:CONFigure:SOURce?	<source>

System

Function	Command	Query	Response
Set Date	:SYSTem:DATE <year>,<month>,<day>,<hour>,<minute>,<second>	:SYSTem:DATE?	<year>,<month>,<day>,<hour>,<minute>,<second>
File List	-----	:MMEMory:CATalog? <path>	<num>,<filelist>
Delete File	:MMEMory:DELete <filepath>	-----	-----
User Area Delete	:MMEMory:DELete:USERarea	-----	-----
License Key Check	-----	:SYSTem:BASE:LICense:CHECk? <filename>	<result>
License Key Install	:SYSTem:BASE:LICense:INSTall	-----	-----
Firmware Update	:SYSTem:BASE:UPDate:FIRMware <target>[,<option>]	:SYSTem:BASE:UPDate:FIRMware?	<result>
Check Firmware Update Package	-----	:SYSTem:BASE:UPDate:FIRMware:CHECk? <filename>	<result>
Beep	:SYSTem:BEEPer <sw>	:SYSTem:BEEPer?	<sw>
Go to Local	:SYSTem:COMMunicate:GTL	-----	-----
System Error	-----	:SYSTem:ERRor?	<code>,<msg>
System All Errors	-----	:SYSTem:ERRor:ALL?	<code>,<msg>
System All Error Codes	-----	:SYSTem:ERRor:CODE:ALL?	<code>
System Error Code	-----	:SYSTem:ERRor:CODE[:NEXT]?	<code>
System Error Count	-----	:SYSTem:ERRor:COUNt?	<num>
Language Selection of Remote Command	:SYSTem:LANGuage <mode>	:SYSTem:LANGuage?	<mode>
System Last Error	-----	:SYSTem:LAST:ERRor?	<msg>
Reboot	:SYSTem:REBoot	-----	-----
System Test	-----	:SYSTem:SELF:TEST?	<status>

System (Cont'd)

Function	Command	Query	Response
System Test Error Last Time	-----	:SYSTem:SELF:TEST:ERRor:LASTtime?	<year>,<month>,<day>,<hour>,<min>
System Test Error Count	-----	:SYSTem:SELF:TEST:ERRor:COUNt?	<count>
Shutdown	:SYSTem:SHUTdown	-----	-----

Interface Setting

Function	Command	Query	Response
Set Direction of RF Connector	:ROUTe:PORT:CONNect:DIRection <input>,<output>	:ROUTe:PORT:CONNect:DIRection?	<input>,<output>
Set In/Out of Trigger Connector	:ROUTe:TRIGger:PORT <trigger>	:ROUTe:TRIGger:PORT?	<trigger>
Select Source of Trigger Input	:ROUTe:TRIGger:INPut:SOURce <source>	:ROUTe:TRIGger:INPut:SOURce?	<source>
Select Source of Trigger Output	:ROUTe:TRIGger:OUTPut:SOURce <source>	:ROUTe:TRIGger:OUTPut:SOURce?	<source>

Loss Correction

Function	Command	Query	Response
List Count in Loss Table	-----	:CALCulate:EXTLoss:TABLe:COUNt? <index>	<num>
Delete Loss Value from Cable Loss List	:CALCulate:EXTLoss:TABLe:DELete [<a>[,<b>]]	-----	-----
Loss Table	:CALCulate:EXTLoss:TABLe:SETTing <index>	:CALCulate:EXTLoss:TABLe:SETTing?	<index>
Register Loss Value in Cable Loss List	:CALCulate:EXTLoss:TABLe:VALue <frequency>,<Port1,2/out>,<Port1,2/in>,<Port3,4/out>[,<Port3,4/in>]	:CALCulate:EXTLoss:TABLe:VALue? <n>	<frequency>,<Port1,2/out>,<Port1,2/in>,<Port3,4/out>,<Port3,4/in>
Register Loss Value into All Cable Loss List	:CALCulate:EXTLoss:TABLe:VALue:ALL <frequency>,<Port1out>,<Port1in>,<Port2out>,<Port2in>,<Port3out>,<Port3in>,<Port4out>,<Port4in>	:CALCulate:EXTLoss:TABLe:VALue:ALL? <n>	<frequency>,<Port1out>,<Port1in>,<Port2out>,<Port2in>,<Port3out>,<Port3in>,<Port4out>,<Port4in>
External Loss On/Off	[:ROUTe]:EXTLoss:TABLe:SWITch <sw>	[:ROUTe]:EXTLoss:TABLe:SWITch?	<sw>

5.1.2 VSG commands

AWGN

Function	Command	Query	Response
AWGN C/N Ratio	:SOURce:GPRF:GENerator:ARB:NOISe:CN <cn_ratio>	:SOURce:GPRF:GENerator:ARB:NOISe:CN?	<cn_ratio>
AWGN Switch	:SOURce:GPRF:GENerator:ARB:NOISe:STATe <on_off>	:SOURce:GPRF:GENerator:ARB:NOISe:STATe?	<on_off>

Baseband

Function	Command	Query	Response
Spectrum Reverse	:SOURce:GPRF:GENerator:RFSettings:DM:POLarity <mode>	:SOURce:GPRF:GENerator:RFSettings:DM:POLarity?	<mode>

Baseband ARB

Function	Command	Query	Response
Waveform File Loading	:SOURce:GPRF:GENerator:ARB:FILE:LOAD <file_name>	:SOURce:GPRF:GENerator:ARB:FILE:LOAD? <file_name>	<status>
Waveform File Loading Cancel	:SOURce:GPRF:GENerator:ARB:FILE:LOAD:CANCel	-----	-----
Waveform Load Status	-----	:SOURce:GPRF:GENerator:ARB:FILE:LOAD:STATus?	<status_res>
Waveform Group Information	-----	:SOURce:GPRF:GENerator:ARB:FILE:PATTern? <file_name>,<group_index>	<group_number>
Waveform Group Count	-----	:SOURce:GPRF:GENerator:ARB:FILE:PATTern:COUNt? <file_name>	<count>
Waveform Pattern name	-----	:SOURce:GPRF:GENerator:ARB:FILE:PATTern:NAME? <file_name>,<group_number>	<pattern_name>

Baseband ARB (Cont'd)

Function	Command	Query	Response
Waveform Version	-----	:SOURce:GPRF:GENerator:ARB:FILE:VERSion? <file_name>	<version>
ARB Memory Loaded Files Count	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:COUNt?	<count>
ARB Memory Defrag	:SOURce:GPRF:GENerator:ARB:WAVeform:DEFRag	-----	-----
Delete Waveform File in ARB Memory	:SOURce:GPRF:GENerator:ARB:WAVeform:DELete <file_name>	-----	-----
Delete All Waveform Files in ARB Memory	:SOURce:GPRF:GENerator:ARB:WAVeform:DELete:ALL	-----	-----
ARB Memory Free Space	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:FREE? <target>	<blank>,<consecutive_blank>,<total>
Waveform Generate for Zero Vector Signal Pattern	:SOURce:GPRF:GENerator:ARB:WAVeform:GENerate:ZVSPattern <file_name>,<sampling_rate>,<sample_num>	-----	-----
ARB Memory Loaded File Name	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:NAME? <numeric_value>	<file_name>
ARB Memory Group Information	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern? <file_name>,<group_index>	<group_number>
ARB Memory Group Count	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:COUNt? <file_name>	<count>

Baseband ARB (Cont'd)

Function	Command	Query	Response
ARB Memory Pattern Gap	:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:GAP <file_name>,<group_number>,<gap_length>	:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:GAP? <file_name>,<group_number>	<gap_length>
ARB Memory Pattern Name	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:NAME? <file_name>,<group_number>	<pattern_name>
ARB Memory Pattern Note	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:NOTE? <file_name>,<group_number>	<note>
ARB Memory Pattern Selection	:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect <file_name>[,<group_number>[,<length>]]	:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect?	<file_name>,<group_number>,<length>
ARB Restart	:SOURce:GPRF:GENerator:ARB:WAVeform:RESTart	-----	-----
ARB Sampling Rate	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:SCLock:RATE?	<sampling_rate>
ARB Subsection Switch	:SOURce:GPRF:GENerator:ARB:WAVeform:SSWitch <switch>[,<subsection_number>]	:SOURce:GPRF:GENerator:ARB:WAVeform:SSWitch?	<switch>,<subsection_number>
Waveform Synchronization Select Pattern	:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect <mode>,<filename>[,<group_number>[,<length>]]	:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect?	<filename>,<group_number>,<length>

Baseband ARB (Cont'd)

Function	Command	Query	Response
Waveform Synchronization State	-----	:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:START:STATe?	<status>
Waveform Synchronization Cancel	:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:CANCel	-----	-----

Common

Function	Command	Query	Response
Modulation	:SOURce:GPRF:GENerator:BBMode <mode>	:SOURce:GPRF:GENerator:BBMode?	<mode>
VSG Mode	:SOURce:GPRF:GENerator:MODE <switch>	:SOURce:GPRF:GENerator:MODE?	<switch>
RF Output	:SOURce:GPRF:GENerator:STATe <on_off>	:SOURce:GPRF:GENerator:STATe?	<on_off_res>

Frequency

Function	Command	Query	Response
Frequency	:SOURce:GPRF:GENerator:RFSettings:FREQuency <freq>	:SOURce:GPRF:GENerator:RFSettings:FREQuency?	<freq>

Level

Function	Command	Query	Response
Level	:SOURce:GPRF:GENerator:RFSettings:LEVel <level>	:SOURce:GPRF:GENerator:RFSettings:LEVel?	<level>
Level Setting	-----	:SOURce:GPRF:GENerator:RFSettings:LEVel:SETTing?	<status>

Sequence

Function	Command	Query	Response
Sequence Execution Cancel	:SOURce:GPRF:GENerator:SEQuence:CANCel	-----	-----
Sequence Table Combination Selection	:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern <num>,<table_n>	:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern?	<num>,<table_n>
Sequence Synchronization	:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize <mode>	-----	-----
Sequence Execution	:SOURce:GPRF:GENerator:SEQuence:EXECute [<mode>]	-----	-----
Load Sequence Parameter	:SOURce:GPRF:GENerator:SEQuence:LOAD <list_table_no>,<file_name>	-----	-----
Sequence Reinitialization	:SOURce:GPRF:GENerator:SEQuence:REINitialization <on_off>	:SOURce:GPRF:GENerator:SEQuence:REINitialization?	<on_off>
Sequence Re-execution	:SOURce:GPRF:GENerator:SEQuence:RLISt	-----	-----
Baseband Mode of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:BBMode <list_table_no>,<segment_no>,<mode>	:SOURce:GPRF:GENerator:SEQuence:RX:BBMode? <list_table_no>,<segment_no>	<mode>
Baseband Mode Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:BBMode:ALL <list_table_no>,<mode_n>	:SOURce:GPRF:GENerator:SEQuence:RX:BBMode:ALL? <list_table_no>	<mode_n>
End Condition of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition <list_table_no>,<segment_no>,<condition>	:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition? <list_table_no>,<segment_no>	<condition>

Sequence (Cont'd)

Function	Command	Query	Response
End Condition Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition:ALL <list_table_no>,<condition_n>	:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition:ALL? <list_table_no>	<condition_n>
Sequence Error Check	-----	:SOURce:GPRF:GENerator:SEQuence:RX:ERRor? <list_table_no>	<error_count>,<error_x_res>
Frequency of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency <list_table_no>,<segment_no>,<freq>	:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency? <list_table_no>,<segment_no>	<freq>
Frequency Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency:ALL <list_table_no>,<freq_n>	:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency:ALL? <list_table_no>	<freq_n>
Sequence Start Index for continuous mode	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:GOTO <list_table_no>,<index>	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:GOTO? <list_table_no>	<index>
Sequence Repetition Mode	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:REPetition <list_table_no>,<mode>	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:REPetition? <list_table_no>	<mode>
Sequence Start and Stop Segment	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:SSTop <list_table_no>,<start_seg>,<stop_seg>	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:SSTop? <list_table_no>	<start_seg>,<stop_seg>
Sequence Start Segment	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STARt <list_table_no>,<start_seg>	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STARt? <list_table_no>	<start_seg>

Sequence (Cont'd)

Function	Command	Query	Response
Sequence Step Length	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STEP:LENGth <list_table_no>,<mode>,<time>	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STEP:LENGth? <list_table_no>	<mode>,<time>
Sequence Stop Segment	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STOP <list_table_no>,<stop_seg>	:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STOP? <list_table_no>	<stop_seg>
Level of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:LEVel <list_table_no>,<segment_no>,<level>	:SOURce:GPRF:GENerator:SEQuence:RX:LEVel? <list_table_no>,<segment_no>	<level>
Level Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:LEVel:ALL <list_table_no>,<level_n>	:SOURce:GPRF:GENerator:SEQuence:RX:LEVel:ALL? <list_table_no>	<level_n>
Loop Control of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl <list_table_no>,<segment_no>,<control>	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl? <list_table_no>,<segment_no>	<control>
Loop Control Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:ALL <list_table_no>,<control_n>	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:ALL? <list_table_no>	<control_n>
Loop Count of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt <list_table_no>,<segment_no>,<count>	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt? <list_table_no>,<segment_no>	<count>

Sequence (Cont’d)

Function	Command	Query	Response
Loop Count Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt:ALL <list_table_no>,<count_n>	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt:ALL? <list_table_no>	<count_n>
Segment Number of Segment in Sequence for Loop Control	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment <list_table_no>,<segment_no>,<jump_segment >	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment? <list_table_no>,<segment_no>	<jump_segment>
Loop Jump Segment Group of All Segments in Sequence for Loop Control	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment:ALL <list_table_no>,<jump_segment_n>	:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment:ALL? <list_table_no>	<jump_segment_n>
Output Port of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe <list_table_no>,<segment_no>,<port>	:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe? <list_table_no>,<segment_no>	<port>
Output Port Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe:ALL <list_table_no>,<port_n>	:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe:ALL? <list_table_no>	<port_n>
Steps of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt <list_table_no>,<segment_no>,<count>	:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt? <list_table_no>,<segment_no>	<count>
Steps Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt:ALL <list_table_no>,<count_n>	:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt:ALL? <list_table_no>	<count_n>

Sequence (Cont'd)

Function	Command	Query	Response
Trigger Delay of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay <list_table_no>,<segment_no>,<delay>	:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay? <list_table_no>,<segment_no>	<delay>
Trigger Delay Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay:ALL <list_table_no>,<delay_n>	:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay:ALL? <list_table_no>	<delay_n>
Trigger Source of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce <list_table_no>,<segment_no>,<trigger>	:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce? <list_table_no>,<segment_no>	<trigger>
Trigger Source Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce:ALL <list_table_no>,<trigger_n>	:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce:ALL? <list_table_no>	<trigger_n>
Waveform Change Trigger of Segment in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger <list_table_no>,<segment_no>,<on_off>	:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger? <list_table_no>,<segment_no>	<on_off_res>
Waveform Change Trigger Group of All Segments in Sequence	:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger:ALL <list_table_no>,<on_off_n>	:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger:ALL? <list_table_no>	<on_off_n_res>
Sequence Status	-----	:SOURce:GPRF:GENerator:SEQuence:STATus?	<status>
Save Sequence Parameter	:SOURce:GPRF:GENerator:SEQuence:STORe <list_table_no>,<file_name>	-----	-----

Sequence (Cont'd)

Function	Command	Query	Response
Pattern End Condition of Sequence Waveform	:SOURce:GPRF:GENerator:SEQuence:WAVeform:ENDCondition <list_table_no>,<index>,<condition>	:SOURce:GPRF:GENerator:SEQuence:WAVeform:ENDCondition? <list_table_no>,<index>	<condition>
Sequence Waveform Spectrum Reverse	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:DM:POLarity <list_table_no>,<mode>	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:DM:POLarity? <list_table_no>	<mode>
Number of Registered Waveform	-----	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:PCOunt? <list_table_no>	<count_res>
Trigger Delay of Sequence Waveform Pattern	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:DELay <list_table_no>,<delay>	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:DELay? <list_table_no>	<delay>
Trigger Source for Sequence Waveform Pattern	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce <list_table_no>,<trigger>	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce? <list_table_no>	<trigger>
Group End Trigger of Sequence Waveform Pattern	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GETRigger <list_table_no>,<index>,<on_off>	:SOURce:GPRF:GENerator:SEQuence:WAVeform:GETRigger? <list_table_no>,<index>	<on_off_res>
Repetition Count of Sequence Waveform Pattern	:SOURce:GPRF:GENerator:SEQuence:WAVeform:IREPetition <list_table_no>,<index>,<count>	:SOURce:GPRF:GENerator:SEQuence:WAVeform:IREPetition? <list_table_no>,<index>	<count>
Manual Trigger for Sequence Waveform	:SOURce:GPRF:GENerator:SEQuence:WAVeform:MTEXecute	-----	-----

Sequence (Cont'd)

Function	Command	Query	Response
Sequence Waveform Pattern Delete	:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:DELete <list_table_no>,<index>	-----	-----
Sequence Waveform Pattern Configuration	:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:SELect <list_table_no>,<index>,<file_name>[,<group_number>]	:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:SELect? <list_table_no>,<index>	<file_name>,<group_number>
Sequence Waveform Synchronization State	-----	:SOURce:GPRF:GENerator:SEQuence:WAVeform:SYNChronize:STARt:STATe?	<status>

5.2 Details of Commands

This section explains the SCPI commands in alphabetical order.

■ Viewing the command table

EX .... Command name (header)

Example .... Command function name

Function .... Command function

Command................................. Programming command syntax

Query .... Query syntax

Response .... Response syntax

Parameter .... Parameter definition

Details .... Command restrictions and others

Example of Use................................ Command usage example

Related Commands ....Introduction of related commands

■ Suffix code list

Suffix code	Unit	Suffix code	Unit
DBM	dBm	MHZ	MHz
DBUVEMF	dBμV (emf)	MS	ms
DBUVTERM	dBμV (Term)	MZ	MHz
GZ	GHz	NS	ns
GHZ	GHz	S	s
HZ	Hz	US	μs
KZ	kHz	Z	Hz
KHZ	kHz		

5.2.1 Fundamental operation commands

:CALCulate:CALibration:BAND:ERRor:COUNt?

Band Calibration Error Count

Function

Queries Band Calibration failure count.

Query

```
:CALCulate:CALibration:BAND:ERRor:COUNt?
```

Response

<count>

Parameter

<count>	Band Calibration failure count
Range	0 to

Example of Use

To query the Band Calibration failure count:
:CALC:CAL:BAND:ERR:COUN?
> 0

:CALCulate:CALibration:BAND:ERRor:LASTtime?

BAND Calibration Error Last Time

Function

Queries date and time of last Band Calibration failure.

Query

```
:CALCulate:CALibration:BAND:ERRor:LASTtime?
```

Response

<year>,<month>,<day>,<hour>,<min>

Parameters

<year>	Year (from 2001)
<month>	Month (1 to 12)
<day>	Day (1 to 31)
<hour>	Hour (0 to 23)
<min>	Minute (0 to 59)

Example of Use

To query the time of the last Band Calibration failure:

:CALC:CAL:BAND:ERR:LAST?

> 2011,09,13,21,45

Remarks

The response is *** when Band Calibration has never failed.

:CALCulate:CALibration:BAND:LASTtime?

BAND Calibration Last Time

Function

Queries date and time of last Band Calibration.

Query

```
:CALCulate:CALibration:BAND:LASTtime?
```

Response

<year>,<month>,<day>,<hour>,<min>

Parameters

<year>	Year (from 2001)
<month>	Month (1 to 12)
<day>	Day (1 to 31)
<hour>	Hour (0 to 23)
<min>	Minute (0 to 59)

Example of Use

To query the time of the last Band Calibration:

:CALC:CAL:BAND:LAST?

> 2011,09,13,11,45

Remarks

The response is "***" when Band Calibration has never been executed.

:CALCulate:CALibration:BAND:RESult?

Result of Band Calibration

Function

Queries Band Calibration result.

Query

```
:CALCulate:CALibration:BAND:RESult?
```

Response

<Result>

Parameters

<Result>	Band Calibration result
PASS	Calibration succeeded
FAIL	Calibration failed
UNEXECUTED	Calibration not executed

Example of Use

To query the Band Calibration result:

:CALC:CAL:BAND:RES?

> PASS

:CALCulate:CALibration:BAND:STARt

Execute BAND Calibration

Function

Executes band calibration.

Command

```
:CALCulate:CALibration:BAND:STARt
```

Example of Use

To execute the band calibration:
:CALC:CAL:BAND:STAR

:CALCulate:CALibration:BAND:STARt:TEMPerature

BAND Calibration Temperature

Function

Band Calibration is performed if the internal temperature changes by more than the setting when Band Calibration was executed previously. It is not performed if the internal temperature change is within the setting.

Command

```
:CALCulate:CALibration:BAND:STARt:TEMPerature <temperature>
```

Parameters

<temperature>	Temperature
Range	1.0 to 50.0°C
Resolution	0.1°C

Example of Use

To perform Band Calibration when the internal temperature changes by 2°C or more compared to the previous Band Calibration.:
:CALC:CAL:BAND:STAR:TEMP 2.0

:CALCulate:CALibration:ELAPsed:TIME?

Elapsed Time after Full Calibration

Function

Queries elapsed time after last Full Calibration.

Query

```
:CALCulate:CALibration:ELAPsed:TIME?
```

Response

<time>

Parameter

<time>	Elapsed time in seconds

Example of Use

To query the elapsed time after the last Full Calibration:

:CALC:CAL:ELAP:TIME?

> 1280

Remarks

The response is *** when Full Calibration has never been executed.

:CALCulate:CALibration:FULL:ERRor:COUNt?

Full Calibration Error Count

Function

Queries Full Calibration failure count.

Query

`:CALCulate:CALibration:FULL:ERRor:COUNt?`

Response

<count>

Parameter

<count>	Full Calibration failure count
Range	0 or more

Example of Use

To query the Full Calibration failure count:

:CALC:CAL:FULL:ERR:COUN?

> 0

:CALCulate:CALibration:FULL:ERRor:LASTtime?

FULL Calibration Error Last Time

Function

Queries date and time of last Full Calibration failure.

Query

```
:CALCulate:CALibration:FULL:ERRor:LASTtime?
```

Response

<year>,<month>,<day>,<hour>,<min>

Parameters

<year>	Year (from 2001)
<month>	Month (1 to 12)
<day>	Day (1 to 31)
<hour>	Hour (0 to 23)
<min>	Minute (0 to 59)

Example of Use

To query the time of the last Full Calibration failure:

:CALC:CAL:FULL:ERR:LAST?

> 2012,03,10,11,45

Remarks

The response is "***" when Full Calibration has never failed.

:CALCulate:CALibration:FULL:LASTtime?

FULL Calibration Last Time

Function

Queries date and time of last Full Calibration.

Query

```
:CALCulate:CALibration:FULL:LASTtime?
```

Response

<year>,<month>,<day>,<hour>,<min>

Parameters

<year>	Year (from 2001)
<month>	Month (1 to 12)
<day>	Day (1 to 31)
<hour>	Hour (0 to 23)
<min>	Minute (0 to 59)

Example of Use

To query the time of the last Full Calibration:

:CALC:CAL:FULL:LAST?

> 2012,03,13,11,45

Remarks

The response is "***" when Full Calibration has never been executed.

:CALCulate:CALibration:FULL:RESult?

Result of Full Calibration

Function

Queries Full Calibration result.

Query

```
:CALCulate:CALibration:FULL:RESult?
```

Response

<Result>

Parameter

<Result>	Full Calibration result
PASS	Calibration succeeded
FAIL	Calibration failed
UNEXECUTED	Calibration unexecuted

Example of Use

To query the Full Calibration result.

:CALC:CAL:FULL:RESult?

> PASS

:CALCulate:CALibration:FULL:STARt

Execute FULL Calibration

Function

Executes the calibration for full bandwidth.

Command

```
:CALCulate:CALibration:FULL:STARt
```

Example of Use

To execute the Full Bandwidth calibration:

:CALC:CAL:FULL:STAR

:CALCulate:CALibration:INITial

Initial Calibration

Function

Initialize the values of Full Calibration / Band Calibration.

Command

```
:CALCulate:CALibration:INITial
```

Example of Use

To initialize the values of Full Calibration / Band Calibration:
:CALC:CAL:INIT

Remarks

Latest date and the execution results of Full Calibration / Band Calibration are also initialized.
The failure count, and date and time of last calibration failure are not initialized.

:CALCulate:CALibration:TEMPerature?

Temperature at Full Calibration Execution

Function

Queries module temperature at last Full Calibration after power-on.

Query

```
:CALCulate:CALibration:TEMPerature?
```

Response

<temp>

Parameter

<temp>	Temperature (°C)

Example of Use

To query the module temperature at the last Full Calibration after power-on:
:CALC:CAL:TEMP?
> 35.31

Remarks

The response is *** when Full Calibration has never been executed after power-on.

:CALCulate:EXTLoss:TABLe:COUNt?

List Count in Loss Table

Function

Queries number of lists set in Loss Correction table.

Query

```
:CALCulate:EXTLoss:TABLe:COUNt? <index>
```

Response

<num>

Parameters

<index>	Index number of Loss Correction table
Range	1 to 16
Resolution	1
<num>	Number of lists registered in Level Frequency Correction table
Range	0 to 1000

Example of Use

:CALCulate:EXTLoss:TABLe:COUNt? 1

> 3

:CALCulate:EXTLoss:TABLe:DELete

Delete Loss Value from Cable Loss List

Function

Deletes correction value set in the loss correction table.

The deletion range corresponds to the registered No. in the loss correction table, and the range to be deleted is specified.

If the deletion range is not specified, all values in the loss correction table are deleted.

Command

```
:CALCulate:EXTLoss:TABLe:DELete [<a>[,<b>]]
```

Parameters

<a>	Specify the deletion range
Range	1 to 1000
Resolution	1
<b>	Specify the deletion range
Range	a to 1000
Resolution	1

Example of Use

:CALCulate:EXTLoss:TABLe:DELete 1,1000

:CALCulate:EXTLoss:TABLe:SETTing

Loss Table

Function

Sets or queries Loss Correction table index.

Command

```
:CALCulate:EXTLoss:TABLe:SETTing <index>
```

Query

```
:CALCulate:EXTLoss:TABLe:SETTing?
```

Response

<index>

Parameter

<index>	Correction table index
Range	1 to 16
Resolution	1
Default	1

Example of Use

:CALCulate:EXTLoss:TABLe:SETTing 1
:CALCulate:EXTLoss:TABLe:SETTing?
> 1

:CALCulate:EXTLoss:TABLe:VALue

Register Loss Value into Cable Loss List

Function

Sets/queries value in the loss correction table.
The same frequency in the loss correction table is overwritten.
If the setting of [Port3,4/in] is omitted, the value of [Port3,4/out] is set at the loss correction table.

An arbitrary frequency point loss value is input for the loss correction table and the correction for the frequency between the setting points is determined automatically by linear interpolation.
The same correction values are set for Port1 and Port 2.Also, the same correction values are set for Port 3 and Port 4.

Command

```
:CALCulate:EXTLoss:TABLe:VALue
<frequency>,<Port1,2/out>,<Port1,2/in>,<Port3,4/out>[,<Port3,4/in>]
```

Query

```
:CALCulate:EXTLoss:TABLe:VALue? <n>
```

Response

<frequency>,<Port1,2/out>,<Port1,2/in>,<Port3,4/out>,<Port3,4/in>

Parameters

<frequency>	Frequency point to be set
Range	10 MHz to 6000 MHz
Resolution	1 Hz
Suffix code	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
<Port1,2/out>	Port1/2 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port1,2/in>	Port1/2 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port3,4/out>	Port3/4 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port3,4/in>	Port3/4 UL loss value

Range –100.00 to 100.00 dB
Resolution 0.01 dB
Suffix code DB (uses DB when omitted)

<n> Correction table point to query
Range 1 to 1000
Resolution 1

Example of Use

To set the values of the table below in the loss correction table.

No	Frequency	Port1 (dB)		Port2 (dB)		Port3 (dB)		Port4 (dB)	
		out	in	out	in	out	in	out	in
1	1000000000	10.00	11.00	10.00	11.00	12.00	13.00	12.00	13.00

```
:CALCulate:EXTLoss:TABLe:VALue 1GHz,10,11,12,13
:CALCulate:EXTLoss:TABLe:VALue? 1
> 1000000000,10.00,11.00,12.00,13.00
```

Remarks

The loss correction table is held as a correction value common for each measurement software.
The loss correction table values are arranged automatically in ascending order for frequencies.

:CALCulate:EXTLoss:TABLe:VALue:ALL

Register Loss Value into All Cable Loss List

Function

Sets or queries a value in the loss correction table. The same frequency in the loss correction table is overwritten.

An arbitrary frequency point loss value is input for the loss correction table and the correction for the frequency between the setting points is determined automatically by linear interpolation.
Different correction values can be set for Port1,Port2, Port3, and Port4.

Command

```
:CALCulate:EXTLoss:TABLe:VALue:ALL
<frequency>,<Port1out>,<Port1in>,<Port2out>,<Port2in>,<Port3out>,<Port3in
>,<Port4out>,<Port4in>
```

Query

```
:CALCulate:EXTLoss:TABLe:VALue:ALL? <n>
```

Response

<frequency>,<Port1out>,<Port1in>,<Port2out>,<Port2in>,<Port3out>,<Port3in>,<Port4out>,<
Port4in>

Parameters

<frequency>	Frequency point to be set
Range	10 MHz to 6000 MHz
Resolution	1 Hz
Suffix code	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
<Port1out>	Port1 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port1in>	Port1 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port2out>	Port2 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port2in>	Port2 UL loss value
Range	–100.00 to 100.00 dB

5

SCPI Command Reference

Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port3out>	Port3 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port3in>	Port3 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port4out>	Port4 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port4in>	Port4 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<n>	Correction table point to query
Range	1 to 1000
Resolution	1

Example of Use

To set the values of the table below in the loss correction table.

No	Frequency	Port1 (dB)		Port2 (dB)		Port3 (dB)		Port4 (dB)	
		out	In	out	in	out	in	out	in
1	1000000000	10.00	11.00	12.00	13.00	14.00	15.00	16.00	17.00

:CALCulate:EXTLoss:TABLe:VALue:ALL 1GHZ,10,11,12,13,14,15,16,17
:CALCulate:EXTLoss:TABLe:VALue:ALL? 1
> 1000000000,10.00,11.00,12.00,13.00,14.00,15.00,16.00,17.00

Remarks

The loss correction table is maintained as correction values common to each measurement software.
The loss correction table values are arranged automatically in ascending order for frequencies.

:MAINtenance:SYSTem:INFormation:DEVice:MODel?

Module Model Number

Function

Queries module model number.

Query

```
:MAINtenance:SYSTem:INFormation:DEVice:MODel?
```

Response

<model>

Parameter

<model>	Model number

Example of Use

:MAINtenance:SYSTem:INFormation:DEVice:MODel?
> MU887000A

:MAINtenance:SYSTem:INFormation:MAINframe:MODel?

Main Frame Model Number

Function

Queries MT8870A model number

Query

```
:MAINtenance:SYSTem:INFormation:MAINframe:MODel?
```

Response

<model>

Parameter

<model>	Model number

Example of Use

:MAINtenance:SYSTem:INFormation:MAINframe:MODel?
> MT8870A

:MMEMory:CATalog?

File List

Function

Queries list of files saved in specified path.

Query

```
:MMEMory:CATalog? <path>
```

Response

<num>,<filelist>

Parameters

<path>	Path
<num>	Number of files
<filelist>	File list

Details

When the response list is 10485760 characters or more, "..." is appended to the list.
A file extension such as "*.h" can be specified for the path.

Example of Use

:MMEM:CAT? "CELLULAR/parameter"
> 3,setup.000,setup.001,setup.002

:MMEM:CAT? "CELLULAR/parameter/*001"
> 1,setup.001

:MMEMory:DELete

Delete File

Function

Deletes specified file.

Command

```
:MMEMory:DELete <filepath>
```

Parameter

<filepath>	File name (path information included)

Details

Specify the file name in a format including the path.

Example of Use

:MMEM:CAT? "CELLULAR/parameter"
> 3,setup.000,setup.001,setup.002
:MMEM:DEL "CELLULAR/parameter/setup.002"
:MMEM:CAT? "CELLULAR/parameter"
> 2,setup.000,setup.001

:MMEMory:DELete:USERarea

User Area Delete

Function

Deletes user area.

Command

```
:MMEMory:DELete:USERarea
```

Details

Data stored in the memory of FTP-accessible modules is erased.
All user files, such as waveform data, are erased.

Example of Use

:MMEMory:DELete:USERarea

:MMEMory:LOG:CLEar

Log Clear

Function

Clears log.

Command

```
:MMEMory:LOG:CLEar
```

Details

Clears modular log file

Example of Use

:MMEMory:LOG:SIZE?

> 2147483647

:MMEMory:LOG:CLEar

:MMEMory:LOG:LOAD?

Log Reading

Function

Queries log information.

Query

```
:MMEMory:LOG:LOAD? [<line>]
```

Response

#XY<Y byte binary data><terminator>

X and Y are ASCII characters, and the value X is a digit number of the value Y. For example, when Y is 12500, X is 5.

The binary data <Y byte binary data> is a word string in the log file.

Parameter

<line>	Line number of the latest acquired log
Range	1 to 999999
<terminator>	Terminator

Details

When the line of acquiring log is specified by the parameter, the log is acquired by the specified number of lines. When omitted, the entire log is acquired.

The linefeed codes in the log file are replaced by tabs.

Example of Use

:MMEMory:LOG:LOAD?
> CELLULAR Measurement Software License expired.\n

:MMEM:LOAD:LOG? 2
> CELLULAR Measurement Software License expired.\n
> R[*IDN?]\n

:MMEMory:LOG:SIZE?

Log Size Reading

Function

Queries log size.

Query

`:MMEMory:LOG:SIZE?`

Response

<size>

Parameter

<size>	Log size

Details

The response is the byte size of the log file.

Example of Use

:MMEMory:LOG:SIZE?

> 48

:MMEMory:SIZE?

Drive Size Reading

Function

Queries free space of the memory.

Query

```
:MMEMory:SIZE? [<target>]
```

Response

<total>,<blank>

Parameters

<target>	Area to check the free space.
LOG	Area for log records
USER	Area for waveform files and installer files
Default	USER
<total>	Total space of the specified area (byte)
<blank>	Free space of the specified area (byte)

Example of Use

:MMEMory:SIZE? LOG
> 2097152,734002

[:ROUTe]:EXTLoss:TABLe:SWITch

External Loss On/Off

Function

Sets or queries whether level corrected using all ports DL/UL/ loss data in the loss correction table.

Command

```
[:ROUTe]:EXTLoss:TABLe:SWITch <sw>
```

Query

```
[:ROUTe]:EXTLoss:TABLe:SWITch?
```

Response

<sw>

Parameter

<sw>	All ports DL/UL loss value enable/disable
COMMON	The correction value is used.
ON	The correction value is used.
OFF	The correction value is not used.
Default	OFF

Example of Use

[:ROUTe]:EXTLoss:TABLe:SWITch ON
[:ROUTe]:EXTLoss:TABLe:SWITch?
> ON

Remarks

To obtain the correction value for a frequency not set in the loss correction table, the value is calculated by linear interpolation using the correction values for preceding and succeeding frequencies.
The correction value for the minimum frequency is used for a lower frequency lower than the minimum frequency. The correction value for the maximum frequency is used for a higher frequency than the maximum frequency.

:ROUTe:PORT:CONNect:DIRection

Set Direction of RF Connector

Function

Sets or queries connectors for RF signals input/output.

Command

```
:ROUTe:PORT:CONNect:DIRection <input>,<output>
```

Query

```
:ROUTe:PORT:CONNect:DIRection?
```

Response

<input>,<output>

Parameters

<input>	Test Port number
PORT1	Test Port1
PORT2	Test Port2
PORT3	Test Port3
PORT4	Test Port4
Default	PORT1
<output>	Test Port number
PORT1	Test Port1
PORT2	Test Port2
PORT3	Test Port3
PORT4	Test Port4
Default	PORT1

Details

Test Port1 and Test Port2 can be set for both input and output.
Test Port3 and Test Port4 can be set for either input or output.

Example of Use

To set Test Port1 for input RF signals and Test Port2 for output RF signals:
:ROUTe:PORT:CONNect:DIRection PORT1,PORT2
:ROUTe:PORT:CONNect:DIRection?
> PORT1,PORT2

:ROUTe:TRIGger:INPut:SOURce

Select Source of Trigger Input

Function

Sets or queries the signal used as external trigger.

Command

```
:ROUTe:TRIGger:INPut:SOURce <source>
```

Query

```
:ROUTe:TRIGger:INPut:SOURce?
```

Response

<source>

Parameter

<source>	Signal used as external trigger
TRGIN	Trigger input to the trigger connector of the slot with the module to control installed.
TRGIN1	Trigger input to Trigger Connector 1.
SLOT1	Trigger signal output from the Slot 1 module.
Default	TRGIN

Details

When selecting TRGIN, set the I/O setting of the trigger connector to input.
The signal used as external trigger may be changed by using the commands below.

:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect
:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize

Example of Use

:ROUTe:TRIGger:INPut:SOURce SLOT1
:ROUTe:TRIGger:INPut:SOURce?
> SLOT1

Related Command

:ROUTe:TRIGger:PORT

:ROUTe:TRIGger:OUTPut:SOURce

Select Source of Trigger Output

Function

Sets and queries signal output at Trigger connector on MT8870A rear panel.

Command

```
:ROUTe:TRIGger:OUTPut:SOURce <source>
```

Query

```
:ROUTe:TRIGger:OUTPut:SOURce?
```

Response

<source>

Parameter

<source>	Signal output at Trigger connector
WFM1	SG waveform marker 1
WFM2	SG waveform marker 2
WFM3	SG waveform marker 3
SGSYNC	SG synchronization start signal
NONE	No signal output
Default	NONE

Details

When setting output at the Trigger connector, the signal set by this command is output at the Trigger connector.

The signal output to the trigger connector may be changed by using the commands below.

:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect
:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize

Example of Use

:ROUTe:TRIGger:OUTPut:SOURce WFM1
:ROUTe:TRIGger:OUTPut:SOURce?
> WFM1

Related Command

:ROUTe:TRIGger:PORT

:ROUTe:TRIGger:PORT

Set In/Out of Trigger Connector

Function

Sets or queries input/output of trigger connector on MT8870A rear panel.

Command

```
:ROUTe:TRIGger:PORT <trigger>
```

Query

```
:ROUTe:TRIGger:PORT?
```

Response

<trigger>

Parameter

<trigger>	Module
INPUT	Input
OUTPUT	Output
Default	INPUT

Details

The trigger connector input/output can be set for each module.
The I/O of the trigger connector may be changed by using the commands below.

:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect
:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize

Example of Use

:ROUTe:TRIGger:PORT INPUT
:ROUTe:TRIGger:PORT?
> INPUT

:STATus:OPERation:CONDition?

Operation Status Condition Register Query

Function

Queries content of operation status condition register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:OPERation:CONDition?
```

Response

<oscr>
For the register bit allocation, refer to the description for :STATus:OPERation[:EVENt].

Parameter

<oscr>	Operation status condition register
Range	0 to 65535

Details

The sum of the values for bits of the occurring event from the values $2^0 = 1$, $2^1 = 2$ to $2^{15} = 32768$, that correspond to the operation status condition register bits 0, 1 to 15 becomes the response.

Example of Use

To query content of operation status condition register:
:STAT:OPER:COND?
>0

:STATus:OPERation:ENABle

Operation Status Enable Register

Function

Sets or queries operation status enable register value.

Command

```
:STATus:OPERation:ENABle <oser>
```

Query

```
:STATus:OPERation:ENABle?
```

Response

<oser>

For register bit allocation, refer to the description for :STATus:OPERation[:EVENt].

Parameter

<oser>	Operation status enable register
Range	0 to 65535
Resolution	1
Default	0

Details

The bit for outputting detection results in set in the status byte register.

Example of Use

To set to read-only bit 4 of the operation status event register when $2^4 = 16$ is set at the operation status enable register:

:STAT:OPER:ENAB 16

To query the operation status condition register value:

:STAT:OPER:ENAB?

>16

:STATus:OPERation[:EVENt]?

Operation Status Register Query

Function

Queries contents of operation status register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:OPERation[:EVENt]?
```

Response

<osr> Operation status register

Value = bit0 + bit1 + ... + bit15

bit0 = 2^0 = 1	Unused
bit1 = 2^1 = 2	Unused
bit2 = 2^2 = 4	Unused
bit3 = 2^3 = 8	Unused
bit4 = 2^4 = 16	Unused
bit5 = 2^5 = 32	Unused
bit6 = 2^6 = 64	Unused
bit7 = 2^7 = 128	Unused
bit8 = 2^8 = 256	Measurement
bit9 = 2^9 = 512	Signal generator
bit10 = 2^{10} = 1024	Unused
bit11 = 2^{11} = 2048	Unused
bit12 = 2^{12} = 4096	Unused
bit13 = 2^{13} = 8192	Unused
bit14 = 2^{14} = 16384	Unused
bit15 = 2^{15} = 32768	Unused

Parameter

<osr> Operation status register

Range 0 to 65535

Details

The sum of the values for bits of the occurring event from the values 2^0 = 1, 2^1 = 2 to 2^{15} = 32768, that correspond to the operation status register bits 0, 1 to 15 becomes the response.

Example of Use

To query content of operation status:

:STAT:OPER?

>512

:STATus:OPERation:GENerator:CONDition?

Generator Operation Status Condition Register Query

Function

Queries content of signal generator status condition register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:OPERation:GENerator:CONDition?
```

Response

<goscr>
For register bit allocation, refer to the description
for :STATus:OPERation:GENerator[:EVENt].

Parameter

<goscr>	Signal generator status condition register
Range	0 to 65535

Details

Same as :STATus:OPERation:CONDition

:STATus:OPERation:GENerator:ENABle

Generator Operation Status Enable Register

Function

Sets or queries signal generator status register value.

Command

```
:STATus:OPERation:GENerator:ENABle <goser>
```

Query

```
:STATus:OPERation:GENerator:ENABle?
```

Response

<goser>

For register bit allocation, refer to the description for :STATus:OPERation:GENerator[:EVENt].

Parameter

<goser>	Signal generator status register
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:ENABle.

:STATus:OPERation:GENerator[:EVENt]?

Generator Operation Status Register Query

Function

Queries signal generator operation status register value.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:OPERation:GENerator[:EVENt]?
```

Response

<gosr>

Value = bit0 + bit1 + ... + bit15

bit0 = 2^0 = 1	The file is being loaded.
bit1 = 2^1 = 2	Defragmentation in progress.
bit2 = 2^2 = 4	Preparation of synchronized output of waveform pattern in

progress.

bit3 = 2^3 = 8	Unused
bit4 = 2^4 = 16	Unused
bit5 = 2^5 = 32	Unused
bit6 = 2^6 = 64	Unused
bit7 = 2^7 = 128	Unused
bit8 = 2^8 = 256	Unused
bit9 = 2^9 = 512	Unused
bit10 = 2^{10} = 1024	Unused
bit11 = 2^{11} = 2048	Unused
bit12 = 2^{12} = 4096	Unused
bit13 = 2^{13} = 8192	Unused
bit14 = 2^{14} = 16384	Unused
bit15 = 2^{15} = 32768	Unused

Parameter

<gosr>	Signal generator operation status register
Range	0 to 65535

Details

The sum of the values for bits of the occurring event from the values $2^0 = 1$, $2^1 = 2$ to $2^{15} = 32768$, that correspond to the signal generator operation status register bits 0, 1 to 15 becomes the response.

Example of Use

To query content of signal generator operation status register:

:STAT:OPER:GEN?

> 1

:STATus:OPERation:GENerator:NTRansition

Generator Operation Status Negative Transition Filter

Function

Sets or queries signal generator status register transition filter (negative transition).

Command

```
:STATus:OPERation:GENerator:NTRansition <gosr_ntf>
```

Query

```
:STATus:OPERation:GENerator:NTRansition?
```

Response

<gosr_ntf>

For register bit allocation, refer to the description for :STATus:OPERation:GENerator[:EVENt].

Parameter

<gosr_ntf>	Signal generator status register transition filter (negative transition)
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:NTRansition

:STATus:OPERation:GENerator:PTRansition

Generator Operation Status Positive Transition Filter

Function

Sets or queries signal generator status register transition filter (positive transition).

Command

```
:STATus:OPERation:GENerator:PTRansition <gosr_ptf>
```

Query

```
:STATus:OPERation:GENerator:PTRansition?
```

Response

<gosr_ptf>

For register bit allocation, refer to the description for :STATus:OPERation:GENerator[:EVENt].

Parameter

<gosr_ptf>	Signal generator status register transition filter (positive transition)
Range	0 to 65535
Resolution	1
Default	65535

Details

Same as :STATus:OPERation:PTRansition

:STATus:OPERation:MEASure:CONDition?

Measurement Operation Status Condition Register Query

Function

Queries content of measurement operation status condition register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:OPERation:MEASure:CONDition?
```

Response

<moscr>
For register bit allocation, refer to the description for :STATus:OPERation:MEASure[:EVENt].

Parameter

<moscr>	Measurement status condition register
Range	0 to 65535

Details

Same as :STATus:OPERation:CONDition

Example of Use

To query content of measurement operation status condition register:
:STAT:OPER:MEAS:COND?
> 1

:STATus:OPERation:MEASure:ENABle

Measurement Operation Status Enable Register

Function

Sets or queries measurement status enable register value.

Command

```
:STATus:OPERation:MEASure:ENABle <moser>
```

Query

```
:STATus:OPERation:MEASure:ENABle?
```

Response

<moser>

For register bit allocation, refer to the description for :STATus:OPERation:MEASure[:EVENt].

Parameter

<moser>	Measurement status enable register
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:ENABle

:STATus:OPERation:MEASure[:EVENt]?

Measurement Operation Status Register Query

Function

Queries content of measurement operation status register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:OPERation:MEASure[:EVENt]?
```

Response

<mosr>

Value = bit0 + bit1 + ... + bit15

bit0 = 2^0 = 1	Measurement in progress
bit1 = 2^1 = 2	Unused
bit2 = 2^2 = 4	Unused
bit3 = 2^3 = 8	Unused
bit4 = 2^4 = 16	Unused
bit5 = 2^5 = 32	Unused
bit6 = 2^6 = 64	Unused
bit7 = 2^7 = 128	Unused
bit8 = 2^8 = 256	Unused
bit9 = 2^9 = 512	Unused
bit10 = 2^{10} = 1024	Unused
bit11 = 2^{11} = 2048	Unused
bit12 = 2^{12} = 4096	Unused
bit13 = 2^{13} = 8192	Unused
bit14 = 2^{14} = 16384	Unused
bit15 = 2^{15} = 32768	Unused

The bit allocation depends on the application.
Refer to the application software operation manual.

Parameter

<mosr>	Measurement status register
Range	0 to 65535

Details

The sum of the values for bits of the occurring event from the values 2^0 = 1, 2^1 = 2 to 2^{15} = 32768, that correspond to the measurement operation status register bits 0, 1 to 15 becomes the response.

5

SCPI Command Reference

Example of Use

To query content of measurement operation status register:

:STAT:OPER:MEAS?

> 1

:STATus:OPERation:MEASure:NTRansition

Measurement Operation Status Negative Transition Filter

Function

Sets or queries measurement status register transition filter (negative transition).

Command

```
:STATus:OPERation:MEASure:NTRansition <mosr_ntr>
```

Query

```
:STATus:OPERation:MEASure:NTRansition?
```

Response

<mosr_ntr>

For register bit allocation, refer to the description for :STATus:OPERation:MEASure[:EVENt].

Parameter

<mosr_ntr>	Measurement status register transition filter (negative transition)
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:NTRansition

:STATus:OPERation:MEASure:PTRansition

Measurement Operation Status Positive Transition Filter

Function

Sets or queries measurement status register transition filter (positive transition).

Command

```
:STATus:OPERation:MEASure:PTRansition <mosr_ptr>
```

Query

```
:STATus:OPERation:MEASure:PTRansition?
```

Response

<mosr_ptr>

For register bit allocation, refer to the description for :STATus:OPERation:MEASure[:EVENt].

Parameters

<mosr_ptr>	Measurement status register transition filter (positive transition)
Range	0 to 65535
Resolution	1
Default	65535

Details

Same as :STATus:OPERation:PTRansition

:STATus:OPERation:NTRansition

Operation Status Negative Transition Filter

Function

Sets or queries operation status register transition filter (negative transition).

Command

```
:STATus:OPERation:NTRansition <osr_ntf>
```

Query

```
:STATus:OPERation:NTRansition?
```

Response

<osr_ntf>

For register bit allocation, refer to the description for :STATus:OPERation[:EVENt].

Parameters

<osr_ntf>	Operation status register transition filter (negative transition)
Range	0 to 65535
Resolution	1
Default	0

Details

The bit to detect the change of 1 to 0 of the operation status condition register value is set to 1.

Example of Use

To output the change of 1 to 0 of the operation status condition register bit 8 to the operation status event register when 28 = 256 is set at the transition filter (negative transition):

:STAT:OPER:NTR 256

To query the operation status register transition filter (negative transition) value:

:STAT:OPER:NTR?

>256

:STATus:OPERation:PTRansition

Operation Status Positive Transition Filter

Function

Sets or queries operation status register transition filter (positive transition).

Command

```
:STATus:OPERation:PTRansition <osr_ptf>
```

Query

```
:STATus:OPERation:PTRansition?
```

Response

<osr_ptf>

For register bit allocation, refer to the description for :STATus:OPERation[:EVENt].

Parameter

<osr_ptf>	Operation status register transition filter (positive transition)
Range	0 to 65535
Resolution	1
Default	65535

Details

The bit to detect the change of 0 to 1 of the operation status condition register value is set to 1.

Example of Use

To output the change of 0 to 1 of the operation status condition register bit 9 to the operation status event register when set $2^9 = 512$ is set at the transition filter (positive transition):

:STAT:OPER:PTR 512

To query the operation status register transition filter (positive transition) value:

:STAT:OPER:PTR?

>512

:STATus:PRESet

Preset Status

Function

Inializes each enable register and each register transition filter (positive and negative transition).

Command

```
:STATus:PRESet
```

Details

The execution status register, questionable register, measurement status register, measurement questionable register, signal generator status register, signal generator questionable register, and each enable register transition filter (positive and negative transition) are initialized.
The setting for each enable register becomes 0.
The setting for each transition filter (positive transition) becomes 65535.
The setting for each transition filter (negative transition) becomes 0.
Each type of event register and condition register is not initialized.
Status bye and reference event status registers are not initialized.

Example of Use

:STAT:PRES

:STATus:QUEStionable:CONDition?

Questionable Status Condition Register Query

Function

Queries content of questionable status condition register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:QUEStionable:CONDition?
```

Response

<qscr>
For register bit allocation, refer to the description for :STATus:QUEStionable[:EVENt].

Parameter

<qscr>	Questionable status condition register
Range	0 to 65535

Details

Same as :STATus:OPERation:CONDition

:STATus:QUEStionable:ENABle

Questionable Status Enable Register

Function

Sets or queries questionable enable register value.

Command

```
:STATus:QUEStionable:ENABle <qser>
```

Query

```
:STATus:QUEStionable:ENABle?
```

Response

<qser>

For register bit allocation, refer to the description for :STATus:QUEStionable[:EVENt].

Parameter

<qser>	Questionable enable register
Range	0 to 65535
Resolution	1
Default	0

Details

The bit for outputting detection results is set in the status byte register.

:STATus:QUEStionable[:EVENt]?

Questionable Status Register Query

Function

Queries content of questionable status register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:QUEStionable[:EVENt]?
```

Response

<qsr>

Value = bit0 + bit1 + ... + bit15

bit0 = 2^0 = 1	Unused
bit1 = 2^1 = 2	Unused
bit2 = 2^2 = 4	Unused
bit3 = 2^3 = 8	Unused
bit4 = 2^4 = 16	Unused
bit5 = 2^5 = 32	Unused
bit6 = 2^6 = 64	Unused
bit7 = 2^7 = 128	Unused
bit8 = 2^8 = 256	Unused
bit9 = 2^9 = 512	Measurement
bit10 = 2^{10} = 1024	Signal generator
bit11 = 2^{11} = 2048	Unused
bit12 = 2^{12} = 4096	Unused
bit13 = 2^{13} = 8192	Unused
bit14 = 2^{14} = 16384	Unused
bit15 = 2^{15} = 32768	Unused

Parameter

<qsr>	Questionable status register
Range	0 to 65535

Details

The sum of the values for bits of the occurring event from the values 2^0 = 1, 2^1 = 2 to 2^{15} = 32768, that correspond to the questionable status register bits 0, 1 to 15 becomes the response.

Example of Use

To query content of questionable status register:
:STAT:QUES?
> 512

:STATus:QUEStionable:GENerator:CONDition?

Generator Questionable Status Condition Register Query

Function

Queries content of signal generator questionable status condition register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:QUEStionable:GENerator:CONDition?
```

Response

<gqscr>
For register bit allocation, refer to the description
for :STATus:QUEStionable:GENerator[:EVENt].

Parameter

<gqscr>	Signal generator questionable status condition register
Range	0 to 65535

Details

Same as :STATus:OPERation:CONDition

:STATus:QUEStionable:GENerator:ENABle

Generator Questionable Status Enable Register

Function

Sets or queries signal generator questionable enable register value.

Command

```
:STATus:QUEStionable:GENerator:ENABle <gqser>
```

Query

```
:STATus:QUEStionable:GENerator:ENABle?
```

Response

<gqser>
For register bit allocation, refer to the description
for :STATus:QUEStionable:GENerator[:EVENt].

Parameter

<gqser>	Signal generator questionable enable register
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:ENABle

:STATus:QUEStionable:GENerator[:EVENt]?

Generator Questionable Status Register Query

Function

Queries contents of signal generator questionable status register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:QUEStionable:GENerator[:EVENt]?
```

Response

<gqsr> Value = bit0 + bit1 + ... + bit15

bit0 = 2^0 = 1	The loaded file has an error.
bit1 = 2^1 = 2	Unused
bit2 = 2^2 = 4	Unused
bit3 = 2^3 = 8	Unused
bit4 = 2^4 = 16	Unused
bit5 = 2^5 = 32	Unused
bit6 = 2^6 = 64	Unused
bit7 = 2^7 = 128	Unused
bit8 = 2^8 = 256	Unused
bit9 = 2^9 = 512	Unused
bit10 = 2^{10} = 1024	Unused
bit11 = 2^{11} = 2048	Unused
bit12 = 2^{12} = 4096	Unused
bit13 = 2^{13} = 8192	Unused
bit14 = 2^{14} = 16384	Unused
bit15 = 2^{15} = 32768	Unused

Parameter

<gqsr>	Signal generator questionable status register
Range	0 to 65535

Details

The sum of the values for bits of the occurring event from the values 2^0 = 1, 2^1 = 2 to 2^{15} = 32768, that correspond to the signal generator questionable status register bits 0, 1 to 15 becomes the response.

Example of Use

To query content of signal generator questionable status register:

:STAT:QUES:GEN?

> 1

:STATus:QUEStionable:GENerator:NTRansition

Generator Questionable Status Negative Transition Filter

Function

Sets or queries signal generator questionable register transition filter (negative transition).

Command

```
:STATus:QUEStionable:GENerator:NTRansition <gqsr_ntf>
```

Query

```
:STATus:QUEStionable:GENerator:NTRansition?
```

Response

<gqsr_ntf>
For register bit allocation, refer to the description for :STATus:QUEStionable:GENerator[:EVENt].

Parameter

<gqsr_ntf>	Signal generator questionable register transition filter (negative transition)
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:NTRansition

:STATus:QUEStionable:GENerator:PTRansition

Generator Questionable Status Positive Transition Filter

Function

Sets or queries signal generator questionable register transition filter (positive transition).

Command

```
:STATus:QUEStionable:GENerator:PTRansition <gqsr_ptf>
```

Query

```
:STATus:QUEStionable:GENerator:PTRansition?
```

Response

<gqsr_ptf>
For register bit allocation, refer to the description
for :STATus:QUEStionable:GENerator[:EVENt].

Parameter

<gqsr_ptf>	Signal generator questionable register transition filter (positive transition)
Range	0 to 65535
Resolution	1
Default	65535

Details

Same as :STATus:OPERation:PTRansition

:STATus:QUEStionable:MEASure:CONDition?

Measurement Questionable Status Condition Register Query

Function

Queries contents of the measurement questionable status condition register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:QUEStionable:MEASure:CONDition?
```

Response

<mqscr>
For register bit allocation, refer to the description
for :STATus:QUEStionable:MEASure[:EVENt].

Parameter

<mqscr>	Measurement questionable status condition register
Range	0 to 65535

Details

Same as :STATus:OPERation:CONDition

:STATus:QUEStionable:MEASure:ENABle

Measurement Questionable Status Enable Register

Function

Sets or queries measurement questionable enable register value.

Command

```
:STATus:QUEStionable:MEASure:ENABle <mqser>
```

Query

```
:STATus:QUEStionable:MEASure:ENABle?
```

Response

<mqser>

For register bit allocation, refer to the description for :STATus:QUEStionable:MEASure[:EVENt].

Parameter

<mqser>	Measurement questionable enable register
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:ENABle

:STATus:QUEStionable:MEASure[:EVENt]?

Measurement Questionable Status Register Query

Function

Queries content of measurement questionable status register.
The event occurrence can be identified using the retrieved value.

Query

```
:STATus:QUEStionable:MEASure[:EVENt]?
```

Response

<mqsr>

Value = bit0 + bit1 + ... + bit15

bit0 = 2^0 = 1	Level over
bit1 = 2^1 = 2	Level under
bit2 = 2^2 = 4	Timeout
bit3 = 2^3 = 8	Unused
bit4 = 2^4 = 16	Unused
bit5 = 2^5 = 32	Unused
bit6 = 2^6 = 64	Unused
bit7 = 2^7 = 128	Unused
bit8 = 2^8 = 256	Unused
bit9 = 2^9 = 512	Unused
bit10 = 2^{10} = 1024	Unused
bit11 = 2^{11} = 2048	Unused
bit12 = 2^{12} = 4096	Unused
bit13 = 2^{13} = 8192	Unused
bit14 = 2^{14} = 16384	Unused
bit15 = 2^{15} = 32768	Unused

Bit allocation depends on the application.
Refer to the application software operation manual.

Parameter

<mqsr>	Measurement questionable status register
Range	0 to 65535

Details

The sum of the values for bits of the occurring event from the values 2^0 = 1, 2^1 = 2 to 2^{15} = 32768, that correspond to the measurement questionable status register bits 0, 1 to 15 becomes the response.

Example of Use

To query content of measurement operation status register:
:STAT:QUES:MEAS?
> 1

:STATus:QUEStionable:MEASure:NTRansition

Measurement Questionable Status Negative Transition Filter

Function

Sets or queries measurement questionable register transition filter (negative transition).

Command

```
:STATus:QUEStionable:MEASure:NTRansition <mqsr_ntr>
```

Query

```
:STATus:QUEStionable:MEASure:NTRansition?
```

Response

<mqsr_ntr>
For register bit allocation, refer to the description
for :STATus:QUEStionable:MEASure[:EVENt].

Parameter

<mqsr_ntr>	Measurement questionable register transition filter (negative transition)
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:NTRansition

:STATus:QUEStionable:MEASure:PTRansition

Measurement Questionable Status Positive Transition Filter

Function

Sets or queries measurement questionable register transition filter (positive transition).

Command

```
:STATus:QUEStionable:MEASure:PTRansition <mqsr_ptr>
```

Query

```
:STATus:QUEStionable:MEASure:PTRansition?
```

Response

<mqsr_ptr>
For register bit allocation, refer to the description
for :STATus:QUEStionable:MEASure[:EVENt].

Parameter

<mqsr_ptr>	Measurement questionable register transition filter (positive transition)
Range	0 to 65535
Resolution	1
Default	65535

Details

Same as :STATus:OPERation:PTRansition

:STATus:QUEStionable:NTRansition

Questionable Status Negative Transition Filter

Function

Sets or queries questionable register transition filter (negative transition).

Command

```
:STATus:QUEStionable:NTRansition <qsr_ntf>
```

Query

```
:STATus:QUEStionable:NTRansition?
```

Response

<qsr_ntf>

For register bit allocation, refer to the description for :STATus:QUEStionable[:EVENt].

Parameter

<qsr_ntf>	Questionable register transition filter (negative transition)
Range	0 to 65535
Resolution	1
Default	0

Details

Same as :STATus:OPERation:NTRansition

:STATus:QUEStionable:PTRansition

Questionable Status Positive Transition Filter

Function

Sets or queries the questionable register transition filter (positive transition).

Command

```
:STATus:QUEStionable:PTRansition <qsr_ptf>
```

Query

```
:STATus:QUEStionable:PTRansition?
```

Response

<qsr_ptf>

For register bit allocation, refer to the description for :STATus:QUEStionable[:EVENt].

Parameter

<qsr_ptf>	Questionable register transition filter (positive transition)
Range	0 to 65535
Resolution	1
Default	65535

Details

Same as :STATus:OPERation:PTRansition

:SYSTem:BASE:LICense:CHECk?

License Key Check

Function

Checks license and prepares installation if normal.

Query

```
:SYSTem:BASE:LICense:CHECk? <filename>
```

Response

<result>

Parameters

<filename>	License file name
<result>	Check result for license file
–1	Cannot be installed because executed for other module in same main frame
0	Normal
1	Specified file not found
2	Incorrect file
3	Damaged file or another MT8870 file
4	License already exists (Temporary license only)
5	Outside license period
6	Maximum number of licenses reached

Details

A software license file must be sent by FTP to a specified folder (/install/license/) in advance. Installation preparation is disabled if the unit is rebooted or the power is turned off before installing using the :SYSTem:BASE:LICense:INSTall command.

Example of Use

:SYSTem:BASE:LICense:CHECk? "Licenses.xml"
>0
:SYSTem:BASE:LICense:INSTall

:SYSTem:BASE:LICense:INFormation?

License Information

Function

Queries license expiration limit.

Query

```
:SYSTem:BASE:LICense:INFormation? <model>
```

Response

<kind>,<days>

Parameters

<model>	License name	
<kind>	License type	
0	Normal license	
1	Temporary license	
2	Trial license	
<days>	Remaining days	
	1 or more:	When the number of remaining days is 1, the license is on
the		the last day.
	0:	Expired license
	–1:	No limit (normal license)

Details

Specify the product model name (options included) in the license expiration limit query of the license name.

Example: MX887011A,MX887012A

Example of Use

:SYST:BASE:LIC:INF? "MX887012A"
>0,-1

:SYSTem:BASE:LICense:INSTall

License Key Install

Function

Installs license.

Command

```
:SYSTem:BASE:LICense:INSTall
```

Details

Installation is prepared using the SYSTem:BASE:LICense:CHECk? command. Remote connection of all modules is cut by the system reboot after executing this command.

Example of Use

:SYSTem:BASE:LICense:CHECk? "Licenses.xml"
>0
:SYSTem:BASE:LICense:INSTall

Remarks

The hardware option license is stored in the module. Changing the options for a specific module in the same MT8870A will stop the application starting. The licenses for software, software options, and waveforms are stored in the MT8870A. When a module is moved to another MT8870A, the application may not start due to the different license.

:SYSTem:BASE:REFerence:FREQuency

Frequency Reference

Function

Sets or queries frequency reference signal source.

Command

```
:SYSTem:BASE:REFerence:FREQuency <source>
```

Query

```
:SYSTem:BASE:REFerence:FREQuency?
```

Response

<source>

Parameter

<source>	Internal reference signal or external reference signal
INT	MT8870A internal reference signal
AUTO	Reference signal input to Ref Input connector at rear panel (uses internal reference signal if external signal not detected)
Default	AUTO

Details

The reference signal source is set for all modules installed in the MT8870A. It cannot be set for each module.

Example of Use

:SYST:BASE:REF:FREQ INT
:SYST:BASE:REF:FREQ?
> INT

Related Command

:SYSTem:BASE:REFerence:FREQuency:CONFigure

Remarks

Not initialized by *RST command

:SYSTem:BASE:REFerence:FREQuency:ADJust

Adjust Reference Clock

Function

Sets or queries the reference frequency correction value.

Command

```
:SYSTem:BASE:REFerence:FREQuency:ADJust <val>
```

Query

```
:SYSTem:BASE:REFerence:FREQuency:ADJust?
```

Response

<val>

Parameter

<val>	Adjustment value
Range	0 to 1023
Resolution	1
Default	512

Details

This is a common setting for all modules installed in the MT8870A.

Example of Use

:SYST:BASE:REF:FREQ:ADJ 512
:SYST:BASE:REF:FREQ:ADJ?
> 512

Remarks

No initialized by *RST command

:SYSTem:BASE:REFerence:FREQuency:ADJust:DEFault?

Adjust Reference Clock Default

Function

Queries factory default reference frequency correction value.

Query

```
:SYSTem:BASE:REFerence:FREQuency:ADJust:DEFault?
```

Response

<val>

Parameter

<val>

Range	0 to 1023
Resolution	1
Default	512

Details

This is a common value for all modules installed in the MT8870A.

Example of Use

:SYST:BASE:REF:FREQ:ADJ:DEF?
> 512

:SYSTem:BASE:REFerence:FREQuency:CONFigure

Frequency Reference

Function

Sets or queries frequency reference signal source.

Command

```
:SYSTem:BASE:REFerence:FREQuency:CONFigure <source>
```

Query

```
:SYSTem:BASE:REFerence:FREQuency:CONFigure?
```

Response

<source_res>

Parameter

<source>	Internal reference signal or external reference signal
10MHZINT	MT8870A internal reference signal
10MHZEXT	Reference signal input to Ref Input connector at rear panel (Internal reference signal is used if external signal not detected.)
13MHZEXT	Reference signal input to Ref Input connector at rear panel (Internal reference signal is used if external signal not detected.)
AUTO	Reference signal input to Ref Input connector at rear panel (Internal reference signal is used if external signal not detected.)
<source_res>	Internal reference signal or external reference signal
10MHZINT	MT8870A internal reference signal
AUTO	Reference signal input to Ref Input connector at rear panel (Internal reference signal is used if external signal not detected.)
Default	AUTO

Details

The reference signal source is set for all modules installed in the MT8870A. It cannot be set for each module.

Example of Use

:SYSTem:BASE:REFerence:FREQuency:CONFigure 10MHZINT
:SYSTem:BASE:REFerence:FREQuency:CONFigure?
> 10MHZINT

Related Command

:SYSTem:BASE:REFerence:FREQuency

Remarks

Not initialized by *RST command

:SYSTem:BASE:REFerence:FREQuency:CONFigure:SOURce?

Reference Signal Query

Function

Queries status of frequency reference signal source.

Query

```
:SYSTem:BASE:REFerence:FREQuency:CONFigure:SOURce?
```

Response

<source>

Parameter

<source>	Reference signal status
INT	Locked to internal reference signal source
VALI	Locked to external reference signal source
INVALID	Not locked to external reference signal source

Details

Same for all slots

Example of Use

:SYSTem:BASE:REFerence:FREQuency:CONFigure:SOURce?
> INT

Related Command

:SYSTem:BASE:REFerence:FREQuency:SOURce

:SYSTem:BASE:REFerence:FREQuency:SOURce?

Reference Signal Query

Function

Queries status of frequency reference signal source.

Query

```
:SYSTem:BASE:REFerence:FREQuency:SOURce?
```

Response

<source>

Parameter

<source>	Reference signal status
INT	Locked to internal reference signal source
EXT	Locked to external reference signal source
EXTU	Not locked to external reference signal source

Details

Same for all slots

Related Command

:SYSTem:BASE:REFerence:FREQuency:CONFigure:SOURce

Example of Use

:SYST:BASE:REF:FREQ:SOUR?

> EXT

:SYSTem:BASE:UPDate:FIRMware

Firmware Update

Function

Updates firmware package or queries update result.

Command

```
:SYSTem:BASE:UPDate:FIRMware <target>[,<option>]
```

Query

```
:SYSTem:BASE:UPDate:FIRMware?
```

Response

<result>

Parameter

<target>	Update target module
SINGLE	Single module
ALL	All modules in MT8870A
<option>	Option
DIFF	Updates only when version different
FORCE	Always updates
<result>	Firmware installation result
0	Normal
1	Failed
2	Aborted

Details

Updating must be prepared for using the :SYSTem:BASE:UPDate:FIRMware:CHECk? command, which reboots the system.
Modules cannot be controlled during rebooting. Choose whether to update all modules in the MT8870A, or a single module. The firmware update result can be checked using :SYSTem:BASE:UPDate:FIRMware?

Example of Use

:SYSTem:BASE:UPDate:FIRMware:CHECk? "package.inst"
>0
:SYSTem:BASE:UPDate:FIRMware ALL,DIFF

:SYSTem:BASE:UPDate:FIRMware:CHECk?

Check Firmware Update Package

Function

Checks firmware package prepares update if normal.

Query

```
:SYSTem:BASE:UPDate:FIRMware:CHECk? <filename>
```

Response

<result>

Parameters

<filename>	Firmware package file name
<result>	Result of firmware package file check
0	Normal
1	Specified file not found
2	Incorrect file
3	File damaged or not installer for machine
4	File version normal but not supported by this machine

Details

The firmware package file must be sent by FTP to a specified destination folder (/install/package/) in advance.

Updating preparation is disabled if the machine is rebooted or the power is turned off before updating the firmware using the :SYSTem:BASE:UPDate:FIRMware command.

Example of Use

:SYSTem:BASE:UPDate:FIRMware:CHECk? "package.inst"
>0
:SYSTem:BASE:UPDate:FIRMware ALL,DIFF

:SYSTem:BEEPer

Beep

Function

Sets or queries buzzer On/Off.

Command

```
:SYSTem:BEEPer <sw>
```

Query

```
:SYSTem:BEEPer?
```

Response

<sw>

Parameter

<sw>	Buzzer On/Off
ON	Buzzer enabled
OFF	Buzzer disabled

Details

It can be set for each module.

When the buzzer is enabled, audible signals are emitted in the following cases.

When an error is detected during self-diagnostic at boot-time:	One long audible signal
When the software starts at boot-time:	Three short audible signals
When the power is shut down or `:SYSTem:REBoot` or `SYSTem:SHUTdown` command is sent:	Two short audible signals
When an error is present in the sent command:	One short audible signal

Example of Use

:SYST:BEEP ON

:SYST:BEEP?

>ON

:SYSTem:COMMunicate:GPIB:ADDR

GPIB Address

Function

Sets or queries GPIB address.

Command

```
:SYSTem:COMMunicate:GPIB:ADDR <gp_address>
```

Query

```
:SYSTem:COMMunicate:GPIB:ADDR?
```

Response

<gp_address>	GPIB address

Parameter

<gp_address>	GPIB address	
Range	1 to 30	
Default	Slot 1	1
	Slot 2	1
	Slot 3	1
	Slot 4	1

Details

A GPIB address can be set for each module in each slot of the MT8870A.
When a module is moved to another slot, the slot GPIB address applies.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart.
The GPIB address changes to the initial value when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

```
:SYST:COMM:GPIB:ADDR 8
:SYST:COMM:NET:REST

:SYST:COMM:GPIB:ADDR?
> 8
```

:SYSTem:COMMunicate:GPIB:ADDR:ALL

GPIB Address

Function

Sets or queries the GPIB addresses of all modules.

Command

```
:SYSTem:COMMunicate:GPIB:ADDR:ALL
<address1>,<address2>,<address3>,<address4>
```

Query

```
:SYSTem:COMMunicate:GPIB:ADDR:ALL?
```

Response

<address1,address2,address3,address4>

Parameter

<address1>	GPIB address of Module 1
Range	1 to 30
Default	1
<address2>	GPIB address of Module 2
Range	1 to 30
Default	1
<address3>	GPIB address of Module 3
Range	1 to 30
Default	1
<address4>	GPIB address of Module 4
Range	1 to 30
Default	1

Details

This command can set the GPIB addresses of all the modules.
GPIB address is set for the slot of MT8870A.
When a module is moved to another slot, GPIB address of the slot is applied.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
Pressing the IP Reset button on the MT8870A panel at power-on resets the GPIB addresses to the default.

Example of Use

:SYST:COMM:GPIB:ADDR:ALL 1,2,3,4
:SYST:COMM:NET:REST

:SYST:COMM:GPIB:ADDR:ALL?

> 1,2,3,4

:SYSTem:COMMunicate:GPIB[:SELF]:DELimiter

Delimiter

Function

Sets or queries delimiter (characters added to message end).

Command

```
:SYSTem:COMMunicate:GPIB[:SELF]:DELimiter <code>
```

Query

```
:SYSTem:COMMunicate:GPIB[:SELF]:DELimiter?
```

Response

<code>

Parameter

<code>	End code
LF	LF (Line Feed)
CRLF	CR/LF (Carriage Return + Line Feed)
NONE	None (only EOI)
Default	CRLF

Details

It can be set for each module.
The delimiter changes to the initial value when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

:SYST:COMM:GPIB:DEL CRLF
:SYST:COMM:GPIB:DEL?
>CRLF

Remarks

The processing is the same as :SYSTem:COMMunicate:GPIB[:SELF]:TERMinator.
The *RST, :SYSTem:COMMunicate:NET:RESTart, and :SYSTem:REBoot commands do not perform initialization.
When the parameter is set to NONE in Ethernet connection, the delimiter of the response message from the main unit is set to CRLF.

:SYSTem:COMMunicate:GPIB[:SELF]:TERMinator

Terminator

Function

Sets or queries terminator (characters added to message end).

Command

```
:SYSTem:COMMunicate:GPIB[:SELF]:TERMinator <code>
```

Query

```
:SYSTem:COMMunicate:GPIB[:SELF]:TERMinator?
```

Response

<code>	End code

Parameter

<code>	End code
LF	LF (Line Feed)
CRLF	CR/LF (Carriage Return + Line Feed)
NONE	None (only EOI)
Default	CRLF

Details

It can be set for each module.

Pressing the IP Reset button on the MT8870A panel at power-on switches the termination code to the initial value.

Example of Use

:SYST:COMM:GPIB:TERM CRLF
:SYST:COMM:GPIB:TERM?
>CRLF

Remarks

Same processing as :SYSTem:COMMunicate:GPIB[:SELF]:DELimiter

No initialization with *RST, :SYSTem:REBoot and :SYSTem:COMMunicate:NET:RESTart commands.

When the parameter is set to NONE in Ethernet connection, the terminator of the response message from the main unit is set to CRLF.

:SYSTem:COMMunicate:GTL

Go to Local

Function

Disconnects remote connection.

Command

`:SYSTem:COMMunicate:GTL`

Details

The Remote lamp on the module panel goes off.

Example of Use

:SYST:COMM:GTL

:SYSTem:COMMunicate:NET:DNS:DOMain

Domain Name

Function

Sets or queries domain name.

Command

```
:SYSTem:COMMunicate:NET:DNS:DOMain <name>
```

Query

```
:SYSTem:COMMunicate:NET:DNS:DOMain?
```

Response

<name>	Domain name

Parameter

<name> Domain name

Maximum	63 bytes
Default	"DOMAIN"

A parameter error occurs if the naming convention is not followed at setup.

- The first character must be an alphabetic character.
- Intermediate characters must alphanumeric characters or a hyphen.
- The last character must be an alphanumeric character.
- Characters must be enclosed in double quotation marks ("").

Details

The domain name is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart.
The domain name changes to the initial value when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

```
:SYST:COMM:NET:DNS:DOM "Domain"
:SYST:COMM:NET:REST

:SYST:COMM:NET:DNS:DOM?
> Domain
```

:SYSTem:COMMunicate:NET:DNS:HOST

Host Name

Function

Sets or queries host name.

Command

```
:SYSTem:COMMunicate:NET:DNS:HOST <host>
```

Query

```
:SYSTem:COMMunicate:NET:DNS:HOST?
```

Response

<host>	Host name

Parameter

<host> Host name

Minimum	1 byte
Maximum	61 byte
Default	"HOST"

A parameter error occurs if the naming convention is not followed at setup.

- The first character must be an alphanumeric character.
- Intermediate characters must alphanumeric characters or a hyphen.
- The last character must be an alphanumeric character.
- Characters must be enclosed in double quotation marks ("").

Details

The slot number is added to the end of the set host name (HOSTNAME_- Slot number).
The host name is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart.
The host name changes to the initial value when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

```
:SYST:COMM:NET:DNS:HOST "ANRITSU-PC"
:SYST:COMM:NET:REST

:SYST:COMM:NET:DNS:HOST?
> ANRITSU-PC-1
```

:SYSTem:COMMunicate:NET:FTP:USER:CATalog?

FTP User List

Function

Queries all FTP users names.

Query

```
:SYSTem:COMMunicate:NET:FTP:USER:CATalog?
```

Response

<name>,<name>,.....

Parameter

<name>	User name

Example of Use

:SYST:COMM:NET:FTP:USER:CAT?

> user1,user2,user3

:SYSTem:COMMunicate:NET:FTP:USER:CHANge:NAME

Change FTP User Name

Function

Changes FTP user name.

Command

```
:SYSTem:COMMunicate:NET:FTP:USER:CHANge:NAME <before>,<after>
```

Parameters

<before>	User name before change
<after>	User name after change

Details

The user name character string must be 1 to 31 alphanumeric characters.
Set to enclose in double quotation marks ("").
The head character may be an upper or lower-case alphabetic character, a period (.) or an underscore (_). Second and subsequent characters may be upper or lower-case alphanumerics, periods (.) or underscores (_). A name composed entirely numeric values is prohibited.

Example of Use

To change user1 to user2.
:SYST:COMM:NET:FTP:USER:CHAN:NAME "user1","user2"

:SYSTem:COMMunicate:NET:FTP:USER:CHANGe:PASS

Change FTP User Password

Function

Changes FTP user password.

Command

```
:SYSTem:COMMunicate:NET:FTP:USER:CHANGe:PASS <name>,<before>,<after>
```

Parameters

<name>	User name for password change
<before>	Password before change
<after>	Password after change

Details

The user name password must be 1 to 16 upper or lower-case alphanumeric characters and symbols (! #$%'()-=^~@[{;+]},.?_).
Set to enclose in double quotation marks ("").

Example of Use

:SYST:COMM:NET:FTP:USER:CHAN:PASS "user1","abcde","fghij"

:SYSTem:COMMunicate:NET:HWADdress?

MAC Address

Function

Queries MAC address.

Query

```
:SYSTem:COMMunicate:NET:HWADdress?
```

Response

<macaddr>

Parameter

<macaddr> MAC address

Details

Queries MAC address of MU887000A communicating currently.

Example of Use

:SYST:COMM:NET:HWAD?

> 00-00-00-00-00-00

:SYSTem:COMMunicate:NET:IPVersion

IP Version

Function

Sets or queries IPv4 and IPV6 On/Off.
When On is set, the module communicates using that IP version.

Command

```
:SYSTem:COMMunicate:NET:IPVersion <ipver>
```

Query

```
:SYSTem:COMMunicate:NET:IPVersion?
```

Response

<ipver>

Parameter

<ipver>	IP Version
V4	Only IPv4 set to On
V6	Only IPv6 set to On
V4V6	Both IPv4 and IPv6 set to On
Default	IPv4

Details

The IP version is set to On/Off for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
Pressing the IP Reset button on the MT8870A panel at power-on switches the IP version to the initial value.

Example of Use

:SYST:COMM:NET:IPV V4V6
:SYST:COMM:NET:REST

After restart,
:SYST:COMM:NET:IPV?
> V4V6

Remarks

Not initialized by *RST and :SYSTem:REBoot commands

:SYSTem:COMMunicate:NET:IPVFour:ADDRess:TYPE

IPv4 Type

Function

Sets or queries IPv4 address type.

Command

```
:SYSTem:COMMunicate:NET:IPVFour:ADDRess:TYPE <ipv4type>
```

Query

```
:SYSTem:COMMunicate:NET:IPVFour:ADDRess:TYPE?
```

Response

<ipv4type>

Parameter

<ipv4type>	IPv4 address type
STATIC	Fixed IP
DHCP	Acquires IPv4 from DHCP server and set
Default	STATIC

Details

The IPv4 address type is set for all slots in the MT8870A.
When STATIC is set, set the address, subnet mask, and default gateway using the following commands:
:SYSTem:COMMunicate:NET:IPVFour:STATic:IPADdress
:SYSTem:COMMunicate:NET:IPVFour:STATic:SMASk
:SYSTem:COMMunicate:NET:IPVFour:STATic:GIP

To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
The IPv4 address type is set for the MT8870A slot.
When a module is moved to another slot, IPv4 address type of the slot is applied.
Pressing the IP Reset button on the MT8870A panel at power-on sets the IPV4 address type to the initial value.

Example of Use

:SYST:COMM:NET:IPVF:ADDR:TYPE DHCP
:SYST:COMM:NET:REST

After restart
:SYST:COMM:NET:IPVF:ADDR:TYPE TYPE?
> DHCP

:SYSTem:COMMunicate:NET:IPVFour:CURRent:GIP?

Get Current Default Gateway

Function

Queries current default gateway address.

Query

```
:SYSTem:COMMunicate:NET:IPVFour:CURRent:GIP?
```

Response

<gateway>

Parameter

<gateway>	Default gateway address

Details

The default gateway address for the module is acquired.
The response is 0 when the module IPv4 address is set to Off.

Example of Use

:SYST:COMM:NET:IPVF:CURR:GIP?
> 192.168.0.1

:SYSTem:COMMunicate:NET:IPVFour:CURRent:IPADdress?

Get Current IPv4 Address

Function

Queries module IPv4 address.

Query

```
:SYSTem:COMMunicate:NET:IPVFour:CURRent:IPADdress?
```

Response

<num>	Number of set IP address
<ipaddr>	IP address

Parameters

<num>	Number of set IP address
<ipaddr>	IP address

Details

The module IPv4 address is acquired.
The response is 0 when the module IPv4 address is set to Off.

Example of Use

:SYST:COMM:NET:IPVF:CURR:IPAD?
> 2,192.168.0.1,172.16.0.31

:SYSTem:COMMunicate:NET:IPVFour:CURRent:SMASk?

Get Current Subnet Mask

Function

Queries current subnet mask.

Query

`:SYSTem:COMMunicate:NET:IPVFour:CURRent:SMASk?`

Response

<mask> Subnet mask

Details

The subnet mask of the module is acquired.
The response is 0 when the module IPv4 address is set to Off.

Example of Use

:SYST:COMM:NET:IPVF:CURR:SMAS?
> 255.255.255.0

:SYSTem:COMMunicate:NET:IPVFour:DNS:AUTO:ENABle

DNS Server Auto

Function

Sets and queries whether to acquire DNS server address automatically.

Command

```
:SYSTem:COMMunicate:NET:IPVFour:DNS:AUTO:ENABle <switch>
```

Query

```
:SYSTem:COMMunicate:NET:IPVFour:DNS:AUTO:ENABle?
```

Response

<switch>	DNS server auto-address acquisition function On/Off setting

Parameter

<switch>	Sets DNS server auto-address acquisition function On/Off
ON	ON set
OFF	Off set

Details

This setting is applied to all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
The DNS server auto-address acquisition function setting changes to OFF when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

```
:SYST:COMM:NET:IPVF:DNS:AUTO:ENAB ON
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVF:DNS:AUTO:ENAB?
>ON
```

:SYSTem:COMMunicate:NET:IPVFour:DNS:PRIMary

DNS Primary Address

Function

Sets or queries DNS primary address (IPv4).

Command

```
:SYSTem:COMMunicate:NET:IPVFour:DNS:PRIMary <ip4addr>
```

Query

```
:SYSTem:COMMunicate:NET:IPVFour:DNS:PRIMary?
```

Response

<ip4addr>

Parameter

<ip4addr>	DNS primary address

Details

The DNS primary address is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
Pressing the IP Reset button on the MT8870A panel at power-on switches the DNS primary address to 0.

Example of Use

:SYST:COMM:NET:IPVF:DNS:PRIM 192.168.10.1
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVF:DNS:PRIM?
>192.168.10.1

:SYSTem:COMMunicate:NET:IPVFour:DNS:SECondary

DNS Secondary Address

Function

Sets or queries DNS secondary address (IPv4).

Command

```
:SYSTem:COMMunicate:NET:IPVFour:DNS:SECondary <ip4addr>
```

Query

```
:SYSTem:COMMunicate:NET:IPVFour:DNS:SECondary?
```

Response

<ip4addr>

Parameter

<ip4addr>	DNS secondary address

Details

The DNS secondary address is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
Pressing the IP Reset button on the MT8870A panel at power-on switches the DNS secondary address to 0.

Example of Use

```
:SYST:COMM:NET:IPVFr:DNS:SEC 192.168.10.2
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVF:DNS:SEC?
>192.168.10.2
```

:SYSTem:COMMunicate:NET:IPVFour:STATic:GIP

Default Gateway

Function

Sets or queries fixed IPv4 setting default gateway.

Command

```
:SYSTem:COMMunicate:NET:IPVFour:STATic:GIP <gateway>
```

Query

```
:SYSTem:COMMunicate:NET:IPVFour:STATic:GIP?
```

Response

<gateway>

Parameter

<gateway>	Default gateway address xxx.xxx.xxx.xxx

Details

The default gateway is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
If a blank space is specified in the parameters, the default gateway setting is initialized.
Pressing the IP Reset button on the MT8870A panel at power-on switches the default gateway to 0.

Example of Use

:SYST:COMM:NET:IPVF:STAT:GIP "192.168.0.1"
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVF:STAT:GIP?
> 192.168.0.1

:SYSTem:COMMunicate:NET:IPVFour:STATic:IPADdress

IPv4 Address

Function

Sets or queries module fixed IPv4 address.

Command

```
:SYSTem:COMMunicate:NET:IPVFour:STATic:IPADdress
<ipv4_addr1>,<ipv4_addr2>,<ipv4_addr3>,<ipv4_addr4>
```

Query

```
:SYSTem:COMMunicate:NET:IPVFour:STATic:IPADdress?
```

Response

<ipv4_addr1>,<ipv4_addr2>,<ipv4_addr3>,<ipv4_addr4>

Parameter

<ipv4_addr1>	IPv4 address of Slot 1
<ipv4_addr2>	IPv4 address of Slot 2
<ipv4_addr3>	IPv4 address of Slot 3
<ipv4_addr4>	IPv4 address of Slot 4

Details

The IPv4 address of each slot of the MT8870A is set. It can be set without an inserted module.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
The IPv4 address is set for the MT8870A slot.
When a module is moved to another slot, the IPv4 address of the slot applies.
Pressing the IP Reset button on the MT8870A panel at power-on does not clear the command setting.

Example of Use

```
:SYST:COMM:NET:IPVF:STAT:IPAD "192.168.1.1","192.168.1.2","192.168.1.3","192.168.1.4"
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVF:STAT:IPAD?
> 192.168.1.1,192.168.1.2,192.168.1.3,192.168.1.4
```

:SYSTem:COMMunicate:NET:IPVFour:STATic:SMASk

Subnet Mask

Function

Sets or queries fixed IPv4 address subnet mask.

Command

```
:SYSTem:COMMunicate:NET:IPVFour:STATic:SMASk <mask>
```

Query

```
:SYSTem:COMMunicate:NET:IPVFour:STATic:SMASk?
```

Response

<mask>	Subnet mask

Parameter

<mask>	Subnet mask

Details

The subnet mask is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
Pressing the IP Reset button on the MT8870A panel at power-on switches the subnet mask to 255.255.255.0

Example of Use

:SYST:COMM:NET:IPVF:STAT:SMAS "255.255.255.0"
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVF:STAT:SMAS?
> 255.255.255.0

:SYSTem:COMMunicate:NET:IPVSix:ADDRess:TYPE

IPv6 Type

Function

Sets or queries IPv6 address type.

Command

```
:SYSTem:COMMunicate:NET:IPVSix:ADDRess:TYPE <ipv6type>
```

Query

```
:SYSTem:COMMunicate:NET:IPVSix:ADDRess:TYPE?
```

Response

<ipv6type>

Parameter

<ipv6type>	IPv6 address type
STATIC	Fixed IP
RA_EUI64	Automatic allocation by RA based on EUI-64 (RFC4862)
RA_RANDOM	Automatic allocation by RA based on privacy expansion (RFC4941)
DHCPV6	Acquires address from DHCP server and sets
Default	STATIC

Details

The IPv6 address type is set for all slots in the MT8870A.
Set the address using the following commands when setting STATIC.
:SYSTem:COMMunicate:NET:IPVSix:STATic:IPADdress
:SYSTem:COMMunicate:NET:IPVSix:STATic:DEFRouter

To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
The IPv6 address type is set for the MT8870A slots.
When a module is moved to another slot, the IPv6 address type is applied to that slot.
Pressing the IP Reset button on the MT8870A panel at power-on switches the IPv6 address type to the initial value.

Example of Use

```
:SYST:COMM:NET:IPVS:ADDR:TYPE RA_EUI64
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVS:ADDR:TYPE?
> RA_EUI64
```

:SYSTem:COMMunicate:NET:IPVSix:CURRent:DEFRouter?

Get Default Router Address

Function

Queries current default router address.

Query

```
:SYSTem:COMMunicate:NET:IPVSix:CURRent:DEFRouter?
```

Response

<ip6addr> Default router address

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

Details

The IPv6 address/prefix length is queried for the module being communicated with.

The response is 0 when the module IPv6 address is set to Off.

Example of Use

:SYST:COMM:NET:IPVS:CURR:DEFR?

> 2001:DB8::1

:SYSTem:COMMunicate:NET:IPVSix:CURRent:IPADdress?

Get Current IPv6 Address and Prefix

Function

Queries current IPv6 address/prefix length.

Query

```
:SYSTem:COMMunicate:NET:IPVSix:CURRent:IPADdress?
```

Response

<num> Number of set IP address

<ip6addr> IPv6 address and prefix length

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx

Details

The IPv6 address and prefix length of the module are queried.

The response is 0 when the module IPv6 address is set to Off.

Example of Use

:SYST:COMM:NET:IPVS:CURR:IPAD?

> 2,2001:DB8::1/64,2001:DB8::2/64

:SYSTem:COMMunicate:NET:IPVSix:CURRent:LOCaladdress?

Get Link Local Address

Function

Queries current link local address/prefix length.

Query

```
:SYSTem:COMMunicate:NET:IPVSix:CURRent:LOCaladdress?
```

Response

<ip6addr>

Parameter

<ip6addr>	Link local address/prefix length xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx

Details

The link local address and prefix length are set for the module being communicated with.
The response is 0 when the module IPv6 address is set to Off.

Example of Use

:SYST:COMM:NET:IPVS:CURR:LOC?
> 2001:DB8::1/64

:SYSTem:COMMunicate:NET:IPVSix:DNS:AUTO:ENABle

DNS Server Auto

Function

Sets and queries DNS server auto-address acquisition.

Command

```
:SYSTem:COMMunicate:NET:IPVSix:DNS:AUTO:ENABle <switch>
```

Query

```
:SYSTem:COMMunicate:NET:IPVSix:DNS:AUTO:ENABle?
```

Response

<switch>	DNS server auto-address acquisition On/Off setting

Parameter

<switch>	Sets DNS server auto-address acquisition On/Off
ON	On set
OFF	Off set

Details

This setting is applied to all modules installed in the MT8870A.
To enable this command, send the :SYSTem:COMMunicate:NET:RESTart command.
The DNS server auto-address acquisition function setting changes to OFF when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

```
:SYST:COMM:NET:IPVS:DNS:AUTO:ENAB ON
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVS:DNS:AUTO:ENAB?
>ON
```

:SYSTem:COMMunicate:NET:IPVSix:DNS:PRIMary

DNS Primary Address

Function

Sets or queries DNS primary address (IPv6).

Command

```
:SYSTem:COMMunicate:NET:IPVSix:DNS:PRIMary <ip6addr>
```

Query

```
:SYSTem:COMMunicate:NET:IPVSix:DNS:PRIMary?
```

Response

<ip6addr>	DNS primary address

Parameter

<ip6addr>	DNS primary address

Details

The DNS primary address is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
Pressing the IP Reset button on the MT8870A panel at power-on sets the DNS primary address to primary address to 0.

Example of Use

```
:SYST:COMM:NET:IPVS:DNS:PRIM "fe80::215:c5ff:febb:42cf/128"
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVS:DNS:PRIM?
> fe80::215:c5ff:febb:42cf/128
```

5

SCPI Command Reference

:SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary

DNS Secondary Address

Function

Sets or queries DNS secondary address (IPv6).

Command

```
:SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary <ip6addr>
```

Query

```
:SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary?
```

Response

<ip6addr>	DNS secondary address

Parameter

<ip6addr>	DNS secondary address

Details

The DNS secondary address is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
Pressing the IP Reset button on the MT8870A panel at power-on sets the DNS secondary address to primary address to 0.

Example of Use

:SYST:COMM:NET:IPVSix:DNS:SEC "fe80::215:c5ff:febb:42ce"
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVSix:DNS:SEC?
> fe80::215:c5ff:febb:42ce

:SYSTem:COMMunicate:NET:IPVSix:STATic:DEFRouter

Default Router

Function

Sets or queries fixed IPv6 setting default router address.

Command

```
:SYSTem:COMMunicate:NET:IPVSix:STATic:DEFRouter <ip6addr>
```

Query

```
:SYSTem:COMMunicate:NET:IPVSix:STATic:DEFRouter?
```

Response

<ip6addr>

Parameter

<ip6addr>	Default router address

Details

The default router address is set for all modules installed in the MT8870A.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
If a blank space is specified in the parameters, the default router address setting is initialized.
Pressing the IP Reset button on the MT8870A panel at power-on switches the default router address to 0.

Example of Use

```
:SYST:COMM:NET:IPVS:STAT:DEFR "2001:DB8::1"
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVS:STAT:DEFR?
> 2001:DB8::1
```

:SYSTem:COMMunicate:NET:IPVSix:STATic:IPADdress

IPv6 Address

Function

Sets or queries fixed IPv6 setting IPv6 address/prefix length.

Command

```
:SYSTem:COMMunicate:NET:IPVSix:STATic:IPADdress
<ipv6_addr1>,<ipv6_addr2>,<ipv6_addr3>,<ipv6_addr4>
```

Query

```
:SYSTem:COMMunicate:NET:IPVSix:STATic:IPADdress?
```

Response

<ipv6_addr1>,<ipv6_addr21>,<ipv6_addr3>,<ipv6_addr4>

Parameters

<ipv6_addr1>	IPv6 address/prefix length to Slot 1
<ipv6_addr2>	IPv6 address/prefix length to Slot 2
<ipv6_addr3>	IPv6 address/prefix length to Slot 3
<ipv6_addr4>	IPv6 address/prefix length to Slot 4

Details

The IPv6 address is set for a slot of the MT8870A. It can be set for a slot without an inserted module.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.
The IPv6 address is set for the MT8870A slot.
When a module is moved to another slot, the IPv6 address is applied to that slot.

Example of Use

```
:SYST:COMM:NET:IPVS:STAT:IPAD
"2001:DB8::1/64","2001:DB8::2/64","2001:DB8::3/64","2001:DB8::4/64"
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVS:STAT:IPAD?
> 2001:DB8::1/64,2001:DB8::2/64,2001:DB8::3/64,2001:DB8::4/64
```

:SYSTem:COMMunicate:NET:PORT:SETTing

Raw Socket Port Number

Function

Sets or queries port number at RawSocket communication.

Command

```
:SYSTem:COMMunicate:NET:PORT:SETTing <port>
```

Query

```
:SYSTem:COMMunicate:NET:PORT:SETTing?
```

Response

<port>	Port number

Parameter

<port>	Port number
Range	5001 to 60000
Default	56001

Details

The TCP/IP port number is set when performing RawSocket communication with the MT8870A.
The port number is common to IPv4 and IPv6.
To enable this command setting, send the :SYSTem:COMMunicate:NET:RESTart command.

Example of Use

```
:SYSTem:COMMunicate:NET:PORT:SETTing 56001
:SYSTem:COMMunicate:NET:RESTart
:SYSTem:COMMunicate:NET:PORT:SETTing?
> 56001
```

:SYSTem:COMMunicate:NET:RESTart

Network Restart

Function

Restarts network.

Command

```
:SYSTem:COMMunicate:NET:RESTart
```

Details

Sending this command reboots the system. Remote control of all modules is disabled at restart. To continue remote control, wait a few seconds after sending this command and then reconnect remote control.

Example of Use

:SYST:COMM:NET:REST

Remarks

Communication setting not updated by *RST and :SYSTem:REBoot

:SYSTem:DATE

Set Date

Function

Sets or queries date.

Command

```
:SYSTem:DATE <year>,<month>,<day>,<hour>,<minute>,<second>
```

Query

```
:SYSTem:DATE?
```

Response

<year>,<month>,<day>,<hour>,<minute>,<second>

Parameters

<year>	Year
Range	2012 to 2027
Resolution	1
<month>	Month
Range	1 to 12
Resolution	1
<day>	Day
Range	1 to 31
Resolution	1
<hour>	Hour
Range	0 to 23
Resolution	1
<minute>	Minutes
Range	0 to 59
Resolution	1
<second>	Second
Range	0 to 59
Resolution	1

Example of Use

:SYSTem:DATE 2012,9,13,11,6,17

:SYSTem:DATE?

> 2012,9,13,11,6,17

Remarks

The response is "***" when the date and time setting fails.

:SYSTem:ERRor?

System Error

Function

Queries error messages in error/event queue.

Query

`:SYSTem:ERRor?`

Response

<code>,<msg>

Parameters

<code>	Error code
0	"No Error"
–113	"Undefined header"
–200	"Execution error"
–220	"Parameter error"
–310	"Option license is required to execute the command"
–370	"Main frame upgrade is required to execute the command"
<msg>	Event message corresponding to error/event number

Details

The response for no error/event queue is "0, "No Error"".

Example of Use

:SYST:ERR?

> -220,"Parameter error"

:SYSTem:ERRor:ALL?

System All Errors

Function

Queries error messages in error/event queue.

Query

```
:SYSTem:ERRor:ALL?
```

Response

<code>,<msg>,<code>,<msg>.......

Parametera

<code>	Error code
0	"No Error"
–113	"Undefined header"
–200	"Execution error"
–220	"Parameter error"
–310	"Option license is required to execute the command"
–370	"Main frame upgrade is required to execute the command"
<msg>	Event message corresponding to error/event number

Details

The response for no error/event queue is "0, "No Error"".
After executing this command, the error/event queue is emptied.

Example of Use

:SYST:ERR:ALL?
> -113,"Undefined header",-220,"Parameter error"

:SYSTem:ERRor:CODE:ALL?

System All Error Codes

Function

Queries all error codes in error/event queue.

Query

`:SYSTem:ERRor:CODE:ALL?`

Response

<code>,<code>,....

Parameter

<code>	Error code
0	"No Error"
–113	"Undefined header"
–200	"Execution error"
–220	"Parameter error"
–310	"Option license is required to execute the command"
–370	"Main frame upgrade is required to execute the command"

Details

The response for no error/event queue is 0.
After executing this command, the error/event queue is emptied.

Example of Use

:SYST:ERR:CODE:ALL?
> -113,-220

:SYSTem:ERRor:CODE[:NEXT]?

System Error Code

Function

Queries error codes in error/event queue.

Query

```
:SYSTem:ERRor:CODE[:NEXT]?
```

Response

<code>

Parameter

<code>	Error code
0	"No Error"
–113	"Undefined header"
–200	"Execution error"
–220	"Parameter error"
–310	"Option license is required to execute the command"
–370	"Main frame upgrade is required to execute the command"

Details

The response for no error/event queue is 0.

Example of Use

:SYST:ERR:CODE?
> -220

:SYSTem:ERRor:COUNt?

System Error Count

Function

Queries number of all errors/events in error/event queue.

Query

```
:SYSTem:ERRor:COUNt?
```

Response

<num>

Parameter

<num>	Number of errors/events

Details

Up to four errors are saved in the error event queue.
If a fifth error occurs, it is saved in the error event queue by deleting the oldest error in the queue.
The :SYSTem:ERRor? and :SYSTem:ERRor:CODE? commands can be used to delete errors in the event queue sequentially from oldest first.
The :SYSTem:ERRor:ALL? and :SYSTem:ERRor:CODE:ALL? commands can be used to delete all errors and empty the event queue.

Example of Use

:SYST:ERR:COUN?
> 3

:SYSTem:INFormation?

Module Information

Function

Queries module information.

Query

```
:SYSTem:INFormation? <index>
```

Response

When parameter is 1

model name	Product name (≤30 characters)

When parameter is 2

model number	Model name (≤20 characters)

When parameter is 3

BIOS version	BIOS (IPL) version (≤10 characters)

When parameter is 4

OS version	OS version (≤255 characters)

When parameter is 5

number of HW,name/rev,name/rev,...,name/rev

number of HW:	Number of hardware units
name:	Hardware name (≤30 characters, bundled with " ")
rev:	Hardware revision (≤10 characters)

When parameter is 6

number of FPGA,name/ver,name/ver,...,name/ver

number of FPGA:	Number of FPGAs
name:	FPGA name (≤30 characters, bundled with " ")
ver:	FPGA version (“xx.xx” is described)

When parameter is 7

number of HW option,num/name/sw,num/name/sw,...,num/name/sw

number of HW option:	Number of hardware options
num:	Option number (0 to 999)
name:	Option name (≤50 characters, bundled with " ")
sw:	Option switch
ON :	With options
OFF :	Without options

When parameter is 8

0 always

When parameter is 9

0 always

When parameter is 10

0 always

When parameter is 0 or omitted

Character strings of all above responses comma separated

Parameter

<index>	Index of information to acquire
Range	1 to 10

Example of Use

:SYST:INF? 1

> TRX Test Module

:SYST:INF? 2

> MU887000A

:SYST:INF? 3

> 1.0

:SYST:INF? 4

> 2.6

:SYST:INF? 5

> 2, "Main Board/01", "Option Board/FE"

:SYST:INF? 6

> 2, "High Speed Control FPGA/01.00","Wide Band FPGA/FF.00"

:SYST:INF? 7

> 2,001/"Rubidium"/ON,009/"GPIB Interface"/ON

:SYST:INF? 8

> 0

:SYST:INF? 9

> 0

:SYST:INF? 10

> 0

Related Command

:SYSTem:INFormation:MAINframe?

Remarks

If the installed temporary license has expired, the license information cannot be obtained using this command.

:SYSTem:INFormation:DEVice:ID?

Module Serial No.

Function

Queries module serial number.

Query

```
:SYSTem:INFormation:DEVice:ID?
```

Response

<serial>

Parameter

<serial>	Serial number 10-digit number Example 6200103056

Example of Use

:SYST:INF:DEV:ID?
> 6123456789

:SYSTem:INFormation:MAINframe?

Main Frame Information

Function

Queries MT8870A information.

Query

```
:SYSTem:INFormation:MAINframe? <index>
```

Response

When parameter is 1

model name	Product name (≤30 characters)

When parameter is 2

model number	Model name (≤20 characters)

When parameter is 3

BIOS version	0 always

When parameter is 4

OS version	0 always

When parameter is 5

number of HW,name/rev,name/rev,...,name/rev

number of HW:	Number of hardware units

name: Hardware name (≤30 characters, bundled with " ")
rev: Hardware revision (≤10 characters)

When parameter is 6
number of FPGA,name/ver,name/ver,...,name/ver
number of FPGA: Number of FPGAs
name: FPGA name (≤30 characters, bundled with " ")
ver: FPGA version (≤10 characters)

When parameter is 7
number of HW option,num/name/sw,num/name/sw,...,num/name/sw
number of HW option: Number of hardware options
num: Option number (0 to 999)
name: Option name (≤50 characters)
sw: Option switch
    ON : With options
    OFF : Without options

When parameter is 8
number of Software license,model/serial/name/ver,model/serial/name/ver,...,model/serial/name/ver
model: Software name (≤30 characters)
serial: Serial number (10 characters)
name: Software license name (≤50 characters, bundled with " ")
ver: Software version ("xx.xx" is described)

When parameter is 9
number of SW option,num/name/sw,num/name/sw,...,num/name/sw
number of SW option: Number of software options
num: Option name (≤20 characters MX887000A-001)
name: Option name (≤50 characters, bundled with " ")
sw: Option switch
    ON : With options
    OFF : Without options

When parameter is 10
number of SG wave license,model/serial/name/maxver,model/serial/name/maxver,...,model/serial/name/maxver
number of SG wave license: Number of SG waveform licenses
model: SG waveform license name (≤30 characters)
serial: Serial number (10 characters)
name: SG waveform license name (≤50 characters, bundled with " ")
maxver: Usable maximum version (00.00 to 99.99)

When parameter is 0 or omitted
Character strings of all above responses comma separated

Parameter

<index> Index of information to acquire

Example of Use

:SYST:INF:MAIN? 1
> > Universal Wireless Test Set
:SYST:INF:MAIN? 2
> MT8870A
:SYST:INF:MAIN? 3
> 1.0
:SYST:INF:MAIN? 4
> 2.6
:SYST:INF:MAIN? 5
> 2, "Main Board/01","Option Board/FE"
:SYST:INF:MAIN? 6
> 2, "High Speed Control FPGA/01","Wide Band FPGA/FF"
:SYST:INF:MAIN? 7
> 2,001/"Rubidium"/ON,009/"GPIB Interface"/ON
:SYST:INF:MAIN? 8
> 2,MX887010A/6123456789/"Cellular Measurement Software"/01.01,MX887020A/6123456789/"Vector SG Software"/01.00
:SYST:INF:MAIN? 9
> 2,MX887000A-001/"Extended Digitizing Option"/ON,MX887000A-002/"Phase Noise Measurement"/OFF
:SYST:INF:MAIN? 10
> 1,MX887070A/6123456789/Standard wave license/5

Related Command

:SYSTem:INFormation?

Remarks

If the installed temporary license has expired, the license information cannot be obtained using this command.

:SYSTem:INFormation:MAINframe:DEVice:ID?

Main Frame Serial No.

Function

Queries MT8870A serial number.

Query

```
:SYSTem:INFormation:MAINframe:DEVice:ID?
```

Response

<serial>

Parameter

<serial>	Serial number 10-digit number Example 6200112233

Example of Use

:SYST:INF:MAIN:DEV:ID?
> 6123456789

:SYSTem:INFormation:MAINframe:PACKage:VERSion?

Firmware Version

Function

Queries the firmware version.

Query

```
:SYSTem:INFormation:MAINframe:PACKage:VERSion?
```

Response

<ver>

Parameter

<ver>	Firmware version (Written as "xx.xx.xx")

Example of Use

:SYSTem:INFormation:MAINframe:PACKage:VERSion?
> 1.00.00

:SYSTem:INFormation:MAINframe:POWeron:COUNt?

Chassis Power ON Count

Function

Queries MT8870A power-on count.

Query

```
:SYSTem:INFormation:MAINframe:POWeron:COUNt?
```

Response

<num>

Parameter

<num>	MT8870A power-on count

Example of Use

:SYSTem:INFormation:MAINframe:POWeron:COUNt?
> 1971

:SYSTem:INFormation:MAINframe:RTIMe?

Main Frame Running Time Count Read

Function

Queries MT8870A total running time.

Query

```
:SYSTem:INFormation:MAINframe:RTIMe?
```

Response

<time>

Parameter

<time>	Total running time (minutes)

Example of Use

:SYST:INF:MAIN:RTIM?

> 1234

:SYSTem:INFormation:MAINframe:SOFTware?

Type Name of Software

Function

Queries the software model.

Query

```
:SYSTem:INFormation:MAINframe:SOFTware?
```

Response

<n>,<model>,…,<model>

Parameters

<n>	Number of software
<model>	Software model

Example of Use

:SYST:INF:MAIN:SOFT?

> 3,MX887010A,MX887011A,MX887012A

Remarks

The registered model names in Anritsu's ordering information can be only obtained.

:SYSTem:INFormation:MAINframe:SOFTware:OPTion?

Option Number

Function

Queries the software option number.

Query

```
:SYSTem:INFormation:MAINframe:SOFTware:OPTion? <model>
```

Response

<n>,<num>,...,<num>

Parameters

<model>	Software model
<n>	Number of options
<num>	Option number

Example of Use

If the followings are installed:

- MX887011A
- No option

:SYST:INF:MAIN:SOFT:OPT? MX887011A
> 0

Remarks

The registered model names in Anritsu's ordering information can be only obtained.
A parameter error occurs if an invalid software name is set.

:SYSTem:INFormation:MAINframe:TEMPerature?

Main Frame Temperature

Function

Queries MT8870A temperature.

Query

```
:SYSTem:INFormation:MAINframe:TEMPerature?
```

Response

<temp>	MT8870A temperature (°C)
Resolution:	0.01

Example of Use

:SYST:INF:MAIN:TEMP?

> 47.12

:SYSTem:INFormation:MAINframe:WAVeform?

Type Name of Waveform

Function

Queries the waveform license information.

Query

```
:SYSTem:INFormation:MAINframe:WAVeform?
```

Response

<n>,<model>,...,<model>

Parameters

<n>	Number of waveform licenses
<model>	Waveform license name

Example of Use

:SYST:INF:MAIN:WAV?
\> 2,MV887011A,MV887012A

Remarks

The registered model names in Anritsu's ordering information can be only obtained.

:SYSTem:INFormation:POWeron:COUNt?

Module Power ON Count

Function

Queries module power-on count.

Query

`:SYSTem:INFormation:POWeron:COUNt?`

Response

<num>

Parameter

<num>	Module power-on count

Example of Use

:SYSTem:INFormation:POWeron:COUNt?
> 2525

:SYSTem:INFormation:PLUG:COUNt?

Count of Module Insertion

Function

Queries module insertion/removal information.

Query

```
:SYSTem:INFormation:PLUG:COUNt?
```

Response

<num>

Parameter

<num> Module insertion/removal count

Details

Note:

When a module is moved to another slot, the insertion/removal is counted.

When a module is inserted/removed in the same slot of the same MT8870A, insertion/removal is not counted.

Example of Use

:SYSTem:INFormation:PLUG:COUNt?

> 40

:SYSTem:INFormation:RTIMe?

Module Running Time Count Read

Function

Queries module total running time.

Query

`:SYSTem:INFormation:RTIMe?`

Response

<time>

Parameter

<time>	Total running time (minutes)

Example of Use

:SYST:INF:RTIM?

> 1234

:SYSTem:INFormation:SLOT?

Module Slot Number

Function

Queries number of slot where module installed.

Query

`:SYSTem:INFormation:SLOT?`

Response

<slot>

Parameter

<slot>

Range	1 to 4

Example of Use

:SYST:INF:SLOT?

> 1

:SYSTem:INFormation:TEMPerature?

Module Temperature

Function

Queries module temperature.

Query

```
:SYSTem:INFormation:TEMPerature?
```

Response

<temp>

Parameter

<temp>	Module temperature (°C)
Resolution:	0.01

Example of Use

:SYST:INF:TEMP?

> 47.12

:SYSTem:LANGuage

Language Selection of Remote Command

Function

Switches remote control command language mode.

Command

```
:SYSTem:LANGuage <mode>
```

Query

```
:SYSTem:LANGuage?
```

Response

<mode>

Parameter

<mode>	Language mode
NAT	Native
SCPI	SCPI

Example of Use

SYST:LANG NAT
SYST:LANG?
>NAT

:SYSTem:LAST:ERRor?

System Last Error

Function

Queries last error.

Query

```
:SYSTem:LAST:ERRor?
```

Response

<msg>

Parameter

<msg>	Abnormal contents
NO ERROR	No error
TemperatureAlarm	Forced shutdown due to temperature abnormality

Details

Executing this command does not clear the last error information.

Example of Use

:SYSTem:LAST:ERRor?
>NO ERROR

:SYSTem:REBoot

Reboot

Function

Reboots system.

Command

```
:SYSTem:REBoot
```

Details

Sending this command reboots the system. Remote control of all modules installed in the MT8870A is disconnected while rebooting.

When the MT8870A Error lamp goes off, remote control can be performed.

Example of Use

:SYST:REB

Related Command

:SYSTem:SHUTdown

:SYSTem:SELF:TEST?

System Test

Function

Queries time when module self-diagnostics executed.

Query

```
:SYSTem:SELF:TEST?
```

Response

<status1>,<status2>,<status3>,...,<year>,<month>,<day>,<hour>,<min>
The detail of <status> is the format as follows:

If MU887000A-002 Audio Measurement Hardware is not installed,
"Chassis EEPROM: xx","Module EEPROM: xx","RAM: xx","Correction ROM: xx","FPGA Configuration: xx"
If MU887000A-002 Audio Measurement Hardware is installed,
"Chassis EEPROM: xx","Module EEPROM: xx","RAM: xx","Correction ROM: xx","FPGA Configuration: xx", "Audio Memory: xx", "Audio Correction ROM: xx", "Audio FPGA Configuration: xx"

"Pass" or "Fail" is entered to xx.

Parameters

<status>	Last executed self-diagnostics test
<year>	Year (2001 to)
<month>	Month (1 to 12)
<day>	Day (1 to 31)
<hour>	Time (0 to 23)
<min>	Minute (0 to 59)

Example of Use

:SYST:SELF:TEST?
If MU887000A-002 Audio Measurement Hardware is not installed,
> "Chassis EEPROM: Pass","Module EEPROM:Pass","RAM: Pass","Correction ROM: Pass","FPGA Configuration:Pass",2012,4,5,12,34
If MU887000A-002 Audio Measurement Hardware is installed,
> "Chassis EEPROM: Pass","Module EEPROM: Pass ","RAM: Fail","Correction ROM: Pass","FPGA Configuration: Pass", "Audio Memory: Pass", "Audio Correction ROM: Pass", "Audio FPGA Configuration: Pass", 2012,4,5,12,34

:SYSTem:SELF:TEST:ERRor:COUNt?

System Test Error Count

Function

Queries number of self-diagnostics test errors.

Query

`:SYSTem:SELF:TEST:ERRor:COUNt?`

Response

<count>

Parameter

<count>	Number of self-diagnostics errors
Range	0 or more

Example of Use

:SYST:SELF:TEST:ERR:COUN?
> 0

:SYSTem:SELF:TEST:ERRor:LASTtime?

System Test Error Last Time

Function

Queries time of last self-diagnostics error.

Query

```
:SYSTem:SELF:TEST:ERRor:LASTtime?
```

Response

<year>,<month>,<day>,<hour>,<min>

Parameters

<year>	year (2001 or later)
<month>	month (1 to 12)
<day>	month (1 to 31)
<hour>	hour (0 to 23)
<min>	minute (0 to 59)

Example of Use

:SYST:SELF:TEST:ERR:LAST?

> 2012,1,2,3,45

Remarks

The response is *** when there has never been a self-diagnostics error.

:SYSTem:SHUTdown

Shutdown

Function

Shutdowns system.

Command

```
:SYSTem:SHUTdown
```

Details

Sending this command disconnects remote connection of all modules.

Example of Use

:SYST:SHUT

Related Command

:SYSTem:REBoot

:SYSTem:VERSion?

Package Version

Function

Queries package version.

Query

```
:SYSTem:VERSion?
```

Response

<ver>

Parameter

<ver>	Package version (xx.xx.xx)

Example of Use

:SYST:VERS?
> 1.00.00

5.2.2 VSG commands

:SOURce:GPRF:GENerator:ARB:FILE:LOAD

Waveform File Loading

Function

Loads waveform file saved in storage media to waveform memory.

Command

```
:SOURce:GPRF:GENerator:ARB:FILE:LOAD <file_name>
```

Query

```
:SOURce:GPRF:GENerator:ARB:FILE:LOAD? <file_name>
```

Response

<status>

Parameters

<file_name>	Waveform file name
<status>	Waveform file status
0	Already loaded to waveform memory
1	Can be loaded to waveform memory
2	Requires license to use waveform file
3	No corresponding waveform file
4	Insufficient waveform memory free space
5	Can be loaded if waveform memory optimized
6	Parameter error of waveform file
7	Loading to waveform memory
8	Optimizing waveform memory

Details

To use this command, at least one waveform file listed in Table 1.3.3-2 "Waveforms" is required. Refer to the following descriptions when specifying the file name to <file_name>.
Chapter 3 "Wavefrom File Details" in *Waveform File for Cellular Application Operation Manual*
Chapter 3 "Wavefrom File Details" in *Waveform File for GNSS & Broadcast Operation Manual*
Chapter 4 "Wavefrom Files" in *MV887030A/31A/32A/33A/40A Short Range WirelessWaveform Options for the MT8870A Operation Manual*

If the waveform file is loaded using this command while the waveform pattern is being reproduced, the reproduction is aborted. When the loading of the waveform file is completed, the reproduction is resumed from the beginning of the waveform pattern automatically.

5 SCPI Command Reference

Example of Use

To load the waveform file MV887011A_WCDMA_0002.xml in the storage media to the waveform memory:

:SOUR:GPRF:GEN:ARB:FILE:LOAD "MV887011A_WCDMA_0002.xml"

To query the status of the waveform file MV887013A_LTEFDD_0004.xml in the storage memory.

:SOUR:GPRF:GEN:ARB:FILE:LOAD? "MV887013A_LTEFDD_0004.xml"

> 0

Related Command

:SOURce:GPRF:GENerator:ARB:WAVeform:DEFRag

Remarks

An execution error occurs if the operation mode is:

- Sequence mode

Commands other than cancel loading and query loading cannot be executed during waveform loading (execution error). The file extension can be omitted.
A parameter error occurs if the specified file does not exist.
The extension of the waveform file (xml) can be omitted.

:SOURce:GPRF:GENerator:ARB:FILE:LOAD:CANCel

Waveform File Loading Cancel

Function

Cancels loading of file to waveform memory.

Command

```
:SOURce:GPRF:GENerator:ARB:FILE:LOAD:CANCel
```

Example of Use

To cancels loading of file to waveform memory:
:SOUR:GPRF:GEN:ARB:FILE:LOAD:CANC

Remarks

An execution error occurs if the operation mode is:
- Sequence mode
Nothing can be executed during waveform loading.

:SOURce:GPRF:GENerator:ARB:FILE:LOAD:STATus?

Waveform Load Status

Function

Queries waveform file loading status.

Query

```
:SOURce:GPRF:GENerator:ARB:FILE:LOAD:STATus?
```

Response

<status_res>

Parameter

<status_res>	Waveform file loading status
0	Not loading (Loading not executed or already executed)
1	Loading

Example of Use

To query waveform file loading status:
:SOUR:GPRF:GEN:ARB:FILE:LOAD:STAT?
> 1

:SOURce:GPRF:GENerator:ARB:FILE:PATTern?

Waveform Group Information

Function

Queries the group number of waveform file saved in storage media.

Query

```
:SOURce:GPRF:GENerator:ARB:FILE:PATTern? <file_name>,<group_index>
```

Response

<group_number>

Parameters

<file_name>	Waveform file name
<group_index>	Order of waveform pattern stored in waveform file
Range	1 to Number of waveform patterns included in waveform file
<group_number>	Group number

Example of Use

To query the second group number of waveform pattern in the waveform files saved in storage media:

:SOUR:GPRF:GEN:ARB:FILE:PATT? "MV887070A_FMRDS_0001.xml",2

> 2

Remarks

The extension of the waveform file (xml) can be omitted.

A parameter error occurs if the specified file does not exist.

:SOURce:GPRF:GENerator:ARB:FILE:PATTern:COUNt?

Waveform Group Count

Function

Queries the number of waveform patterns in waveform file saved in storage media.

Query

```
:SOURce:GPRF:GENerator:ARB:FILE:PATTern:COUNt? <file_name>
```

Response

<count>

Parameters

<file_name>	Waveform file name
<count>	Number of waveform patterns

Example of Use

To query the number of waveform patterns in waveform file saved in storage media:

:SOUR:GPRF:GEN:ARB:FILE:PATT:COUN? "MV887013A_LTEFDD_0002.xml"

\> 7

Remarks

The extension of the waveform file (xml) can be omitted.

:SOURce:GPRF:GENerator:ARB:FILE:PATTern:NAME?

Waveform Pattern name

Function

Queries the pattern name (title) with specification of the group number in the waveform file saved in the storage media.

Query

```
:SOURce:GPRF:GENerator:ARB:FILE:PATTern:NAME? <file_name>,<group_number>
```

Response

<pattern_name>

Parameters

<file_name>	Waveform file name
<group_number>	Group number
Range	Range of group number defined by waveform file
<pattern_name>	Pattern name (title)

Example of Use

To query the pattern name (title) described in the group number 2 in the waveform file "MV887111A_ISDBT_0001.xml" saved in the storage media:

:SOUR:GPRF:GEN:ARB:FILE:PATT:NAME? "MV887111A_ISDBT_0001.xml",2

> Mode:3, GI:1/8, Data:PN23fix, (A:1segPR,QPSK), (B:12seg,64QAM)

Remarks

The extension of the waveform file (xml) can be omitted.

A parameter error occurs if the specified file or group number does not exist.

:SOURce:GPRF:GENerator:ARB:FILE:VERSion?

Waveform Version

Function

Queries version of waveform file saved to storage media.

Query

```
:SOURce:GPRF:GENerator:ARB:FILE:VERSion? <file_name>
```

Response

<version>

Parameters

<file_name>	Waveform file name
<version>	Version number

Example of Use

To query the version of the waveform file "MV887030A_b_2_100L.xml" saved to the storage media:

:SOUR:GPRF:GEN:ARB:FILE:VERS? "MV887030A_b_2_100L.xml"

> 01.00.00

Remarks

The file extension (xml) of waveform file name can be ignored.
A parameter error occurs if the specified file does not exist.

The response is –999 when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:NOISe:CN

AWGN C/N Ratio

Function

Sets or queries output level ratio (C/N) of carrier and AWGN when AWGN On.

Command

```
:SOURce:GPRF:GENerator:ARB:NOISe:CN <cn_ratio>
```

Query

```
:SOURce:GPRF:GENerator:ARB:NOISe:CN?
```

Response

<cn_ratio>

Unit	dB

Parameter

<cn_ratio>	C/N setting value
Range	–12 to 40 dB
Resolution	1 dB
Suffix Code	DB (uses DB when omitted)
Default	40 dB

Example of Use

To set the output level ratio (C/N) of the carrier and AWGN to –10 dB:

:SOUR:GPRF:GEN:ARB:NOIS:CN -10

To query the set output level ratio (C/N) of the carrier and AWGN:

:SOUR:GPRF:GEN:ARB:NOIS:CN?

> -10

Remarks

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:NOISe:STATe

AWGN Switch

Function

Sets or queries AWGN output On/Off.

Command

```
:SOURce:GPRF:GENerator:ARB:NOISe:STATe <on_off>
```

Query

```
:SOURce:GPRF:GENerator:ARB:NOISe:STATe?
```

Response

<on_off>

Parameters

<on_off>	AWGN output On/Off
0	Off
1	On
OFF	Off
ON	On
Default	0

Example of Use

To output AWGN:
:SOUR:GPRF:GEN:ARB:NOIS:STAT ON

To query the status of the AWGN output:
:SOUR:GPRF:GEN:ARB:NOIS:STAT?
> 1

Related Command

:SOURce:GPRF:GENerator:ARB:NOISe:CN

Remarks

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

5

SCPI Command Reference

:SOURce:GPRF:GENerator:ARB:WAVeform:COUNt?

ARB Memory Loaded files count

Function

Queries number of waveform files loaded into memory.

Query

`:SOURce:GPRF:GENerator:ARB:WAVeform:COUNt?`

Response

<count>

Parameter

<count>	Number of waveform files loaded into memory

Example of Use

To query number of waveform files loaded into memory:
:SOUR:GPRF:GEN:ARB:WAV:COUN?
> 2

:SOURce:GPRF:GENerator:ARB:WAVeform:DEFRag

ARB Memory Defrag

Function

Optimizes waveform memory to increase contiguous free space.

Command

`:SOURce:GPRF:GENerator:ARB:WAVeform:DEFRag`

Example of Use

To optimize the waveform memory:
:SOUR:GPRF:GEN:ARB:WAV:DEFR

Remarks

An execution error occurs in the following cases:
- Sequence mode
- A command other than a query is executed during waveform fragmentation.
- The waveform file loaded into the waveform memory is not found in the memory medium.

:SOURce:GPRF:GENerator:ARB:WAVeform:DELete

Deleting Waveform File in ARB Memory

Function

Deletes specified waveform file in waveform memory.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:DELete <file_name>
```

Parameter

<file_name>	Waveform file name

Details

Specify the name of file that has been loaded to the waveform memory.
To get the name of the loaded file, use the following commands.
:SOURce:GPRF:GENerator:ARB:WAVeform:COUNt?
:SOURce:GPRF:GENerator:ARB:WAVeform:NAME?

Example of Use

To delete the waveform file MV887100A_GPS_0002.xml in the waveform memory:
:SOUR:GPRF:GEN:ARB:WAV:DEL "MV887100A_GPS_0002.xml"

Remarks

An execution error occurs if the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

An execution error occurs if the specified file does not exist.
The extension of the waveform file (xml) can be omitted.

5

SCPI Command Reference

:SOURce:GPRF:GENerator:ARB:WAVeform:DELete:ALL

Deleting All Waveform Files in ARB Memory

Function

Deletes all waveform files in waveform memory.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:DELete:ALL
```

Example of Use

To delete all waveform files in the waveform memory:

:SOUR:GPRF:GEN:ARB:WAV:DEL:ALL

Remarks

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:WAVeform:FREE?

ARB Memory Free Space

Function

Queries waveform memory free space.

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:FREE? <target>
```

Response

<blank>,<consecutive_blank>,<total>

Parameters

<target>	Memory type for free space reading
ALL	Free space of all memories read
GROUPINFO	Group information free space read
SECTIONINFO	Section information free space read
ARB	ARB memory free space read
<blank>	Free space (bytes)
<consecutive_blank>	Contiguous free space (bytes)
<total>	Total waveform memory size (bytes)

Example of Use

To query the waveform memory free space:

:SOUR:GPRF:GEN:ARB:WAV:FREE? GROUPINFO
> 1000,2000,3000
:SOUR:GPRF:GEN:ARB:WAV:FREE? SECTIONINFO
> 10000,20000,30000
:SOUR:GPRF:GEN:ARB:WAV:FREE? ARB
> 100,200,300
:SOUR:GPRF:GEN:ARB:WAV:FREE? ALL
> 1000,2000,3000,10000,20000,30000,100,200,300

Remarks

If ALL is set, the response is returned in the order of group information space, section information space, and ARB memory.

The response is –999 when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:WAVeform:GENerate:ZVSPattern

Waveform Generate for Zero Vector Signal Pattern

Function

Generates waveform pattern for zero waveform to load to waveform memory.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:GENerate:ZVSPattern
<file_name>,<sampling_rate>,<sample_num>
```

Parameters

<file_name>	File name of waveform pattern to generate (xml extension)
<sampling_rate>	Sampling rate
Range	2 to 200 MHz
Resolution	1 Hz
Suffix Code	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
<sample_num>	Number of samples
Range	512 to 100,000
Resolution	1

Example of Use

To generate the waveform pattern for the zero waveform to load to the waveform memory:

:SOUR:GPRF:GEN:ARB:WAV:GEN:ZVSP "ZeroWave.xml", 2MHZ, 512

Remarks

An execution error occurs if the operation mode is:

- Sequence mode running

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

The file extension (xml) of waveform file name can be ignored.
An execution error occurs when there is already a file with the same name.

:SOURce:GPRF:GENerator:ARB:WAVeform:NAME?

ARB Memory Loaded file name

Function

Queries name of waveform file loaded to waveform memory.

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:NAME? <numeric_value>
```

Response

<file_name>

Parameters

<numeric_value>	Arbitrary number allocated to waveform file
<file_name>	Waveform file name

Example of Use

To query the name of the waveform file loaded to the waveform memory:

:SOUR:GPRF:GEN:ARB:WAV:NAME? 1

> MV887040A_BLE

Remarks

A parameter error occurs when the set number is bigger than the number of the waveform files that are loaded in the waveform memory.

The waveform file name of response is a character string without the extension (.xml).

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern?

ARB Memory Group Information

Function

Queries group number of waveform file loaded to waveform memory.

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern? <file_name>,<count>
```

Response

<group_number>

Parameters

<file_name>	Waveform file name
<group_index>	Order of waveform pattern stored in waveform file
Range	1 to Number of waveform patterns stored in waveform file
<group_number>	Group number

Example of Use

To query the second number of the waveform pattern in the group of waveform files loaded to the waveform memory:

```
:SOUR:GPRF:GEN:ARB:WAV:PATT? "MV887016A_EVDO_0002",2
> 2
```

Remarks

The file name is a character string without the extension (xml).
An execution error occurs when the specified file is not loaded.
The response is –999 when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:COUNt?

ARB Memory Group Count

Function

Queries number of waveform patterns in waveform file loaded to waveform memory.

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:COUNt? <file_name>
```

Response

<count>

Parameters

<file_name>	Waveform file name
<count>	Number of waveform patterns

Example of Use

To query the number of waveform patterns in the waveform file loaded to the waveform memory:

:SOUR:GPRF:GEN:ARB:WAV:PATT:COUN? "MV887111A_ISDBT_0004"

> 5

Remarks

The file name is a character string without the extension (xml).

An execution error occurs when the specified file is not loaded.

The response is –999 when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:GAP

ARB Memory Gap Length

Function

Sets gap length of waveform pattern in waveform file loaded to waveform memory.

Queries gap length of waveform pattern in waveform file loaded to waveform memory.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:GAP
<file_name>,<group_nunber>,<gap_length>
```

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:GAP?
<file_name>,<group_nunber>
```

Response

<gap_length>	
Unit	μs

Parameters

<file_name>	Waveform file name
<group_number>	Group number
Range	Range of group number defined by waveform file
Resolution	1
<gap_length>	Gap length (f:Sampling rate of waveform)
Range	0, 3 to 10000 μs (200 MHz)
	0, 30 to 10000 μs (20≤ f <200 MHz)
	0, 300 to 100000 μs (2≤ f <20 MHz)
	0, 3 to 1000 ms (200≤ f <2000 kHz)
	0, 30 to 10000 ms (20≤ f <200 kHz)
Resolution	1 μs (2≤ f ≤200 MHz)
	1 ms (20≤ f <2000 kHz)
Suffix Code	S,MS,US,NS (uses US when omitted)
Default	0

Example of Use

To set 10-μs gap length to the waveform pattern of the group number 1 in the waveform file "MV887030A_b_11_1024L.xml":

```
:SOUR:GPRF:GEN:ARB:WAV:PATT:GAP "MV887030A_b_11_1024L",1,10
```

To query the gap length applied to waveform pattern of the group number 1 in the waveform file "MV887030A_b_11_1024L.xml":

```
:SOUR:GPRF:GEN:ARB:WAV:PATT:GAP? "MV887030A_b_11_1024L",1
> 10
```

Remarks

The file name is a character string without the extension (xml).
An execution error occurs when the specified file is not loaded.
A parameter error occurs if the specified group number does not exist.

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:NAME?

ARB Memory Pattern name

Function

Queries the pattern name (title) with specification of the group number in the waveform file loaded to the waveform memory.

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:NAME?
<file_name>,<group_number>
```

Response

<pattern_name>

Parameters

<file_name>	Waveform file name
<group_number>	Group number
Range	Range of group number defined by waveform file
<pattern_name>	Pattern name (title)

Example of Use

To query the pattern name (title) described in the group number 2 in the waveform file "MV887015A_C2K_0002.xml" loaded to the waveform memory:

:SOUR:GPRF:GEN:ARB:WAV:PATT:NAME? "MV887015A_C2K_0002",2

> FL Wave RL Request : PCB +1dB

Remarks

The file name is a character string without the extension (xml).

An execution error occurs when the specified file is not loaded.

The response is *** when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

A parameter error occurs if the specified group number does not exist.

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:NOTE?

ARB Memory Pattern Note

Function

Queries pattern information with specification of group number in waveform file loaded to waveform memory.

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:NOTE?
<file_name>,<group_number>
```

Response

<note>

Parameters

<file_name>	Waveform file name
<group_number>	Group number
Range	Range of group number defined by waveform file
<note>	Memo

Remarks

The file name is a character string without the extension (xml).
An execution error occurs when the specified file is not loaded.
The response is *** when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

A parameter error occurs if the specified group number does not exist.

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect

ARB Memory Pattern Selection

Function

Selects pattern to be played in waveform file loaded to waveform memory.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect
<file_name>[,<group_number>[,<length>]]
```

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect?
```

Response

<file_name>,<group_number>,<length>

Parameters

<file_name>	Waveform file name
<group_number>	Group number
Range	Range of group number defined by waveform file
Default	1
<length>	Number of waveform patterns played continuously
Range	1 to Number of waveform patterns included in waveform file
Default	1

Example of Use

To set the pattern of group number 2 in the waveform file "MV887012A_GSM_0002.xml" to the play pattern:

:SOUR:GPRF:GEN:ARB:WAV:PATT:SEL "MV887012A_GSM_0002",2,1

To query the information of the pattern currently being played:

:SOUR:GPRF:GEN:ARB:WAV:PATT:SEL?

> "MV887012A_GSM_0002",2,1

Remarks

The file name is a character string without the extension (xml).

An execution error occurs when the following processes are being executed:

- Loading waveform
- Defragmenting waveform memory

The response is "NO FILE",0,0 when no waveform pattern is selected.

A parameter error occurs if the specified group number does not exist.

:SOURce:GPRF:GENerator:ARB:WAVeform:RESTart

ARB Restart

Function

Plays selected pattern from beginning.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:RESTart
```

Example of Use

To play the selected pattern from the beginning:

:SOUR:GPRF:GEN:ARB:WAV:REST

Remarks

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:WAVeform:SCLock:RATE?

ARB Sampling Rate

Function

Queries sampling rate of selected pattern.

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:SCLock:RATE?
```

Response

<sampling_rate>

Unit	Hz

Parameter

<sampling_rate>	Sampling rate
Resolution	0.001 Hz

Example of Use

To query the sampling rate of the selected pattern:

:SOUR:GPRF:GEN:ARB:WAV:SCL:RATE?

> 3250000.000

Remarks

The response for no pattern selection is 0.000 Hz.

The response is –999.999 when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:WAVeform:SSWitch

ARB Subsection Switch

Function

Sets or queries waveform pattern subsection selection method.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:SSWitch
<switch>[,<subsection_number>]
```

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:SSWitch?
```

Response

<switch>,<subsection_number>

Parameters

<switch>	Subsection selection method
FIXED	Fixed
ALTERNATE	Subsection 0 and 1 alternately
ILPCONTROL	Inner Loop Power Control
Default	FIXED

<subsection_number>	Subsection number
Range	0 to 3
Resolution	1
Default	0

Example of Use

To set the waveform pattern subsection selection method to FIXED and the subsection number to 2:

:SOUR:GPRF:GEN:ARB:WAV:SSW FIXED,2

To query the waveform pattern subsection selection method:

:SOUR:GPRF:GEN:ARB:WAV:SSW?

> FIXED,2

Remarks

An execution error occurs if the operation mode is:

- Sequence mode running

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:CANCel

Waveform Synchronization Cancel

Function

Cancels synchronized output start stand-by state of waveform patterns.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:CANCel
```

Example of Use

To cancel synchronized output start stand-by state of waveform patterns.

:SOUR:GPRF:GEN:ARB:WAV:SYNChronize:STARt:CANCel

:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect

Waveform Synchronization Select Pattern

Function

Selects waveform pattern from the waveform file loaded in the waveform memory, sets the synchronization method, and starts synchronization. Also, queries the waveform pattern.

Command

```
:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect
<mode>,<filename>[,<group_number>[,<length>]]
```

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect?
```

Response

<filename>,<group_number>,<length>

Parameters

<mode>	Synchronization Mode:
MASTER	Starts pattern synchronized output as a master. (Only Slot 1 module can be set to this.)
INTSLAVE	Prepares pattern synchronized output as a slave. (Synchronizes with the MT8870A signal.)
EXTSLAVE	Prepares pattern synchronized output as a slave. (Synchronizes with external trigger.)
<file_name>	Waveform file name
<group_number>	Group No.
Range	Group No. range defined by the waveform file.

Default 1

<length> Number of waveform patterns played continuously

Range 1 to the waveform pattern number included in the waveform file

Default 1

Example of Use

To set a waveform pattern of the waveform file MV887012A_GSM_0002.xml to replay pattern with Synchronization mode set to MASTER.

:SOUR:GPRF:GEN:ARB:WAV:SYNC:STAR:PATT:SEL MASTER,"MV887012A_GSM_0002"

To query the information of the pattern currently being played:

:SOUR:GPRF:GEN:ARB:WAV:SYNC:STAR:PATT:SEL?

> MASTER,"MV887012A_GSM_0002",1,1

Remarks

Available Synchronization modes vary according to the slot to which the module is installed.

Slot No. / Synchronization Mode	Slot1	Slot2	Slot3	Slot4
MASTER	✓	–	–	–
INTSLAVE	–	✓	✓	✓
EXTSLAVE	✓	✓	✓	✓

✓:Available –: Unavailable

By executing this command, the waveform pattern set by the below command is changed.

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect

By executing this command, the setting values of the below commands relating to external trigger are changed.

:ROUTe:TRIGger:PORT

:ROUTe:TRIGger:OUTPut:SOURce

:ROUTe:TRIGger:INPut:SOURce

By setting <mode>, the external trigger settings of the modules are changed as below.

<mode> / Command	MASTER	INTSLAVE	EXTSLAVE
:ROUTe:TRIGger:PORT	OUTPUT	INPUT	INPUT*
:ROUTe:TRIGger:OUTPut:SOURce	SGSYNC	(No Change)	(No Change)
:ROUTe:TRIGger:INPut:SOURce	(No Change)	SLOT1	TRGIN1

*: When set to EXTSLAVE, the external trigger of Slot 1 is changed to INPUT.

Related Command

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect
:SOURce:GPRF:GENerator:ARB:WAVeform:RESTart
:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:STATe
:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:CANCel

:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:STATe?

Waveform Synchronization State

Function

Queries synchronized output start stand-by state of waveform patterns.

Query

```
:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:STATe?
```

Response

<status>

Parameter

<status>	Synchronization stand-by state
0	Synchronized output start stand-by state
1	Synchronized output start stand-by cancel state

Example of Use

To query synchronized output start stand-by state of waveform pattern.

:SOUR:GPRF:GEN:ARB:WAV:SYNC:STAT:STAT?
> 1

:SOURce:GPRF:GENerator:BBMode

Modulation

Function

Sets baseband mode to modulation wave output or CW output.

Command

```
:SOURce:GPRF:GENerator:BBMode <mode>
```

Query

```
:SOURce:GPRF:GENerator:BBMode?
```

Response

<mode>

Parameter

<mode>	Baseband mode
CW	CW output set
ARB	ARB waveform modulation wave output set
Default	CW

Example of Use

To set the baseband mode setting to modulation wave output:

:SOUR:GPRF:GEN:BBM ARB

To query the baseband mode:

:SOUR:GPRF:GEN:BBM?

> ARB

Remarks

When a waveform file is not selected by the commands below, the test ports of the panel do not output signals even after set to ARB.

:SOURce:GPRF:GENerator:ARB:WAVeform:PATTern:SELect

:SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt:PATTern:SELect

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:MODE

VSG Mode

Function

Sets or queries VSG operation mode.

Command

```
:SOURce:GPRF:GENerator:MODE <switch>
```

Query

```
:SOURce:GPRF:GENerator:MODE?
```

Response

<switch>

Parameter

<switch>	Operation mode
NORMAL	Normal mode
SEQUENCE	Sequence mode
Default	NORMAL

Example of Use

To set the VSG operation mode to Normal mode:

:SOUR:GPRF:GEN:MODE NORMAL

To query the VSG operation mode:

:SOUR:GPRF:GEN:MODE?

> NORMAL

Remarks

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:RFSettings:DM:POLarity

Spectrum Reverse

Function

Sets or queries output waveform IQ inversion.

Command

```
:SOURce:GPRF:GENerator:RFSettings:DM:POLarity <mode>
```

Query

```
:SOURce:GPRF:GENerator:RFSettings:DM:POLarity?
```

Response

<mode>

Parameter

<mode>	Mode
NORMAL	IQ not inverted
INVERT	IQ inverted
Default	NORMAL

Example of Use

To invert the output waveform IQ:

:SOUR:GPRF:GEN:RFS:DM:POL INVERT

To query the output waveform IQ inversion status:

:SOUR:GPRF:GEN:RFS:DM:POL?

> INVERT

Remarks

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:RFSettings:FREQuency

Frequency

Function

Sets or queries RF frequency.

Command

```
:SOURce:GPRF:GENerator:RFSettings:FREQuency <freq>
```

Query

```
:SOURce:GPRF:GENerator:RFSettings:FREQuency?
```

Response

<freq>

Unit	Hz

Parameter

<freq>	Frequency
Range	1 MHz to 3.8 GHz (*1)
	1 MHz to 6.0 GHz (*2)
Resolution	1 Hz
Suffix Code	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
Default	1 GHz

Example of Use

To set the RF frequency to 3 GHz:
:SOUR:GPRF:GEN:RFS:FREQ 3GHZ

To query the RF frequency setting value:
:SOUR:GPRF:GEN:RFS:FREQ?
> 3000000000

Remarks

(*1) Setting range without frequency expansion option
(*2) Setting range with frequency expansion option

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:RFSettings:LEVel

Level

Function

Sets or queries RF output level.

Command

```
:SOURce:GPRF:GENerator:RFSettings:LEVel <level>
```

Query

```
:SOURce:GPRF:GENerator:RFSettings:LEVel?
```

Response

<level>

Unit dBm

Parameter

<level> Output Level

Range

RF frequency f (MHz)	Port	Range
1 ≤ f ≤ 3800	Port1, Port2	(–130.0 dBm + CableLoss) to (–10.0 dBm + CableLoss – AWGN_RF_Adjust_Gain)
	Port3, Port4	(–120.0 dBm + CableLoss) to (0.0 dBm + CableLoss – AWGN_RF_Adjust_Gain)
3800 < f ≤ 6000	Port1, Port2	(–130.0 dBm + CableLoss) to (–18.0 dBm + CableLoss – AWGN_RF_Adjust_Gain)
	Port3, Port4	(–120.0 dBm + CableLoss) to (–8.0 dBm + CableLoss – AWGN_RF_Adjust_Gain)

Resolution 0.1 dBm

Suffix Code DBM, DBUVEMF, DBUVTERM (uses DBM when omitted)

Default –120.0 dBm

Details

CableLoss

As shown in the following table, this value depends on the External Loss setting (On/Off).

Setting	COMMON, ON	Loss value [dB](≥0) This is a value obtained by compensating the value set by Register Loss Value into Cable Loss List.
	OFF	0 [dB]

AWGN_RF_Adjust_Gain

As shown in the following table, this value depends on the AWGN Switch setting.

Setting	ON	Adjust value [dB](≥0) This is a value that depends on the C/N value set by AWGN C/N Ratio and the waveform currently being reproduced.
	OFF	0 [dB]

Example of Use

To set the RF output level to –50.0 dBm:

:SOUR:GPRF:GEN:RFS:LEV -50.0DBM

To query the RF output level setting value:

:SOUR:GPRF:GEN:RFS:LEV?

> -50.0

:SOURce:GPRF:GENerator:RFSettings:LEVel:SETTing?

Level Setting

Function

Queries whether output level in Normal mode within guaranteed accuracy range.

Query

```
:SOURce:GPRF:GENerator:RFSettings:LEVel:SETTing?
```

Response

<status>

Parameter

<status>	Output level status
NORMAL	Normal or status without waveform
UNLEVELD	Out of guaranteed level accuracy range

Example of Use

To query the output level status in use in the Normal mode:

:SOUR:GPRF:GEN:RFS:LEV:SETT?

> NORMAL

:SOURce:GPRF:GENerator:SEQuence:CANCel

Sequence Execution Cancel

Function

Cancels sequence processing.

Command

```
:SOURce:GPRF:GENerator:SEQuence:CANCel
```

Example of Use

To cancel the sequence processing:

:SOUR:GPRF:GEN:SEQ:CANC

Remarks

An execution error occurs if the operation mode is:

- Normal mode

:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern

Sequence Table Combination Selection

Function

Sets or queries number of sequence tables executed and execution order of sequence tables.

Command

```
:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern <num>,<table_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern?
```

Response

<num>,<table_n>

Parameters

<num>	Number of sequence tables to execute	
Range	1 to 16	
Default	1	
<table_n>	Sequence table number	
	<table1>	1st sequence table number to execute
	<table2>	2nd sequence table number to execute
	....	
	<table16>	16th sequence table number to execute
	Range	1 to 16
	Default	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

Example of Use

To set the number of sequence tables to execute to 3 and the execution order of the sequence tables to 1,2,3:

:SOUR:GPRF:GEN:SEQ:COMB:PATT 3,1,2,3

To query the number of sequence tables to execute and the execution order of the sequence tables:

:SOUR:GPRF:GEN:SEQ:COMB:PATT?

> 3,1,2,3

Remarks

An execution error the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize

Sequence Synchronization

Function

Sets start timing of waveform list table when executing sequence synchronization.

Command

```
:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize <mode>
```

Parameter

<mode>	Synchronization Mode
NONE	No synchronization with other modules.
MASTER	Starts waveform list table as a master. (Available only for the module installed to Slot 1.)
INTSLAVE	Starts waveform list table as a slave. (Synchronizes with the MT8870A signal.)
EXTSLAVE	Starts waveform list table as a slave. (Synchronizes with external trigger.)
Default	NONE

Example of Use

To set waveform list table start timing to external trigger when executing sequence table.

:SOUR:GPRF:GEN:SEQ:COMB:PATT:SYNC EXTSLAVE

Remarks

Available Synchronization modes vary according to the slot to which the module is installed.

Slot No. / Synchronization Mode	Slot1	Slot2	Slot3	Slot4
NONE	✓	✓	✓	✓
MASTER	✓	–	–	–
INTSLAVE	–	✓	✓	✓
EXTSLAVE	✓	✓	✓	✓

✓:Available –: Unavailable

By executing this command, the setting values of the below commands relating to external trigger are changed.

:ROUTe:TRIGger:PORT
:ROUTe:TRIGger:OUTPut:SOURce
:ROUTe:TRIGger:INPut:SOURce

By setting <mode>, the external trigger settings of the modules are changed as below.

Command \ <mode>	NONE	MASTER	INTSLAVE	EXTSLAVE
:ROUTe:TRIGger:PORT	(No Change)	OUTPUT	INPUT	INPUT*
:ROUTe:TRIGger:OUTPut:SOURce	(No Change)	SGSYNC	(No Change)	(No Change)
:ROUTe:TRIGger:INPut:SOURce	(No Change)	(No Change)	SLOT1	TRGIN1

*: When set to EXTSLAVE, the external trigger of Slot 1 is changed to INPUT.

:SOURce:GPRF:GENerator:SEQuence:EXECute

Sequence Execution

Function

Starts the sequence from the 1st SG sequencer data in the execution order.

Command

```
:SOURce:GPRF:GENerator:SEQuence:EXECute [<mode>]
```

Parameters

<mode>	Execution mode of sequence
NORMAL	Normal Execution mode
FORCED	Forced Execution mode
	Handled as NORMAL when omitted.

Details

In Normal Execution mode, the sequence execution starts when no error is found in the SG sequencer data.
In Forced Execution mode, the sequence execution starts even if the level of SG sequencer data is set to out-of-range value. In this case, the signal is output at the highest level possible if the set value exceeds the actual upper limit and at the lowest level possible if the set value goes under the actual lower limit.
If the level set value is out of range, the response to :SOURce:GPRF:GENerator:SEQuence:RX:ERRor? is 8. If there is an error in addition to the level error, the sequence execution does not start even in Forced Execution mode.

Example of Use

Starts the sequence from the 1st SG sequencer data in the execution order.

:SOUR:GPRF:GEN:SEQ:EXEC

Related Command

:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern

Remarks

The sequence does not restart even if this command is sent while the sequence is operating. In this case, this command is ignored.
The number and order of the SG sequencer data to execute is set by :SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern.
Use :SOURce:GPRF:GENerator:SEQuence:RLISt when the sequence is restarted from segment number 1 while the sequence is operating.
The operation for Normal Execution mode and Forced Execution mode are the same when no error is found in the SG sequencer data.
An execution error occurs when the sequence is executed in Forced Execution mode while the

level set value is in out-of-range error.

An execution error occurs if the operation mode is:
- Normal mode

:SOURce:GPRF:GENerator:SEQuence:LOAD

Load Sequence Parameter

Function

Loads SG sequencer data from file to specified sequence table.

Command

```
:SOURce:GPRF:GENerator:SEQuence:LOAD <list_table_no>,<file_name>
```

Parameters

<list_table_no>	Number of sequence table for SG sequencer data to load
Range	1 to 16
<file_name>	SG sequencer data file name

Example of Use

To load the SG sequencer data of the file "SeqFile.xml" to the sequence table 1:

:SOUR:GPRF:GEN:SEQ:LOAD 1,"SeqFile.xml"

Remarks

The extension of the waveform file (xml) can be omitted.

An execution error occurs if: the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

A parameter error occurs when the specified file does not exist.

:SOURce:GPRF:GENerator:SEQuence:REINitialization

Sequence Reinitialization

Function

Sets or queries operation mode after sequence completed.

Command

```
:SOURce:GPRF:GENerator:SEQuence:REINitialization <on_off>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:REINitialization?
```

Response

<on_off>

Parameter

<on_off>	Status after sequence executed
0	Sequence mode continued
1	Changed to Normal mode
OFF	Sequence mode continued
ON	Changed to Normal mode
Default	0

Example of Use

To change the operation mode to Normal mode after sequence execution:

:SOUR:GPRF:GEN:SEQ:REIN ON

To query the operation mode setting after sequence execution:

:SOUR:GPRF:GEN:SEQ:REIN ON?

> 1

Remarks

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RLISt

Sequence Re-Execution

Function

Stops the sequence operation and resumes the sequence from the 1st SG sequencer data in the execution order.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RLISt
```

Example of Use

To restart the sequence execution:

:SOUR:GPRF:GEN:SEQ:RLIS

Remarks

When this command is set while the sequence is operating, sequence operation in the Sequence mode is stopped.
After that, the sequence is resumed from the sequence top in the execution mode specified by :SOURce:GPRF:GENerator:SEQuence:EXECute.

An execution error occurs if the operation mode is:.
- Normal mode

:SOURce:GPRF:GENerator:SEQuence:RX:BBMode

Base Band Mode of Segment in Sequence

Function

Sets or queries baseband mode of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:BBMode
<list_table_no>,<segment_no>,<mode>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:BBMode? <list_table_no>,<segment_no>
```

Response

<mode>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<mode>	Baseband mode
CW	CW output set
ARB	ARB waveform modulation wave output set
Default	CW

Example of Use

To set the baseband mode of Segment 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:BBM 1,10,CW

To query the baseband mode of Segment 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:BBM? 1,10

> CW

Remarks

An execution error occurs if the operation mode is:

- Sequence mode running

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

5 SCPI Command Reference

:SOURce:GPRF:GENerator:SEQuence:RX:BBMode:ALL

Base Band Mode Group of All Segments in Sequence

Function

Sets or collectively queries baseband mode of each segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:BBMode:ALL <list_table_no>,<mode_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:BBMode:ALL? <list_table_no>
```

Response

<mode_n>

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<mode_n>		
	<mode_1>	Baseband mode of Segment 1
	<mode_2>	Baseband mode of Segment 2
	…	
	<mode_10001>	Baseband mode of Segment 10001
	CW	CW output set
	ARB	ARB waveform output set
	Default	CW

Example of Use

To set the baseband mode of each segment of Sequence table #1 collectively:

:SOUR:GPRF:GEN:SEQ:RX:BBM:ALL 1,CW,ARB,ARB,CW

To query the baseband mode of each segment of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:BBM:ALL? 1

> CW,ARB,ARB,CW,CW,...,CW

Remarks

The command parameter <mode_n> can be set to 10001 at maximum. If ten parameters are set, the baseband mode is set for Segment 1 to Segment 10. In this case, the baseband modes for Segment 11 and subsequent segments are not changed.

An execution occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

• Loading waveform

:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition

End Condition of Segment in Sequence

Function

Sets or queries end condition of specified one segment of SG sequencer tabledata.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition
<list_table_no>,<segment_no>,<condition>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition?
<list_table_no>,<segment_no>
```

Response

<condition>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<condition>	End condition
SNUMBER	Ends when time of specified number of steps elapsed
TRIGGER	Ends when trigger received
Default	SNUMBER

Example of Use

To set the end condition of Segment 10 of Sequence table #1 to the number of steps:

:SOUR:GPRF:GEN:SEQ:RX:ENDC 1,10,SNUMBER

To query the end condition of Segment 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:ENDC? 1,10

> SNUMBER

Related Command

:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce

Remarks

An execution error occurs if the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition:ALL

End Condition Group of All Segments in Sequence

Function

Sets or queries end condition of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition:ALL
<list_table_no>,<condition_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:ENDCondition:ALL? <list_table_no>
```

Response

<condition_n>

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<condition_n>		
	<condition_1>	End condition of Segment 1
	<condition_2>	End condition of Segment 2
	….	
	<condition_10001>	End condition of Segment 10001
	SNUMBER	Ends when time of specified number of steps elapses
	TRIGGER	Ends when trigger received
	Default	SNUMBER

Example of Use

To set the end condition of each segment of Sequence table #1 collectively:

:SOUR:GPRF:GEN:SEQ:RX:ENDC:ALL 1,SNUMBER,SNUMBER,TRIGGER

To query the end condition of each segment of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:ENDC:ALL? 1

> SNUMBER,SNUMBER,TRIGGER,SNUMBER,...,SNUMBER

Remarks

The command parameter <condition_n> can be set to 10001 at maximum. If ten parameters are set, the end conditions for Segment 1 to Segment 10 are set. In this case, the end conditions for Segment 11 and subsequent segments are not changed.

An execution error if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:ERRor?

Sequence Error Check

Function

Queries whether SG sequencer data has setting (error) at which operation cannot be guaranteed at sequence execution.

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:ERRor? <list_table_no>
```

Response

<error_count>,<error_x_res>

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<error_count>	Number of segments with error	
<error_x_res>	Evaluation result for each segment	
	<error_n_res>	Evaluation result for segment number where sequence starts
	<error_(n+1)_res>	
	…	
	<error_m_res>	Evaluation result for segment number where sequence ends The evaluation result indicates errors using binary addition (bits).
	1	Loop nest generated
	2	Loop jump destination out of execution sequence range
	4	Loop sent with trigger waiting segment
	8	Level out of setting range
	16	Unusable waveform pattern set

Example of Use

To query the error check result of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:ERR? 1

> 2,0,0,1,3,...0

:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency

Frequency of Segment in Sequence

Function

Sets or queries frequency of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency
<list_table_no>,<segment_no>,<freq>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency?
<list_table_no>,<segment_no>
```

Response

<freq>

Unit	Hz

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<freq>	Frequency
Range	1 MHz to 3.8 GHz (*1)
	1 MHz to 6.0 GHz (*2)
Resolution	1 Hz
Suffix Code	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
Default	1 GHz

Example of Use

To set the frequency of Segment 10 of Sequence table #1:
:SOUR:GPRF:GEN:SEQ:RX:FREQ 1,10,3GHZ

To query the frequency of Segment 10 of Sequence table #1:
:SOUR:GPRF:GEN:SEQ:RX:FREQ? 1,10
> 3000000000

Remarks

(*1) Setting range without frequency expansion option
(*2) Setting range with frequency expansion option

An execution error if the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency:ALL

Frequency Group of All Segments in Sequence

Function

Sets or queries frequency of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency:ALL <list_table_no>,<freq_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:FREQuency:ALL? <list_table_no>
```

Response

<freq_n>

Unit	Hz

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<freq_n>		
	<freq_1>	Frequency of Segment 1
	<freq_2>	Frequency of Segment 2
	…	
	<freq_10001>	Frequency of Segment 10001
	Range	1 MHz to 3.8 GHz (*1) 1 MHz to 6.0 GHz (*2)
	Resolution	1 Hz
	Suffix	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
	Default	1 GHz

Example of Use

To set the frequency column of each segment of Sequence table #1 collectively:

:SOUR:GPRF:GEN:SEQ:RX:FREQ:ALL 1,3GHZ,3.5GHZ,3.8GHZ

To query the frequency column of each segment of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:FREQ:ALL? 1

> 3000000000,3500000000,3800000000,1000000000,1000000000,...,1000000000

5

SCPI Command Reference

Remarks

The command parameter <freq_n> can be set to 10001 at maximum.
If ten parameters are set, the frequencies for Segment 1 to Segment 10 are set.
In this case, the frequencies for Segment 11 and subsequent segments are not changed.

(*1) Setting range without the frequency expansion option
(*2) Setting range with the frequency expansion option

An execution error occurs if the operation mode:
- Sequence mode
An execution error occurs when the following processes are executing:
• Loading waveform
• Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:GOTO

Sequence Start Index for continuous mode

Function

Sets or queries start segment number for second sequence and later in sequence repetition (in Continuous mode).

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:GOTO <list_table_no>,<index>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:GOTO? <list_table_no>
```

Response

<index>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<index>	Segment number
Range	1 to 10001
Default	1

Example of Use

To set the start segment of Sequence table #1 for the second sequence and later to 10:

```
:SOUR:GPRF:GEN:SEQ:RX:GEN:GOTO 1,10
```

To query the start segment of Sequence table #1 for the second sequence and later:

```
:SOUR:GPRF:GEN:SEQ:RX:GEN:GOTO? 1
> 10
```

Remarks

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:REPetition

Sequence Repetition Mode

Function

Sets or queries sequence repetition method.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:REPetition
<list_table_no>,<mode>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:REPetition? <list_table_no>
```

Response

<mode>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<mode>	Sequence repetition method
CONTINUOUS	Sequence repeated
SINGLE	Only once
Default	CONTINUOUS

Example of Use

To set the sequence repetition method of Sequence table #1 to Sequence is repeated:

:SOUR:GPRF:GEN:SEQ:RX:GEN:REP 1,CONTINUOUS

To query the sequence repetition method of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:GEN:REP? 1

> CONTINUOUS

Remarks

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:SSTop

Sequence Start and Stop Segment

Function

Sets or queries both start segment number and end segment number of sequence.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:SSTop
<list_table_no>,<start_seg>,<stop_seg>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:SSTop? <list_table_no>
```

Response

<start_seg>,<stop_seg>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<start_seg>	Start segment number
Range	1 to 10001
Resolution	1
Default	1
<stop_seg>	End segment number
Range	<start_seg> to 10001
Resolution	1
Default	1

Example of Use

To set the start segment to 10 and end segment to 30 for Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:GEN:SST 1,10,30

To query the start segment and end segment for Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:GEN:SST? 1

> 10,30

Remarks

A parameter error occurs when the end segment number is smaller than the start segment number.

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STARt

Sequence Start Segment

Function

Sets or queries start segment of sequence.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STARt
<list_table_no>,<start_seg>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STARt? <list_table_no>
```

Response

<start_seg>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<start_seg>	Start segment number
Range	1 to End segment number
Resolution	1
Default	1

Example of Use

To set the start segment of Sequence table #1 to 10:

:SOUR:GPRF:GEN:SEQ:RX:GEN:STAR 1,10

To query the start segment of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:GEN:STAR? 1

> 10

Remarks

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STEP:LENGth

Sequence Step Length

Function

Sets or queries step length time.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STEP:LENGth
<list_table_no>,<mode>[,<time>]
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STEP:LENGth? <list_table_no>
```

Response

<mode>,<time>

Unit of <time>	ms

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<mode>	Step length setting mode
TIME	Setting with time Set the parameter <time> for the step length time.
GSLOT	GSM slot time (15/26 ms)
GFRAME	GSM frame time (60/13 ms)
WSLOT	WCDMA slot time (10/15 ms)
ESLOT	EVDO slot time (80/48 ms)
EFRAME	EVDO frame time (80/3 ms)
Default	TIME
<time>	Step length
Range	0.5 to 80 ms
Resolution	0.00001 ms
Suffix Code	S, MS, US, NS (uses MS when omitted)
Default	10 ms

Example of Use

To set the step length of Sequence table #1 to 1.00000 ms:

:SOUR:GPRF:GEN:SEQ:RX:GEN:STEP:LENG 1,TIME,1.00000

To query the step length of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:GEN:STEP:LENG? 1

> TIME,1.00000

Remarks

Common setting for SG sequence data. The time set with this command is applied to all segments of the SG sequencer data.

The <time> parameter can be omitted when <mode> is not TIME.
The setting is ignored if the <time> parameter is set when <mode> is not TIME.
When <mode> is set to other than TIME, the step length is as follows:

GSLOT:	0.57692[ms](= 15/26[ms])
GFRAME:	4.61538[ms](= 60/13[ms])
WSLOT:	0.66667[ms] (= 10/15[ms])
ESLOT:	1.66667[ms] (= 80/48[ms])
EFRAME:	26.66667[ms] (= 80/3[ms])

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STOP

Sequence Stop Segment

Function

Sets or queries stop segment of sequence.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STOP
<list_table_no>,<stop_seg>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STOP? <list_table_no>
```

Response

<stop_seg>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<stop_seg>	Stop segment number
Range	Start segment number to 10001
Default	1

Example of Use

To set the stop segment of Sequence table #1 to 30:

:SOUR:GPRF:GEN:SEQ:RX:GEN:STOP 1,30

To query the stop segment of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:GEN:STOP? 1

> 30

Remarks

When the stop segment number is smaller than the start segment number, a parameter error occurs.

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:LEVel

Level of Segment in Sequence

Function

Sets or queries level of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:LEVel
<list_table_no>,<segment_no>,<level>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:LEVel? <list_table_no>,<segment_no>
```

Response

<level>

Unit dBm

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<level>	Level
Range	–230.0 to –110.0 dBm (*1)
Resolution	0.1 dBm
Suffix Code	DBM (uses DBM when omitted)
Default	–120.0 dBm

Example of Use

To set the output level of Segment 10 of Sequence table 1 to –50.0 dBm:

:SOUR:GPRF:GEN:SEQ:RX:LEV 1,10,-50.0DBM

To query the output level setting value of Segment 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:LEV? 1,10

> -50.0

Remarks

(*1) Settable range with combination of ports and cable loss values

An execution error occurs if the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:LEVel:ALL

Level Group of All Segments in Sequence

Function

Sets or queries level of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:LEVel:ALL <list_table_no>,<level_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:LEVel:ALL? <list_table_no>
```

Response

<level_n>

Unit dBm

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<level_n>		
	<level_1>	Level of Segment 1
	<level_2>	Level of Segment 2
	....	
	<level_10001>	Level of Segment 10001
	Range	–230.0 to –110.0 dBm (*1)
	Resolution	0.1 dBm
	Suffix Code	DBM (uses DBM when omitted)
	Default	–120.0 dBm

Example of Use

To set collectively the level of each segment of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:LEV:ALL 1,-50.0DBM,-50.0DBM,-50.0DBM

To query the level of each segment of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:LEV:ALL? 1

> -50.0,-50.0,-50.0,-120.0,-120.0,...,-120.0

Remarks

The command parameter <level_n> can be set to 10001 at maximum. If ten parameters are set, the levels for Segment 1 to Segment 10 are set. In this case, the frequencies for Segment 11 and subsequent segments are not changed.

(*1) Settable range settable with combination of ports and cable loss values

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl

Loop Control of Segment in Sequence

Function

Sets or queries repetition control (Loop) of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl
<list_table_no>,<segment_no>,<control>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl?
<list_table_no>,<segment_no>
```

Response

<control>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<control>	Repetition control
LOOP	Repetition control executed
NSEGMENT	Repetition control not executed, but next segment executed
Default	NSEGMENT

Example of Use

To set the repetition control of Segment 10 of Sequence table #1 to Loop control:

:SOUR:GPRF:GEN:SEQ:RX:NSLC 1,10,LOOP

To query the repetition control setting of Segment 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:NSLC? 1,10

> LOOP

Related Commands

:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt
:SOURce:GPRF:GENerator:SEQuence:RX:ERRor?

Remarks

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:ALL

Loop Control Group of All Segment in Sequence

Function

Sets or queries repetition control (Loop) of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:ALL
<list_table_no>,<control_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:ALL? <list_table_no>
```

Response

<control_n>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16

<control_n>

<control_1>	Repetition control of Segment 1
<control_2>	Repetition control of Segment 2
….	
<control_10001>	Repetition control of Segment 10001
LOOP	Repetition control executed
NSEGMENT	Repetition control not executed, but next segment executed
Default	NSEGMENT

Example of Use

To set repetition control of each segment of Sequence table #1 collectively:

:SOUR:GPRF:GEN:SEQ:RX:NSLC:ALL 1,LOOP,LOOP,NSEGMENT,LOOP

To query the repetition control of each segment of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:NSLC:ALL? 1

>LOOP,LOOP,NSEGMENT,LOOP,NSEGMENT,NSEGMENT,...,NSEGMENT

Remarks

The command parameter <control_n> can be set to 10001 at maximum. If ten parameters are set, the repetition controls for Segment 1 to Segment 10 are set. In this case, the repetition controls for Segment 11 and subsequent segments are not changed.

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt

Loop Count of Segment in Sequence

Function

Sets or queries repetition count at repetition control (Loop) of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt
<list_table_no>,<segment_no>,<count>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt?
<list_table_no>,<segment_no>
```

Response

<count>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<count>	Repetition count
Range	0 to 100 (*1)
Default	0

Example of Use

To set the repetition count of Segment 10 of Sequence table #1 to 2:
:SOUR:GPRF:GEN:SEQ:RX:NSLC:COUN 1,10,2

To query the repetition count of Segment 10 of Sequence table #1:
:SOUR:GPRF:GEN:SEQ:RX:NSLC:COUN? 1,10
> 2

Remarks

(*1) Repetition control is not executed when the repetition count is 0.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
• Loading waveform
• Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt:ALL

Loop Count Group of All Segments in Sequence

Function

Sets or queries repetition count at repetition control (Loop) of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt:ALL
<list_table_no>,<count_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:COUNt:ALL? <list_table_no>
```

Response

<count_n>

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<count_n>		
	<count_1>	Repetition count of Segment 1
	<count_2>	Repetition count of Segment 2
	....	
	<count_10001>	Repetition count of Segment 10001
	Range	0 to 100 (*1)
	Default	0

Example of Use

To set the repetition count of each segment of Sequence table 1 collectively:
:SOUR:GPRF:GEN:SEQ:RX:NSLC:COUN:ALL 1,2,2,2

To query the repetition count of each segment of Sequence table 1:
:SOUR:GPRF:GEN:SEQ:RX:NSLC:COUN:ALL? 1
> 2,2,2,0,0,...,0

Remarks

The command parameter <count_n> can be set to 10001 at maximum. If ten parameters are set, the repetition counts for Segment 1 to Segment 10 are set. In this case, the repetition counts for Segment 11 and subsequent segments are not changed.

(*1) If the repetition count is set to 0, repetition control is not performed.
An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment

Segment Number of Segment in Sequence for Loop Control

Function

Sets or queries jump destination segment number at repetition control (Loop) of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment
<list_table_no>,<segment_no>,<jump_segment>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment?
<list_table_no>,<segment_no>
```

Response

```
<jump_segment>
```

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<jump_segment >	Jump destination segment number
Range	1 to segment_no
Default	1

Example of Use

To set the jump destination segment number on Loop control of Segment 10 of Sequence table #1 to 5:

```
:SOUR:GPRF:GEN:SEQ:RX:NSLC:LSEG 1,10,5
```

To query the jump destination segment number on Loop control of Segment 10 of Sequence table #1:

```
:SOUR:GPRF:GEN:SEQ:RX:NSLC:LSEG? 1,10
> 5
```

Remarks

A parameter error occurs if the jump destination segment number is bigger than the set segment number.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment:ALL

Loop Jump Segment Group of All Segments in Sequence for Loop Control

Function

Sets or queries jump destination segment number on repetition control (Loop) of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment:ALL
<list_table_no>,<jump_segment_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:NSLControl:LSEGment:ALL?
<list_table_no>
```

Response

<jump_segment_n>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16

<jump_segment_n>

<jump_segment_1>	Jump destination segment number of Segment 1
<jump_segment_2>	Jump destination segment number of Segment 2
….	
<jump_segment_10001>	Jump destination segment number of Segment 10001

Range	1 to n (*1)
Default	1

Example of Use

To set the jump destination segment number of each segment of Sequence table #12 collectively:

:SOUR:GPRF:GEN:SEQ:RX:NSLC:LSEG:ALL 12,1,1,2

To query the jump destination segment number of each segment of Sequence table #12:

:SOUR:GPRF:GEN:SEQ:RX:NSLC:LSEG:ALL? 12

> 1,1,2,1,1,...,1

Remarks

The command parameter <jump_segment_n> can be set to 10001 at maximum. If ten parameters are set, the jump destination segment numbers for Segment 1 to Segment 10 are set. In this case, the jump destination segments for Segment 11 and subsequent segments are not changed.

(*1) A parameter error occurs if the jump destination segment number is bigger than the set segment number.

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe

Output Port of Segment in Sequence

Function

Sets or queries output port of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe
<list_table_no>,<segment_no>,<port>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe?
<list_table_no>,<segment_no>
```

Response

<port>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<port>	Port setting
PORT1	Port 1
PORT2	Port 2
PORT3	Port 3
PORT4	Port 4
Default	PORT1

Details

The ports on the front panel that output RF signals are set.

Example of Use

To set the output port of Segment 10 of Sequence table #1 to PORT1:
:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT 1,10,PORT1

To query the output port of Segment 10 of Sequence table #1:
:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT? 1,10
> PORT1

Remarks

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe:ALL

Output Port Group of All Segments in Sequence

Function

Sets or queries output port of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe:ALL
<list_table_no>,<port_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe:ALL? <list_table_no>
```

Response

<port_n>

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<port_n>	Port setting for Segment n	
	<port_1>	Port setting for Segment 1
	<port_2>	Port setting for Segment 2
	….	
	<port_10001>	Port setting for Segment 10001
	PORT1	Port 1
	PORT2	Port 2
	PORT3	Port 3
	PORT4	Port 4
	Default	PORT1

Example of Use

To set the output port of each segment of Sequence table #10 collectively:

:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT:ALL 10,PORT2,PORT3,PORT3

To query the output port of each segment of Sequence table #10:
:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT:ALL? 10
> PORT2,PORT3,PORT3,PORT1,PORT1,...,PORT1

Remarks

The command parameter <port_n> can be set to 10001 at maximum. If ten parameters are set, the output ports for Segment 1 to Segment 10 are set. In this case, the output ports for Segment 11 and subsequent segments are not changed.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt

Steps of Segment in Sequence

Function

Sets or queries step count of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt
<list_table_no>,<segment_no>,<count>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt?
<list_table_no>,<segment_no>
```

Response

<count>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<count>	Segment continuous step count
Range	1 to 100
Default	1

Example of Use

To set the step count of Segment 10 of Sequence table #1 to 2:
:SOUR:GPRF:GEN:SEQ:RX:STEP:COUN 1,10,2

To query the step count of Segment 10 of Sequence table #1:
:SOUR:GPRF:GEN:SEQ:RX:STEP:COUN? 1,10
> 2

Related Command

Set the time per step with :SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STEP:LENGth.

Remarks

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt:ALL

Steps Group of All Segments in Sequence

Function

Sets or queries the step count of each segment of the SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt:ALL
<list_table_no>,<count_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:STEP:COUNt:ALL? <list_table_no>
```

Response

<count_n>

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<count_n>		
	<count_1>	Segment 1 continuous step count
	<count_2>	Segment 2 continuous step count
	….	

<count_10001>	Segment 10001 continuous step count
Range	1 to 100
Default	1

Example of Use

To set the step count of each segment of Sequence table #1 collectively:
:SOUR:GPRF:GEN:SEQ:RX:STEP:COUN:ALL 1,2,2,2

To query the step count of each segment of Sequence table #1:
:SOUR:GPRF:GEN:SEQ:RX:STEP:COUN:ALL? 1
> 2,2,2,1,1,...,1

Remarks

The command parameter <count_n> can be set to 10001 at maximum. If ten parameters are set, the step counts for Segment 1 to Segment 10 are set. In this case, the step counts for Segment 11 and subsequent segments are not changed.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay

Trigger Delay of Segment in Sequence

Function

Sets or queries delay time of the specified one segment in the SG sequencer data.
The delay time is the time from the trigger occurrence, which is the trigger source for the SG sequencer transition, until the trigger reception.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay
<list_table_no>,<segment_no>,<delay>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay?
<list_table_no>,<segment_no>
```

Response

<delay>

5

SCPI Command Reference

Unit	ms

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<delay>	Delay time
Range	0.000 to 1000.000 ms
Resolution	0.001 ms
Suffix Code	S, MS, US, NS (uses MS when omitted)
Default	0 ms

Example of Use

To set the delay time of Segment 10 of Sequence table #1 to 1.000 ms:

```
:SOUR:GPRF:GEN:SEQ:RX:TRIG:DEL 1,10,1.000
```

To query the delay time of Segment 10 of Sequence table #1:

```
:SOUR:GPRF:GEN:SEQ:RX:TRIG:DEL? 1,10
> 1.000
```

Remarks

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay:ALL

Trigger Delay Group of All Segments in Sequence

Function

Sets or queries delay time of each segment in the SG sequencer data collectively.
The delay time is the time from the trigger occurrence, which is the trigger source for the SG sequencer transition, until the trigger reception.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay:ALL
<list_table_no>,<delay_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:DELay:ALL? <list_table_no>
```

Response

<delay_n>

Unit ms

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<delay_n>		
	<delay_1>	Delay time of Segment 1
	<delay_2>	Delay time of Segment 2
	....	
	<delay_10001>	Delay time of Segment 10001
	Range	0.000 to 1000.000 ms
	Resolution	0.001 ms
	Suffix Code	S, MS, US, NS (uses MS when omitted)
	Default	0 ms

Example of Use

To set the delay time of each segment of Sequence table #5 collectively:

:SOUR:GPRF:GEN:SEQ:RX:TRIG:DEL:ALL 5,1.000,2.000,3.000,4.000,5.000

To query the delay time of each segment of Sequence table #5:

:SOUR:GPRF:GEN:SEQ:RX:TRIG:DEL:ALL? 5

> 1.000,2.000,3.000,4.000,5.000,0.000,0.000,...,0.000

Remarks

The command parameter <delay_n> can be set to 10001 at maximum. If ten parameters are set, the delay time values for Segment 1 to Segment 10 are set. In this case, the delay time for Segment 11 and subsequent segments is not changed.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce

Trigger Source of Segment in Sequence

Function

Sets or queries trigger source of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce
<list_table_no>,<segment_no>,<trigger>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce?
<list_table_no>,<segment_no>
```

Response

<trigger>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<trigger>	Trigger source
EXTERNAL	External trigger
VIDEO	Level trigger
WFGEND	Waveform group end
WFM1	Waveform Marker 1
WFM2	Waveform Marker 2
WFM3	Waveform Marker 3
Default	EXTERNAL

Details

The trigger to be used when the segment termination condition is a trigger is selected.

The external trigger is a voltage change in the signal input to the connector on the MT8870A rear panel.
The level trigger is the level of RF signal input to MT8870A exceeding the specified value.
For waveform group end, a trigger is the timing when reproduced waveform pattern is changed. The trigger output when the waveform pattern is changed is set by :SOURce:GPRF:GENerator:SEQuence:WAVeform:GETRigger.
The waveform marker is the signal embedded in the IQ data of the waveform file.

Example of Use

To set the trigger source of Segment 10 of Sequence table #1 to the external trigger:

```
:SOUR:GPRF:GEN:SEQ:RX:TRIG:SOUR 1,10,EXTERNAL
```

To query the trigger source of Segment 10 of Sequence table #1:

```
:SOUR:GPRF:GEN:SEQ:RX:TRIG:SOUR? 1,10
> EXTERNAL
```

Remarks

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce:ALL

Trigger Source Group of All Segments in Sequence

Function

Sets or queries trigger source of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce:ALL
<list_table_no>,<trigger_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce:ALL? <list_table_no>
```

Response

<trigger_n>

Parameters

<list_table_no>	Sequence table number	
Range	1 to 16	
<trigger_n>		
	<trigger_1>	Trigger source of Segment 1
	<trigger_2>	Trigger source of Segment 2
	....	
	<trigger_10001>	Trigger source of Segment 10001
	EXTERNAL	External trigger
	VIDEO	Level trigger
	WFGEND	Waveform group end
	WFM1	Waveform Marker 1
	WFM2	Waveform Marker 2
	WFM3	Waveform Marker 3
	Default	EXTERNAL

Example of Use

To set the trigger source of each segment of Sequence table #7 collectively:

:SOUR:GPRF:GEN:SEQ:RX:TRIG:SOUR:ALL 7,WFM3,WFM2,WFM1,WFM2,WFM3

To query the trigger source of each segment of Sequence table #7:

:SOUR:GPRF:GEN:SEQ:RX:TRIG:SOUR:ALL? 7

> WFM3,WFM2,WFM1,WFM2,WFM3,EXTERNAL,EXTERNAL,...,EXTERNAL

Remarks

The command parameter <trigger_n> can be set to 10001 at maximum.

If ten parameters are set, the trigger source for Segment 1 to Segment 10 are set.

In this case, the trigger source for Segment 11 and subsequent segments is not changed.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger

Waveform Change Trigger of Segment in Sequence

Function

Sets or queries waveform change trigger of specified one segment of SG sequencer data.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger
<list_table_no>,<segment_no>,<on_off>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger?
<list_table_no>,<segment_no>
```

Response

<on_off_res>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<on_off>	Waveform change trigger setting
0	Waveform change trigger not generated
1	Trigger generated at switching to next segment
OFF	Waveform change trigger not generated
ON	Trigger generated at switching to next segment
Default	0
<on_off_res>	Waveform change trigger setting
0	Trigger not generated
1	Trigger generated at switching to next segment

Example of Use

To set the waveform change trigger of Segment 10 of Sequence table #1 to no generation:

:SOUR:GPRF:GEN:SEQ:RX:WCTR 1,10,OFF

To query the waveform change trigger of Segment 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:RX:WCTR? 1,10

> 0

Related Command

To change the modulation waveform with the waveform change trigger, set the waveform pattern trigger source to SGRequest
using :SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce.

Remarks

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

• Loading waveform
• Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger:ALL

Waveform Change Trigger Group of All Segments in Sequence

Function

Sets or queries waveform change trigger of each segment of SG sequencer data collectively.

Command

```
:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger:ALL
<list_table_no>,<on_off_n>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:WCTRigger:ALL? <list_table_no>
```

Response

<on_off_n_res>

Parameters

<list_table_no>	Sequence table number	
<on_off_n>	Waveform change trigger setting	
	<on_off_1>	Waveform change trigger setting of Segment 1
	<on_off_2>	Waveform change trigger setting of Segment 2
	....	
	<on_off_10001>	Waveform change trigger setting of Segment 10001
	0	Trigger not generated
	1	Trigger generated at switching to next segment
	OFF	Trigger not generated
	ON	Trigger generated at switching to next segment
	Default	0
<on_off_n_res>		
	<on_off_1_res>	Waveform change trigger setting of Segment 1
	<on_off_2_res>	Waveform change trigger setting of Segment 2
	....	
	<on_off_10001_res>	Waveform change trigger setting of Segment 10001
	0	Trigger not generated
	1	Trigger generated at switching to next segment

Example of Use

To set the waveform change trigger of each segment of Sequence table #3 collectively:

5

SCPI Command Reference

:SOUR:GPRF:GEN:SEQ:RX:WCTR:ALL 3,OFF,ON,OFF,ON,ON

To query the waveform change trigger of each segment of Sequence table #3:
:SOUR:GPRF:GEN:SEQ:RX:WCTR:ALL? 3
> 0,1,0,1,1,0,0,...,0

Related Command

:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce

Remarks

The command parameter<on_off_n> can be set to 10001 at maximum. If ten parameters are set, the waveform change triggers for Segment 1 to Segment 10 are set. In this case, the waveform change triggers for Segment 11 and subsequent segments are not changed.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:STATus?

Sequence Status

Function

Queries sequence operation status in sequence mode.

Query

```
:SOURce:GPRF:GENerator:SEQuence:STATus?
```

Response

<status>

Parameter

<status>	Sequence operation status in sequence mode
0	Stopped (Inactive)
1	In progress (Active)

Example of Use

To query the sequence operation status in the sequence mode:

:SOUR:GPRF:GEN:SEQ:STAT?

> 1

:SOURce:GPRF:GENerator:SEQuence:STORe

Save Sequence Parameter

Function

Saves SG sequencer information of specified sequence data to file.

Command

```
:SOURce:GPRF:GENerator:SEQuence:STORe <list_table_no>,<file_name>
```

Parameters

<list_table_no>	Sequence table number to save SG sequencer data
Range	1 to 16
<file_name>	SG sequencer data saving file name

Example of Use

To save the SG sequencer data of Sequence table #1 to the file name "SeqFile.xml":
:SOUR:GPRF:GEN:SEQ:STOR 1,"SeqFile.xml"

Remarks

The extension of the waveform file (xml) can be omitted.
An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

An execution error occurs if there is already a file with the same name.

:SOURce:GPRF:GENerator:SEQuence:WAVeform:ENDCondition

Pattern End Condition of Sequence Waveform

Function

Sets or queries condition to change to next registered waveform pattern when waveform pattern of waveform list table finished.

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:ENDCondition
<list_table_no>,<index>,<condition>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:ENDCondition?
<list_table_no>,<index>
```

Response

<condition>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<index>	Waveform index number
Range	1 to 200
<condition>	End condition
NONE	None (waveform not changed)
REPEAT	Changes to next waveform when repetition count finished
TRIGGER	Changes to next waveform with trigger
Default	REPEAT

Example of Use

To set the waveform pattern end condition of Waveform index number 10 of Sequence table #1 to None (waveform not changed):

:SOUR:GPRF:GEN:SEQ:WAV:ENDC 1,10,NONE

To query the waveform pattern end condition of Waveform index number 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:WAV:ENDC? 1,10

> NONE

Remarks

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:DM:POLarity

Sequence Waveform Spectrum Reverse

Function

Sets or queries whether output waveform IQ inverted at sequence operation.

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:DM:POLarity
<list_table_no>,<mode>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:DM:POLarity?
<list_table_no>
```

Response

<mode>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<mode>	Mode
NORMAL	IQ not inverted
INVERT	IQ inverted
Default	NORMAL

Example of Use

To set the output waveform at sequence operation to no IQ inversion:
:SOUR:GPRF:GEN:SEQ:WAV:GEN:DM:POL 1,NORMAL

To query the output waveform at sequence operation:
:SOUR:GPRF:GEN:SEQ:WAV:GEN:DM:POL? 1
> NORMAL

Remarks

Common setting for the waveform list table.
The IQ inversion setting is applied to all waveform patterns during sequence execution.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:PCOunt?

Number of Registered Waveform

Function

Queries the number of waveforms registered contiguously from index 1 in waveform list table.

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:PCOunt? <list_table_no>
```

Response

<count_res>

Parameter

<list_table_no>	Sequence table number
Range	1 to 16
<count_res>	the number of waveforms registered contiguously from index 1 in waveform list table

Example of Use

To query the number of waveforms registered contiguously from index 1 in waveform list table

:SOUR:GPRF:GEN:SEQ:WAV:GEN:PCO? 1

> 3

:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:DELay

Trigger Delay of Sequence Waveform Pattern

Function

Sets or queries trigger delay time of waveform pattern registered in waveform list table.

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:DELay
<list_table_no>,<delay>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:DELay?
<list_table_no>
```

Response

<delay>

Unit	ms

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<delay>	Delay time
Range	0.000 to 1000.000 ms
Resolution	0.001 ms
Suffix Code	S, MS, US, NS (uses MS when omitted)
Default	0 ms

Example of Use

To set the waveform pattern delay time of Sequence table #1 to 100.000 ms:

:SOUR:GPRF:GEN:SEQ:WAV:GEN:TRIG:DEL 1,100.000

To query the waveform pattern delay time of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:WAV:GEN:TRIG:DEL? 1

> 100.000

Remarks

Common setting for the waveform list table.

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce

Trigger Source for Sequence Waveform Pattern

Function

Sets queries trigger source for waveform list table waveform pattern change.

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce
<list_table_no>,<trigger>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce?
<list_table_no>
```

Response

<trigger>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<trigger>	Trigger source
EXTERNAL	External trigger
VIDEO	Level trigger
SGREQUEST	Trigger from SG sequencer
MANUAL	Trigger with user commands
Default	EXTERNAL

Example of Use

To set the waveform pattern trigger source of Sequence table #1 to the external trigger:

:SOUR:GPRF:GEN:SEQ:WAV:GEN:TRIG:SOUR 1,EXTERNAL

To query the waveform pattern trigger source of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:WAV:GEN:TRIG:SOUR? 1

> EXTERNAL

Remarks

Common setting for the waveform list table.

An execution error occurs if the operation mode is:

- Sequence mode running

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:WAVeform:GETRigger

Group End Trigger of Sequence Waveform Pattern

Function

Sets or queries trigger output (Group End Trigger) for SG sequencer at waveform list table waveform pattern change.

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GETRigger
<list_table_no>,<index>,<on_off>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:GETRigger?
<list_table_no>,<index>
```

Response

<on_off_res>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<index>	Waveform index number
Range	1 to 200
<on_off>	Trigger output
0	Trigger not generated
1	Trigger generated
OFF	Trigger not generated
ON	Trigger generated
Default	0
<on_off_res>	Trigger output
0	Trigger not generated
1	Trigger generated

Example of Use

To set the trigger output at waveform pattern end of Waveform index number 10 of Sequence table #1 to no generation:

:SOUR:GPRF:GEN:SEQ:WAV:GETR 1,10,OFF

To query the trigger output on waveform pattern end of Waveform index number 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:WAV:GETR? 1,10
> 0

Related Command

To use the trigger set with this command on the SG sequence table, set the trigger to WFGend using :SOURce:GPRF:GENerator:SEQuence:RX:TRIGger:SOURce.

Remarks

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
• Loading waveform
• Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:WAVeform:IREPetition

Repetition Count of Sequence Waveform Pattern

Function

Sets or queries waveform pattern repetition count.

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:IREPetition
<list_table_no>,<index>,<count>
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:IREPetition?
<list_table_no>,<index>
```

Response

<count>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<index>	Waveform index number
Range	1 to 200
<count>	Repetition count
Range	1 to 65535
Default	1

Example of Use

To set the waveform pattern repetition count of Waveform index number 10 of Sequence table #1 to 5:

:SOUR:GPRF:GEN:SEQ:WAV:IREP 1,10,5

To query the waveform pattern repetition count of Waveform index number 10 of Sequence table #1:

:SOUR:GPRF:GEN:SEQ:WAV:IREP? 1,10

> 5

Remarks

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:WAVeform:MTEXecute

Manual Trigger for Sequence Waveform

Function

Generates trigger to change waveform pattern.
This command can be used when the waveform list table trigger source is set to the trigger with user commands (MANUAL).

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:MTEXecute
```

Example of Use

To generate the trigger to change the waveform pattern:

:SOUR:GPRF:GEN:SEQ:WAV:MTEX

Related Command

:SOURce:GPRF:GENerator:SEQuence:WAVeform:GENeral:TRIGger:SOURce

Remarks

An execution error occurs if the operation mode is:
- Normal mode
- Stopped Sequence mode

:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:DELete

Sequence Waveform Pattern Delete

Function

Deletes waveform pattern set (registered) at waveform list table from list table.

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:DELete
<list_table_no>,<index>
```

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<index>	Waveform index number
Range	1 to 200

Example of Use

To delete the waveform pattern of Waveform index number 10 of Sequence table #1 from the list:

:SOUR:GPRF:GEN:SEQ:WAV:PATT:DEL 1,10

Remarks

An execution error occurs when the operation mode is:

- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:SELect

Sequence Waveform Pattern Configuration

Function

Sets or queries waveform file name and group number with specification of Waveform index number in the waveform list table.

Command

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:SELect
<list_table_no>,<index>,<file_name>[,<group_number>]
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:PATTern:SELect?
<list_table_no>,<index>
```

Response

<file_name>,<group_number>

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<index>	Waveform index number
Range	1 to 200
<file_name>	Waveform file name (character string)
<group_number>	Group number
Range	Range of group number defined by waveform file

Example of Use

To set the waveform pattern and group number of Waveform index number 10 of Sequence table #3 to the file name “MV887011A_WCDMA_0002” and 1:

:SOUR:GPRF:GEN:SEQ:WAV:PATT:SEL 3,10,”MV887011A_WCDMA_0002”,1

To query the waveform file name and group number of Waveform index number 10 of Sequence table #3:

:SOUR:GPRF:GEN:SEQ:WAV:PATT:SEL? 3,10

> "MV887011A_WCDMA_0002",1

Remarks

The file name is a character string without the extension (xml).
When the group number is omitted, 1 is set.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:

- Loading waveform
- Defragmenting waveform memory

A parameter error occurs if the specified group number does not exist.

:SOURce:GPRF:GENerator:SEQuence:WAVeform:SYNChronize:STARt:STATe?

Sequence Waveform Synchronization State

Function

Queries synchronized output stand-by state of waveform pattern.

Query

```
:SOURce:GPRF:GENerator:SEQuence:WAVeform:SYNChronize:STARt:STATe?
```

Response

<status>

Parameters

<status>	Synchronization stand-by state
0	Synchronized output stand-by state
1	Release of synchronized output stand-by state

Example of Use

To query synchronized output stand-by state of waveform pattern.

:SOUR:GPRF:GEN:SEQ:WAV:SYNC:STAR:STAT?

> 1

:SOURce:GPRF:GENerator:STATe

RF Output

Function

Sets or queries On/Off of RF signal output.

Command

```
:SOURce:GPRF:GENerator:STATe <on_off>
```

Query

```
:SOURce:GPRF:GENerator:STATe?
```

Response

<on_off_res>

Parameters

<on_off>	RF output On/Off
0	Off
1	On
OFF	Off
ON	On
Default	0

<on_off_res>	RF output On/Off
0	Off
1	On

Example of Use

To set RF output to On:

:SOUR:GPRF:GEN:STAT ON

To query the status of the RF output:

:SOUR:GPRF:GEN:STAT?

> 1

Remarks

When this setting is OFF, set the output level to the minimum value internally.

An execution error occurs when the operation mode is:
- Sequence mode

An execution error occurs when the following processes are executing:
- Loading waveform
- Defragmenting waveform memory

Chapter 6 Native Command Reference

This chapter describes the details of Native commands.

6.1 List of Commands

6.1.1 IEEE488.2 common commands

Function	Command	Query	Response
Clear Status Command	*CLS	-----	-----
Standard Event Status Enable Command	*ESE register	*ESE?	register
Standard Event Status Register Query	-----	*ESR?	register
Identification Query	-----	*IDN? [index]	ANRITSU,MT8870A[:slot:MU887000A],serial[:serial2],firm[:firm2]
Operation Complete	*OPC	*OPC?	operation
Preset	*RST	-----	-----
Service Request Enable Command	*SRE register	*SRE?	register
Status Byte Register	-----	*STB?	register
Trigger	*TRG	-----	-----
Self Test	-----	*TST?	test_result
Wait to Continue	*WAI	-----	-----

6.1.2 Basic operation commands

Calibration

Function	Command	Query	Response
Execute BAND Calibration	BANDCAL	-----	-----
BAND Calibration Temperature	BANDCAL_TEMP temperature	-----	-----
BAND Calibration Last Time	-----	BANDCAL_TM?	year,month,day,hour,min
BAND Calibration Error Last Time	-----	BANDCALERR_TM?	year,month,day,hour,min
Band Calibration Error Count	-----	BANDCALERRCNT?	count
Result of Band Calibration	-----	BANDCALRESULT?	Result
Temperature at Full Calibration Execution	-----	CALTEMP?	temp
Elapsed Time after Full Calibration	-----	CALTIME?	time
Execute FULL Calibration	FULLCAL	-----	-----
FULL Calibration Last Time	-----	FULLCAL_TM?	year,month,day,hour,min
FULL Calibration Error Last Time	-----	FULLCALERR_TM?	year,month,day,hour,min
Full Calibration Error Count	-----	FULLCALERRCNT?	count
Result of Full Calibration	-----	FULLCALRESULT?	Result

Communications

Function	Command	Query	Response
Delimiter	DELM code	DELM?	code
Domain Name	DOMAINNAME name	DOMAINNAME?	name
GPIB Address	GPIBADDR gp_address	GPIBADDR?	gp_address
GPIB Address	GPIBADDRALL address1,address2,address3,address4	GPIBADDRALL?	address1,address2,address3,address4
Host Name	HOSTNAME host	HOSTNAME?	host
Get Current IPv4 Address	-----	IPV4_CURRENT_ADDR?	num,ipaddr
Get Current Default Gateway	-----	IPV4_CURRENT_DEFAULTGATEWAY?	gateway
Get Current Subnet Mask	-----	IPV4_CURRENT_SUBNETMASK?	mask
DNS Server Auto	IPV4_DNS_AUTO_SW switch	IPV4_DNS_AUTO_SW?	switch
DNS Primary Address	IPV4_DNS_PRIMARY ip4addr	IPV4_DNS_PRIMARY?	ip4addr
DNS Secondary Address	IPV4_DNS_SECONDARY ip4addr	IPV4_DNS_SECONDARY?	ip4addr
IPv4 Address	IPV4_STATIC_ADDR_ALLMODULE ipv4_addr1,ipv4_addr2,ipv4_addr3,ipv4_addr4	IPV4_STATIC_ADDR_ALLMODULE?	ipv4_addr1,ipv4_addr2,ipv4_addr3,ipv4_addr4
Default Gateway	IPV4_STATIC_DEFAULTGATEWAY gateway	IPV4_STATIC_DEFAULTGATEWAY?	gateway
Subnet Mask	IPV4_STATIC_SUBNETMASK mask	IPV4_STATIC_SUBNETMASK?	mask
IPv4 Type	IPV4_TYPE ipv4type	IPV4_TYPE?	ipv4type

Communications (Cont'd)

Function	Command	Query	Response
Get Current IPv6 Address and Prefix	-----	IPV6_CURRENT_ADDR?	num,ipaddr
Get Default Router Address	-----	IPV6_CURRENT_DEFAULTROUTER?	ip6addr
Get Link Local Address	-----	IPV6_CURRENT_LINK_LOCAL_ADDR?	ip6addr
DNS Server Auto	IPV6_DNS_AUTO_SW switch	IPV6_DNS_AUTO_SW?	switch
DNS Primary Address	IPV6_DNS_PRIMARY ip6addr	IPV6_DNS_PRIMARY?	ip6addr
DNS Secondary Address	IPV6_DNS_SECONDARY ip6addr	IPV6_DNS_SECONDARY?	ip6addr
IPv6 Address	IPV6_STATIC_ADDR_ALLMODULE ipv6_addr1,ipv6_addr2,ipv6_addr3,ipv6_addr4	IPV6_STATIC_ADDR_ALLMODULE?	ipv6_addr1,ipv6_addr2,ipv6_addr3,ipv6_addr4
Default Router	IPV6_STATIC_DEFAULTROUTER ip6addr	IPV6_STATIC_DEFAULTROUTER?	ipv6_addr
IPv6 Type	IPV6_TYPE ipv6type	IPV6_TYPE?	ipv6type
IP Version	IPVER ipver	IPVER?	ipver
MAC Address	-----	MACADDR?	macaddr
Network Restart	NETRESTART	-----	-----
Raw Socket Port Number	PORTSETTING port	PORTSETTING?	port
Terminator of Message	TRM code	TRM?	code

Event Status Register

Function	Command	Query	Response
End Event Status (Generator) Enable Command	`ESE0 register`	`ESE0?`	`register`
Error Event Status (Generator) Enable Command	`ESE1 register`	`ESE1?`	`register`
End Event Status (Measure) Enable Command	`ESE2 register`	`ESE2?`	`register`
Error Event Status (Measure) Enable Command	`ESE3 register`	`ESE3?`	`register`
End Event Status (Generator) Register Query	-----	`ESR0?`	`register`
Error Event Status (Generator) Register Query	-----	`ESR1?`	`register`
End Event Status (Measurement) Register Query	-----	`ESR2?`	`register`
Error Event Status (Measurement) Register Query	-----	`ESR3?`	`register`

FTP

Function	Command	Query	Response
FTP User List	-----	`FTPUSER_LIST?`	`name`
Change FTP User Name	`FTPUSER_CHANGE before,after`	-----	-----
Change FTP User Password	`FTPUSER_PASSWD name,before,after`	-----	-----

Information

Function	Command	Query	Response
License Information	-----	KEYINFO? model	kind,days
Log Clear	LOGCLEAR	-----	-----
Log Reading	-----	LOGREAD? [size]	log
Log Size Reading	-----	LOGSIZE?	log
Main Frame Information	-----	MFINFO? [index]	model name, model number,…
Main Frame Model Number	-----	MFMODEL?	model
Main Frame Serial Number	-----	MFSERIAL?	serial
Main Frame Temperature	-----	MFTEMP?	temp
Main Frame Running Time Count Read	-----	MFTMCNT?	time
Module Information	-----	SYSINFO? [index]	-----
Module Model Number	-----	SYSMODEL?	model
Module Serial Number	-----	SYSSERIAL?	serial
Module Slot Number	-----	SYSSLOT?	slot
Module Temperature	-----	SYSTEMP?	temp
Module Running Time Counter Read	-----	SYSTMCNT?	time
Package Version	-----	SYSVER?	ver
Chassis Power-on Count	-----	MFPOWERONCNT?	num
Module Insertion Count	-----	SYSPLUGCNT?	num
Module Power-on Count	-----	SYSPOWERONCNT?	num
Firmware Version	-----	MCFV?	ver
Option Number	-----	MCOPT? model	n,num
Type Name of Software	-----	MCSOPT?	n,model
Type Name of Waveform	-----	MCWOPT?	n,model
Drive Size Reading	-----	DSIZE? [target]	total,blank

Interface Setting

Function	Command	Query	Response
Set In/Out of Trigger Connector	MFTRIGPORT trigger	MFTRIGPORT?	trigger
Set Direction of RF Connector	PORT input,output	PORT?	input,output
Select Source of Trigger Input	SYSTRIGINSRC source	SYSTRIGINSRC?	source
Select Trigger Output Source	SYSTRIGOUTSRC source	SYSTRIGOUTSRC?	source

Loss Correction

Function	Command	Query	Response
Delete Loss Value from Cable Loss List	DELLOSSTBL [a[,b]]	-----	-----
External Loss On/Off	EXTLOSSW sw	EXTLOSSW?	sw
Loss Table	LOSSTBL index	LOSSTBL?	index
Number of Loss Table	-----	LOSSTBLCNT? index	num
Register Loss Value in Cable Loss List	LOSSTBLVAL frequency,Port1,2/out,Port1,2/in,Port3,4/out[,Port3,4/in]	LOSSTBLVAL? n	frequency,Port1,2/out,Port1,2/in,Port3,4/out[,Port3,4/in]
Register Loss Value into All Cable Loss List	LOSSTBLVALALL frequency,Port1out,Port1in,Port2out,Port2in,Port3out,Port3in,Port4out,Port3in,Port4in	LOSSTBLVALALL? n	frequency,Port1out,Port1in,Port2out,Port2in,Port3out,Port3in,Port4out,Port3in,Port4in

Reference Clock

Function	Command	Query	Response
Adjust Reference Clock	`ADJREFCLK val`	`ADJREFCLK?`	`val`
Adjust Reference Clock Default	-----	`ADJREFCLKDEF?`	`val`
Frequency Reference	`FREQREF source`	`FREQREF?`	`source`
Frequency Reference	`REF source`	`REF?`	`source_res`
Reference Signal Query	-----	`FREQREFSOURCE?`	`source`
Reference Signal Query	-----	`EXTREF?`	`source`

System

Function	Command	Query	Response
Set Date	`DATETIME year,month,day,hour,minute,second`	`DATETIME?`	`year,month,day,hour,minute,second`
Beep	`BEEP sw`	`BEEP?`	`sw`
Delete File	`FDEL filepath`	-----	-----
Check Firmware Update Package	-----	`FIRMCHECK? filename`	`result`
Firmware Update	`FIRMUPDATE target[,option]`	`FIRMUPDATE?`	`result`
File List	-----	`FLIST? path`	`num,filelist`
Go to Local	`GTL`	-----	-----
License Key Check	-----	`KEYCHECK? filename`	`result`
License Key Install	`KEYINST`	-----	-----
Reboot	`REBOOT`	-----	-----
Shutdown	`SHUTDOWN`	-----	-----
System Error	-----	`SYSERR?`	`code,msg`
System All Errors	-----	`SYSERRALL?`	`code,msg`
System Error Count	-----	`SYSERRCNT?`	`num`
System Error Code	-----	`SYSERRCODE?`	`code`

System (Cont'd)

Function	Command	Query	Response
System All Error Codes	-----	SYSERRCODEALL?	code
System Last Error	-----	SYSLASTERROR?	msg
Language Selection of Remote Command	SYST:LANG mode	SYST:LANG?	mode
User Area Delete	SYSTEMDEFAULT	-----	-----
System Test	-----	SYSTEST?	status

6.1.3 VSG commands

The abbreviations for SCPI commands are used in the Native command mode.

Example:

:SOURce:GPRF:GENerator:RFSettings:LEVel

SOUR:GPRF:GEN:RFS:LEV

:SOURce:GPRF:GENerator:SEQuence:RX:GENeral:STEP:LENGth

SOUR:GPRF:GEN:SEQ:RX:GEN:STEP:LENG

6.2 Details of Commands

This section explains the Native commands in alphabetical order.

■ Viewing command table

EX ....Command name (header)

Example ....Command function name

Function ....Command function

Command.................................. Programming command syntax

Query ....Query syntax

Response ....Response syntax

Parameter ....Parameter definition

Details .... Command restrictions and others

Example of Use................................. Command usage example

Related Commands ....Introduction of related commands

■ Suffix code list

Suffix code	Unit
GZ	GHz
KZ	kHz
MZ	MHz
Z	Hz

6.2.1 IEEE488.2 common commands

*CLS

Clear Status Command

Function

Clears standard event status register and status byte register (Output queue and MAV bit are excluded)

Command

```
*CLS
```

Details

Sending after program message terminator or before query message unit clears all event status byte registers and event queues

Example of Use

To clear the standard event status register and status byte register (Output queue and MAV bit excluded):

*CLS

Related Commands

*RST Status byte registers and event status registers are not cleared.

*ESE Only standard event status registers are cleared.

*ESE

Standard Event Status Enable Command

Function

Sets standard event status enable register

When a specified event occurs, the end summary bit (ESB) value in bit5 of the status byte register is set to 1 (true).

Command

```
*ESE register
```

Query

```
*ESE?
```

Response

Register	Standard event status enable register
Value = bit0 + bit1 + ... + bit7	
bit0 = 2^0 = 1	Operation complete
bit1 = 2^1 = 2	Request control
bit2 = 2^2 = 4	Query error
bit3 = 2^3 = 8	Device error
bit4 = 2^4 = 16	Execution error
bit5 = 2^5 = 32	Command error
bit6 = 2^6 = 64	User request
bit7 = 2^7 = 128	Power up

Parameter

Register	Standard event status enable register
Range	0 to 255
Resolution	1
Default	0

Details

The sum of the values for bits to be enabled from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, corresponding to the standard event status enable register bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the parameter.

Example of Use

To enable bit 2 of the standard event status enable register 0:

*ESE 2

*ESE?

> 2

*ESR?

Standard Event Status Register Query

Function

Queries standard event status register

The event occurrence can be identified using the retrieved value.

Query

```
*ESR?
```

Response

Register	Standard event status register
Value = bit0 + bit1 + ... + bit7	
bit0 = 2^0 = 1	Operation completion
bit1 = 2^1 = 2	Request control
bit2 = 2^2 = 4	Query error
bit3 = 2^3 = 8	Device error
bit4 = 2^4 = 16	Execution error
bit5 = 2^5 = 32	Command error
bit6 = 2^6 = 64	User request
bit7 = 2^7 = 128	Power up

Parameter

register	Standard event status register
Range	0 to 255

Details

The sum of the values for bits of the occurring event from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, that correspond to the standard event status register (measurement) bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the response.

Example of Use

To query the standard event status register value:

*ESR?

*IDN?

Identification Query

Function

Queries device identification information

Query

```
*IDN? [index]
```

Response

When parameter is 1

ANRITSU,MT8870A,serial,firm

ANRITSU	Company name (Anritsu Corporation)
MT8870A	Model name
serial	Serial number (the measuring instrument-specific number)
firm	FPGA version of the measuring instrument

When parameter is 2

ANRITSU,MT8870A:slot:MU887000A,serial1:serial2,firm1:firm2

ANRITSU	Company name (Anritsu Corporation)
MT8870A	Model name
slot	Number of slot where module installed
MU887000A	Model name of the module
serial1	Serial number (the measuring instrument-specific number)
serial2	Serial number (the module-specific number)
firm1	FPGA version of the measuring instrument
firm2	Package version of the module

When parameter is omitted

ANRITSU,MU887000A,serial,firm

ANRITSU	Company name (Anritsu Corporation)
MU887000A	Model name of the module
serial	Serial number (the module-specific number)
firm	Package version of the module

Parameter

index	Index of information to acquire
Range	1 to 2

Example of Use

To query the device identification information:

*IDN?

*OPC

Operation Complete

Function

Sets bit0 in standard event status register or operation completion when all selected pending device operations completed

The *OPC? query sets the output queue to 1 and waits until MAV occurs when all the selected pending operations are completed.

Command

```
*OPC
```

Query

```
*OPC?
```

Response

Operation Operation completion flag

Example of Use

To query operation completion:

*OPC?

> 0

*RST

Preset

Function

Executes initialization

Command

*RST

Details

Initialization may be disabled depending on the connection status with terminals.

Settings related to communications (IPv4, IPv6, and GPIB), date, and time are not initialized by *RST.
When the operation is in Sequence mode, it quits the sequence processing, and changes into Normal mode.
When the operation is in synchronization stand-by state of waveform patterns among modules, the synchronization stand-by state is cancelled.

Example of Use

To execute initialization:
*RST

Related Commands

REBOOT	Same function as *RST
SHUTDOWN	Same function as *RST

*SRE

Service Request Enable Command

Function

Sets service request enable register
When an event set with the service request enable register occurs, the status byte register MSS (bit6) is set to 1 (true).

Command

```
*SRE register
```

Query

```
*SRE?
```

Response

Register Service request enable register
Value = bit0 + bit1 + ... + bit7

	SCPI command mode	Native command mode
bit0 = 2^0 = 1	Unused	END event status register (Signal generator)
bit1 = 2^1 = 2	Unused	ERR event status register (Signal generator)
bit2 = 2^2 = 4	Error event queue	END event status register (Measurement)
bit3 = 2^3 = 8	Questionable status register	ERR event status register (Measurement)
bit4 = 2^4 = 16	MAV	MAV
bit5 = 2^5 = 32	ESB	ESB
bit6 = 2^6 = 64	Unused	Unused
bit7 = 2^7 = 128	Operation status register	Unused

Parameter

Register	Service request enable register
Range	0 to 255
Resolution	1
Default	0

Details

Set the sum of the values for bits to be enabled to the parameter, from the values 2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32, 2^6 = 64, and 2^7 = 128, corresponding to the service request enable register bits 0, 1, 2, 3, 4, 5, 6, and 7.

Example of Use

To set MSS bit to 1 (true) at END event:
*SRE 4

*STB?

Status Byte Register

Function

Queries status byte register value

The status byte register bit6 is retrieved as MSS (Master Summary Status).

Query

*STB?

Response

Register Status byte register

Value = bit0 + bit1 + ... + bit7

	SCPI command mode	Native command mode
bit0 = 2^0 = 1	Unused	END event status register (Signal generator)
bit1 = 2^1 = 2	Unused	ERR event status register (Signal generator)
bit2 = 2^2 = 4	Error event queue	END event status register (Measurement)
bit3 = 2^3 = 8	Questionable status register	ERR event status register (Measurement)
bit4 = 2^4 = 16	MAV	MAV
bit5 = 2^5 = 32	ESB	ESB
bit6 = 2^6 = 64	MSS	MSS
bit7 = 2^7 = 128	Operation status register	Unused

Parameter

register	Status byte register
Range	0 to 255

Details

Set the sum of the values for bits to be enabled to the parameter, from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, corresponding to the service request enable register bits 0, 1, 2, 3, 4, 5, 6, and 7.

Example of Use

To query the status byte register value:

*STB?

> 0

*TRG

Trigger

Function

Applies trigger

Command

```
*TRG
```

*TST?

Self Test

Function

Queries self-diagnostics result

Query

```
*TST?
```

Response

test_result	Self-diagnostics result
0	Normal end
1	Failure end

Example of Use

To query the self-diagnostics result:

*TST?

> 0

Related Command

SYSTEST, :SYSTem:SELF:TEST

*WAI

Wait to Continue

Function

Makes next command wait while command executing

Command

```
*WAI
```

Details

The next command processing is not started until all active processes are completed.

Example of Use

To wait for completion of measurement with *WAI:

SNGLS;*WAI

CDMA2K_TXPWR?

6.2.2 Commands in alphabetical order

ADJREFCLK

Adjust Reference Clock

Function

Sets or queries reference frequency correction value

Command

```
ADJREFCLK val
```

Query

```
ADJREFCLK?
```

Response

val

Parameter

val	Adjustment value
Range	0 to 1023
Resolution	1
Default	512

Details

This is a common setting for all modules installed in the MT8870A.

Example of Use

ADJREFCLK 512
ADJREFCLK?
> 512

Remarks

Not initialized by *RST command

ADJREFCLKDEF

Adjust Reference Clock Default

Function

Acquires factory default reference frequency correction value

Query

`ADJREFCLKDEF?`

Response

val	Factory default adjustment value

Parameter

val	Factory default adjustment value
Range	0 to 1023
Resolution	1

Details

This is a common value for all modules mounted in the MT8870A.

Example of Use

ADJREFCLKDEF?

> 512

BANDCAL

Execute BAND Calibration

Function

Executes band calibration

Command

```
BANDCAL
```

Example of Use

To execute the band calibration:

BANDCAL

BANDCALERRCNT?

Band Calibration Error Count

Function

Queries Band Calibration failure count

Query

```
BANDCALERRCNT?
```

Response

count

Parameter

count	Band Calibration failure count
Range	0 or more

Example of Use

To query the Band Calibration failure count:

BANDCALERRCNT?

> 0

BANDCALERR_TM?

BAND Calibration Error Last Time

Function

Queries date and time of last Band Calibration failure

Query

`BANDCALERR_TM?`

Response

year,month,day,hour,min

Parameters

year	Year (from 2001)
month	Month (1 to 12)
day	Day (1 to 31)
hour	Hour (0 to 23)
min	Minute (0 to 59)

Example of Use

To query the time of the last Band Calibration failure:

BANDCALERR_TM?

> 2011,09,13,21,45

Remarks

The response is *** if Band Calibration has never failed.

BANDCALRESULT?

Result of Band Calibration

Function

Queries Band Calibration result

Query

```
BANDCALRESULT?
```

Response

Result

Parameter

Result	Band Calibration result
PASS	Calibration succeeded
FAIL	Calibration failed
UNEXECUTED	Calibration unexecuted

Example of Use

To query the Band Calibration result:

BANDCALRESULT?

> PASS

BANDCAL_TEMP

BAND Calibration Temperature

Function

Band Calibration is performed if the internal temperature changes by more than the setting when Band Calibration was executed previously. It is not performed if the internal temperature change is within the setting.

Command

```
BANDCAL_TEMP temperature
```

Parameter

temperature	Temperature
Range	1.0 to 50.0°C
Resolution	0.1°C

Example of Use

To perform Band Calibration when the internal temperature changes by 2°C or more compared to the previous Band Calibration.:
BANDCAL_TEMP 2.0

BANDCAL_TM?

BAND Calibration Last Time

Function

Queries date and time of last Band Calibration

Query

```
BANDCAL_TM?
```

Response

year,month,day,hour,min

Parameters

year	Year (from 2001)
month	Month (1 to 12)
day	Day (1 to 31)
hour	Hour (0 to 23)
min	Minute (0 to 59)

Example of Use

To query the time of the last Band Calibration:

BANDCAL_TM?
> 2011,09,13,11,45

Remarks

The response is *** when Band Calibration has never been executed.

BEEP

Beep

Function

Sets or queries buzzer On/Off

Command

`BEEP sw`

Query

`BEEP?`

Response

sw

Parameter

sw	Buzzer On/Off
ON	Buzzer enabled
OFF	Buzzer disabled

Details

It can be set for each module.
When the buzzer is enabled, audible signals are emitted in the following cases.

When an error is detected during self-diagnostic at boot-time:	One long audible signal
When the software starts at boot-time:	Three short audible signals
When the power is shut down or `:SYSTem:REBoot` or `SYSTem:SHUTdown` command is sent:	Two short audible signals
When an error is present in the sent command:	One short audible signal

Example of Use

BEEP ON
BEEP?
>ON

Remarks

It is not initialized by the *RST command.

CALTEMP?

Temperature at Full Calibration Execution

Function

Queries module temperature at the last Full Calibration after power-on.

Query

`CALTEMP?`

Response

temp

Parameter

temp	Temperature (°C)

Example of Use

To query the module temperature at the last Full Calibration after power-on:

CALTEMP?

> 32.31

Related Command

The response is *** when Full Calibration has never been executed after power-on.

CALTIME?

Elapsed Time after Full Calibration

Function

Queries elapsed time after last Full Calibration

Query

```
CALTIME?
```

Response

time

Parameter

time	Elapsed time in seconds

Example of Use

To query the elapsed time after the last Full Calibration:

CALTIME?

> 1280

Remarks

The response is *** when Full Calibration has never been executed.

DATETIME

Set Date

Function

Sets or queries date

Command

`DATETIME year,month,day,hour,minute,second`

Query

`DATETIME?`

Response

year,month,day,hour,minute,second

Parameters

year	Year
Range	2012 to 2027
Resolution	1
month	Month
Range	1 to 12
Resolution	1
day	Day
Range	1 to 31
Resolution	1
hour	Hour
Range	0 to 23
Resolution	1
minute	Minutes
Range	0 to 59
Resolution	1
second	Second
Range	0 to 59
Resolution	1

Example of Use

DATETIME 2012,9,13,11,6,17
DATETIME?
> 2012,9,13,11,6,17

Remarks

The response is *** when the date and time setting fails.

DELLOSSTBL

Delete Loss Value from Cable Loss List

Function

Deletes correction value set in the loss correction table.

The deletion range corresponds to the registered No. in the loss correction table, and the range to be deleted is specified. If the deletion range is unspecified, all values in the loss correction table are deleted.

Command

```
DELLOSSTBL [a[,b]]
```

Parameters

a		Specifies delete range
	Range	1 to 1000
	Resolution	1
b		Specifies delete range
	Range	a to 1000
	Resolution	1

Example of Use

DELLOSSTBL 1,1000

DELM

Delimiter

Function

Sets or queries delimiter (characters added to message end)

Command

```
DELM code
```

Query

```
DELM?
```

Response

code

Parameter

code	End code
LF	LF (Line Feed)
CRLF	CR/LF (Carriage Return + Line Feed)
NONE	None (only EOI)
Default	CRLF

Details

It can be set for each module.
The delimiter changes to the initial value when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

DELM CRLF
DELM?
>CRLF

Remarks

The processing is the same as TRM.
The *RST, REBOOT, and NETRESTART commands do not perform initialization.
When the parameter is set to NONE in Ethernet connection, the delimiter of the response message from the main unit is set to CRLF.

DOMAINNAME

Domain Name

Function

Sets or queries domain name

Command

```
DOMAINNAME name
```

Query

```
DOMAINNAME?
```

Response

name	Domain name

Parameter

name — Domain name

Maximum	63 bytes
Default	"DOMAIN"

A parameter error occurs is the following name convention is not followed at setup.

- The first character must be an alphabetic character.
- Intermediate characters must be alphanumerics or a hyphen.
- The last character must be alphanumeric.
- Characters must be enclosed in double quotation marks ("").

Details

The domain name is set for all modules mounted in the MT8870A.
To enable this command setting, send the NETRESTART command.
The Domain name changes to the initial value when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

DOMAINNAME "Domain"
NETRESTART
DOMAINNAME?
> Domain

DSIZE?

Drive Size Reading

Function

Queries free space of the memory.

Query

```
DSIZE? [target]
```

Response

total,blank

Parameters

target	Area to check the free space.
LOG	Area for log records
USER	Area for waveform files and installer files
Default	USER
total	Total space of the specified area (byte)
blank	Free space of the specified area (byte)

Example of Use

DSIZE? LOG

> 2097152,734002

ESE0

End Event Status (Generator) Enable Command

Function

Sets end event status (signal generator) enable register

Command

```
ESE0 register
```

Query

```
ESE0?
```

Response

register Value = bit0 + bit1 + ... + bit7

See response of ESR0? for bit allocation.

Parameter

register	End event status (signal generator) enable register
Range	0 to 255
Resolution	1
Default	0

Details

The sum of the values for bits to be enabled from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, corresponding to the end event status 0 enable register bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the parameter.

Example of Use

To enable bit 0 of the end event status enable register:

ESE0 1

ESE0?

> 1

ESE1

Error Event Status (Generator) Enable Command

Function

Sets error event status (signal generator) enable register

Command

```
ESE1 register
```

Query

```
ESE1?
```

Response

register Value = bit0 + bit1 + ... + bit7

See response of ESR1? for bit allocation.

Parameter

register	Error event status (signal generator) enable register
Range	0 to 255
Resolution	1
Default	0

Details

The sum of the values for bits to be enabled from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, corresponding to the error event status enable register bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the parameter.

Example of Use

To enable bit 2 of the error event status enable register:

ESE1 4

ESE1?

> 4

ESE2

End Event Status (Measure) Enable Command

Function

Sets end event status (measurement) enable register

Command

```
ESE2 register
```

Query

```
ESE2?
```

Response

register		Value = bit0 + bit1 + ... + bit7

Bit allocation depends on the application.

Refer to ESR2? in the application software operation manual.

Parameter

register	End event status (measurement) enable register
Range	0 to 255
Resolution	1
Default	0

Details

The sum of the values for bits to be enabled from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, corresponding to the end event status enable register bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the parameter.

Example of Use

To enable bit 4 of the end event status enable register 2:

ESE2 16

ESE2?

> 16

ESE3

Error Event Status (Measure) Enable Command

Function

Sets error event status (measurement) enable register

Command

```
ESE3 register
```

Query

```
ESE3?
```

Response

register	Value = bit0 + bit1 + ... + bit7

Bit allocation depends on the application.

Refer to ESR3? in the application software operation manual.

Parameter

register	Error event status enable register
Range	0 to 255
Resolution	1
Default	0

Details

The sum of the values for bits to be enabled from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, corresponding to the error event status enable register bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the parameter.

Example of Use

To enable bit 6 of the error event status enable register 0:

ESE3 64

ESE3?

> 64

ESR0?

End Event Status (Generator) Register Query

Function

Queries end event status register (signal generator)
The event occurrence can be identified using the retrieved value.

Query

`ESR0?`

Response

register	Value = bit0 + bit1 + ... + bit7
bit0 = 2^0 = 1	File loading completed
bit1 = 2^1 = 2	Defragmentation completed.
bit2 = 2^2 = 4	Preparing synchronized output of waveform patterns.
bit3 = 2^3 = 8	Unused (reserved for application use)
bit4 = 2^4 = 16	Unused (reserved for application use)
bit5 = 2^5 = 32	Unused (reserved for application use)
bit6 = 2^6 = 64	Unused (reserved for application use)
bit7 = 2^7 = 128	Unused (reserved for application use)

Parameter

register	End event status register (Signal generator)
Range	0 to 255

Details

The sum of the values for bits of the occurring event from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, that correspond to the end event status register (signal generator) bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the response.

Example of Use

To query the end event status register (signal generator) value:
ESR0?
> 1

6

Native Command Reference

ESR1?

Error Event Status (Generator) Register Query

Function

Queries error event status register (signal generator)
The event occurrence can be identified using the retrieved value.

Query

ESR1?

Response

register	Value = bit0 + bit1 + ... + bit7
bit0 = 2^0 = 1	The loaded file has an error.
bit1 = 2^1 = 2	Unused (reserved for application use)
bit2 = 2^2 = 4	Unused (reserved for application use)
bit3 = 2^3 = 8	Unused (reserved for application use)
bit4 = 2^4 = 16	Unused (reserved for application use)
bit5 = 2^5 = 32	Unused (reserved for application use)
bit6 = 2^6 = 64	Unused (reserved for application use)
bit7 = 2^7 = 128	Unused (reserved for application use)

Parameter

register	Error event status register (signal generator)
Range	0 to 255

Details

The sum of the values for bits of the occurring event from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, that correspond to the error event status register (signal generator) bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the response.

Example of Use

To query the error event status register (signal generator) value.
ESR1?
> 0

ESR2?

End Event Status (Measurement) Register Query

Function

Queries end event status register (measurement)
The event occurrence can be identified using the retrieved value.

Query

ESR2?

Response

register	Value = bit0 + bit1 + ... + bit7
bit0 = 2^0 = 1	End of measurement
bit1 = 2^1 = 2	Unused (reserved for application use)
bit2 = 2^2 = 4	Unused (reserved for application use)
bit3 = 2^3 = 8	Unused (reserved for application use)
bit4 = 2^4 = 16	Unused (reserved for application use)
bit5 = 2^5 = 32	Unused (reserved for application use)
bit6 = 2^6 = 64	Unused (reserved for application use)
bit7 = 2^7 = 128	Unused (reserved for application use)

The bit allocation depends on the application.
Refer to ESR2? in the application software operation manual.

Parameter

register	End event status register (measurement)
Range	0 to 255

Details

The sum of the values for bits of the occurring event from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, that correspond to the end event status register (measurement) bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the response.

Example of Use

To query the end event status register (measurement) value:
ESR2?
> 0

ESR3?

Error Event Status (Measurement) Register Query

Function

Queries error event status register (measurement)

The event occurrence can be identified using the retrieved value.

Query

ESR3?

Response

register	Value = bit0 + bit1 + ... + bit7
bit0 = 2^0 = 1	Level over
bit1 = 2^1 = 2	Level under
bit2 = 2^2 = 4	Timeout
bit3 = 2^3 = 8	Unused (reserved for application use)
bit4 = 2^4 = 16	Unused (reserved for application use)
bit5 = 2^5 = 32	Unused (reserved for application use)
bit6 = 2^6 = 64	Unused (reserved for application use)
bit7 = 2^7 = 128	Unused (reserved for application use)

The bit allocation depends on the application.

Refer to ESR3? in the application software operation manual.

Parameter

register	Error event status register (measurement)
Range	0 to 255

Details

The sum of the values for bits of the occurring event from the values $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, and $2^7 = 128$, that correspond to the error event status register (measurement) bits 0, 1, 2, 3, 4, 5, 6, and 7 becomes the response.

Example of Use

To query the error event status register (measurement) value:

ESR3?

> 4

EXTLOSSW

External Loss On/Off

Function

Sets or queries whether level corrected using all ports DL/UL/ loss data in the loss correction table

Command

`EXTLOSSW sw`

Query

`EXTLOSSW?`

Response

sw

Parameter

sw	All ports DL/UL loss value enable/disable
COMMON	The correction value is used.
ON	The correction value is used.
OFF	The correction value is not used.
Default	OFF

Example of Use

EXTLOSSW ON
EXTLOSSW?
> ON

Remarks

To obtain the correction value for a frequency not set in the loss correction table, the value is calculated by linear interpolation using the correction values for preceding and succeeding frequencies.
The correction value for the minimum frequency is used for a lower frequency than the minimum frequency. The correction value for the maximum frequency is used for a higher frequency than the maximum frequency.

EXTREF?

Reference Signal Query

Function

Queries status of frequency reference signal source

Query

EXTREF?

Response

source

Parameter

source	Reference signal status
INT	Locked to internal reference signal source
VALI	Locked to external reference signal source
INVALID	Not locked to external reference signal source

Details

Same for all slots

Example of Use

EXTREF?
> INT

Related Command

FREQREFSOURCE

FDEL

Delete File

Function

Deletes specified file

Command

```
FDEL filepath
```

Parameter

filepath	File name (path information included) Up to 1024 characters The following characters cannot be used: \ : * ? " < > |

Details

Specify the file name in a format including the path.

Example of Use

FLIST? "CELLULAR/parameter"
> 3,setup.000,setup.001,setup.002
FDEL "CELLULAR/parameter/setup.002"
FLIST? "CELLULAR/parameter"
> 2,setup.000,setup.001

FIRMCHECK?

Check Firmware Update Package

Function

Checks firmware package prepares update if normal

Query

`FIRMCHECK? filename`

Response

result

Parameter

filename	Firmware package file name
result	Result of firmware package file check
0	Normal
1	Specified file not found
2	Incorrect file
3	File damaged or not installer for machine
4	File version normal but not supported by this machine

Details

The firmware package file must be sent by FTP to a specified destination folder (/install/package/) in advance.

Updating preparation is disabled if the machine reboots or the power is turned off before updating the firmware using the FIRMUPDATE command.

Example of Use

FIRMCHECK? "package.inst"

>0

FIRMUPDATE

Firmware Update

Function

Updates firmware package or queries update result

Command

```
FIRMUPDATE target[,option]
```

Query

```
FIRMUPDATE?
```

Response

result

Parameters

target	Update target module
SINGLE	Single module
ALL	All modules in MT8870A
option	Option
DIFF	Updates only when version different
FORCE	Always updates
result	Firmware installation result
0	Normal
1	Failed
2	Aborted

Details

Updating must be prepared for using the FIRMCHECK? command, which reboots the system. Modules cannot be controlled during rebooting. Choose whether to update all modules in the MT8870A, or a single module. The firmware update result can be checked using FIRMUPDATE?

Example of Use

FIRMUPDATE ALL,DIFF

FLIST?

File List

Function

Queries list of files saved in specified path

Query

```
FLIST? path
```

Response

num,filelist

Parameters

path	Path
num	Number of files
filelist	File list

Details

When the response list is 10485760 characters or more, "…" is appended to the list.
A file extension such as "*.h" can be specified for the path.

Example of Use

FLIST? "CELLULAR/parameter"
> 3,setup.000,setup.001,setup.002
FLIST? "CELLULAR/parameter/*001"
> 1,setup.001

FREQREF

Frequency Reference

Function

Sets or queries frequency reference signal source

Command

```
FREQREF source
```

Query

```
FREQREF?
```

Response

source

Parameters

source	Internal reference signal or external reference signal
INT	MT8870A internal reference signal
AUTO	Reference signal input to Ref Input connector at rear panel (Uses internal reference signal if external signal not detected.)
Default	AUTO

Details

The reference signal source is set for all modules installed in the MT8870A. It cannot be set for each module.

Example of Use

FREQREF INT
FREQREF?
> INT

Related Command

REF

Remarks

Not initialized by *RST command

FREQREFSOURCE?

Reference Signal Query

Function

Queries status of frequency reference signal source

Query

`FREQREFSOURCE?`

Response

source

Parameters

source	Reference signal status
INT	Locked to internal reference signal source
EXT	Locked to external reference signal source
EXTU	Not locked to external reference signal source

Details

Same for all slots

Example of Use

FREQREFSOURCE?
> EXT

Related Command

EXTREF

FTPUSER_CHANGE

Change FTP User Name

Function

Changes FTP user name

Command

```
FTPUSER_CHANGE before,after
```

Parameters

before	User name before change
after	User name after change

Details

The user name character string must be 1 to 31 alphanumeric characters.
Set to enclose in double quotation marks ("").
The head character may be an upper or lower-case alphabetic character, a period (.) or an underscore (_). Second and subsequent characters may be upper or lower-case alphanumerics, periods (.) or underscores (_). A name composed entirely numeric values is prohibited.

Example of Use

To change user1 to user2:
FTPUSER_CHANGE "user1","user2"

FTPUSER_LIST?
FTP User List

Function
Queries all FTP user names.

Query
```
FTPUSER_LIST?
```

Response
name,name,..... User name

Example of Use
FTPUSER_LIST?

> user1,user2,user3

FTPUSER_PASSWD
Change FTP User Password

Function
Changes FTP user password

Command
```
FTPUSER_PASSWD name,before,after
```

Parameter
name	User name for password change
before	Password before change
after	Password after change

Details
The user name password must be 1 to 16 upper or lower-case alphanumeric characters and symbols (! #$%'()-=^~@[{;+]},.?_).

Set to enclose in double quotation marks ("").

Example of Use
FTPUSER_PASSWD "user1","abcde","fghij"

FULLCAL

Execute FULL Calibration

Function

Executes for full bandwidth calibration

Command

```
FULLCAL
```

Example of Use

To execute the calibration for full bandwidth:

FULLCAL

FULLCALERRCNT?

Full Calibration Error Count

Function

Queries Full Calibration failure count

Query

```
FULLCALERRCNT?
```

Response

count

Parameter

count	Full Calibration failure count
Range	0 or more

Example of Use

To query the Full Calibration failure count:

FULLCALERRCNT?

> 0

FULLCALERR_TM?

FULL Calibration Error Last Time

Function

Queries date and time of last Full Calibration failure

Query

```
FULLCALERR_TM?
```

Response

year,month,day,hour,min

Parameters

year	Year (from 2001)
month	Month (1 to 12)
day	Day (1 to 31)
hour	Hour (0 to 23)
min	Minute (0 to 59)

Example of Use

To query time of last Full Calibration failure:

FULLCALERR_TM?

> 2012,03,10,11,45

Remarks

The response is *** if Full Calibration has never failed.

FULLCALRESULT?

Result of Full Calibration

Function

Queries Full Calibration result

Query

`FULLCALRESULT?`

Response

Result

Parameter

Result	Full Calibration result
PASS	Calibration succeeded
FAIL	Calibration failed
UNEXECUTED	Calibration not executed

Example of Use

To query Full Calibration result:
FULLCALRESULT?
> PASS

FULLCAL_TM?

FULL Calibration Last Time

Function

Queries date and time of last Full Calibration

Query

```
FULLCAL_TM?
```

Response

year,month,day,hour,min

Parameters

year	Year (from 2001)
month	Month (1 to 12)
day	Day (1 to 31)
hour	Hour (0 to 23)
min	Minute (0 to 59)

Example of Use

To query the time of the last Full Calibration:

FULLCAL_TM?

> 2012,03,13,11,45

Remarks

The response is *** if Full Calibration has never been executed.

GPIBADDR

GPIB Address

Function

Sets or queries GPIB address

Command

```
GPIBADDR gp_address
```

Query

```
GPIBADDR?
```

Response

gp_address	GPIB address

Parameter

gp_address	GPIB address	
Range	1 to 30	
Default	Slot 1	1
	Slot 2	1
	Slot 3	1
	Slot 4	1

Details

A GPIB address can be set for each module in each slot of the MT8870A.
When a module is moved to another slot, the slot GPIB address applies.
To enable this command setting, send the NETRESTART command.
The GPIB address changes to the initial value when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

```
GPIBADDR 8
NETRESTART

GPIBADDR?
> 8
```

GPIBADDRALL

GPIB Address

Function

Sets or queries the GPIB addresses of all modules.

Command

```
GPIBADDRALL address1,address2,address3,address4
```

Query

```
GPIBADDRALL?
```

Response

address1,address2,address3,address4

Parameter

address1	GPIB address of Module 1
Range	1 to 30
Default	1
address2	GPIB address of Module 2
Range	1 to 30
Default	1
address3	GPIB address of Module 3
Range	1 to 30
Default	1
address4	GPIB address of Module 4
Range	1 to 30
Default	1

Details

This command can set the GPIB addresses of all the modules.
GPIB address is set for the slot of MT8870A.
When a module is moved to another slot, the slot GPIB address applies.
To enable this command setting, send the NETRESTART command.
The GPIB address is reset to the default when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

GPIBADDRALL 1,2,3,4
NETRESTART

GPIBADDRALL?
> 1,2,3,4

GTL

Go to Local

Function

Enables panel operation

Command

```
GTL
```

Details

The Remote lamp on the panel goes off.

Example of Use

GTL

HOSTNAME

Host Name

Function

Sets or queries host name

Command

```
HOSTNAME host
```

Query

```
HOSTNAME?
```

Response

host	Host name

Parameter

host

Host name

Minimum	1 byte
Maximum	63 byte
Default	"HOST"

A parameter error occurs if the naming convention is not followed at setup.

- The first character must be an alphabetic character.
- Intermediate characters must alphanumeric characters or a hyphen.
- The last character must be an alphanumeric character.
- Characters must be enclosed in double quotation marks ("").

Details

The slot number is added to the end of the set host name (HOSTNAME_- Slot number).
The host name is set for all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
The host name changes to the initial value when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

HOSTNAME "ANRITSU-PC"
NETRESTART
HOSTNAME?
> ANRITSU-PC-1

Remarks

The host name set by this command is displayed at the Windows PC network control panel.

IPV4_CURRENT_ADDR?

Get Current IPv4 Address

Function

Queries module IPv4 address

Query

`IPV4_CURRENT_ADDR?`

Response

num	Number of set IP addresses
ipaddr	IP address

Parameter

num	Number of set IP addresses
ipaddr	IP address

Details

The module IPv4 address is acquired.
The response is 0 when the module IPv4 address is set to Off.

Example of Use

IPV4_CURRENT_ADDR?
> 2,192.168.0.1,172.16.0.31

IPV4_CURRENT_DEFAULTGATEWAY?

Get Current Default Gateway

Function

Queries current default gateway address

Query

```
IPV4_CURRENT_DEFAULTGATEWAY?
```

Response

gateway

Parameter

gateway	Default gateway address

Details

The default gateway address for the module is acquired.
The response is 0 when the module IPv4 address is set to Off.

Example of Use

IPV4_CURRENT_DEFAULTGATEWAY?
> 192.168.0.1

IPV4_CURRENT_SUBNETMASK?

Get Current Subnet Mask

Function

Queries current subnet mask

Query

```
IPV4_CURRENT_SUBNETMASK?
```

Response

mask

Parameter

mask	Subnet mask

Details

The subnet mask of the module is acquired.
The response is 0 when the module IPv4 address is set to Off.

Example of Use

IPV4_CURRENT_SUBNETMASK?
> 255.255.255.0

IPV4_DNS_AUTO_SW

DNS Server Auto

Function

Sets and queries whether to acquire DNS server address automatically

Command

```
IPV4_DNS_AUTO_SW switch
```

Query

```
IPV4_DNS_AUTO_SW?
```

Response

switch	Sets DNS server auto-address acquisition function On/Off

Parameter

switch	Sets DNS server auto-address acquisition function On/Off
ON	On set
OFF	Off set
Default	1

Details

This setting is applied to all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
The DNS server auto-address acquisition function setting changes to OFF when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

IPV4_DNS_AUTO_SW ON
NETRESTART

IPV4_DNS_AUTO_SW?
>ON

IPV4_DNS_PRIMARY

DNS Primary Address

Function

Sets or queries DNS primary address (IPv4)

Command

```
IPV4_DNS_PRIMARY ip4addr
```

Query

```
IPV4_DNS_PRIMARY?
```

Response

ip4addr	DNS primary address

Parameter

ip4addr	DNS primary address Set in format xxx.xxx.xxx.xxx

Details

The DNS primary address is set for all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
Pressing the IP Reset button on the MT8870A panel at power-on switches the DNS primary address to 0.

Example of Use

IPV4_DNS_PRIMARY "192.168.10.1"
NETRESTART
IPV4_DNS_PRIMARY?
>192.168.10.1

IPV4_DNS_SECONDARY

DNS Secondary Address

Function

Sets or queries DNS secondary address (IPv4).

Command

```
IPV4_DNS_SECONDARY ip4addr
```

Query

```
IPV4_DNS_SECONDARY?
```

Response

ip4addr

Parameter

ip4addr	DNS secondary address Set in format xxx.xxx.xxx.xxx

Details

The DNS secondary address is set for all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
Pressing the IP Reset button on the MT8870A panel at power-on switches the DNS secondary address to 0.

Example of Use

IPV4_DNS_SECONDARY "192.168.10.2"
NETRESTART
IPV4_DNS_SECONDARY?
>192.168.10.2

IPV4_STATIC_ADDR_ALLMODULE

IPv4 Address

Function

Sets or queries module fixed IPv4 address

Command

```
IPV4_STATIC_ADDR_ALLMODULE ipv4_addr1,ipv4_addr2,ipv4_addr3,ipv4_addr4
```

Query

```
IPV4_STATIC_ADDR_ALLMODULE?
```

Response

ipv4_addr1,ipv4_addr2,ipv4_addr3,ipv4_addr4

Parameters

ipv4_addr1	IPv4 address of Slot 1 xxx.xxx.xxx.xxx
ipv4_addr2	IPv4 address of Slot 2 xxx.xxx.xxx.xxx
ipv4_addr3	IPv4 address of Slot 3 xxx.xxx.xxx.xxx
ipv4_addr4	IPv4 address of Slot 4 xxx.xxx.xxx.xxx

Details

The IPv4 address of each slot of the MT8870A is set. It can be set without an inserted module.
To enable this command setting, send the NETRESTART command.
The IPv4 address is set for the MT8870A slot.
When a module is moved to another slot, the IPv4 address of the slot applies.
Pressing the IP Reset button on the MT8870A panel at power-on does not clear the command setting.

Example of Use

```
IPV4_STATIC_ADDR_ALLMODULE
"192.168.1.101","192.168.1.102","192.168.1.103","192.168.1.104"
NETRESTART

IPV4_STATIC_ADDR_ALLMODULE?
> 192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.104
```

IPV4_STATIC_DEFAULTGATEWAY

Default Gateway

Function

Sets or queries fixed IPv4 setting default gateway

Command

```
IPV4_STATIC_DEFAULTGATEWAY gateway
```

Query

```
IPV4_STATIC_DEFAULTGATEWAY?
```

Response

gateway

Parameter

gateway	Default gateway address xxx.xxx.xxx.xxx

Details

The default gateway is set for all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
If a blank space is specified in the parameters, the default gateway setting is initialized.
Pressing the IP Reset button on the MT8870A panel at power-on switches the default gateway to 0.

Example of Use

IPV4_STATIC_DEFAULTGATEWAY "192.168.0.1"
NETRESTART

IPV4_STATIC_DEFAULTGATEWAY?
> 192.168.0.1

IPV4_STATIC_SUBNETMASK

Subnet Mask

Function

Sets or queries fixed IPv4 address subnet mask

Command

```
IPV4_STATIC_SUBNETMASK mask
```

Query

```
IPV4_STATIC_SUBNETMASK?
```

Response

mask	Subnet mask

Parameter

mask	Subnet mask Set in format xxx.xxx.xxx.xxx

Details

The subnet mask is set for all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
Pressing the IP Reset button on the MT8870A panel at power-on switches the subnet mask to 255.255.255.0.

Example of Use

IPV4_STATIC_SUBNETMASK "255.255.255.0"
NETRESTART

IPV4_STATIC_SUBNETMASK?
> 255.255.255.0

IPV4_TYPE

IPv4 Type

Function

Sets or queries IPv4 address type

Command

```
IPV4_TYPE ipv4type
```

Query

```
IPV4_TYPE?
```

Response

ipv4type

Parameter

ipv4type	IPv4 address type
STATIC	Fixed IP
DHCP	Acquires IPv4 from DHCP server and set
Default	STATIC

Details

The IPv4 address type is set for all slots in the MT8870A.
When STATIC is set, set the address, subnet mask, and default gateway using the following commands:
IPV4_STATIC_ADDR_ALLMODULE
IPV4_STATIC_SUBNETMASK
IPV4_STATIC_DEFAULTGATEWAY

To enable this command setting, send the NETRESTART command.
The IPv4 address type is set for the MT8870A slots.
If a module is moved to another slot, the IPv4 address type applies to the slot.
Pressing the IP Reset button on the MT8870A panel at power-on sets the IPV4 address type to the initial value.

Example of Use

IPV4_TYPE DHCP
NETRESTART

After restart,
IPV4_TYPE?
> DHCP

IPV6_CURRENT_ADDR?

Get Current IP Address and Prefix

Function

Queries current IPv6 address/prefix length

Query

`IPV6_CURRENT_ADDR?`

Response

num	Number of set IP addresses
ipaddr	IP address/prefix length

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx

Parameters

num	Number of set IP addresses
ipaddr	IP address/prefix length xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx

Details

The IPv6 address/prefix length is queried for the module being communicated with.
The response is 0 when the module IPv6 address is set to Off.

Example of Use

IPV6_CURRENT_ADDR?
> 2,2001:DB8::1/64,2001:DB8::2/64

IPV6_CURRENT_DEFAULTROUTER?

Get Default Router Address

Function

Queries current default router address

Query

```
IPV6_CURRENT_DEFAULTROUTER?
```

Response

ip6addr

Parameter

ip6addr	Default router address xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

Details

The IPv6 address/prefix length is queried for the module being communicated with.
The response is 0 when the module IPv6 address is set to Off.

Example of Use

IPV6_CURRENT_DEFAULTROUTER?
> 2001:DB8::1

IPV6_CURRENT_LINK_LOCAL_ADDR?

Get Link Local Address

Function

Queries current link local address/prefix length

Query

```
IPV6_CURRENT_LINK_LOCAL_ADDR?
```

Response

ip6addr

Parameter

ip6addr	Link local address/prefix length xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx

Details

The link local address and prefix length are set for the module being communicated with.
The response is 0 when the module IPv6 address is set to Off.

Example of Use

IPV6_CURRENT_LINK_LOCAL_ADDR?
> 2001:DB8::1/64

IPV6_DNS_AUTO_SW

DNS Server Auto

Function

Sets and queries DNS server auto-address acquisition

Command

```
IPV6_DNS_AUTO_SW switch
```

Query

```
IPV6_DNS_AUTO_SW?
```

Response

switch Sets DNS server auto-address acquisition On/Off

Parameter

switch	Sets DNS server auto-address acquisition On/Off
ON	On set
OFF	Off set
Default	OFF

Details

This setting is applied to all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
The DNS server auto-address acquisition function setting changes to OFF when the IP Reset button on the MT8870A panel is pressed at power-on.

Example of Use

IPV6_DNS_AUTO_SW ON
NETRESTART

IPV6_DNS_AUTO_SW?
>ON

IPV6_DNS_PRIMARY

DNS Primary Address

Function

Sets or queries DNS primary address (IPv6)

Command

```
IPV6_DNS_PRIMARY ip6addr
```

Query

```
IPV6_DNS_PRIMARY?
```

Response

ip6addr	DNS primary address

Parameter

ip6addr	DNS primary address xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

Details

The DNS primary address is set for all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
Pressing the IP Reset button on the MT8870A panel at power-on sets the DNS primary address to primary address to 0.

Example of Use

IPV6_DNS_PRIMARY "fe80::215:c5ff:febb:42cf"
NETRESTART

IPV6_DNS_PRIMARY?
> fe80::215:c5ff:febb:42cf

IPV6_DNS_SECONDARY

DNS secondary address

DNS Secondary Address

Function

Sets or queries DNS secondary address (IPv6)

Command

```
IPV6_DNS_SECONDARY ip6addr
```

Query

```
IPV6_DNS_SECONDARY?
```

Response

ip6addr	DNS secondary address

Parameter

ip6addr	DNS secondary address xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

Details

The DNS secondary address is set for all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
Pressing the IP Reset button on the MT8870A panel at power-on sets the DNS secondary address to primary address to 0.

Example of Use

IPV6_DNS_SECONDARY "fe80::215:c5ff:febb:42ce"
NETRESTART

IPV6_DNS_SECONDARY?
> fe80::215:c5ff:febb:42ce

IPV6_STATIC_ADDR_ALLMODULE

IPv6 Address

Function

Sets or queries fixed IPv6 setting IPv6 address/prefix length

Command

```
IPV6_STATIC_ADDR_ALLMODULE ipv6_addr1,ipv6_addr2,ipv6_addr3,ipv6_addr4
```

Query

```
IPV6_STATIC_ADDR_ALLMODULE?
```

Response

ipv6_addr1,ipv6_addr2,ipv6_addr3,ipv6_addr4

Parameters

ipv6_addr1	IPv6 address/prefix length to Slot 1 xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx
ipv6_addr2	IPv6 address/prefix length to Slot 2 xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx
ipv6_addr3	IPv6 address/prefix length to Slot 3 xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx
ipv6_addr4	IPv6 address/prefix length to Slot 4 xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx

Details

The IPv6 address is set for a slot of the MT8870A. It can be set for a slot without an inserted module.
To enable this command setting, send the NETRESTART command.
The IPv6 address is set for the MT8870A slot.
When a module is moved to another slot, the IPv6 address is applied to that slot.

Example of Use

IPV6_STATIC_ADDR_ALLMODULE
"2001:DB8::1/64","2001:DB8::2/64","2001:DB8::3/64","2001:DB8::4/64"
NETRESTART

IPV6_STATIC_ADDR_ALLMODULE?
> 2001:DB8::1/64,2001:DB8::2/64,2001:DB8::3/64,2001:DB8::4/64

6

Native Command Reference

IPV6_STATIC_DEFAULTROUTER

Default Router

Function

Sets or queries fixed IPv6 setting default router address

Command

```
IPV6_STATIC_DEFAULTROUTER ip6addr
```

Query

```
IPV6_STATIC_DEFAULTROUTER?
```

Response

ip6addr

Parameters

ip6addr	Default router address xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

Details

The default router address is set for all modules installed in the MT8870A.
To enable this command setting, send the NETRESTART command.
If a blank space is specified in the parameters, the default router address setting is initialized.
Pressing the IP Reset button on the MT8870A panel at power-on switches the default router address to 0.

Example of Use

IPV6_STATIC_DEFAULTROUTER "2001:DB8::1"
NETRESTART

IPV6_STATIC_DEFAULTROUTER?
> 2001:DB8::1

IPV6_TYPE

IPv6 Type

Function

Sets or queries IPv6 address type

Command

```
IPV6_TYPE ipv6type
```

Query

```
IPV6_TYPE?
```

Response

ipv6type

Parameter

ipv6type	IPv6 address type
STATIC	Fixed IP
RA_EUI64	Automatic allocation by RA based on EUI-64 (RFC4862)
RA_RANDOM	Automatic allocation by RA based on privacy expansion (RFC4941)
DHCPV6	Acquires address from DHCP server and sets
Default	STATIC

Details

The IPv6 address type is set for all slots in the MT8870A.
Set the address using the following commands when setting STATIC.
IPV6_STATIC_ADDR_ALLMODULE
IPV6_STATIC_DEFAULTROUTER

To enable this command setting, send the NETRESTART command.
The IPv6 address type is set for the MT8870A slot.
When a module is moved to another slot, the IPv6 address type is applied to that slot.
Pressing the IP Reset button on the MT8870A panel at power-on switches the IPv6 address type to the initial value.

Example of Use

```
IPV6_TYPE RA_EUI64
NETRESTART

IPV6_TYPE?
> RA_EUI64
```

IPVER

IP Version

Function

Sets or queries IPv4 and IPV6 On/Off

When On is set, the module communicates using that IP version.

Command

`IPVER ipver`

Query

`IPVER?`

Response

ipver

Parameter

ipver	IP Version
V4	Only IPv4 set to On
V6	Only IPv6 set to On
V4V6	Both IPv4 and IPv6 set to On
Default	V4

Details

The IP version is set to On/Off for all modules installed in the MT8870A.

To enable this command setting, send the NETRESTART command.

Pressing the IP Reset button on the MT8870A panel at power-on switches the IP version to the initial value.

Example of Use

IPVER V4V6

NETRESTART

After restart,

IPVER?

> V4V6

Remarks

Not initialized by *RST and REBOOT commands

KEYCHECK?

License Key Check

Function

Checks license and prepares installation if normal

Query

```
KEYCHECK? filename
```

Response

result

Parameter

filename	License file name
result	Check result for license file
–1	Cannot be installed because executed for another module in same main frame
0	Normal
1	Specified file not found
2	Incorrect file
3	Damaged file or another MT8870 file
4	License already exists (Temporary license only)
5	Outside license period
6	Maximum number of licenses reached

Details

A software license file must be sent by FTP to a specified folder (/install/license/) in advance. Installation preparation is disabled if the unit is rebooted or the power is turned off before installing using the KEYINST command.

Example of Use

KEYCHECK? "Licenses.xml"
>0
KEYINST

KEYINFO?

License Information

Function

Queries license expiration limit

Query

```
KEYINFO? model
```

Response

kind,days

Parameters

model	License name
	- Software license
	- Software option license
	- SG waveform license
	The above licenses can be specified.
kind	License type
0	Normal license
1	Temporary license
2	Trial license
days	Remaining days
	1 or more: When the number of remaining days is 1, the license is on the last day.
	0: Expired license
	–1: No license limit (normal license)

Details

Specify the product model name (options included) in the license expiration limit query of the license name.
Example: MX887011A, MX887012A

Example of Use

KEYINFO? "MX887012A"
>0,-1

KEYINST

License Key Install

Function

Installs license

Command

```
KEYINST
```

Details

Installation is prepared using KEYCHECK? Remote connection of all modules is cut by the system reboot after executing this command.

Example of Usage

KEYCHECK? "Licenses.xml"
>0
KEYINST

Remarks

The hardware option license is stored in the module. Changing the options for a specific module in the same MT8870A will stop the application starting.
The licenses for software, software options, and waveforms are stored in the MT8870A. When a module is moved to another MT8870A, the application may not start due to the different license.

LOGCLEAR

Log Clear

Function

Clears log

Command

```
LOGCLEAR
```

Details

Clears modular log file

Example of Use

LOGCLEAR

LOGREAD?

Log Reading

Function

Queries log information

Query

```
LOGREAD? [line]
```

Response

#XY<Y byte binary data><terminator>

X and Y are ASCII characters, and the value X is a digit number of the value Y.
For example, when Y is 12500, X is 5.
The binary data <Y byte binary data> is a word string in the log file.

Parameter

line	Line number of the latest acquired log
Range	1 to 999999
terminator	Terminator

Details

When the line number of acquiring log is specified by the parameter, the log is acquired by the specified number of lines. When the log acquiring size is omitted, all logs are acquired.
The linefeed codes in the log file are replaced by tabs.

Example of Use

LOGREAD?
> CELLULAR Measurement Software License expired.\n

LOGREAD? 2
> CELLULAR Measurement Software License expired.\n
> R[*IDN?]\n

LOGSIZE?

Log Size Reading

Function

Queries log size

Query

`LOGSIZE?`

Response

size

Parameter

size	Log size

Details

The response is the byte size of the log file.

Example of Use

LOGSIZE?

> 48

LOSSTBL

Loss Table

Function

Sets or queries loss correction table index

Command

```
LOSSTBL index
```

Query

```
LOSSTBL?
```

Response

index

Parameter

index	Correction table index
Range	1 to 16
Resolution	1
Default	1

Example of Use

LOSSTBL 1

LOSSTBL?

> 1

LOSSTBLCNT?

List Count in Loss Table

Function

Queries number of lists set in loss correction table

Query

`LOSSTBLCNT? index`

Response

num

Parameters

index		Index number of table to correct
	Range	1 to 16
	Resolution	1
num		Number of lists registered in level frequency correction table
	Range	0 to 1000

Example of Use

LOSSTBLCNT? 1
> 3

LOSSTBLVAL

Register Loss Value in Cable Loss List

Function

Sets/queries value in the loss correction table.
The same frequency in the loss correction table is overwritten.
If the setting of [Port3,4/in] is omitted, the value of [Port3,4/out] is set at the loss correction table.

An arbitrary frequency point loss value is input for the loss correction table, and the correction for the frequency between the setting points is determined automatically by linear interpolation.
The same correction values are set for Port1 and Port 2.
Also, the same correction values are set for Port 3 and Port 4.

Command

```
LOSSTBLVAL frequency,Port1,2/out,Port1,2/in,Port3,4/out[,Port3,4/in]
```

Query

```
LOSSTBLVAL? n
```

Response

frequency,Port1,2/out,Port1,2/in,Port3,4/out,Port3,4/in

Parameters

frequency	Frequency point to be set
Range	10 MHz to 6000 MHz
Resolution	1 Hz
Suffix Code	HZ, Z,KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
Port1,2/out	Port1/2 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port1,2/in	Port1/2 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port3,4/out	Port3/4 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port3,4/in	Port3/4 UL loss value
Range	–100.00 to 100.00 dB

Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
n	Correction table point to query
Range	1 to 1000
Resolution	1

Example of Use

To set the values of the table below in the loss correction table

No	Frequency	Port1 (dB)		Port2 (dB)		Port3 (dB)		Port4 (dB)	
		out	in	out	in	out	in	out	in
1	1000000000	10.00	11.00	10.00	11.00	12.00	13.00	12.00	13.00

LOSSTBLVAL 1GHz,10,11,12,13
LOSSTBLVAL? 1
> 1000000000,10.00,11.00,12.00,13.00

Remarks

The loss correction table is held as a correction value for each measurement software.
The loss correction table values are arranged automatically in ascending order for frequencies.

LOSSTBLVALALL

Register Loss Value into All Cable Loss List

Function

Sets or queries values in the loss correction table.
The same frequency in the loss correction table is overwritten.

An arbitrary frequency point loss value is input for the loss correction table and the correction for the frequency between the setting points is determined automatically by linear interpolation.
Different correction values can be set for Port1,Port2,Port3, and Port4.

Command

```
LOSSTBLVALALL
frequency,Port1out,Port1in,Port2out,Port2in,Port3out,Port3in,Port4out,Por
t4in
```

Query

```
LOSSTBLVALALL? n
```

Response

frequency,Port1out,Port1in,Port2out,Port2in,Port3out,Port3in,Port4out,Port4in

Parameters

frequency	Frequency point to be set
Range	10 MHz to 6000 MHz
Resolution	1 Hz
Suffix Code	HZ, Z,KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
Port1out	Port1 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port1in	Port1 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port2out	Port2 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port2in	Port2 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port3out	Port3 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port3in	Port3 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port4out	Port4 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port4in	Port4 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
n	Correction table point to query
Range	1 to 1000
Resolution	1

Example of Use

To set the values of the table below in the loss correction table.

No	Frequency	Port1 (dB)		Port2 (dB)		Port3 (dB)		Port4 (dB)	
		out	in	out	in	out	in	out	in
1	1000000000	10.00	11.00	12.00	13.00	14.00	15.00	16.00	17.00

LOSSTBLVALALL 1GHZ,10,11,12,13,14,15,16,17

LOSSTBLVALALL? 1

> 1000000000,10.00,11.00,12.00,13.00,14.00,15.00,16.00,17.00

Remarks

The loss correction table is maintained as correction values common to each measurement software.

The loss correction table values are arranged automatically in ascending order for frequencies.

MACADDR?

MAC Address

Function

Queries MAC address

Query

```
MACADDR?
```

Response

macaddr

Parameter

macaddr	MAC address

Details

The MAC address of a module to communicate with is queried.

Example of Use

MACADDR?

> 00-00-00-00-00-00

MCFV?

Firmware Version

Function

Queries the firmware version.

Query

`MCFV?`

Response

ver

Parameter

ver	Firmware version ("xx.xx.xx")

Example of Use

MCFV?
> 1.00.00

MCOPT?

Option Number

Function

Queries the software option number.

Query

`MCOPT? model`

Response

n,num,...,num

Parameter

model	Software name
n	Number of options
num	Option number

Example of Use

If the followings are installed:

- MX887011A
- No option

MCOPT? MX887011A

> 0

Remarks

The registered model names in your ordering information can be only obtained.

A parameter error occurs if an invalid software name is set.

MCSOPT?

Type Name of Software

Function

Queries the software name.

Query

```
MCSOPT?
```

Response

n,model,...,model

Parameter

n	Number of software
model	Software name

Example of Use

MCSOPT?
> 3,MX887010A,MX887011A,MX887012A

Remarks

The registered model names in your ordering information can be only obtained.

MCWOPT?

Type Name of Waveform

Function

Queries the waveform license information.

Query

```
MCWOPT?
```

Response

n,model,...,model

Parameter

n	Number of waveform licenses
model	Waveform license name

Example of Use

MCWOPT?
> 2,MV887011A,MV887012A

Remarks

The registered model names in your ordering information can be only obtained.

MFINFO?

Main Frame Information

Function

Queries MT8870A information

Query

```
MFINFO? [index]
```

Response

When parameter is 1

model name	Product name (≤30 characters)

When parameter is 2

model number	Model name (≤20 characters)

When parameter is 3

BIOS version	0 always

When parameter is 4

OS version	0 always

When parameter is 5

number of HW,name/rev,name/rev,...,name/rev

number of HW:	Number of hardware units
name:	Hardware name (≤30 characters, bundled with " ")
rev:	Hardware revision (≤10 characters)

When parameter is 6

number of FPGA,name/ver,name/ver,...,name/ver

number of FPGA:	Number of FPGAs
name:	FPGA name (≤30 characters, bundled with " ")
ver:	FPGA version (≤10 characters)

When parameter is 7

number of HW option,num/name/sw,num/name/sw,...,num/name/sw

number of HW option:	Number of hardware options
num:	Option number (0 to 999)
name:	Option name (≤50 characters, bundled with " ")
sw:	Option switch
ON :	With options
OFF :	Without options

When parameter is 8

number of Software license
,model/serial/name/ver,model/serial/name/ver,...,model/serial/name/ver

number of Software license:	Number of software licenses
model:	Software name (≤30 characters)
serial:	Serial number (10 characters)
name:	Software license name (≤50 characters, bundled with " ")

ver: Software version ("xx.xx" is described)

When parameter is 9

number of SW option,num/name/sw,num/name/sw,...,num/name/sw

number of SW option:	Number of software options
num:	Option name (≤20 characters) Example: MX887000A-001
name:	Option name (≤50 characters, bundled with " ")
sw:	Option switch
ON :	With options
OFF :	Without options

When parameter is 10

number of SG wave license

,model/serial/name/maxver,model/serial/name/maxver,...,model/serial/name/maxver

number of SG wave license:	Number of SG waveform licenses
model:	SG waveform license name (≤30 characters)
serial:	Serial number (10 characters)
name:	SG waveform license name (≤50 characters, bundled with " ")
maxver:	Usable maximum version (00.00 to 99.99)

When parameter is 0 or omitted

Character strings of all above responses comma separated

Parameters

index	Index of information to acquire
Range	0 to 10

Example of Use

MFINFO? 1

> Universal Wireless Test Set

MFINFO? 2

> MT8870A

MFINFO? 3

> 1.0

MFINFO? 4

> 2.6

MFINFO? 5

> 2,"Main Board"/01,"Option Board"/FE

MFINFO? 6

> 2,"High Speed Control FPGA"/01,"Wide Band FPGA"/FF

MFINFO? 7

> 2,001/"Rubidium"/ON,009/"GPIB Interface"/ON

MFINFO? 8

> 2,MX887010A/6123456789/"Cellular Measurement Software"/01.01,

MX887020A/6123456789/"Vector SG Software"/01.00

MFINFO? 9

> 2,MX887000A-001/"Extended Digitizing Option"/ON,
MX887000A-002/"Phase Noise Measurement"/OFF
MFINFO? 10
> 1,MX887070A/6123456789/"Standard wave license"/5

Related Command

SYSINFO?

Remarks

If the installed temporary license has expired, the license information cannot be obtained using this command.

MFMODEL?

Main Frame Model Number

Function

Queries MT8870A model number

Query

`MFMODEL?`

Response

model

Parameter

model	Model number

Example of Use

MFMODEL?
> MT8870A

MFPOWERONCNT?

Chassis Power ON Count

Function

Queries MT8870A power-on count

Query

```
MFPOWERONCNT?
```

Response

num

Parameter

num	MT8870A power-on count

Example of Use

MFPOWERONCNT?
> 1971

MFSERIAL?

Main Frame Serial No.

Function

Queries MT8870A serial number

Query

```
MFSERIAL?
```

Response

serial

Parameter

serial	Serial number 10-digit number Example 6200112233

Example of Use

MFSERIAL?
> 6123456789

MFTEMP?

Main Frame Temperature

Function

Queries MT8870A temperature

Query

`MFTEMP?`

Response

temp

Parameter

temp	MT8870A temperature (°C) resolution: 0.01

Example of Use

MFTEMP?
> 47.12

MFTMCNT?

Main Frame Running Time Count Read

Function

Queries MT8870A total running time

Query

`MFTMCNT?`

Response

time

Parameter

time	Total running time (minutes)

Example of Use

MFTMCNT?
> 1234

MFTRIGPORT

Set In/Out of Trigger Connector

Function

Sets or queries input/output of Trigger connector on MT8870A rear panel

Command

`MFTRIGPORT trigger`

Query

`MFTRIGPORT?`

Response

trigger

Parameters

trigger	Trigger connector input/output setting
INPUT	Input
OUTPUT	Output
Default	INPUT

Details

The trigger connector input/output can be set for each module.
The I/O of the trigger connector may be changed by using the commands below.

SOUR:GPRF:GEN:ARB:WAV:SYNC:STAR:PATT:SEL
SOUR:GPRF:GEN:SEQ:COMB:PATT:SYNC

Example of Use

MFTRIGPORT INPUT
MFTRIGPORT?
> INPUT

NETRESTART

Network Restart

Function

Restarts network

Command

`NETRESTART`

Details

Sending this command reboots the system. Remote control of all modules is disabled at restart. To continue remote control, wait a few seconds after sending this command and then reconnect remote control.

Example of Use

NETRESTART

Remarks

Communication settings not updated by *RST and :SYSTem:REBoot.

PORT

Set Direction of RF Connector

Function

Sets or queries connectors for RF signal input/output

Command

`PORT input,output`

Query

`PORT?`

Response

input,output

Parameters

input	Test Port number
PORT1	Test Port1
PORT2	Test Port2
PORT3	Test Port3
PORT4	Test Port4
Default	PORT1
output	Test Port number
PORT1	Test Port1
PORT2	Test Port2
PORT3	Test Port3
PORT4	Test Port4
Default	PORT1

Details

Test Port1 and Test Port2 can be set for both input and output.
Test Port3 and Test Port4 can be set for either input or output.

Example of Use

To set Test Port1 for input RF signals and Test Port2 for output RF signals:

PORT PORT1,PORT2
PORT?
> PORT1,PORT2

PORTSETTING

Raw Socket Port Number

Function

Sets or queries port number at RawSocket communication

Command

```
PORTSETTING port
```

Query

```
PORTSETTING?
```

Response

port	Port number

Parameter

port	Port number
Range	5001 to 60000
Default	56001

Details

The TCP/IP port number is set when performing RawSocket communication with the MT8870A.
The port number is common to IPv4 and IPv6.
To enable this command setting, send the NETRESTART command.

Example of Use

PORTSETTING 56001
NETRESTART
PORTSETTING?
> 56001

REBOOT

Reboot

Function

Reboots system

Command

```
REBOOT
```

Details

Sending this command reboots the system. Remote control of all modules installed in the MT8870A is disconnected while rebooting.
When the MT8870A Error lamp goes off, remote control can be performed.

Example of Use

REBOOT

Related Command

SHUTDOWN

REF

Frequency Reference

Function

Sets or queries frequency reference signal source

Command

```
REF source
```

Query

```
REF?
```

Response

source_res

Parameter

source	Internal reference signal or external reference signal
10MHZINT	MT8870A internal reference signal
10MHZEXT	Reference signal input to Ref Input connector at rear panel (Uses internal reference signal if external signal not detected.)
13MHZEXT	Reference signal input to Ref Input connector at rear panel (Uses internal reference signal if external signal not detected.)

AUTO	Reference signal input to Ref Input connector at rear panel (Uses internal reference signal if external signal not detected.)
source_res	Internal reference signal or external reference signal
10MHZINT	MT8870A internal reference signal
AUTO	Reference signal input to Ref Input connector at rear panel (Uses internal reference signal if external signal not detected.)
Default	AUTO

Details

The reference signal source is set for all modules installed in the MT8870A. It cannot be set for each module.

Example of Use

REF 10MHZINT
REF?
> 10MHZINT

Related Command

FREQREF

Remarks

Not initialized by *RST command

SHUTDOWN

Shutdown

Function

Shutdowns system

Command

```
SHUTDOWN
```

Details

Sending this command disconnects remote connection of all modules.

Example of Use

SHUTDOWN

Related Command

REBOOT

SYSERR?

System Error

Function

Queries error messages in error/event queue

Query

SYSERR?

Response

code,msg

Parameters

code	Error code
0	"No Error"
–113	"Undefined header"
–200	"Execution error"
–220	"Parameter error"
–310	"Option license is required to execute the command"
–370	"Main frame upgrade is required to execute the command"
msg	Event message corresponding to error/event number

Details

The response for no error/event queue is “0, “No Error””.

Example of Use

SYSERR?

> –220, “Parameter error”

SYSERRALL?

System All Errors

Function

Queries error messages in error/event queue

Query

```
SYSERRALL?
```

Response

code,msg,code,msg,.....

Parameters

code	Error code
0	"No Error"
–113	"Undefined header"
–200	"Execution error"
–220	"Parameter error"
–310	"Option license is required to execute the command"
–370	"Main frame upgrade is required to execute the command"
msg	Event message corresponding to error/event number

Details

The response for no error/event queue is “0, “No Error””.
After executing this command, the error/event queue is emptied.

Example of Use

SYSERRALL?
> –113, “Undefined header”, –220, “Parameter error”

SYSERRCNT?

System Error Count

Function

Queries number of all errors/events in error/event queue

Query

```
SYSERRCNT?
```

Response

num

Parameter

num	Number of errors/events

Details

Up to four errors are saved in the error event queue.
If a fifth error occurs, it is saved in the error event queue by deleting the oldest error in the queue.
The SYSERR? and SYSERRCODE? commands can be used to delete errors in the event queue sequentially from oldest first.
The SYSERRALL? and SYSERRCODEALL? commands can be used to delete all errors and empty the event queue.

Example of Use

SYSERRCNT?
> 3

SYSERRCODE?

System Error Code

Function

Queries error code in error/event queue

Query

```
SYSERRCODE?
```

Response

code

Parameter

code	Error code
0	"No Error"
–113	"Undefined header"
–200	"Execution error"
–220	"Parameter error"
–310	"Option license is required to execute the command"
–370	"Main frame upgrade is required to execute the command"

Details

The response for no error/event queue is 0.

Example of Use

SYSERRCODE?
> –220

SYSERRCODEALL?

System All Error Codes

Function

Queries number of all error codes in error/event queue

Query

```
SYSERRCODEALL?
```

Response

code,code,....

Parameter

code	Error code
0	"No Error"
–113	"Undefined header"
–200	"Execution error"
–220	"Parameter error"
–310	"Option license is required to execute the command"
–370	"Main frame upgrade is required to execute the command"

Details

The response for no error/event queue is 0.

After executing this command, the error/event queue is emptied.

Example of Use

SYSERRCODEALL?

> –113,–220

SYSINFO?

Module Information

Function

Queries module information

Query

```
SYSINFO? [index]
```

Response

When parameter is 1

model name	Product name (≤30 characters)

When parameter is 2

model number	Model name (≤20 characters)

When parameter is 3

BIOS version	BIOS (IPL) version (≤10 characters)

When parameter is 4

OS version	OS version (≤255 characters)

When parameter is 5

number of HW,name/rev,name/rev,...,name/rev

number of HW:	Number of hardware units
name:	Hardware name (≤30 characters, bundled with " ")
rev:	Hardware revision (≤10 characters)

When parameter is 6

number of FPGA,name/ver,name/ver,...,name/ver

number of FPGA:	Number of FPGAs
name:	FPGA name (≤30 characters, bundled with " ")
ver:	FPGA version ("xx.xx" is described)

When parameter is 7

number of HW option,num/name/sw,num/name/sw,...,num/name/sw

number of HW option:	Number of hardware options
num:	Option number (0 to 999)
name:	Option name (≤50 characters, bundled with " ")
sw:	Option switch
ON :	With options
OFF :	Without options

When parameter is 8

0 always

When parameter is 9

0 always

When parameter is 10

0 always

When parameter is 0 or omitted

Character strings of all above responses comma separated

Parameter

index		Index of information to acquire
	Range	1 to 10

Example of Use

SYSINFO? 1
> TRX Test Module
SYSINFO? 2
> MU887000A
SYSINFO? 3
> 1.0
SYSINFO? 4
> 2.6
SYSINFO? 5
> 2,"Main Board"/01,"Option Board"/FE
SYSINFO? 6
> 2,"High Speed Control FPGA"/01.00,"Wide Band FPGA"/FF.00
SYSINFO? 7
> 2,001/"Rubidium"/ON,009/"GPIB Interface"/ON
SYSINFO? 8
> 0
SYSINFO? 9
> 0
SYSINFO? 10
> 0

Related Command

MFINFO?

Remarks

If the installed temporary license has expired, the license information cannot be obtained using this command.

SYSLASTERROR?

System Last Error

Function

Queries last error

Query

```
SYSLASTERROR?
```

Response

msg

Parameter

msg	Abnormal contents
NO ERROR	No error
TemperatureAlarm	Forced shutdown due to temperature abnormality

Details

Executing this command does not clear the last error information.

Example of Use

SYSLASTERROR?
>NO ERROR

SYSMODEL?

Module Model Number.

Function

Queries module model number

Query

```
SYSMODEL?
```

Response

model

Parameter

model	Model number

Example of Use

SYSMODEL?
> MU887000A

SYSPLUGCNT?

Count of Module Insertion

Function

Queries module insertion/removal count

Query

```
SYSPLUGCNT?
```

Response

num

Parameter

num	Module insertion/removal count

Details

Note:

When a module is moved to another slot, the insertion/removal is counted.

When a module is inserted/removed in the same slot of the same MT8870A, insertion/removal is not counted.

Example of Use

SYSPLUGCNT?

> 40

SYSPOWERONCNT

Module Power ON Count

Function

Queries module power-on count

Query

```
SYSPOWERONCNT?
```

Response

num

Parameter

num	Module power-on count

Example of Use

SYSPOWERONCNT?

> 9

SYSSERIAL?

Module Serial No.

Function

Queries module serial number

Query

```
SYSSERIAL?
```

Response

serial

Parameter

serial	Serial number

Example of Use

SYSSERIAL?
> 6123456789

SYSSLOT?

Module Slot Number

Function

Queries number of slot where module installed

Query

```
SYSSLOT?
```

Response

slot

Parameter

slot	Slot number
Range	1 to 4

Example of Use

SYSSLOT?
> 1

SYSTEMDEFAULT

User Area Delete

Function

Deletes user area

Command

```
SYSTEMDEFAULT
```

Details

Data stored in the memory of FTP-accessible modules is erased.
All user files, such as waveform data, are erased.

Example of Use

SYSTEMDEFAULT

SYSTEMP?

Module Temperature

Function

Queries module temperature

Query

```
SYSTEMP?
```

Response

temp	Module temperature (°C)
Resolution:	0.01

Example of Use

SYSTEMP?
> 47.12

SYSTEST?

System Test

Function

Queries time when module self-diagnostics executed

Query

SYSTEST?

Response

status1,status2,status3,...,year,month,day,hour,min
The detail of status is the format as follows:

If MU887000A-002 Audio Measurement Hardware is not installed,
"Chassis EEPROM: xx","Module EEPROM: xx","RAM: xx","Correction ROM: xx","FPGA Configuration: xx"

If MU887000A-002 Audio Measurement Hardware is installed,
"Chassis EEPROM: xx","Module EEPROM: xx","RAM: xx","Correction ROM: xx","FPGA Configuration: xx", "Audio Memory: xx", "Audio Correction ROM: xx", "Audio FPGA Configuration: xx"

“Pass” or “Fail” is entered to xx.

Parameters

status	Last executed self-diagnostics test
year	year (2001 to)
month	month (1 to 12)
day	month (1 to 31)
hour	hour (0 to 23)
min	minute (0 to 59)

Example of Use

SYSTEST?
If MU887000A-002 Audio Measurement Hardware is not installed,
> "Chassis EEPROM: Pass","Module EEPROM: Pass","RAM: Pass","Correction ROM: Pass","FPGA Configuration:Pass",2012,4,5,12,34
If MU887000A-002 Audio Measurement Hardware is installed,
> "Chassis EEPROM: Pass","Module EEPROM: Pass ","RAM: Fail","Correction ROM: Pass","FPGA Configuration: Pass", "Audio Memory: Pass", "Audio Correction ROM: Pass", "Audio FPGA Configuration: Pass", 2012,4,5,12,34

SYSTESTERRCNT?

System Test Error Count

Function

Queries number of self-diagnostics test errors

Query

```
SYSTESTERRCNT?
```

Response

count

Parameters

count	Number of self-diagnostics errors
Range	0 or more

Example of Use

SYSTESTERRCNT?
> 0

SYSTESTERR_TM?

System Test Error Last Time

Function

Queries time of last self-diagnostics error

Query

`SYSTESTERR_TM?`

Response

year,month,day,hour,min

Parameters

year	year (2001 or later)
month>	month (1 to 12)
day	month (1 to 31)
hour	hour (0 to 23)
min	minute (0 to 59)

Example of Use

SYSTESTERR_TM?
> 2012,1,2,3,45

Remarks

The response is *** when there has never been a self-diagnostics error.

SYST:LANG

Language Selection of Remote Command

Function

Switches remote control command language mode

Command

`SYST:LANG mode`

Query

`SYST:LANG?`

Response

mode

Parameter

mode	Language mode
NAT	Native
SCPI	SCPI

Example of Use

SYST:LANG NAT
SYST:LANG?
>NAT

SYSTMCNT?

Module Running Time Count Read

Function

Queries module total running time

Query

`SYSTMCNT?`

Response

time

Parameter

time	Total running time (minutes)

Example of Use

SYSTMCNT?

> 1234

SYSTRIGINSRC

Select Source of Trigger Input

Function

Sets or queries the signal used as external trigger.

Command

```
SYSTRIGINSRC source
```

Query

```
SYSTRIGINSRC?
```

Response

source

Parameter

source	Signal used as external trigger
TRGIN	Trigger input to the trigger connector of the slot with the module to control installed.
TRGIN1	Trigger input to Trigger Connector 1.
SLOT1	Trigger signal output from the Slot 1 module.
Default	TRGIN

Details

When selecting TRGIN, set the I/O setting of the trigger connector to input.
The signal used as external trigger may be changed by using the commands below.
SOUR:GPRF:GEN:ARB:WAV:SYNC:STAR:PATT:SEL
SOUR:GPRF:GEN:SEQ:COMB:PATT:SYNC

Example of Use

SYSTRIGINSRC TRGIN1
SYSTRIGINSRC?
> TRGIN1

Related Command

MFTRIGPORT

SYSTRIGOUTSRC

Select Source of Trigger Output

Function

Sets and queries signal output at Trigger connector on MT8870A rear panel

Command

```
SYSTRIGOUTSRC source
```

Query

```
SYSTRIGOUTSRC?
```

Response

source

Parameter

source	Signal output at Trigger connector
WFM1	SG waveform marker 1
WFM2	SG waveform marker 2
WFM3	SG waveform marker 3
SGSYNC	SG synchronization start signal
NONE	No signal output (initial value)
Default	NONE

Details

When setting output at the Trigger connector, the signal selected by this command is output at the Trigger connector.

The signal output to the trigger connector may be changed by using the commands below.

SOUR:GPRF:GEN:ARB:WAV:SYNC:STAR:PATT:SEL

SOUR:GPRF:GEN:SEQ:COMB:PATT:SYNC

Example of Use

SYSTRIGOUTSRC WFM1

SYSTRIGOUTSRC?

> WFM1

Related Command

MFTRIGPORT

SYSVER?

Package Version

Function

Queries package version

Query

SYSVER?

Response

ver

Parameter

ver	Package version (xx.xx.xx)

Example of Use

SYSVER?
> 1.00.00

TRM

Message Terminator

Function

Sets or queries terminator (characters added to message end)

Command

```
TRM code
```

Query

```
TRM?
```

Response

code	End code

Parameter

code	End code
LF	LF (Line Feed)
CRLF	CR/LF (Carriage Return + Line Feed)
NONE	None (only EOI)
Default	CRLF

Details

This can be set for each module.
Pressing the IP Reset button on the MT8870A panel at power-on switches the termination code to the initial value.

Example of Use

TRM CRLF
TRM?
>CRLF

Remarks

The processing is the same as DELM.
The *RST, REBOOT, and NETRESTART commands do not perform initialization.
When the parameter is set to NONE in Ethernet connection, the terminator of the response message from the main unit is set to CRLF.

Chapter 7 Performance Tests

This chapter explains how to setup the measuring instruments required for the performance tests and the test procedures.
To record the performance test results, use the performance test result sheets in Appendix C.

7.1 Outline

The performance tests are performed to assure that the MU887000A performance does not deteriorate. Test the performance of the MU887000A at the initial acceptance inspection, at periodic inspections, and after repairs. Test important items periodically to assure the performance. This chapter explains the following test items.

- VSWR
- SG Level accuracy
- SG Spurious
- SA Level accuracy
- SA Linearity

We recommend testing the performance periodically once or twice a year. If the test results do not meet the specifications, contact the Anritsu Service and Sales office. Contact information is available in a separate file (for the PDF version), and on the last page of this manual (for the printed version).

CAUTION

Warm-up the MU887000A and the required measuring instruments for at least 30 minutes (except when specified otherwise) to stabilize them. To achieve the highest accuracy, the test should be performed at room temperature using a power supply with as little voltage fluctuation as possible in an environment free from noise, vibration, dust and humidity.

Perform the MU887000A level calibration (full band calibration) before performing the performance test. Refer to Section 3.5.2, "Level calibration" for details of the commands for the level calibration.

7.2 Instruments for Performance Tests

The following table lists the measuring instruments required for testing the MU887000A performance and the specifications for each instrument.

Table 7.2-1 Measuring Instruments for Performance Test

Instrument Name	Instrument Required Performance*[1]	Recommended Instrument
Signal Generator	• Frequency Range: 9.9 to 6000 MHz • Frequency Setting Resolution: 100 kHz • Output Level Range: –30 to 0 dBm • External Reference Input: (10 MHz) • Connector: N type	Vector Signal Generator (MG3700A) Mechanical Attenuator (MG3700A-002) High Frequency 6 GHz (MG3700A-011)
Power Meter	• Main Frame Accuracy: ±0.02 dB • Frequency Range: 9.9 MHz to 6000 MHz • Display Resolution: 0.01 dB	Power Meter (ML2438A)
Power Sensor	• Frequency Range: 9.9 MHz to 6000 MHz • Measured Power Range: –40 to +20 dBm • Input Connector: N type	Power Sensor (MA2442D)
Fixed Attenuator (6 dB)	• Frequency Range: 10 MHz to 6000 MHz • Attenuation: 6 dB • Connector: N type	Fixed Attenuator (MP721B)
Fixed Attenuator (3 dB)*2	• Frequency Range: 9.9 MHz to 6000 MHz • Attenuation: 3 dB • Connector: N type	Fixed Attenuator (MP721A)
Adaptor	• Frequency Range: 9.9 MHz to 6000 MHz • Connector: N type	
50Ω Termination	• Frequency Range: 10 MHz to 6000 MHz • Connector: N type	Termination (MP752A)
Signal Analyzer	• Frequency Range: 10 MHz to 6000 MHz • Frequency Setting Resolution: 1 Hz • Measured Power Range: –140 to +20 dBm • External Reference Input: (10 MHz)	Signal Analyzer (MS2690A Series or MS2830A)
Network Analyzer	• Frequency Range: 10 MHz to 6000 MHz • Input Connector: N type	Network Analyzer (MS4642B)
Application Software		Cellular Standards Sequence Measurement (MX887010A)

*1: The performance covers the test item measurement range

*2: Required two pieces

7

Performance Tests

Table 7.2-2 Measuring Instruments Item and Instrument to use

Instrument Name	Performance Test Item				
	VSWR	SG Level Accuracy	SG Spurious	SA Level Accuracy	SA Linearity
Signal Generator		✓		✓	✓
Power Meter		✓		✓	✓
Power Sensor		✓		✓	✓
Fixed Attenuator (6 dB)		✓			
Fixed Attenuator (3 dB)				✓	✓
Adaptor		✓		✓	✓
50Ω Termination	✓				
Signal Analyzer		✓	✓	✓	✓
Network Analyzer	✓				
Application Software	✓	✓		✓	✓

7.3 Performance Test for Each Measurement

7.3.1 VSWR

The following test related to the VSWR at signal input to the MU887000A connector is performed.

(1) Test target specification

Port1, Port2

Frequency f	Specifications
10 MHz≤f< 400 MHz	<1.5
400 MHz≤f≤2700 MHz	<1.2
2700 MHz<f≤3800 MHz	<1.3
3800 MHz<f≤6000 MHz	<1.5

Port3, Port4

Frequency f	Specifications
10 MHz≤f< 30 MHz	<1.8
30 MHz≤f≤3800 MHz	<1.5
3800 MHz<f≤6000 MHz	<1.6

(2) Measuring instruments

- Network analyzer: MS4642B
- Calibration kit
- 50 Ω Termination

(3) Setup

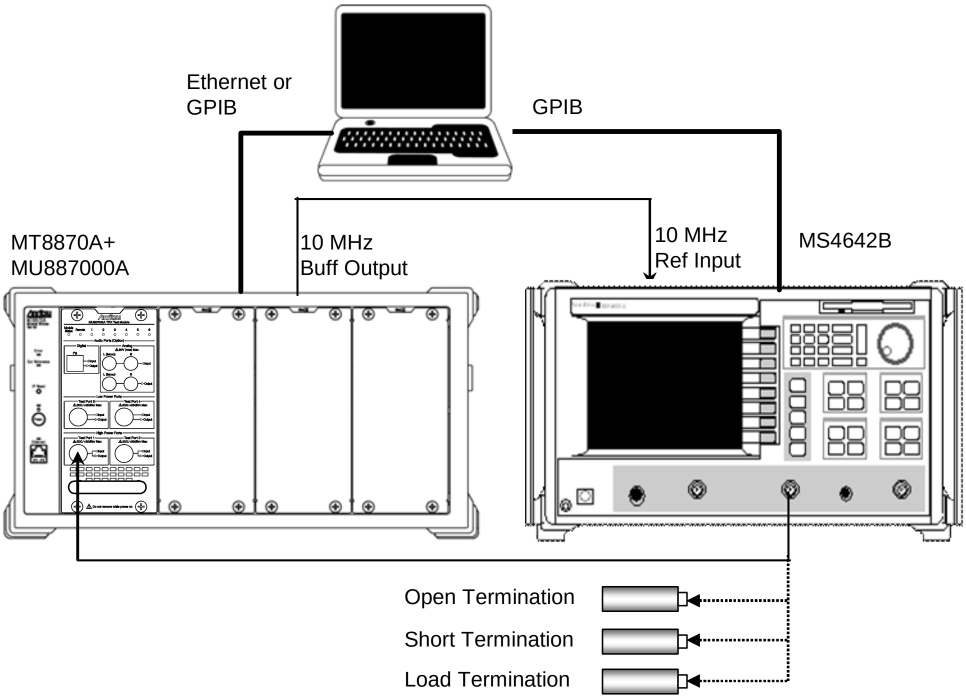


Figure 7.3.1-1 VSWR Test Setup (Port1)

(4) Procedure

Port1

1. Use an N-type cable to connect the network analyzer to the MU887000A.
2. Set the network analyzer as follows:
   - Start Frequency: 10 MHz
   - Stop Frequency: 6000 MHz
3. Calibrate the S_{11} (VSWR) measurement using the Open, Short and Load terminators at the ends of the N-type cable.
4. Connect N-type cable to Port1 of MU887000A.
5. Initialize the settings of MU887000A.
   `*RST`
6. Set the output and input ports to Port1.
   `:ROUTe:PORT:CONNect:DIRection PORT1,PORT1`
7. Set the VSG operation mode to Normal.
   `:SOURce:GPRF:GENerator:MODE NORMAL`
8. Set the VSG modulation to CW.
   `:SOURce:GPRF:GENerator:BBMode CW`
9. Set the VSG output level to –10 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`
10. Output the VSG signal.
    `:SOURce:GPRF:GENerator:STATe ON`
11. Set the VSG output frequency to 1 MHz.
    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 1MHZ`
12. Set the MU887000A input level to –20 dBm.
    `:CONFigure:CELLular:MEASurement:RFSettings:LEVel -20`
13. Measure S_{11} (VSWR) using the network analyzer.

When testing the Port2, read “Port1” in the procedure of Port1 measurement as “Port2”.

Port3

1. Use an N-type cable to connect the network analyzer to the MU887000A.
2. Set the network analyzer as follows:
   - Start Frequency: 10 MHz
   - Stop Frequency: 6000 MHz
3. Calibrate the S_{11} (VSWR) measurement using the Open, Short and Load terminators at the ends of the N-type cable.
4. Connect N-type cable to Port3 of MU887000A.
5. Initialize the settings of MU887000A.
   `*RST`
6. Set the output port to Port3 and the input port to Port4.
   `:ROUTe:PORT:CONNect:DIRection PORT4,PORT3`
7. Set the VSG operation mode to Normal.
   `:SOURce:GPRF:GENerator:MODE NORMAL`
8. Set the VSG modulation to CW.
   `:SOURce:GPRF:GENerator:BBMode CW`
9. Set the VSG output level to 0 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel 0`
10. Output the VSG signal.
    `:SOURce:GPRF:GENerator:STATe ON`
11. Set the VSG output frequency to 1 MHz.
    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 1MHZ`
12. Set the MU887000A input level to –20 dBm.
    `:CONFigure:CELLular:MEASurement:RFSettings:LEVel -20`
13. Connect 50Ω Termination to Port4 of MU887000A.
14. Measure S_{11} (VSWR) using the network analyzer.
15. Set the output port to Port4 and the input port to Port3.
    `:ROUTe:PORT:CONNect:DIRection PORT3,PORT4`
16. Repeat steps 7 thru 14.

When testing the Port4, read “Port3” in the procedure of Port3 measurement as “Port4”, and read “Port4” as “Port3” as well.

7.3.2 SG Level Accuracy

The following test related to the SG level accuracy is performed.

(1) Test target specifications

CW, After calibration, At 10 to 40 °C

Port1,Port2

Frequency f (MHz)	Specifications	Remarks
10≤ f <400	±1.3 dB*	–120≤Output Level
400≤ f ≤3800	±1.0 dB	–120≤Output Level
3800< f ≤6000	±1.3 dB	–100≤Output Level

*: Signal analyzer input level : +15 dBm

Port3,Port4

Frequency f (MHz)	Specifications	Remarks
10≤ f <400	±1.3 dB	–110≤Output Level
400≤ f ≤3800	±1.0 dB	–110≤Output Level
3800< f ≤6000	±1.3dB	–110≤Output Level

(2) Measuring instruments

- Signal generator: MG3700A
- Power meter: ML2438A
- Power sensor: MA2442D
- Signal Analyzer: MS2690A Series or MS2830A
- N-type attenuator (6 dB): MP721B
- Adaptor

(3) Setup

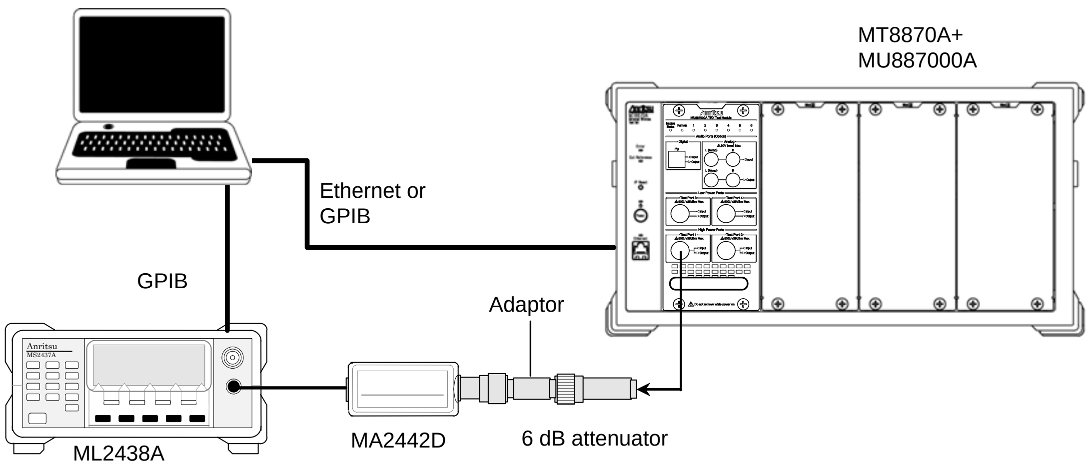


Figure 7.3.2-1 SG Level Accuracy Test Setup (Absolute level measurement)

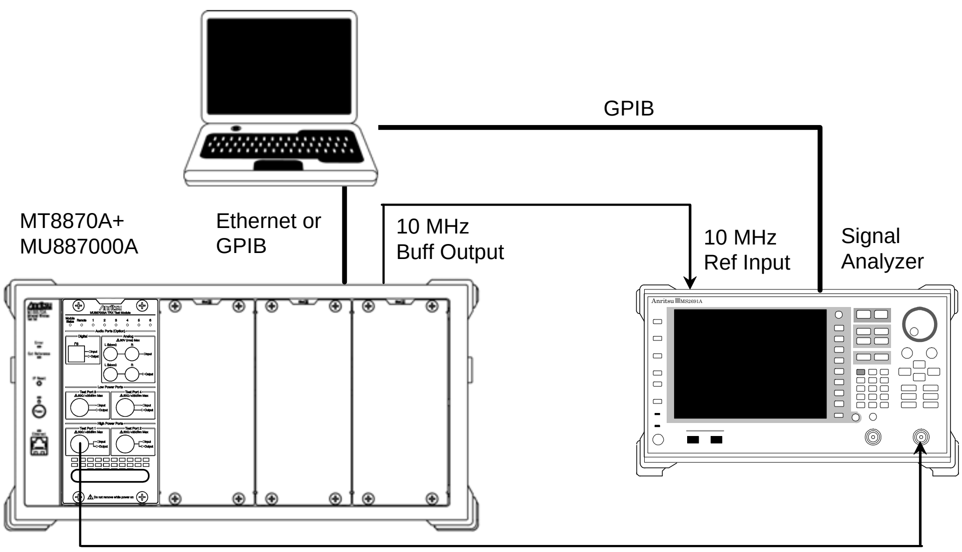


Figure 7.3.2-2 SG Level Accuracy Test Setup (Relative level measurement)

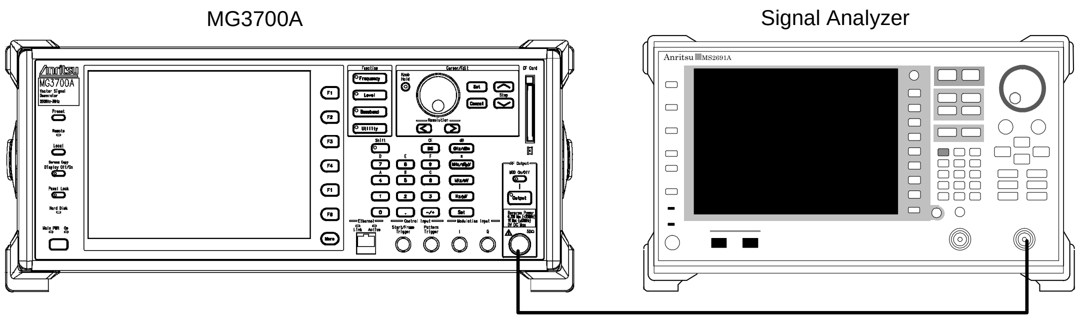


Figure 7.3.2-3 Measurement Level Calibration of Signal Analyzer

Preparation for SG Level Accuracy Measurement

1. Calibrate the power sensor (zero point and sensitivity).
2. Measure the 6 dB attenuator's attenuation value at each frequency shown in “Table 7.3.2-1 Frequency List (Power Meter)”.
3. Connect the signal generator and the signal analyzer as shown in “Figure 7.3.2-3 Measurement Level Calibration of Signal Analyzer”.
4. Set the signal generator output level to –50 dBm.
5. Set the signal generator output frequency to 10.1 MHz.
6. Set the values of #1 in “Table 7.3.2-2 Settings of MS2690 Series or MS2830A” to the signal analyzer.
7. #1 Measured Value is the level measured by the signal analyzer.
8. Set the values of #2 in “Table 7.3.2-2 Settings of MS2690 Series or MS2830A” to the signal analyzer.
9. Calculate the difference between #2 Measured Value and #1 Measured Value. This difference is the error between ranges (D).

   D = (#1 Measured Value) – (#2 Measured Value)
10. Set the signal generator output frequency in “Table 7.3.2-3 Frequency List (Signal Analyzer)” to signal generator and repeat steps 5 thru 9.

Table 7.3.2-1 Frequency List (Power Meter)

Frequency (MHz)	Attenuation of 6 dB attenuator (dB)	Frequency (MHz)	Attenuation of 6 dB attenuator (dB)
10.1		2200.1	
350.1		3000.1	
800.1		3799.9	
1000.1		4000.1	
1500.1		5000.1	
1800.1		5999.9	
2000.1			

Table 7.3.2-2 Settings of MS2690 Series or MS2830A

No.	MU887000A Output level setting (dBm)	Settings of MS2690 Series or MS2830A							
		RBW (Hz)	Zone Width (Hz)	Span (kHz)	ATT (dB)	Preamp	Average	Average Count	Ref Lev (dBm)
#1	≤0dBm ≥–49dBm	100	781.3	25	20	OFF	OFF	2	0
#2	<–49dBm ≥–79dBm	100	781.3	25	0	ON	OFF	2	–40
#3	<–79dBm ≥–100dBm	10	156.3	2.5	0	ON	OFF	2	–40
#4	<–100dBm ≥–110dBm	10	15.63	1	0	ON	ON	2	–70
#5	<–110dBm ≥–120dBm	3	15.63	1	0	ON	ON	4	–70

Table 7.3.2-3 Frequency List (Signal Analyzer)

SG Output Frequency(MHz)	SA Center Frequency(MHz)	#1 Measured Value (dBm)	#2 Measured Value (dBm)	#1–#2 (dB)
10.1	10.1			
350.1	350.1			
800.1	800.1			
1000.1	1000.1			
1500.1	1500.1			
1800.1	1800.1			
2000.1	2000.1			
2200.1	2200.1			
3000.1	3000.1			
3799.9	3799.9			
4000.1	4000.1			
5000.1	5000.1			
5999.9	5999.9			

(4) Procedure

Port1

Absolute level measurement

1. To test Port1, set-up the instruments as shown in Figure 7.3.2-1.
2. Initialize the settings of MU887000A.
   `*RST`
3. Set the output and input ports to Port1.
   `:ROUTe:PORT:CONNect:DIRection PORT1,PORT1`
4. Set the MU887000A input level to +15 dBm.
   `:CONFigure:CELLular:MEASurement:RFSettings:LEVel +15`
5. Set the MU887000A input frequency to 400 MHz.
   `:CONFigure:CELLular:MEASurement:RFSettings:FREQuency 400MHZ`
6. Set the VSG operation mode to Normal.
   `:SOURce:GPRF:GENerator:MODE NORMAL`
7. Set the VSG modulation to CW.
   `:SOURce:GPRF:GENerator:BBMode CW`
8. Set the VSG output level to –20 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel -20`
9. Output the VSG signal.
   `:SOURce:GPRF:GENerator:STATe ON`
10. Set the VSG output frequency to 10.1 MHz.
    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 10.1MHZ`
11. Set the power meter offset to the 6 dB attenuator's attenuation value of the same frequency as the VSG output frequency.
12. Measure the VSG output level at the power meter. This measured value is the absolute level reference of Port1 at 10 MHz (C).
13. Set the VSG frequency to 350.1 thru 3799.9 MHz in “Table 7.3.2-1 Frequency List (Power Meter)” and repeat steps 10 thru 12.
14. Set the VSG output level to –28 dBm.
    `:SOURce:GPRF:GENerator:RFSettings:LEVel -28`
15. Set the VSG output frequency to 4000.1 MHz.
    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 4000.1MHZ`
16. Set the power meter offset to the 6 dB attenuator's attenuation value of the same frequency as the VSG output frequency.

17. Measure the VSG output level at the power meter. This measured value is the absolute level reference of Port1 at 4000 MHz (C).
18. Set the VSG frequency to 5000.1 and 5999.9 MHz in “Table 7.3.2-1 Frequency List (Power Meter)” and repeat steps 15 thru 17.

Relative level measurement

19. Set-up the instruments as shown in Figure 7.3.2-2.
20. Set the VSG output frequency to 10.1 MHz.

    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 10.1MHZ`
21. Set the VSG output level to –20 dBm.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -20`
22. Measure the VSG outputlevel using the signal analyzer and make this measurement *A* dBm.
23. Set the VSG output level to –10 dBm and make this setting *P*.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`
24. Set the signal analyzer as shown in “Table 7.3.2-2 Settings of MS2690 Series or MS2830A” according to the VSG output level.
25. Measure the VSG output level using the signal analyzer and make this measurement *B* dBm.
26. Calculate the level accuracy using the following equation.

    Level Accuracy = B – A + C – P
27. Decrease the VSG output level in 5 dB steps successively such as –15 dBm, –20 dBm, …, –120 dBm and repeat steps 23 thru 26.

    However, if the output level is measured by settings of #2 thru #5 in Table 7.3.2-2, calculate the level accuracy using the following equation. D is the error between ranges of the signal analyzer.

    Level Accuracy = (B + D) – A + C – P
28. Set the VSG frequency to 350.1 thru 3799.9 MHz in “Table 7.3.2-3 Frequency List (Signal Analyzer)” and repeat steps 20 thru 27.
29. Set the VSG output frequency to 4000.1 MHz.

    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 4000.1MHZ`
30. Set the VSG output level to –28 dBm.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -28`
31. Measure the VSG output level using the signal analyzer and make this measurement *A* dBm.

32. Set the VSG output level to –18 dBm and make this setting P.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -18`
33. Set the signal analyzer as shown in “Table 7.3.2-2 Settings of MS2690 Series or MS2830A” according to the VSG output level.
34. Measure the VSG output level using the signal analyzer and make this measurement B dBm.
35. Calculate the level accuracy using the following equation.

    Level Accuracy = B – A + C – P
36. Decrease the VSG output level in 5 dB steps successively such as –15 dBm, –20 dBm, …, –120 dBm and repeat steps 32 thru 35.

    However, if the output level is measured by settings of #2 thru #5 in Table 7.3.2-2, calculate the level accuracy using the following equation. D is the error between ranges of the signal analyzer.

    Level Accuracy = (B + D) – A + C – P
37. Set the VSG frequency to 5000.1 and 5999.9 MHz in “Table 7.3.2-3 Frequency List (Signal Analyzer)” and repeat steps 29 thru 36.

When testing the Port2, read “Port1” in the procedure of Port1 measurement as “Port2".

Port3

Absolute level measurement

1. Connect the power sensor to Port3 as shown in Figure 7.3.2-1.
2. Initialize the settings of MU887000A.
   `*RST`
3. Set the output port to Port3 and the input port to Port4.
   `:ROUTe:PORT:CONNect:DIRection PORT4,PORT3`
4. Set the VSG operation mode to Normal.
   `:SOURce:GPRF:GENerator:MODE NORMAL`
5. Set the VSG modulation to CW.
   `:SOURce:GPRF:GENerator:BBMode CW`
6. Set the VSG output level to –10 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`
7. Output the VSG signal.
   `:SOURce:GPRF:GENerator:STATe ON`
8. Set the VSG output frequency to 10.1 MHz.
   `:SOURce:GPRF:GENerator:RFSettings:FREQuency 10.1MHZ`
9. Set the power meter offset to the 6 dB attenuator's attenuation value of the same frequency as the VSG output frequency.
10. Measure the VSG output level at the power meter. This measured value is the absolute level reference of Port3 at 10 MHz (C).
11. Set the VSG frequency to 350.1 thru 3799.9 MHz in “Table 7.3.2-1 Frequency List (Power Meter)” and repeat steps 8 thru 10.
12. Set the VSG output level to –28 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel -28`
13. Set the VSG output frequency to 4000.1 MHz.
   `:SOURce:GPRF:GENerator:RFSettings:FREQuency 4000.1MHZ`
14. Set the power meter offset to the 6 dB attenuator's attenuation value of the same frequency as the VSG output frequency.
15. Measure the VSG output level at the power meter. This measured value is the absolute level reference of Port3 at 4000 MHz (C).
16. Set the VSG frequency to 5000.1 and 5999.9 MHz in “Table 7.3.2-1 Frequency List (Power Meter)” and repeat steps 13 thru 15.

Relative level measurement

17. Connect the SA to Port3 in Figure 7.3.2-2.

18. Set the VSG output frequency to 10.1 MHz.

    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 10.1MHZ`

19. Set the VSG output level to –10 dBm.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`

20. Measure the VSG output level using the signal analyzer and make this measured value A dBm.

21. Set the VSG output level to 0 dBm and make this setting P.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel 0`

22. Set the signal analyzer as shown in "Table 7.3.2-2 Settings of MS2690 Series or MS2830A" according to the VSG output level.

23. Measure the VSG output level using the signal analyzer and make this measured value B dBm.

24. Calculate the level accuracy using the following equation.

    Level Accuracy = B – A + C – P

25. Decrease the VSG output level in 5 dB steps successively such as –5 dBm, –10 dBm, …, –110 dBm and repeat steps 23 thru 26.

    However, if the output level is measured by settings of #2 thru #5 in Table 7.3.2-2, calculate the level accuracy using the following equation. D is the error between ranges of the signal analyzer.

    Level Accuracy = (B + D) – A + C – P

26. Set the VSG frequency to 350.1 thru 3799.9 MHz in "Table 7.3.2-3 Frequency List (Signal Analyzer)" and repeat steps 18 thru 25.

27. Set the VSG output frequency to 4000.1 MHz.

    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 4000.1MHZ`

28. Set the VSG output level to –18 dBm.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -18`

29. Measure the VSG output level using the signal analyzer and make this measured value A dBm.

30. Set the VSG output level to –8 dBm and make this setting P.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -8`

31. Set the signal analyzer as shown in "Table 7.3.2-2 Settings of MS2690 Series or MS2830A" according to the VSG output level.

32. Measure the VSG output level using the signal analyzer and make this measured value B dBm.

33. Calculate the level accuracy using the following equation.

    Level Accuracy = B – A + C – P

34. Decrease the VSG output level in 5 dB steps successively such as –13 dBm, –18 dBm, …,–108 dBm, –110 dBm and repeat steps 30 thru 33.

    However, if the output level is measured by settings of #2 thru #5 in Table 7.3.2-2, calculate the level accuracy using the following equation. D is the error between ranges of the signal analyzer.

    Level Accuracy = (B + D) – A + C – P

35. Set the VSG frequency to 5000.1 and 5999.9 MHz in “Table 7.3.2-3 Frequency List (Signal Analyzer)” and repeat steps 27 thru 34.

When testing the Port4, read “Port3” in the procedure of Port3 measurement as “Port4”, and read “Port4” as “Port3” as well.

7.3.3 SG Spurious

The following test related to the SG spurious is performed.

(1) Test target specifications

Frequency	Specification
10 MHz to 6000 MHz	Harmonic Distortion < –25 dBc

(2) Measuring instrument

- Signal analyzer: MS2690A series or MS2830A

(3) Setup

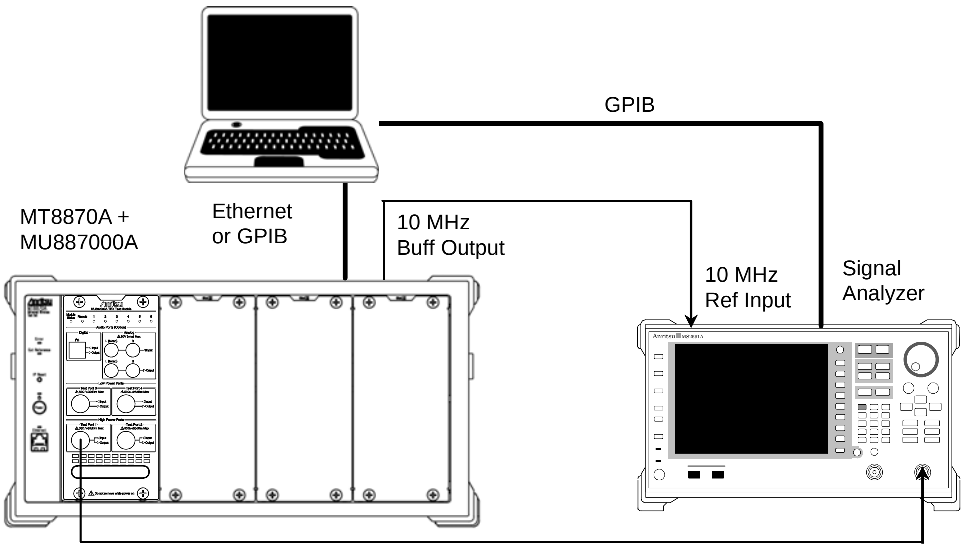


Figure 7.3.3-1 SG Spurious Measurement Setup

(4) Procedure

SG Spurious is measured at Port1 and Port3 only.

Port1

1. To test Port1, set-up the instruments as shown in Figure 7.3.3-1.
2. Initialize the settings of MU887000A.

   `*RST`
3. Set the output and input ports to Port1.

   `:ROUTe:PORT:CONNect:DIRection PORT1,PORT1`
4. Set the VSG operation mode to Normal.

   `:SOURce:GPRF:GENerator:MODE NORMAL`
5. Set the VSG modulation to CW.

   `:SOURce:GPRF:GENerator:BBMode CW`
6. Set the VSG output level to –10 dBm.

   `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`
7. Output the VSG signal

   `:SOURce:GPRF:GENerator:STATe ON`
8. VSG output frequency to 10 MHz.

   `:SOURce:GPRF:GENerator:RFSettings:FREQuency 10MHZ`
9. Measure the 10 MHz signal level (Reference) at the signal analyzer. This value is set to *A* dBm.
10. Measure the 20 MHz signal level (Second harmonics) at the signal analyzer. This value is set to *B* dBm. However, do not measure the level of second-harmonics above 6 GHz.
11. Measure the 30 MHz signal level (Third harmonics) at the signal analyzer. This value is set to *C* dBm. However, do not measure the level of third-harmonics above 6 GHz.
12. Calculate the harmonic distortion as following.

    Second harmonic distortion = B – A (dBc)

    Third harmonic distortion = C – A (dBc)
13. Increase the VSG output frequency in 100 MHz steps successively such as 100 MHz, 200 MHz, …,6000 MHz and repeat steps 8 thru 12.

Port3

1. Connect the signal analyzer to Port3 as shown in Figure 7.3.3-1.
2. Initialize the settings of MU887000A.

   `*RST`
3. Set the output port to Port3 and the input port to Port4.

   `:ROUTe:PORT:CONNect:DIRection PORT4,PORT3`
4. Set the VSG operation mode to Normal.

   `:SOURce:GPRF:GENerator:MODE NORMAL`
5. Set the VSG modulation to CW.

   `:SOURce:GPRF:GENerator:BBMode CW`
6. Set the VSG output level to 0 dBm.

   `:SOURce:GPRF:GENerator:RFSettings:LEVel 0`
7. Output the VSG signal

   `:SOURce:GPRF:GENerator:STATe ON`
8. Set the VSG output frequency to 10 MHz.

   `:SOURce:GPRF:GENerator:RFSettings:FREQuency 10MHZ`
9. Measure the 10 MHz signal level (Reference) at the signal analyzer. This value is set to *A* dBm.
10. Measure the 20 MHz signal level (Second harmonics) at the signal analyzer. This value is set to *B* dBm. However, do not measure the level of second-harmonics above 6 GHz.
11. Measure the 30 MHz signal level (Third harmonics) at the signal analyzer. This value is set to *C* dBm. However, do not measure the level of third-harmonics above 6 GHz.
12. Calculate the harmonic distortion as following.

    Second harmonic distortion = B – A (dBc)

    Third harmonic distortion = C – A (dBc)
13. Increase the VSG output frequency in 100 MHz steps successively such as 100 MHz, 200 MHz, …,6000 MHz and repeat steps 8 thru 12.

7.3.4 SA Level Accuracy

The following test related to the SA level accuracy is performed.

(1) Test target specifications

At 10 to 40 °C

Port1,Port2

Frequency (MHz) / Level (dBm)	400≤f≤3800
–30 ≤level ≤+35	±0.5 dB
–55 ≤level <–30	±0.7 dB
–65 ≤level <–55	±0.9 dB

(2) Measuring instruments

- Signal Generator: MG3700A
- Power meter: ML2438A
- Power sensor: MA2442D
- Signal Analyzer: MS2690A Series or MS2830A
- N-type attenuator (3 dB): MP721A 2 pcs
- Adaptor

(3) Setup

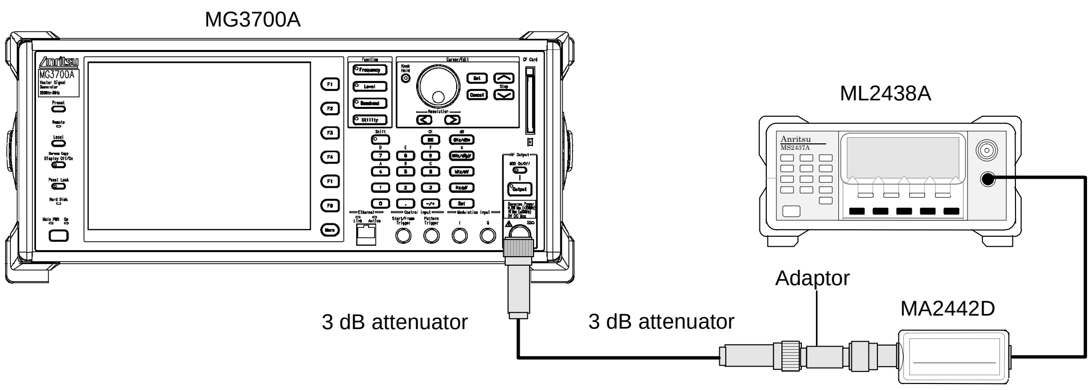


Figure 7.3.4-1 Output Level Setup (0, –25, –30 dBm)

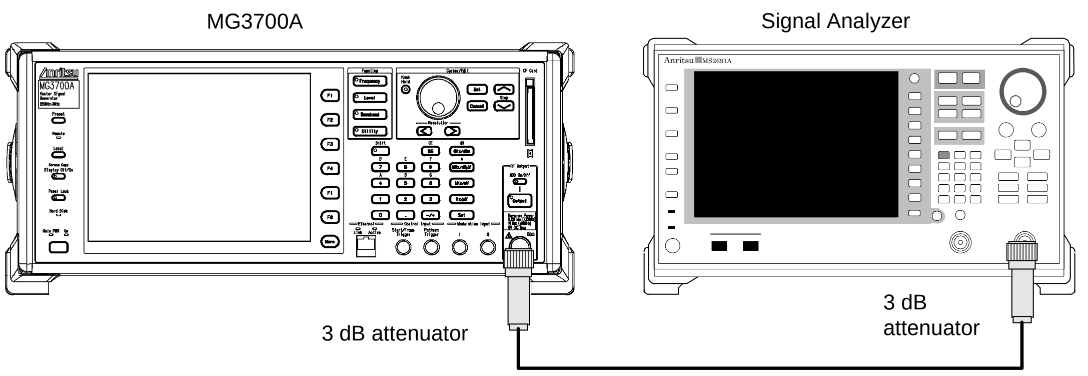


Figure 7.3.4-2 Output Level Setup (–35 dBm)

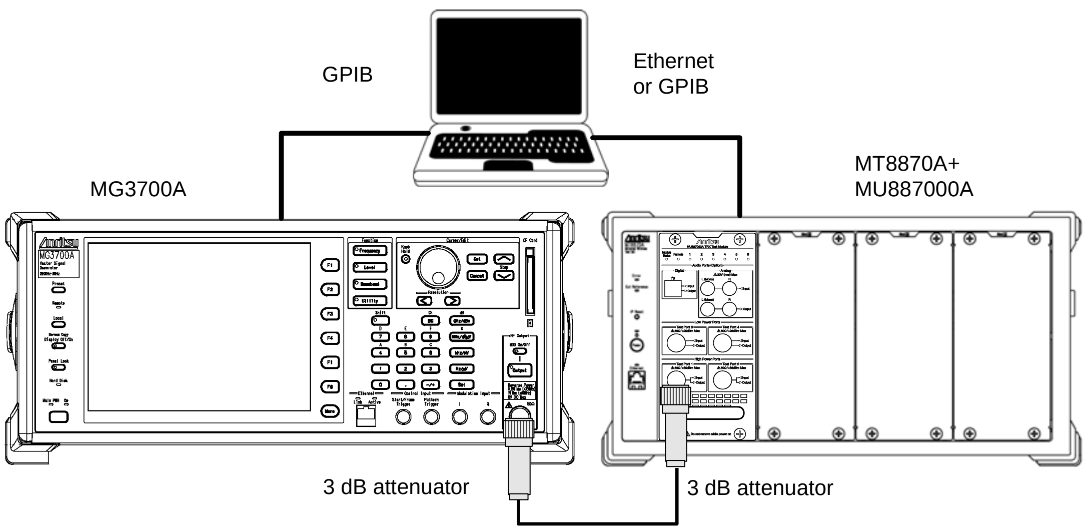


Figure 7.3.4-3 SA Level Accuracy Setup (Port1)

Preparation for SA Level Accuracy Measurement

1. Calibrate the power sensor (zero point and sensitivity).
2. Connect MG3700A and the power meter using the cable with 3 dB attenuator at both edges.
3. Set the MG3700A frequency to 999.9 MHz.
4. Tune the output level of MG3700A so that the power meter displays the value nearest to 0 dBm.

   Make this setting value to MG3700A P0 and the measured value by the power meter C0.
5. Tune the output level of MG3700A so that the power meter displays the value nearest to –25 dBm.

   Make this setting value to MG3700A P25 and the measured value by the power meter C25.
6. Tune the output level of MG3700A so that the power meter displays the value nearest to –30 dBm.

   Make this setting value to MG3700A P30 and the measured value by the power meter C30.
7. Set the MG3700A frequency to the value shown in "Table 7.3.4-1 Frequency List" (1999.9, 2999.9, and 3799.9 MHz), and repeat steps 3 thru 6.
8. Connect MG3700A and the signal analyzer using the cable with 3 dB attenuator at both edges as shown in "Figure 7.3.4-2 Output Level Setup (–35 dBm)".
9. Set the MG3700A frequency to 999.9 MHz.
10. Set the MG3700A output level to the value (P25) obtained in step 5, for outputting the –25 dBm level.
11. Measure the level at the signal analyzer. This value is set to *S*.
12. Set the MG3700A output level to the value (P25–10) which is 10 dB lower than the setting value in the step 10.
13. Measure the level at the signal analyzer. This value is set to *S1*.
14. Set the MG3700A frequency to the value shown in "Table 7.3.4-1 Frequency List" (1999.9, 2999.9, and 3799.9 MHz), and repeat steps 9 thru 13.

7

Performance Tests

Table 7.3.4-1 Frequency List

No.	MU887000A Input Frequency (MHz)	MG3700A Setting Frequency (MHz)
1	1000	999.9
2	2000	1999.9
3	3000	2999.9
4	3800	3799.9

(4) Procedure

SA Level Accuracy is measured at Port1 and Port2 only.

Port1

1. To test Port1, set-up the instruments as shown in Figure 7.3.4-3.
2. Initialize the settings of MU887000A.
   ```
   *RST
   ```
3. Set the output and input ports to Port1.
   ```
   :ROUTe:PORT:CONNect:DIRection PORT1,PORT1
   ```
4. Set the VSG operation mode to Normal.
   ```
   :SOURce:GPRF:GENerator:MODE NORMAL
   ```
5. Set the VSG modulation to CW.
   ```
   :SOURce:GPRF:GENerator:BBMode CW
   ```
6. Set the VSG output level to –120 dBm.
   ```
   :SOURce:GPRF:GENerator:RFSettings:LEVel -120
   ```
7. Stop outputting the VSG signal.
   ```
   :SOURce:GPRF:GENerator:STATe OFF
   ```
8. Set the measurement function of MU887000A to spectrum monitor.
   ```
   :INSTrument[:SELect] CELLULAR
   :CONFigure:CELLular:MEASurement:STANdard COMMON
   :CONFigure:CELLular:MEASurement:SELect SPMON
   ```
9. Set the spectrum monitor to followings.

   Input Frequency:No.1 frequency in “Table 7.3.4-1 Frequency List”
   Span: 50 MHz
   Power measurement bandwidth: 300 kHz
   RBW: 100 kHz
   Detection: rms
   ```
   :CONFigure:CELLular:MEASurement:RFSettings:FREQue
   ncy 1000MHZ
   :CONFigure:CELLular:COMMon:SPMonitor:SPAN 50MHZ
   :CONFigure:CELLular:COMMon:SPMonitor:PMBW 300KHZ
   :CONFigure:CELLular:COMMon:SPMonitor:RBW 100KHZ
   :CONFigure:CELLular:COMMon:SPMonitor:DETect RMS
   ```
10. Set the MG3700A output frequency to 999.9 MHz (No.1 in “Table 7.3.4-1 Frequency List”).
11. Set the MG3700A output level to P0 obtained in “Preparation for SA Level Accuracy Measurement”.
12. Set the input level of MU887000A to 0 dBm.

```
:CONFigure:CELLular:MEASurement:RFSettings:LEVel 0
```

13. Measure the level at MU887000A and make this measurement *A* dBm.

    ```
    :INITiate:CELLular:MEASurement:SINGle
    *WAI
    :FETCh:CELLular:COMMon:SPMonitor:POWer
    ```

14. Calculate the level accuracy using the following equation.

    C0 is the value obtained in "Preparation for SA Level Accuracy Measurement".

    Level Accuracy = A – C0

15. Increase the MU887000A input level in 5 dB steps successively such as +5 dBm, +10 dBm, …,+35 dBm and repeat steps 12 thru 14.

16. Set the MG3700A output level to P30 obtained in "Preparation for SA Level Accuracy Measurement".

17. Set the input level of MU887000A to –30 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel -30
    ```

18. Measure the level at MU887000A and make this measurement *A* dBm.

19. Calculate the level accuracy using the following equation.

    C30 is the value obtained in "Preparation for SA Level Accuracy Measurement".

    Level Accuracy = A – C30

20. Increase the MU887000A input level in 5 dB steps successively such as –25 dBm, –20 dBm, …,–5 dBm and repeat steps 17 thru 19.

21. Set the MG3700A output level to P25–10. P25 is obtained in "Preparation for SA Level Accuracy Measurement".

22. Set the input level of MU887000A to –35 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel -35
    ```

23. Measure the level at MU887000A and make this measurement *A* dBm.

24. Calculate the level accuracy using the following equation.

    C25, S, S1 are the values obtained in "Preparation for SA Level Accuracy Measurement".

    Level Accuracy = A – C25 – (S1 – S)

25. Decrease the MU887000A input level in 5 dB steps successively such as –40 dBm, –45 dBm, …,–65 dBm and repeat steps 22 thru 24.

26. For 2 or later number in “Table 7.3.4-1 Frequency List”, repeat steps 9 thru 25.

When testing the Port2, read “Port1” in the procedure of Port1 measurement as “Port2".

7.3.5 SA Linearity

The following test related to the SA Linearity is performed.

(1) Test target specifications

Port1,Port2

Input Level	Specifications
≥–55 dBm	±0.2 dB
≥–65 dBm	±0.4 dB

(2) Measuring instruments

- Signal Generator: MG3700A
- Power meter: ML2438A
- Power sensor: MA2442D
- Signal Analyzer: MS2690A Series or MS2830A
- N-type attenuator (3 dB): MP721A 2 pcs
- Adaptor

(3) Setup

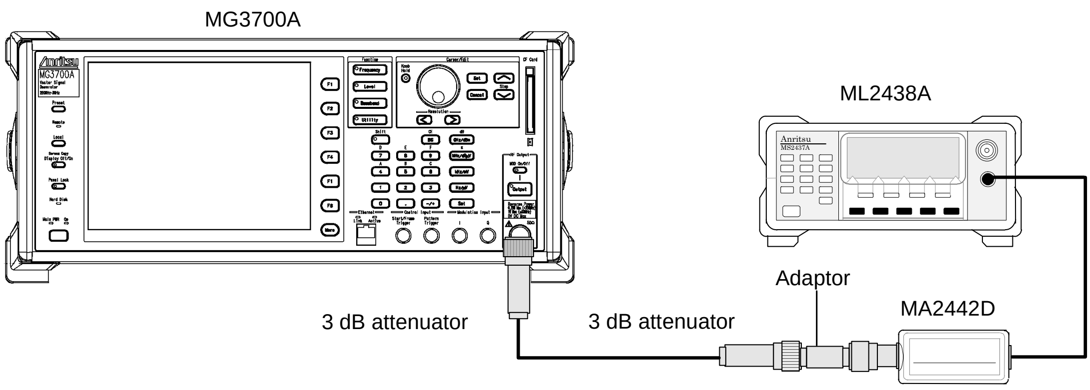


Figure 7.3.5-1 Output Level Setup (0,–25 dBm)

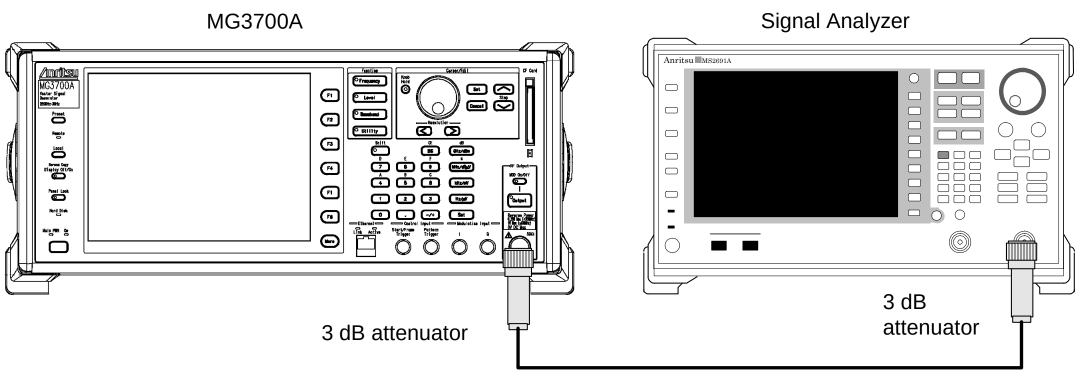


Figure 7.3.5-2 Output Level Setup (0 to –65 dBm)

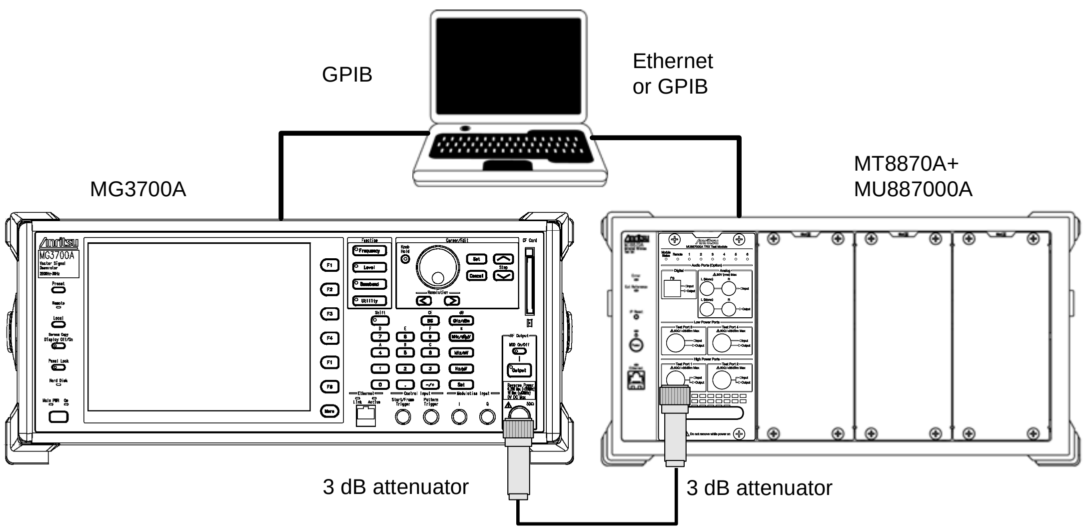


Figure 7.3.5-3 SA Linearity Setup (Port1)

Preparation for SA Linearity Measurement

1. Calibrate the power sensor (zero point and sensitivity).
2. Connect MG3700A and the power meter using the cable with 3 dB attenuator at both edges.
3. Set the MG3700A frequency to 999.9 MHz.
4. Tune the output level of MG3700A so that the power meter displays the value nearest to 0 dBm.

   Make this setting value to MG3700A P0 and the measured value by the power meter C0.
5. Tune the output level of MG3700A so that the power meter displays the value nearest to –25 dBm.

   Make this setting value to MG3700A P25 and the measured value by the power meter C25.
6. Set the MG3700A frequency to the value shown in "Table 7.3.5-1 Frequency List" (1999.9 and 2999.9 MHz), and repeat steps 3 thru 5.
7. Connect MG3700A and the signal analyzer using the cable with 3 dB attenuator at both edges as shown in "Figure 7.3.5-2 Output Level Setup (0 to –65 dBm)".
8. Set the MG3700A frequency to 999.9 MHz.
9. Set the MG3700A output level to P0 obtained in step 4, for outputting the 0 dBm level.
10. Measure the level at the signal analyzer. This value is set to *S*.
11. Set the MG3700A output level to the value (P0–10) which is 10 dB lower than the setting value in the step 9.
12. Measure the level at the signal analyzer. This value is set to *S1*.
13. Set the MG3700A output level to the value (P0–20) which is 20 dB lower than the setting value in the step 9.
14. Measure the level at the signal analyzer. This value is set to *S2*.
15. Set the MG3700A output level to the value (P0–30) which is 30 dB lower than the setting value in the step 9.
16. Measure the level at the signal analyzer. This value is set to *S3*.
17. Set the MG3700A output level to the value (P0–40) which is 40 dB lower than the setting value in the step 9.
18. Measure the level at the signal analyzer. This value is set to *S4*.
19. Set the MG3700A output level to the value (P25) obtained in step 5, for outputting the –25 dBm level, and repeat steps 8 thru 19.

20. For 2 or later number in “Table 7.3.5-1 Frequency List” (1999.9,2999.9), repeat steps 8 thru 19.

Table 7.3.5-1 Frequency List

No.	MU887000A Input Frequency (MHz)	MG3700A Setting Frequency (MHz)
1	1000	999.9
2	2000	1999.9
3	3000	2999.9

(4) Procedure

SA Linearity is measured at Port1 and Port2 only.

Port1

1. To test Port1, set-up the instruments as shown in Figure 7.3.5-3.
2. Initialize the settings of MU887000A.
   `*RST`
3. Set the output and input ports to Port1.
   `:ROUTe:PORT:CONNect:DIRection PORT1,PORT1`
4. Set the VSG operation mode to Normal.
   `:SOURce:GPRF:GENerator:MODE NORMAL`
5. Set the VSG modulation to CW.
   `:SOURce:GPRF:GENerator:BBMode CW`
6. Set the VSG output level to –120 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel -120`
7. Stop outputting the VSG signal.
   `:SOURce:GPRF:GENerator:STATe OFF`
8. Set the measurement function of MU887000A to spectrum monitor.
   ```
   :INSTrument[:SELect] CELLULAR
   :CONFigure:CELLular:MEASurement:STANdard COMMON
   :CONFigure:CELLular:MEASurement:SELect SPMON
   ```
9. Set the spectrum monitor to followings.
   Input Frequency:No.1 frequency in “Table 7.3.5-1 Frequency List”
   Span: 50 MHz
   Power measurement bandwidth: 300 kHz
   RBW: 100 kHz
   Detection: rms
   ```
   :CONFigure:CELLular:MEASurement:RFSettings:FREQuency 1000MHZ
   :CONFigure:CELLular:COMMon:SPMonitor:SPAN 50MHZ
   :CONFigure:CELLular:COMMon:SPMonitor:PMBW 300KHZ
   :CONFigure:CELLular:COMMon:SPMonitor:RBW 100KHZ
   :CONFigure:CELLular:COMMon:SPMonitor:DETect RMS
   ```
10. Set the MG3700A output frequency to 999.9 MHz (No.1 in “Table 7.3.5-1 Frequency List”).
11. Set the MG3700A output level to P0 obtained in “Preparation for SA Linearity Measurement”.
12. Set the input level of MU887000A to 0 dBm.

```
:CONFigure:CELLular:MEASurement:RFSettings:LEVel 0
```

13. Measure the level at MU887000A and make this measurement *A* dBm.

```
:INITiate:CELLular:MEASurement:SINGle
*WAI
:FETCh:CELLular:COMMon:SPMonitor:POWer
```

14. Set the MG3700A output level to the value which is 10 dB lower than the setting value in the step 11.
15. Measure the level at MU887000A and make this measurement *B1* dBm.
16. Calculate the level accuracy using the following equation.

    S, S1 are the values obtained in “Preparation for SA Linearity Measurement”.

    Linearity = (B1 – A) – (S1 – S)
17. Set the MG3700A output level to the value which is 20 dB lower than the setting value in the step 11.
18. Measure the level at MU887000A and make this measurement *B2* dBm.
19. Calculate the level accuracy using the following equation.

    Linearity = (B2 – A) – (S2 – S)
20. Set the MG3700A output level to the value which is 30 dB lower than the setting value in the step 11.
21. Measure the level at MU887000A and make this measurement *B3* dBm.
22. Calculate the level accuracy using the following equation.

    Linearity = (B3 – A) – (S3 – S)
23. Set the MG3700A output level to the value which is 40 dB lower than the setting value in the step 11.
24. Measure the level at MU887000A and make this measurement *B4* dBm.
25. Calculate the level accuracy using the following equation.

    Linearity = (B4 – A) – (S4 – S)
26. Set the MG3700A output level to the value (P25) obtained in “Preparation for SA Linearity Measurement”, for outputting the –25 dBm level, and repeat steps 12 thru 25.
27. For 2 or later number in “Table 7.3.5-1 Frequency List”, repeat steps 9 thru 26.

When testing the Port2, read “Port1” in the procedure of Port1 measurement as “Port2".

7.4 Servicing

If any unit is found to be broken or does not operate as described in the specifications, contact the Anritsu Service and Sales office. Contact information is available in a separate file (for the PDFversion), and on the last page of this manual (for the printed version).

When requesting repair, supply the following information:

(a) Model name and serial number marked on rear panel

(b) Failure symptoms

(c) Person to contact about nature of failure and repair completion notification

(d) Software version

Chapter 8 Utility Tool

This chapter describes how to use the MX887900A Utility Tool.

8.1 Overview

8.1.1 Functions

The MX887900A Utility Tool has the following functions:

Searching for MT8870A

Searches for MT8870A connected via Ethernet or GPIB and displays search results.

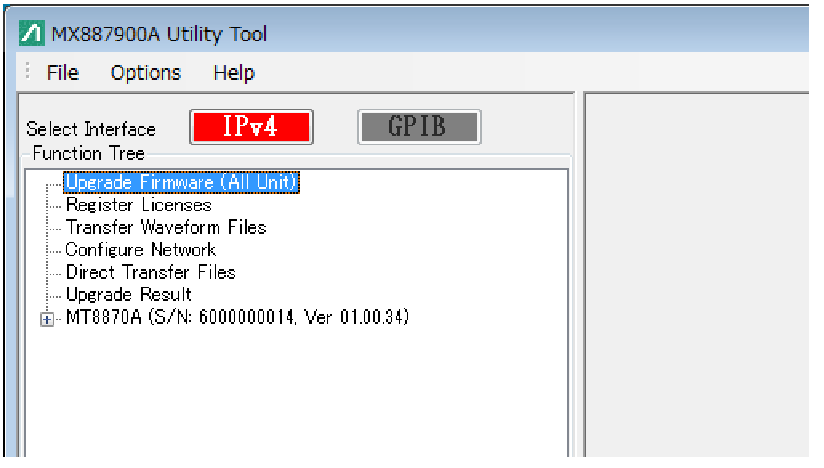


Figure 8.1.1-1 Search Results Display

Configuring network

Specifies network settings, such as module IP address and subnet mask.

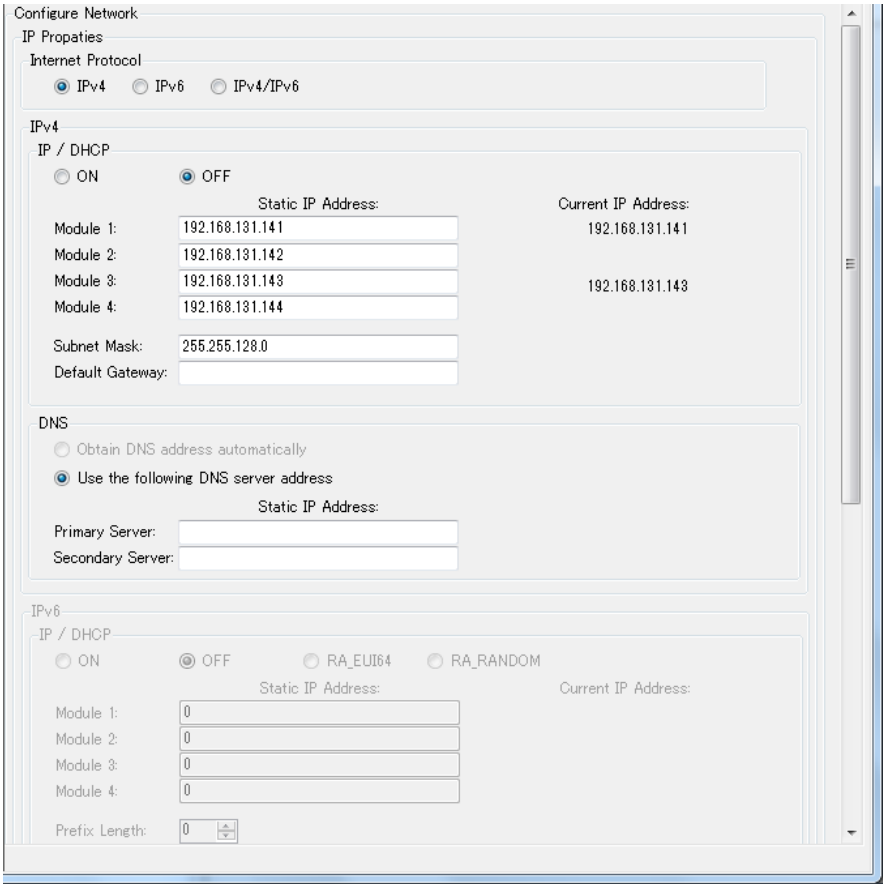


Figure 8.1.1-2 Network Settings Window

8 Utility Tool

Displaying maintenance information

Displays various information, such as self-diagnostic results, run times, power-on count, MT8870A and module calibration results.

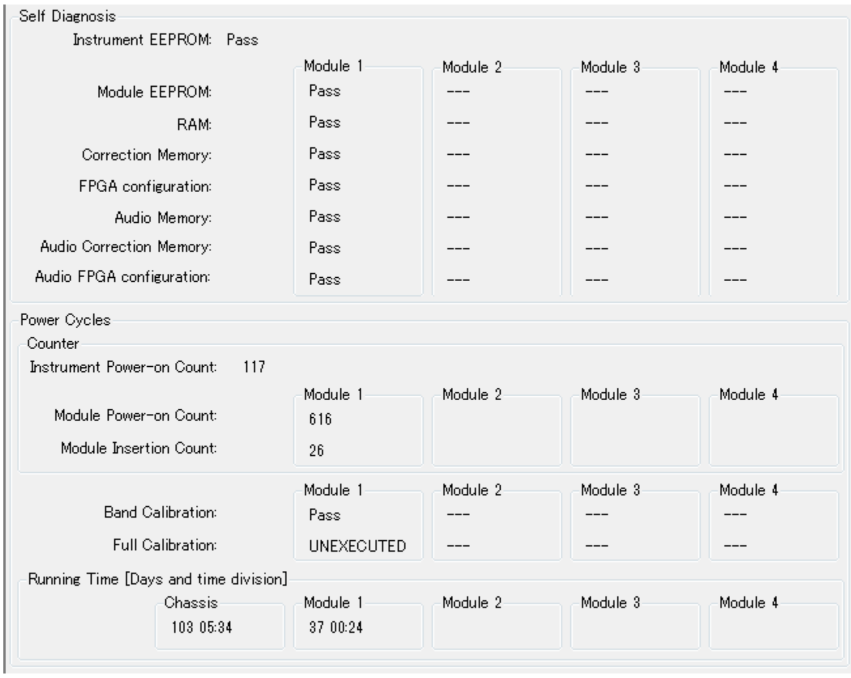


Figure 8.1.1-3 Maintenance Information Window (Self Diagnosis and Run Time)

Figure 8.1.1-4 Maintenance Information Window (Hardware)

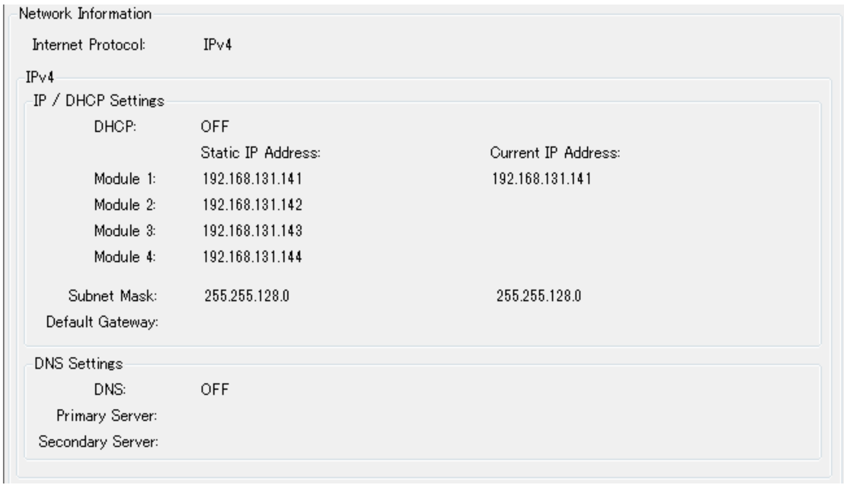


Figure 8.1.1-5 Maintenance Information Window (Communication Settings 1)

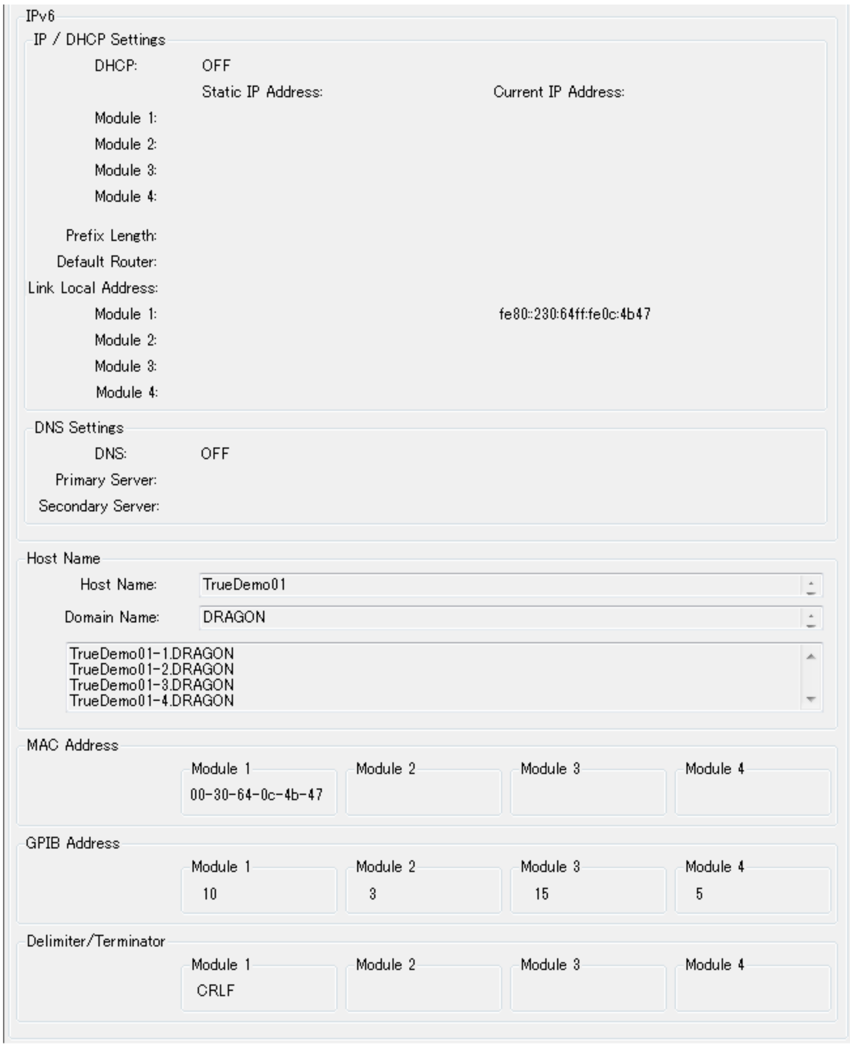


Figure 8.1.1-6 Maintenance Information Window (Communication Settings 2)

8

Utility Tool

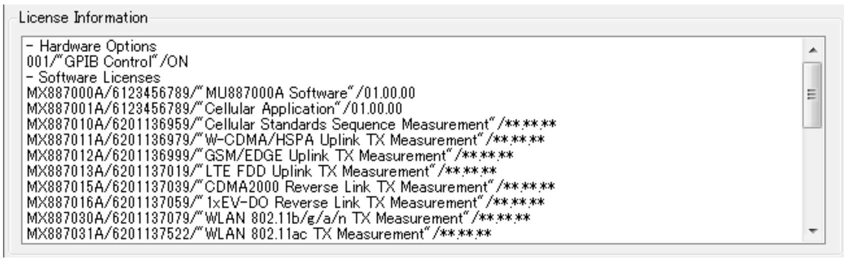


Figure 8.1.1-7 Maintenance Information Window (License and Trigger Settings)

Transferring Files

Transfers files by FTP between PC and module(s).

Upgrading

Upgrades MT8870A and module firmware.

Registering license

Registers MT8870A application software license.

8.1.2 Operating environment

The Utility Tool operates in the following environment:

Table 8.1.2-1 Utility Tool Operating Environment

Item	Specification
OS	Windows XP Professional Service Pack 3 Japanese/English Windows 7 Enterprise Service Pack 1 Japanese/English 32- and 64-bit versions
Display	1024 x 768 or better
Memory	1 GB or more
HDD Free space	200 MB or more
VISA	NI-VISA*1
Interface	Ethernet*2 or GPIB*3

*1: For the version compatibility of OS, .NET Framework, and NI-VISA, refer to Table 8.1.2-2 and Table 8.1.2-3.
To use the Utility Tool of Ver1.1.18 or earlier, installing .NET Framework 4.0 Languages Support is required.

*2: The Utility Tool of Ver1.1.18 or earlier does not operate properly when proxy settings of the network are configured for the FTP Server. Refer to Section 8.2.4 "Setting proxy exceptions" to set the IP address of the MT8870A to exception.

*3: File transfer, software upgrade, and license registration cannot be executed via GPIB.

Table 8.1.2-2 Compatibility Between OS and .NET Framework

OS	.NET Framework 4.0	.NET Framework 4.5
Windows XP	✓	–
Windows 7	✓	✓

✓: Compatible, –: Incompatible

Table 8.1.2-3 Compatibility Between .NET Framework and NI-VISA

NET Framework	NI-VISA 5.0.3 to 5.2	NI-VISA 5.3 to 14.0
.NET Framework 4.0	✓	✓
.NET Framework 4.5	–	✓

✓: Compatible, –: Incompatible

Table 8.1.2-4 NI-VISA Version

Utility Tool Version	NI-VISA Version
Ver 1.1.10 or earlier	Version 5.03 or later, Version 5.2 or earlier
Ver 1.1.17 or later, Ver 1.1.18 or earlier	Version 5.03 or later, Version 5.4 or earlier
Ver 1.2.2 or later	Version 5.03 or later, Version 14.0 or earlier

The table below shows the version compatibility between the Utility Tool and the module. The module versions are displayed in Tree View in Figure 8.3.2-1.

Table 8.1.2-5 Compatible Versions

Utility Tool Version	Module Version
Ver 1.1.0 or earlier	Ver 1.3.16 or earlier
Ver 1.1.7 or later	All versions

8.2 Installing

8.2.1 .NET Framework

Refer to Table 8.1.2-2 “Compatibility Between OS and .NET Framework” to install a version of .NET Framework that is compatible with the OS.

8.2.1.1 .NET Framework 4.0

Installing .NET Framework 4.0

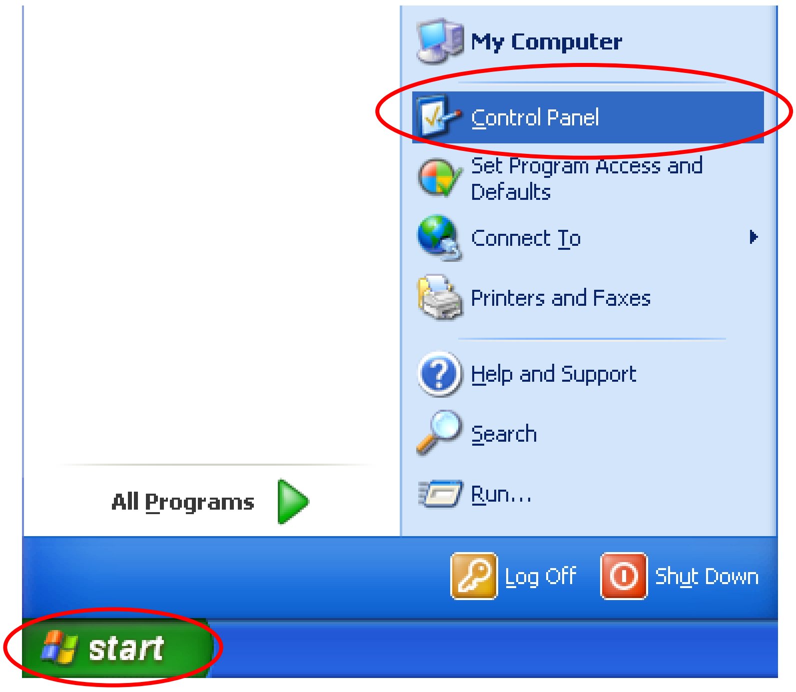
1. Download a full version (not the Client Profile version) of .NET Framework 4.0 from the Microsoft web site.

   http://www.microsoft.com/en-us/download/details.aspx?id=17851

2. Double-click “dotNetFx40_Full_setup.exe” and follow the instructions.

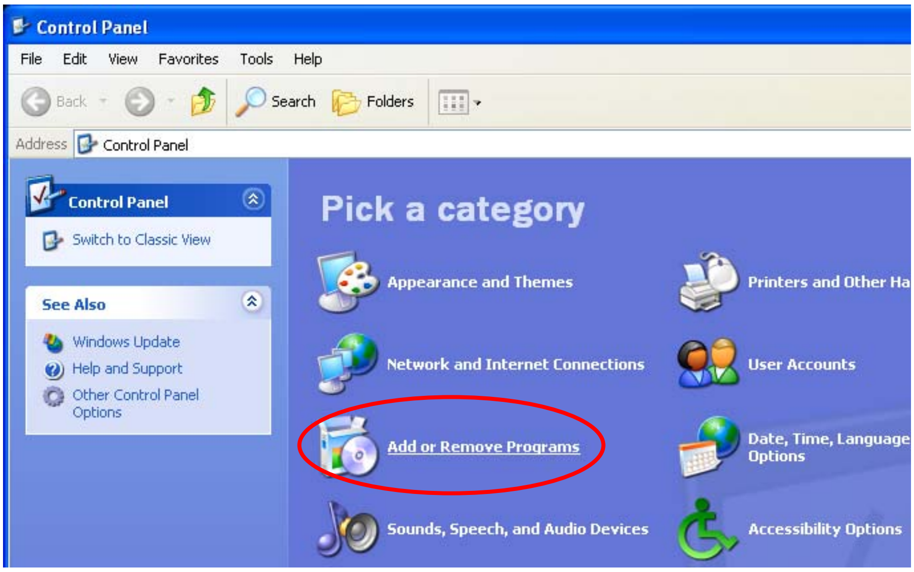
Confirming the installation of .NET Framework 4.0

1. Click **Start** and then **Control Panel**.

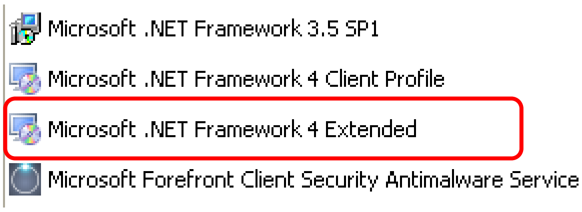


2. Click **Add or Remove Programs**. (Windows XP)

   Click **Uninstall a program** or **Programs and Features**. (Windows 7)



3. Confirm that "Microsoft .NET Framework 4 Extended" is in the list.



When you can find only "Microsoft .NET Framework 4 Client Profile" in the list, overwrite ".NET Framework 4.0 Full Set" to install.

8

Utility Tool

8.2.1.2 .NET Framework 4.5

Installing .NET Framework 4.5

1. Download .NET Framework 4.5 from the Microsoft web site.

   http://www.microsoft.com/en-us/download/details.aspx?id=30653

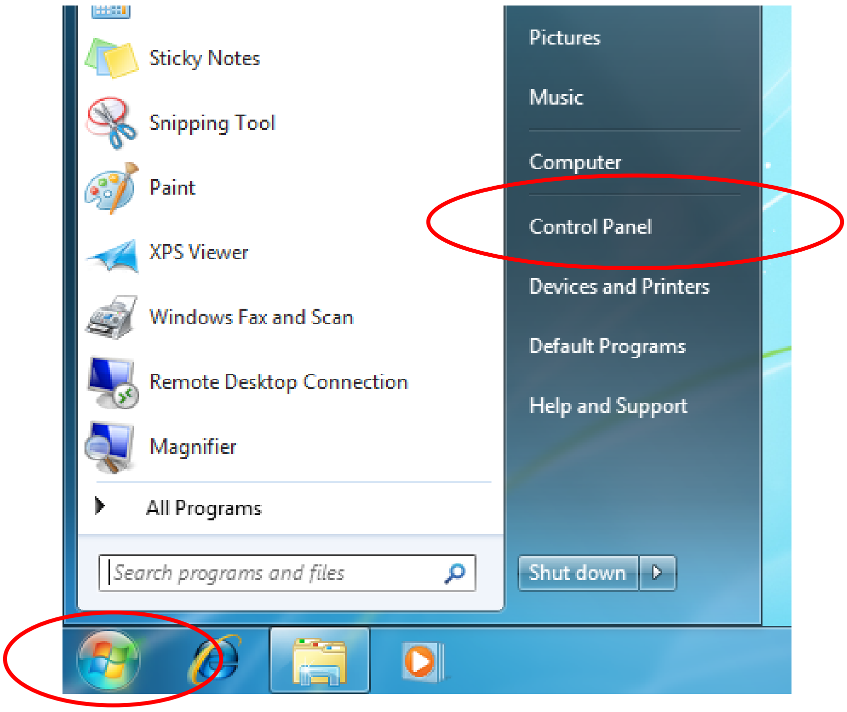
2. Double-click the downloaded execution file. Start installation following the on-screen instructions.

Note:

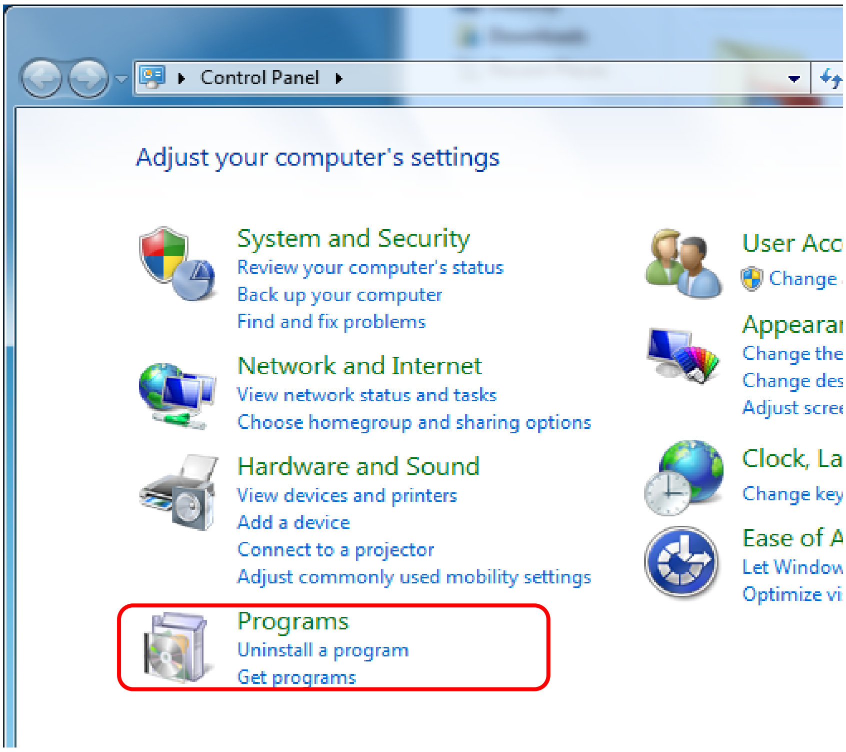
.NET Framework 4.0 is removed when .NET Framework 4.5 is installed.

Confirming installation of .NET Framework 4.5

1. Click **Start** and then **Control Panel**.



2. Click **Uninstall a Program** or **Programs.**



3. Confirm that "Microsoft .NET Framework 4.5" is in the list.

8

Utility Tool

8.2.2 NI-VISA

Installing NI-VISA

1. Download NI-VISA from the National Instruments web site.
2. Double-click the downloaded file and start the install following the on-screen instructions.
3. To use the Utility Tool of Ver1.1.18 or earlier, select .NET Framework 4.0 on the function selection screen regardless of the version of the installed .NET Framework.
   Click **.NET Framework4.0 Languages Support** in the Features list, and then select **Install this feature to a local drive**.
4. Restart the PC when the NI-VISA is successfully installed.

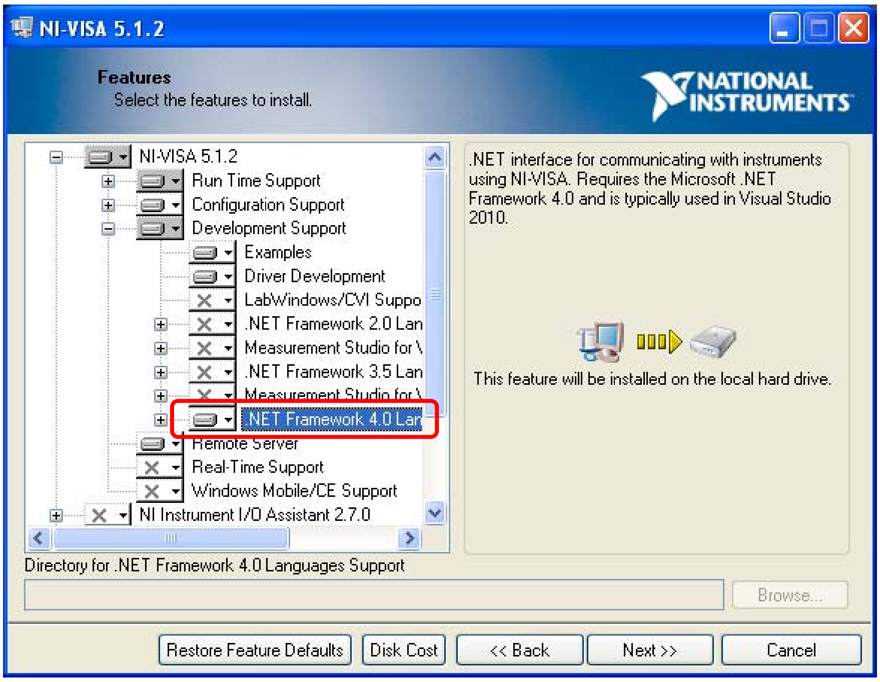
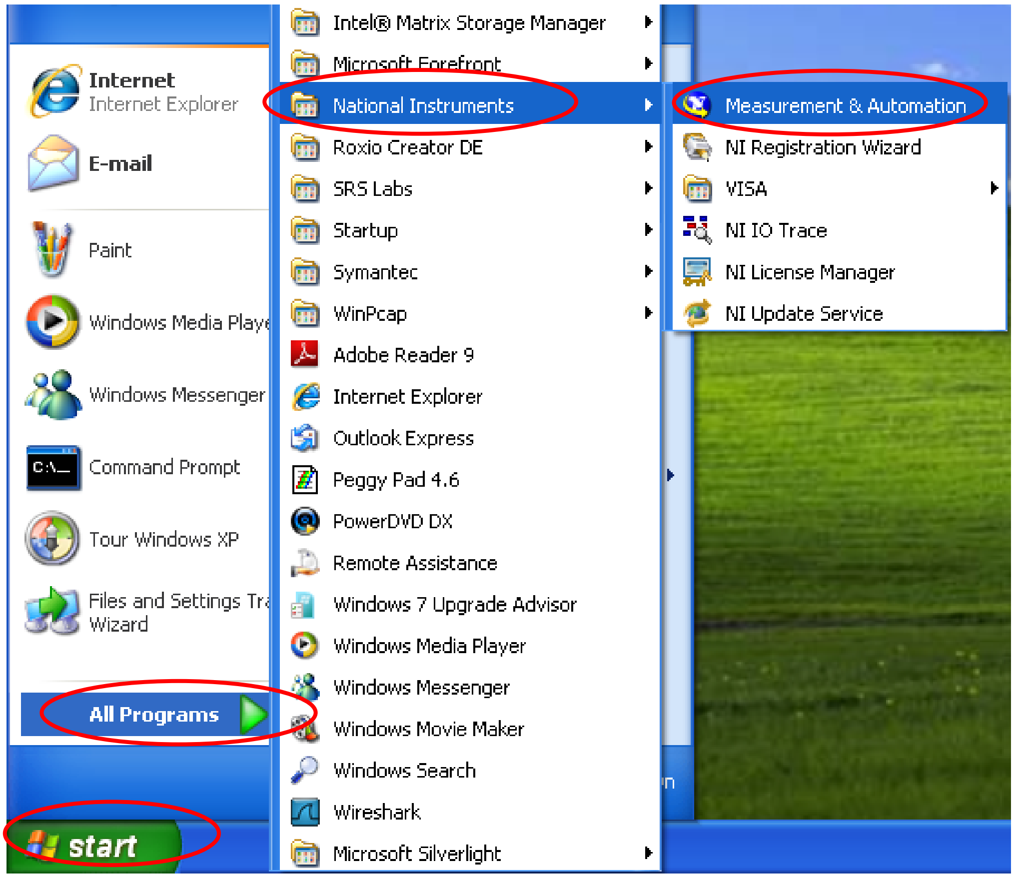


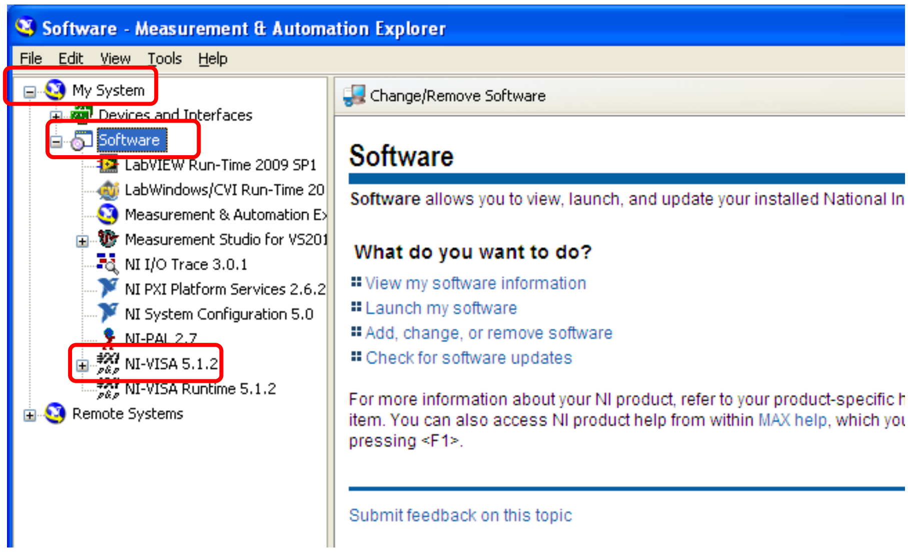
Figure 8.2.2-1 .NET FrameWork4.0 Languages Support Selection Window

Confirming the software version of NI-VISA

1. Click **Start**, **All Program**, **National Instruments**, and then **Measurement & Automation**.



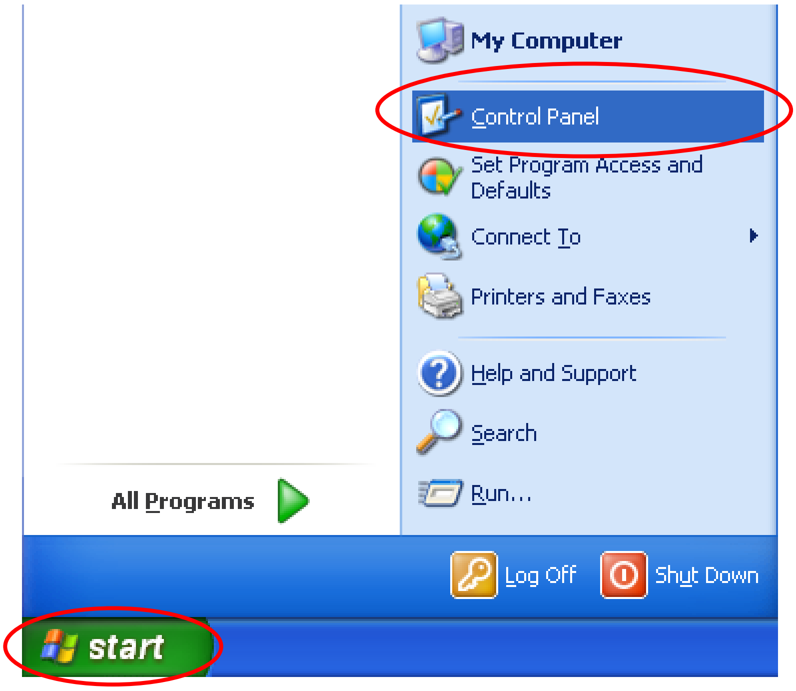
2. Click **My System**, and then **Software**. The versions of the software products are displayed under **Software**.



3. Confirm that the version of NI-VISA is 5.0.3 or later.

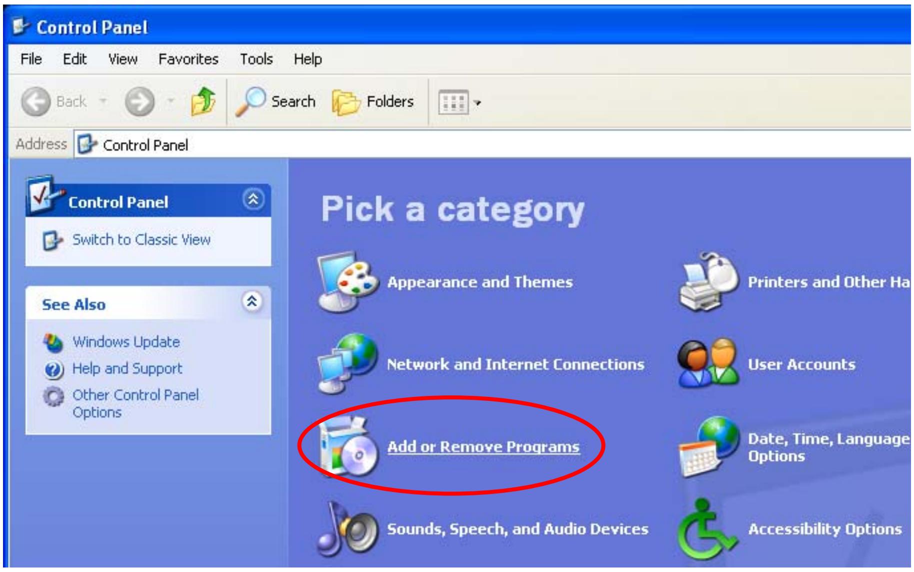
Confirming .NET Framework 4.0 Languages Support of NI-VISA

1. Click **Start** and then **Control Panel**.



2. Click **Add or Remove Programs**. (Windows XP)

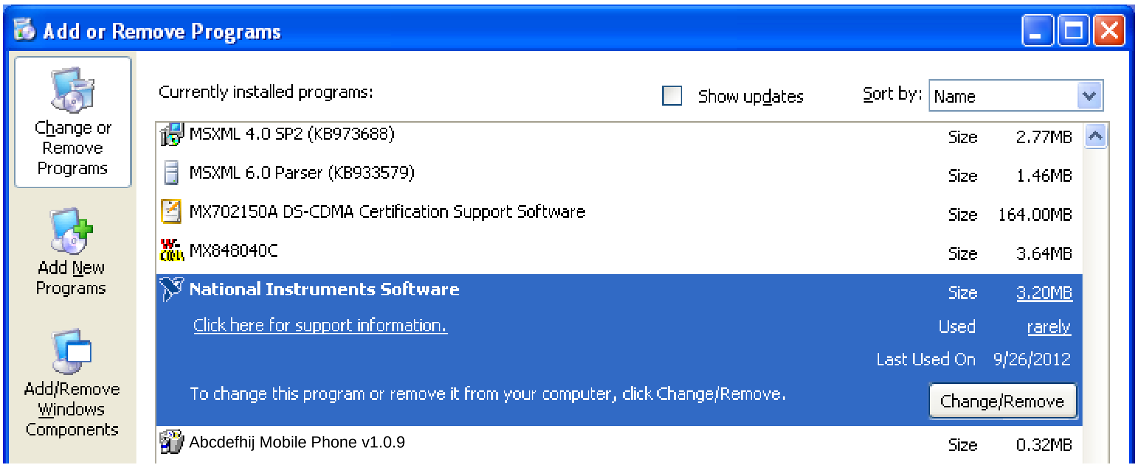
   Click **Uninstall a program** or **Programs and Features**. (Windows 7)

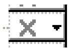


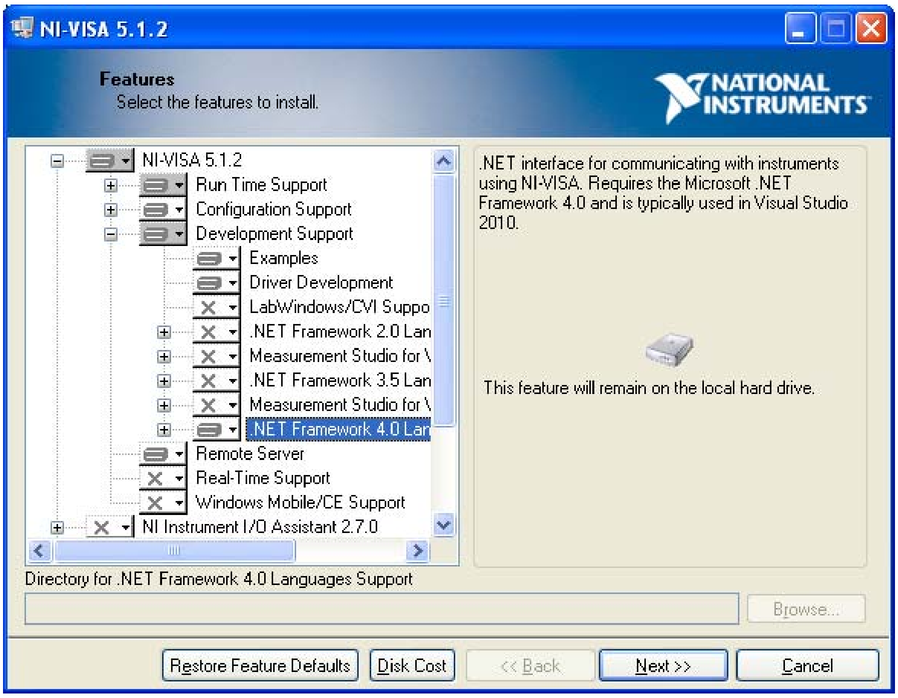
8

Utility Tool

3. Select **National Instruments Software** and then click **Change/Remove**.



4. Confirm that .NET Framework 4.0 Languages Support is installed in the NI-VISA software list. If installed, the icon is other than .



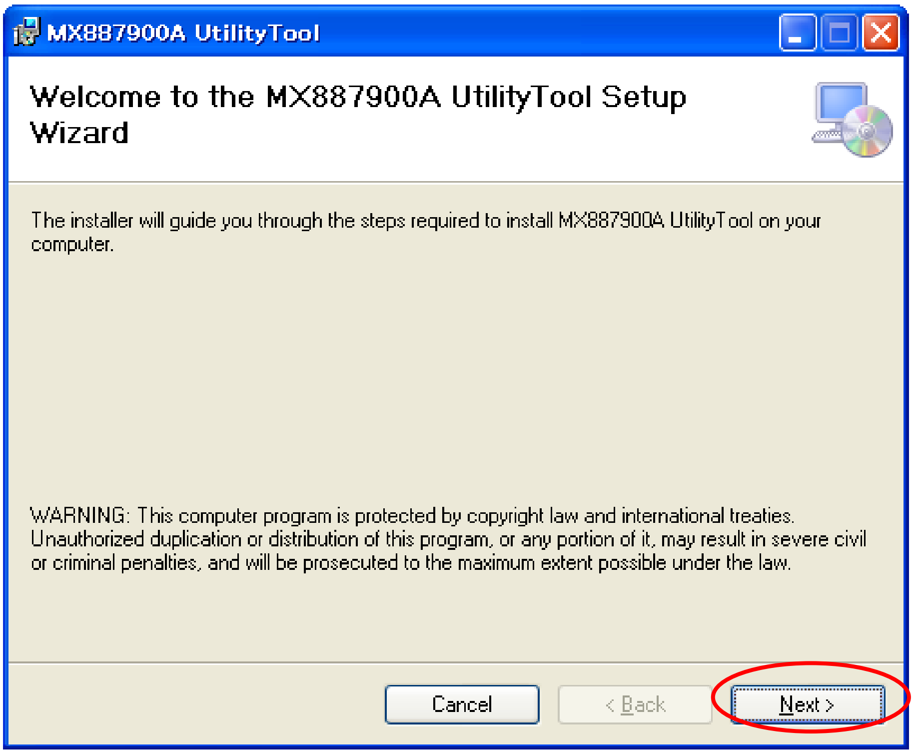
8.2.3 Utility Tool

Installing Utility Tool

1. Double-click the setup program “setup.exe” on the storage media supplied with the MT8870A.

   The setup program is stored in \Installer\MT8870A_Utility Tool.

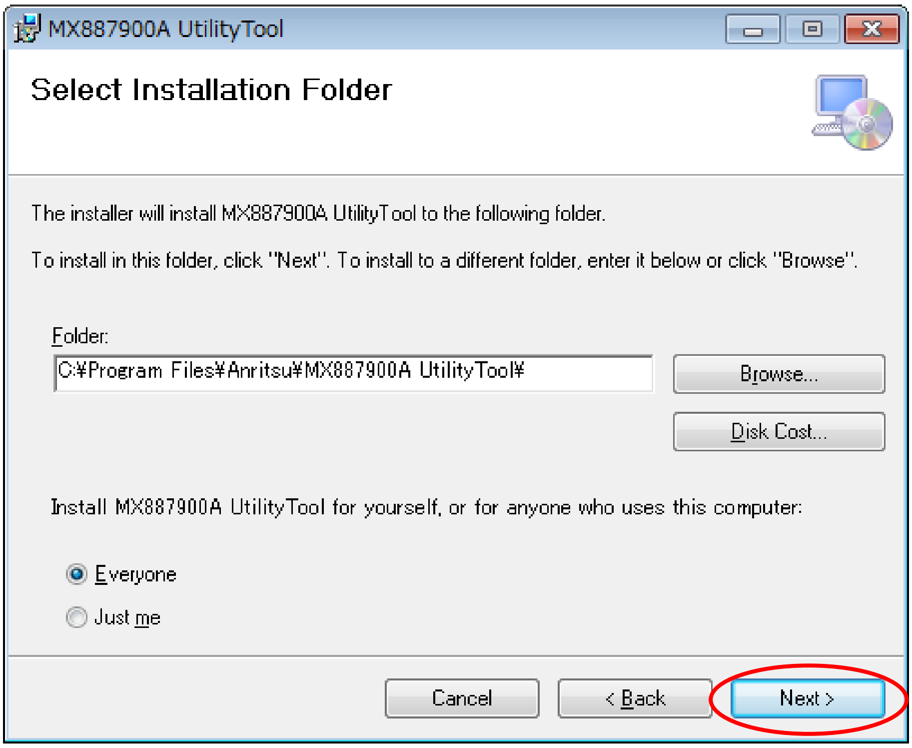
2. Click **Next** in the wizard.



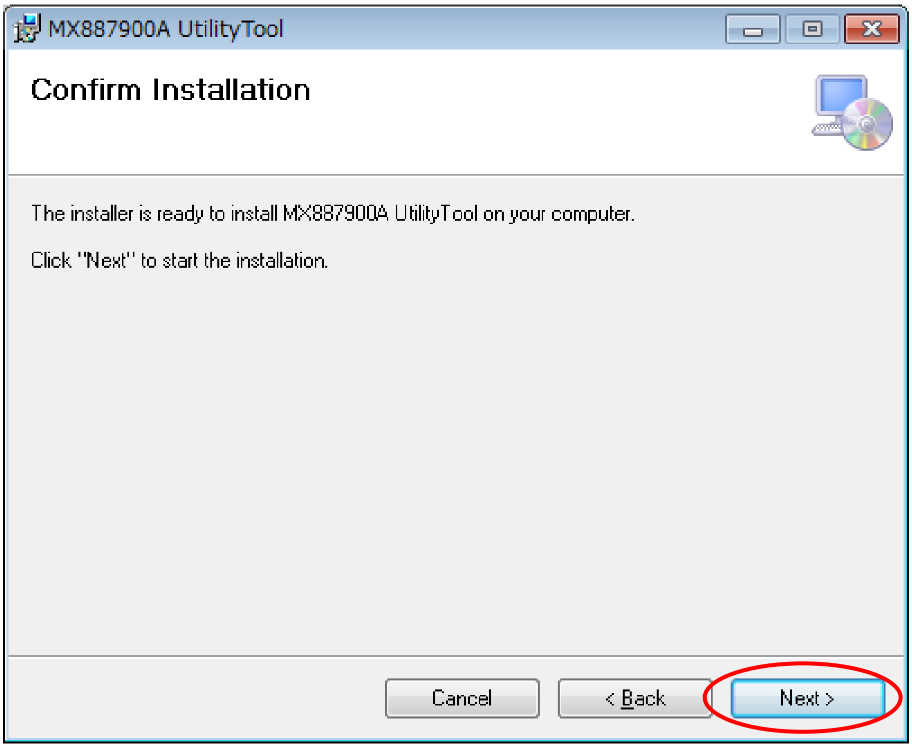
8

Utility Tool

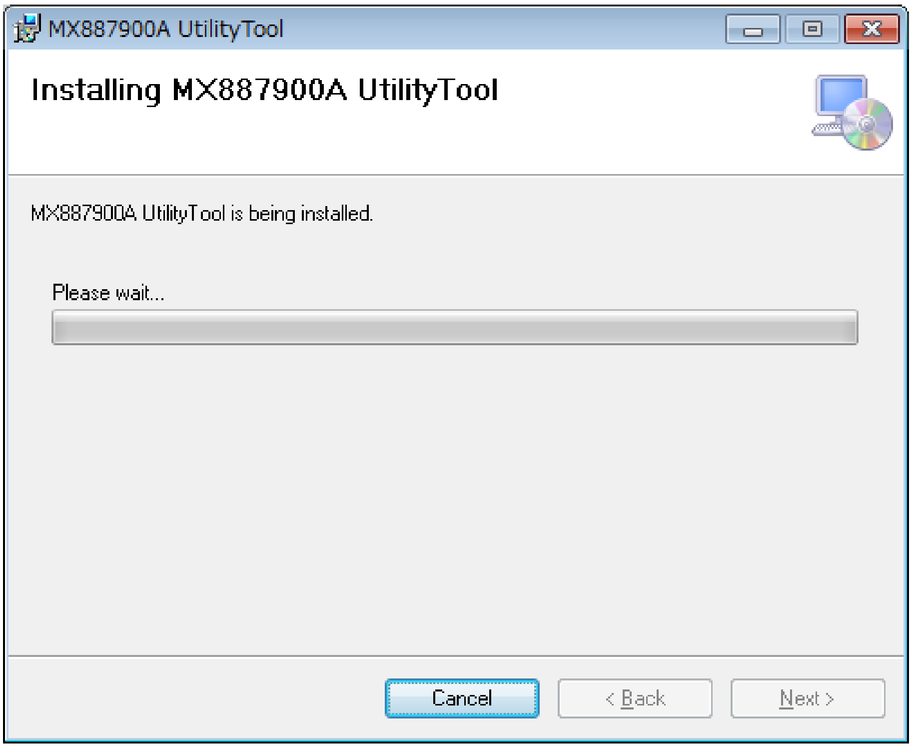
3. Specify the full path to the installation folder, and then click **Next**.



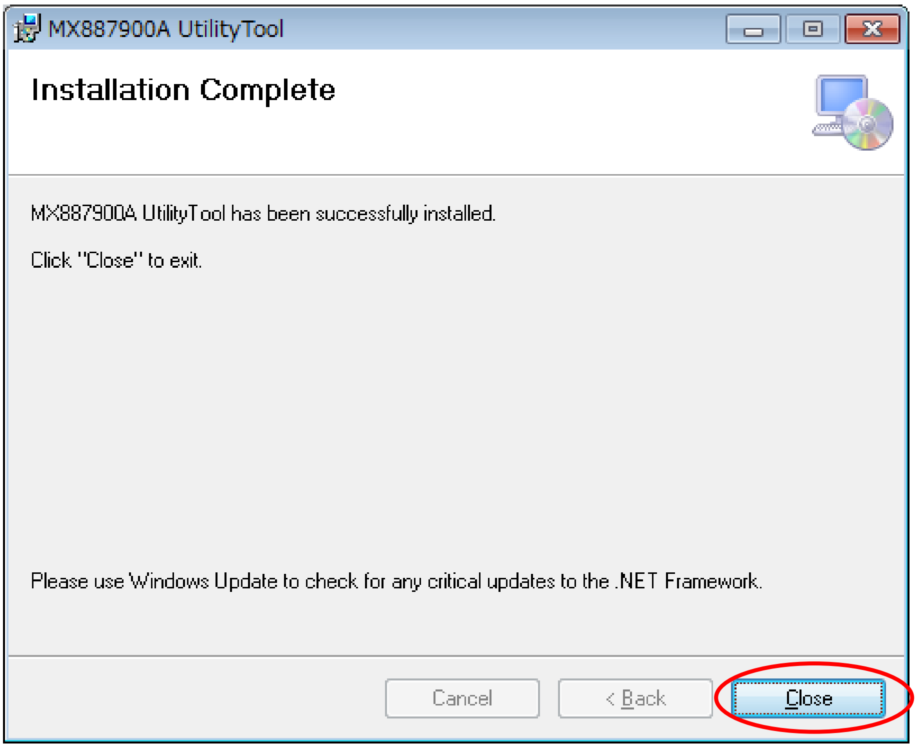
4. Click **Next** when the following message appears.



5. Wait until the installation completes. A progress bar is displayed during installation. To abort the installation, click **Cancel**.



6. Click **Close** to exit the wizard after completing installation.



Uninstalling Utility Tool

1. In Control Panel, click **Uninstall a Program**.
2. Double-click the MX887900A Utility Tool in the program list to start the uninstallation process. Follow the on-screen instructions.

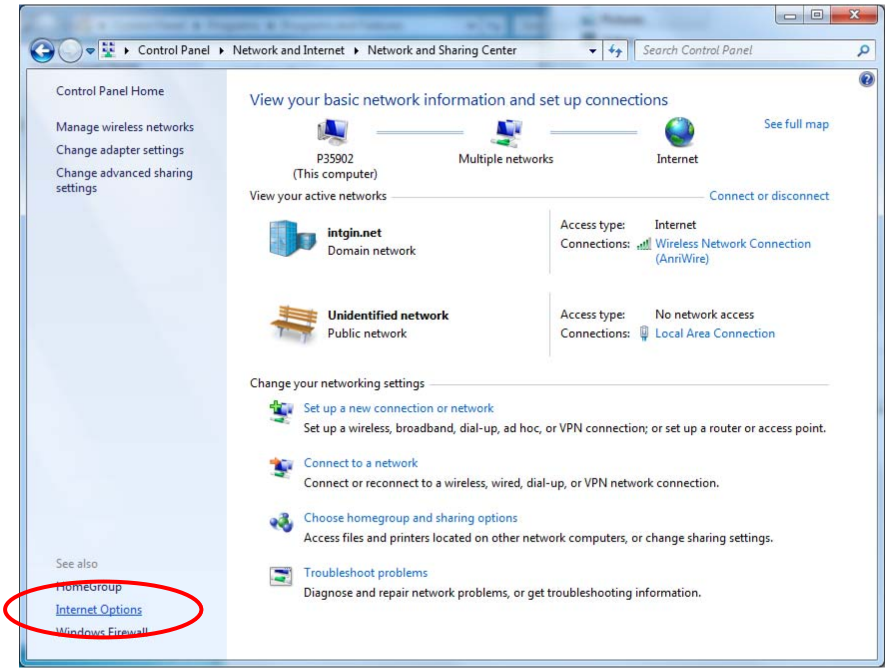
8.2.4 Setting proxy exceptions

This setting is required only for using the Utility Tool of Ver1.1.18 or earlier.

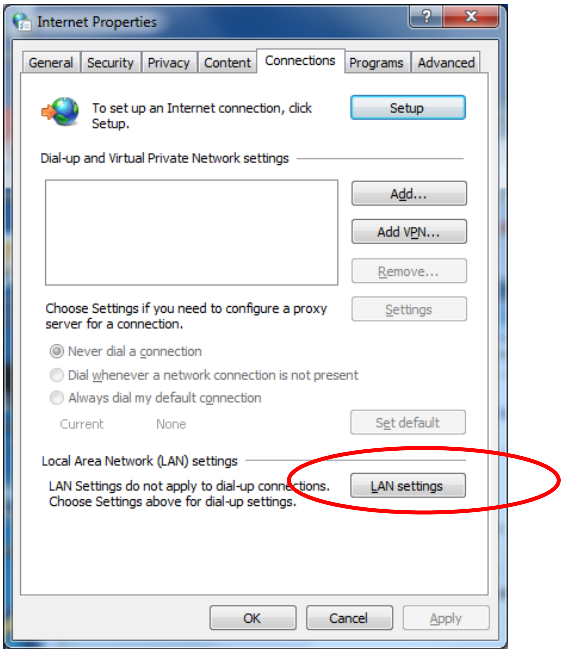
Steps to set proxy exceptions

Perform the following steps to set exception addresses when proxy settings are configured for the FTP Server in the proxy settings of the network.

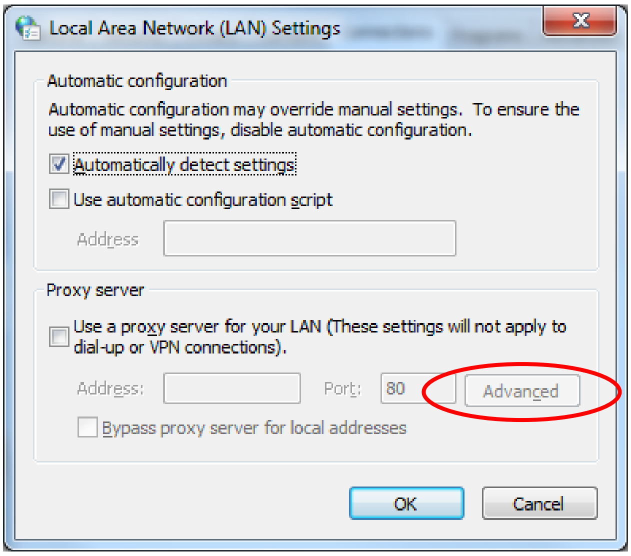
1. In Control Panel, start Network and Sharing Center, and then click **Internet Options**.



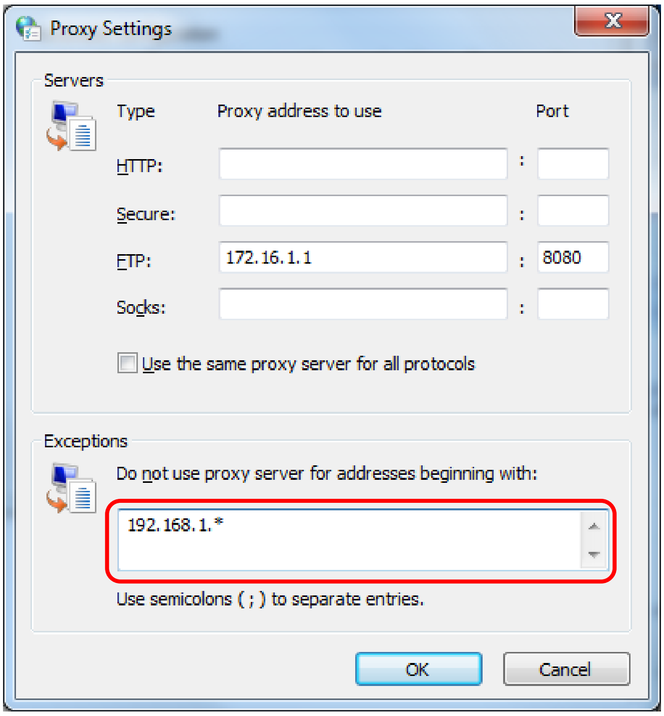
2. In the **Internet Properties** dialog box, click the **Connections** tab, and then click **LAN settings**.



3. In the **Local Area Network (LAN) Settings** dialog box, click **Advanced**.



4. In the **Proxy Settings** dialog box, type the network address of the MT8870A in the **Exceptions** box.



8.3 Operating

In this section, user interface elements are written in bold

For example, “File” in Figure 8.3.1-1 is written as **File**.

8.3.1 Starting and quitting

Starting

1. Open Anritsu in **All Programs** at the Start menu.
   Double-click the MX887900A Utility Tool.
2. Click the button for the interface (**IPv4** or **GPIB**) used to communicate with the MT8870A. The module search message is displayed.
3. The tree view display is updated when the module is detected.

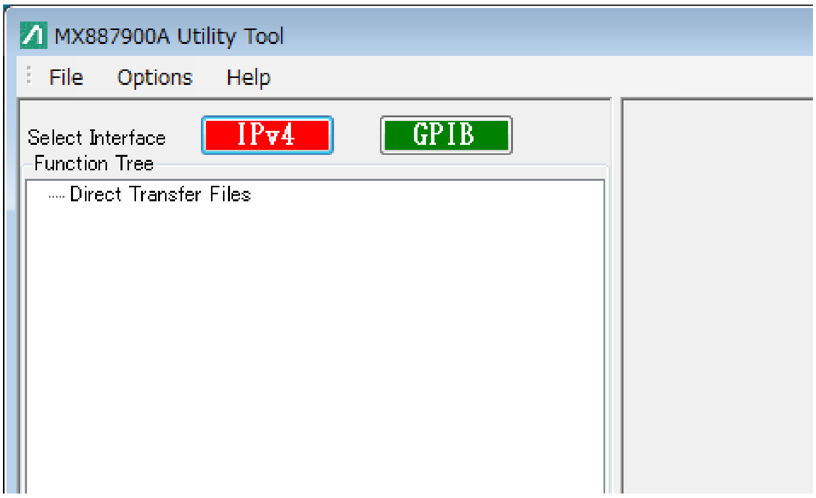


Figure 8.3.1-1 Utility Tool Initial Screen

Quitting

1. Click **File** and then click **Exit**.
2. Click **Yes** in the confirmation dialog box.

8.3.2 Utility Tool screen elements

The screen elements are shown below.

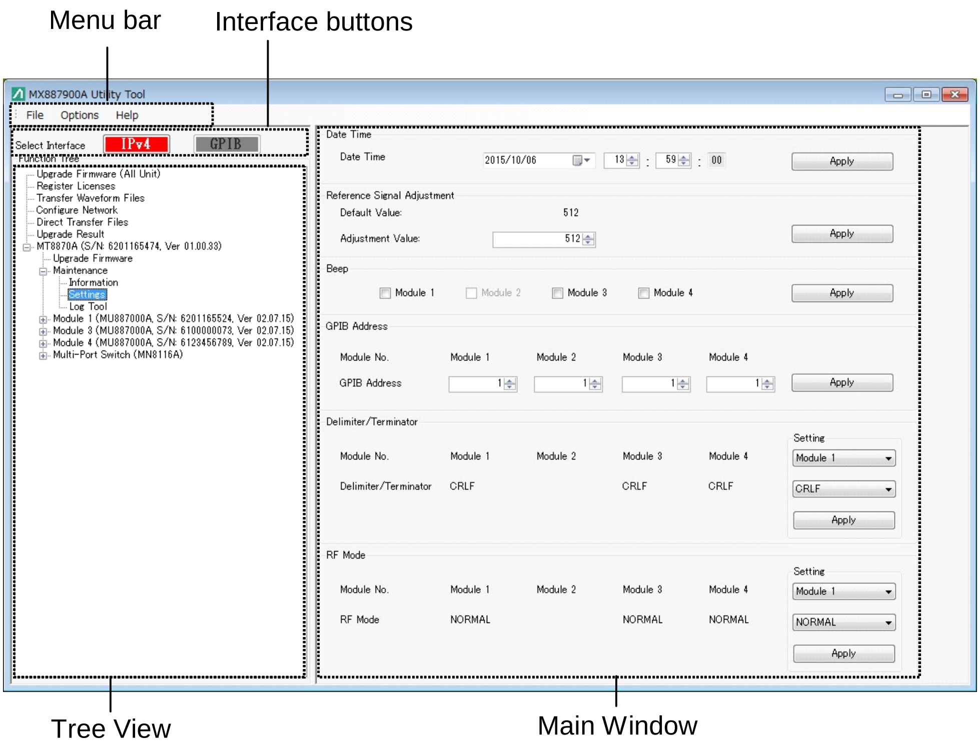


Figure 8.3.2-1 Utility Tool Screen Elements

Table 8.3.2-1 Screen Description

<table>
<tr><th>Name</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">Menu bar</td><td colspan="2">File
Exit: Quits Utility Tool.</td></tr>
<tr><td colspan="2">Options
FIRMUPDATE option
DIFF
Upgrades only when the versions of the built-in program and of the program to load do not match.
FORCE
Upgrades forcibly even when the versions of the built-in program and of the program to load match.</td></tr>
<tr><td colspan="2">Help: Displays version information.</td></tr>
<tr><td rowspan="2">Interface buttons</td><td>IPv4</td><td rowspan="2">Selects interface for communications with MT8870A.</td></tr>
<tr><td>GPIB</td></tr>
<tr><td>Tree view</td><td colspan="2">Displays detected MT8870A(s).
“Module” is followed by a number (1 to 4) representing the slot number of MT8870A’s slot in which the module is installed.
Module 1 to Module 4 are followed by model name, serial number, and package version.
Main window contents are changed by clicking a different tree view item.</td></tr>
<tr><td>Main window</td><td colspan="2">Displays the information and status of MT8870A/module and contains command buttons.</td></tr>
</table>

8.3.3 Searching for devices

This section describes how to search for the MT8870A.

1. Click the **IPv4** or **GPIB** button to select the communications interface.

   When clicking **IPv4**, select a desired search method in the window below.

   - When “Manually enter instrument IP address” is not selected, all MT8870As on the selected network will be searched.
   - When “Manually enter instrument IP address" is selected, the MT8870A equipped with the module of the input IP address will be searched.

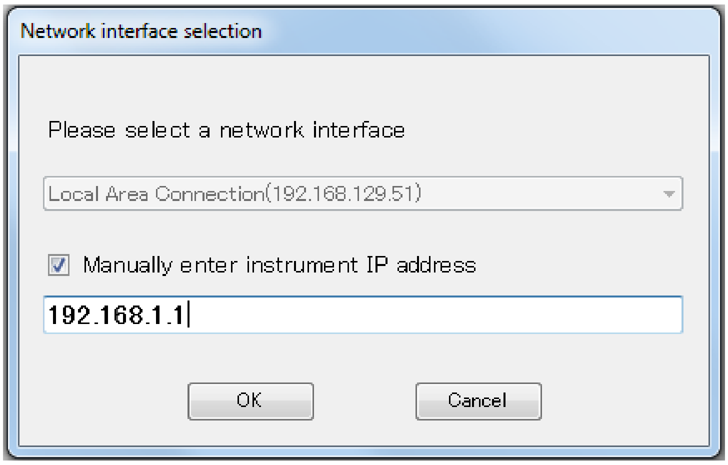


Figure 8.3.3-1 Network Interface Selection Dialog Box

8 Utility Tool

2. When the MT8870A is detected, the model and serial number are displayed in the tree view.

   For the IP4 interface, the tree view displays:

   Update Firmware (All Unit)
   Register Licenses
   Transfer Waveform Files
   Configure Network
   Direct Transfer files
   Update Result

Notes:

- Searching for devices takes a few minutes.
- If no module is installed in the MT8870A, search results will not be displayed in the tree view.

8.3.4 Viewing information on devices

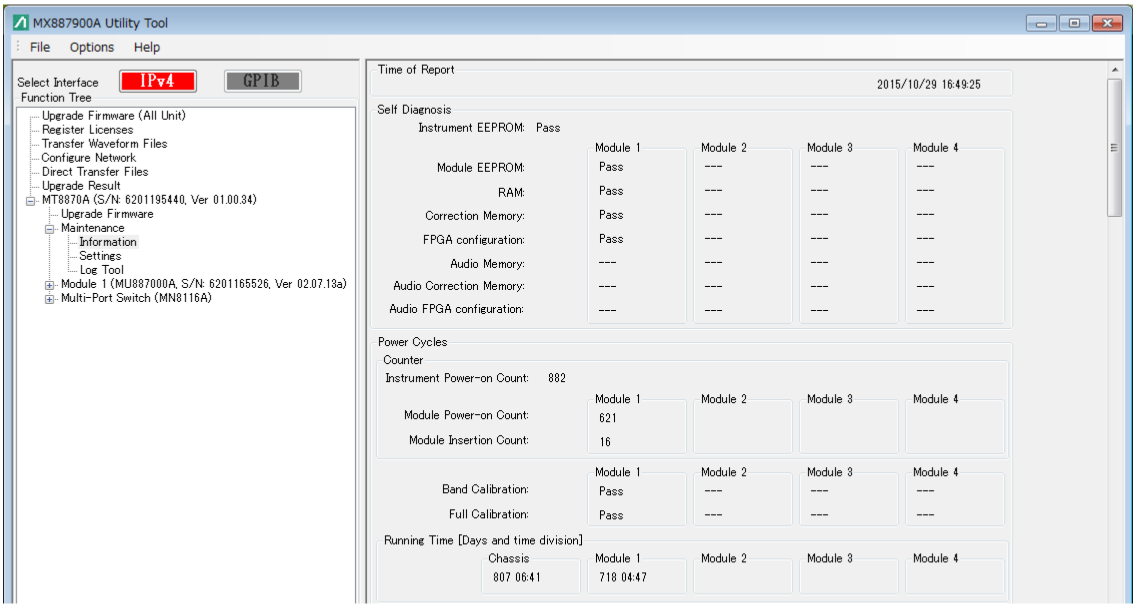
This section describes how to display the Information on the MT8870A and modules.

1. Click [+] to expand MT8870A in the tree view.
2. Click [+] to expand **Maintenance**.
3. Click **Information** to display information on the MT8870A and modules.

Saving displayed information

1. Click **Export** at the bottom of the window to display the **Export Log File** dialog box.
2. Input folder and file names and click **OK** to save the displayed information to a file.

The top of the Maintenance Information window



The bottom of the Maintenance Information window

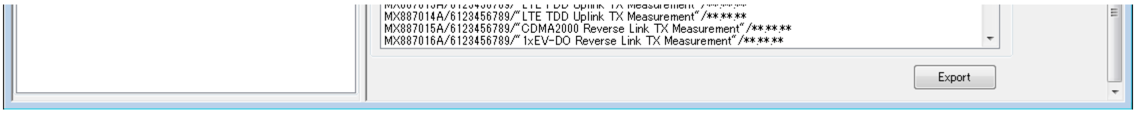


Figure 8.3.4-1 Maintenance Information Window

8

Utility Tool

Table 8.3.4-1 Items of Maintenance Information

Item	Description
Time of Report	The date and time when the maintenance information was obtained. (DD HH MM)
Self-Diagnosis	Self-diagnostic results
Power Cycles	
Counter	Instrument Power-on Count: Number of times MT8870A powered-on Module Power-on Count: Number of times module powered-on Module Insertion Count: Number of times module inserted
Band Calibration	Result of band calibration
Full Calibration	Result of full calibration
Running Time	Time (DD HH MM)
Information	Chassis: Model name, serial number and version Module 1 to Module 4: Model names, serial numbers, versions, application software, and RF Mode (Ver. 1.3.1 or later of the Utility Tool) of modules in slots 1 to 4
Date Time	Time of the MT8870A when getting information. (DD HH MM)
Network Information	
Internet Protocol	IP version
IPv4	DHCP, DNS settings
IPv6	DHCP, DNS settings
Host Name	Host and domain names
MAC Address	MAC addresses of modules in slots 1 to 4
GPIB Address	GPIB addresses of modules in slots 1 to 4
Delimiter/Terminator	Terminator of modules in slots 1 to 4
Reference Signal	MT8870A reference signal
Reference Signal Adjustment	MT8870A reference signal level adjustment
Trigger Settings	Input/Output of MT8870A Trigger connectors
Temperature [degrees]	Temperature of MT8870A and modules (°C)
License Information	License information

8.3.5 Network settings

This section describes how to configure network communication settings. For how to configure GPIB communication settings, refer to 8.3.6 “Setting modules”.

Editing Settings

1. Click **Configure Network** in the tree view to open the Configure Network window.
2. In the tree view, click and select a module you want to configure.
3. Configure settings in the main window. The previous settings can be restored by clicking **Reload**. **Reload** is located at the bottom of the main window.
4. When setting network information for multiple MT8870A units, repeat steps 1 to 3.
5. Click **Apply** to apply changes to the network settings of all modules and close the Network Settings window. To restore the previous settings of all modules, click **Cancel**.
6. As MT8870A reboots automatically when network settings are applied, wait until MT8870A restarts successfully, and then click the **Interface** button to update the tree view.

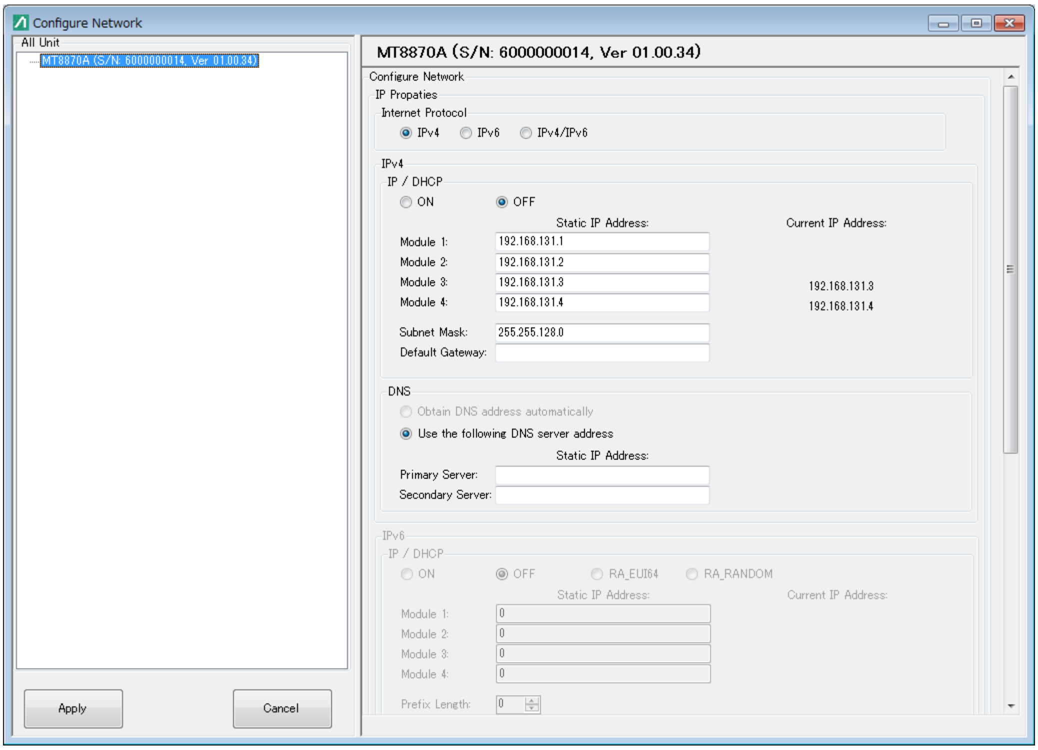


Figure 8.3.5-1 Network Settings Window 1

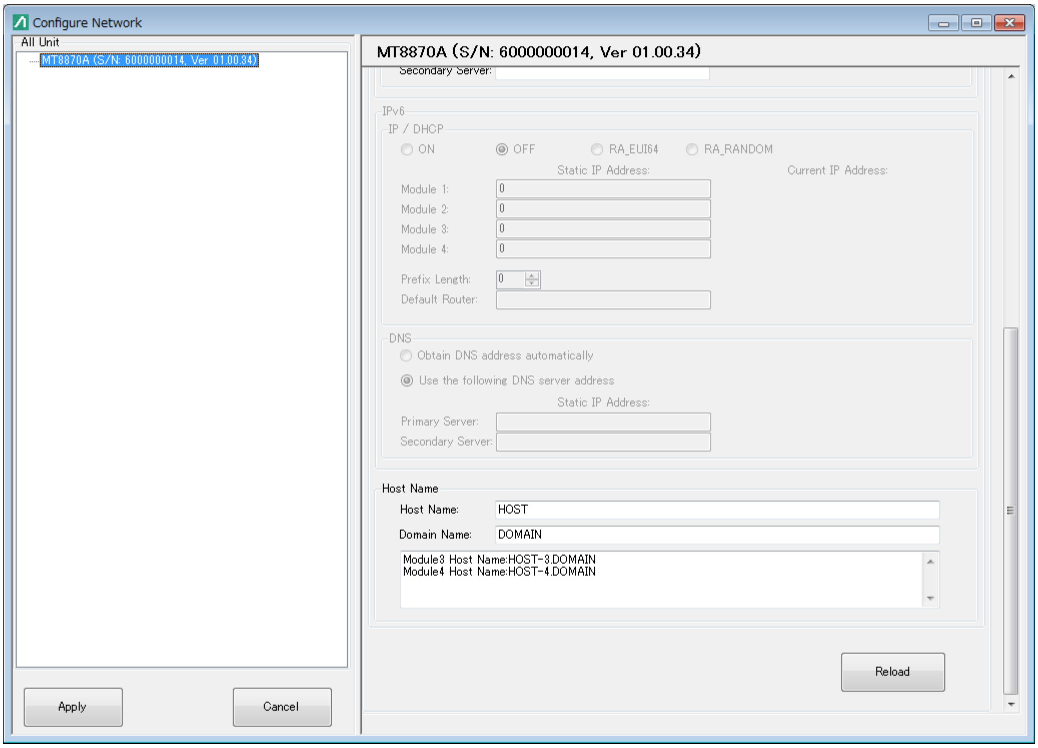


Figure 8.3.5-2 Network Settings Window 2

Table 8.3.5-1 Network Setting Items

Item	Description
IP Properties	
Internet Protocol	IP version
IPv4 DHCP DNS	 IP address auto-setting On/Off Uses set Static IP address when OFF set DNS server address acquisition method Setting Use the following DNS server address uses the Static IP address setting
IPv6 DHCP DNS	 IP address auto-setting On/Off Uses set Static IP address when OFF set DNS server address acquisition method Setting Use the following DNS server address uses the Static IP address setting
Host Name	Host and domain names Displays settings for modules in slots 1 to 4

8.3.6 Setting modules

The time/date, adjustment value of internal reference signal frequency and beeps of modules can be set.

1. Click [+] to expand MT8870A in the tree view.
2. Click [+] to expand **Maintenance**.
3. Click **Settings** to display the module settings in the main window.
4. To set the date and time, input values at Date Time and click **Apply**.
5. To adjust the internal reference signal frequency, input a value at Adjustment value of Reference Signal Adjustment and click **Apply**.
6. To set beeps, select the modules under Beep and click **Apply**.
7. To set RF mode, select a module, select **NORMAL** or **RFSEMAPHORE**, and then click **Apply**. You can set only a single module at a time. This function is available in the Utility Tool Ver. 1.3.1 or later.
8. Set the GPIB addresses and delimiter/terminator according to the following procedure. The procedure depends on the version of the built-in program, as follows:

   For Ver 01.03.16 or later versions

   (a) To set the GPIB addresses, enter the address in each Module box and click **Apply**. You can set all the module addresses at a time. Clicking **Apply** restarts MT8870A.

   (b) To set the delimiter/terminator, select a module and set its delimiter/terminator, then click **Apply**. You can set only a single module at a time.

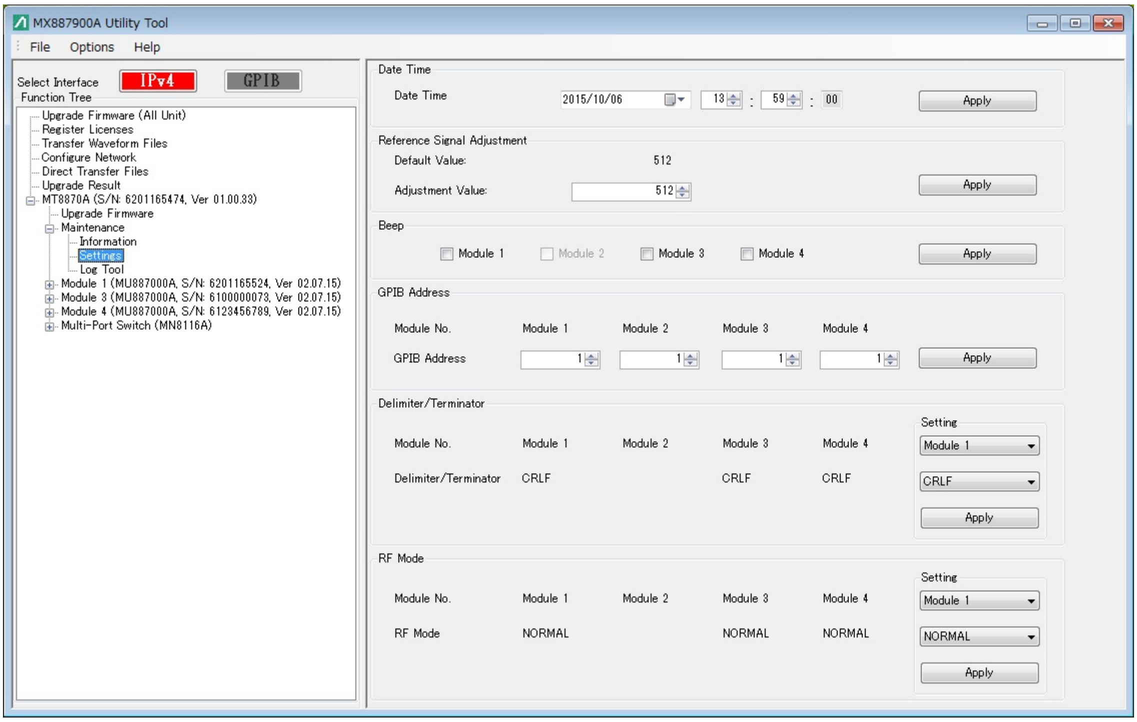


Figure 8.3.6-1 Module Settings Window (Ver 01.03.16 or later)

For versions earlier than Ver 01.03.16

(a) To set the GPIB address and delimiter/terminator, select a module.

(b) Set its GPIB address and delimiter/terminator and click **Apply**. You can set only a single module at a time.

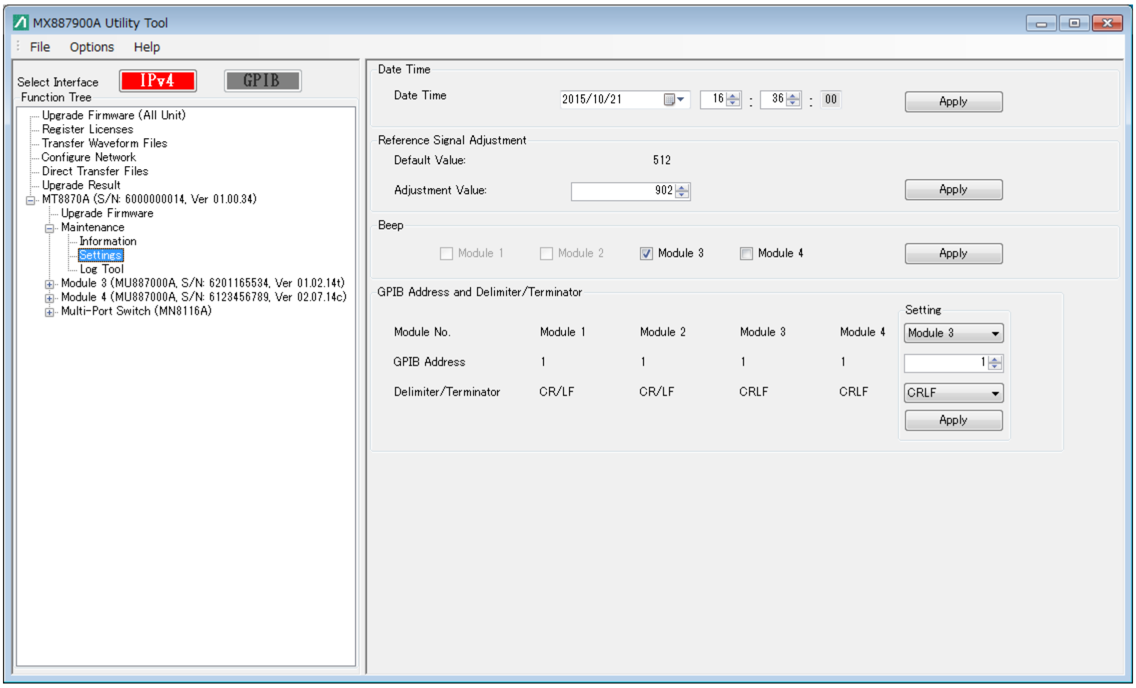


Figure 8.3.6-2 Module Settings Window (Versions Earlier Than Ver 01.03.16)

8.3.7 Log Operation

Retrieve and clear log of each module.

1. Click $\boxed{+}$ in the tree view.
2. Click **Maintenance** – $\boxed{+}$.
3. Click **Log Tool** to display the log operation window of modules.

Retrieving log

1. Click **Export** to display the log file saving dialog box.
2. Input folder and file names and click **OK** to save the displayed information to a file.

Clearing log

1. Click **Clear** to display the confirmation dialog box.
2. Click **OK** to clear the log.

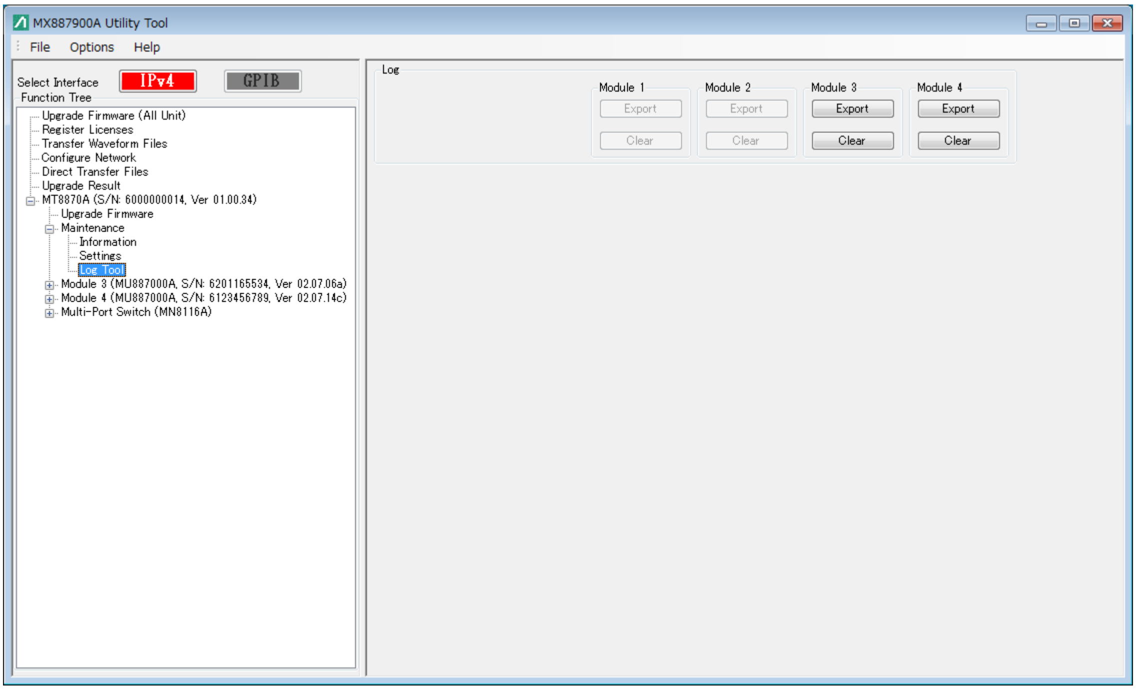


Figure 8.3.7-1 Log Operation Window

8.3.8 Upgrading

The MT8870A firmware can be upgraded when using IPv4 for control. The tree view does not display **Upgrade Firmware** nor **Upgrade Result** when using GPIB for control. To check the version of the built-in program, click **Upgrade Result** of the tree view, using IPv4 for control. The previous upgrade result is displayed, showing the program version.

1. Click + in the tree view.
2. Click **Upgrade Firmware** to display the Upgrade window.
3. Click **Browse** to display the folder selection window.
4. Select the folder containing the file to upgrade and click **OK**.

   The selected file is added to the list.
5. Set a checkmark in the checkboxes for the modules to upgrade displayed under Target Instrument.
6. Click file to upgrade from list and then click **Upgrade**.

   Note:

   The upgrading is interrupted when the error message in Figure 8.3.8-4 is displayed. In this case, delete unnecessary files in the module memory.

   The command to query free space of the module memory is available when the built-in program of the module is in version 01.07.38 or after.

   The upgrading status appears on the Progress window.

   During upgrading, the status lamp 1 to 6 of the MU887000A blink. The blinking lamp varies depending on the file under upgrading. Refer to Appendix D “Status indication of lamps” for details.
7. When upgrading completes, the MT8870A reboots. Rebooting takes a few seconds.
8. When rebooting of the MT8870A completes, the result is displayed on the Progress window. Click **OK** to close the Progress window.

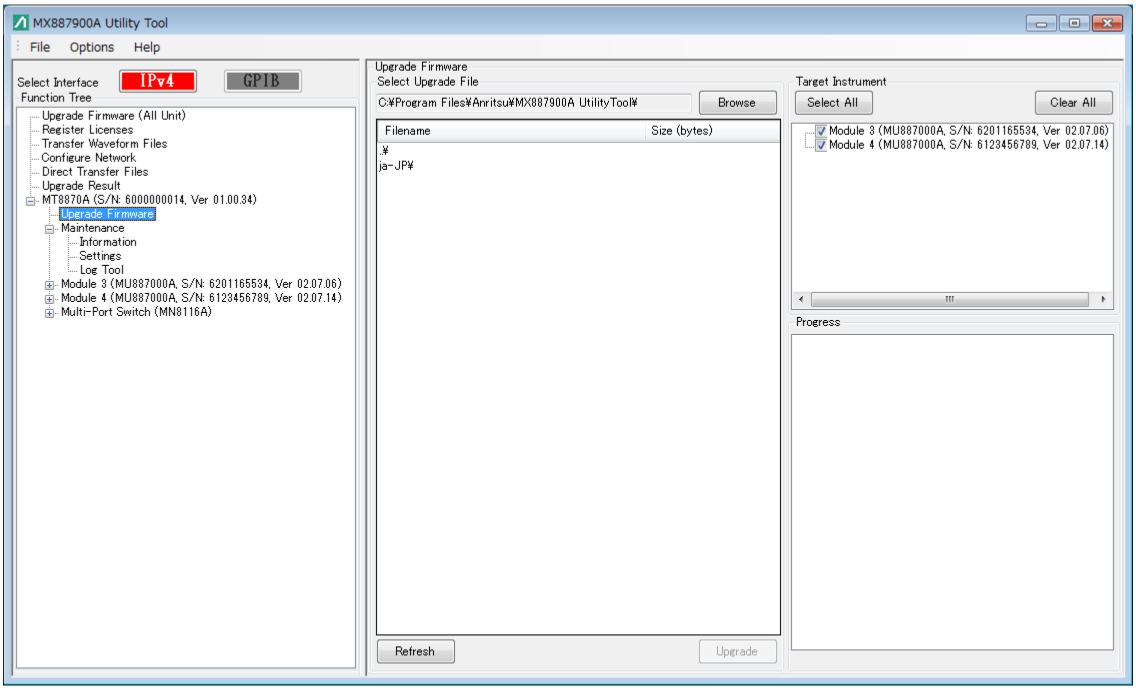


Figure 8.3.8-1 Upgrade Window

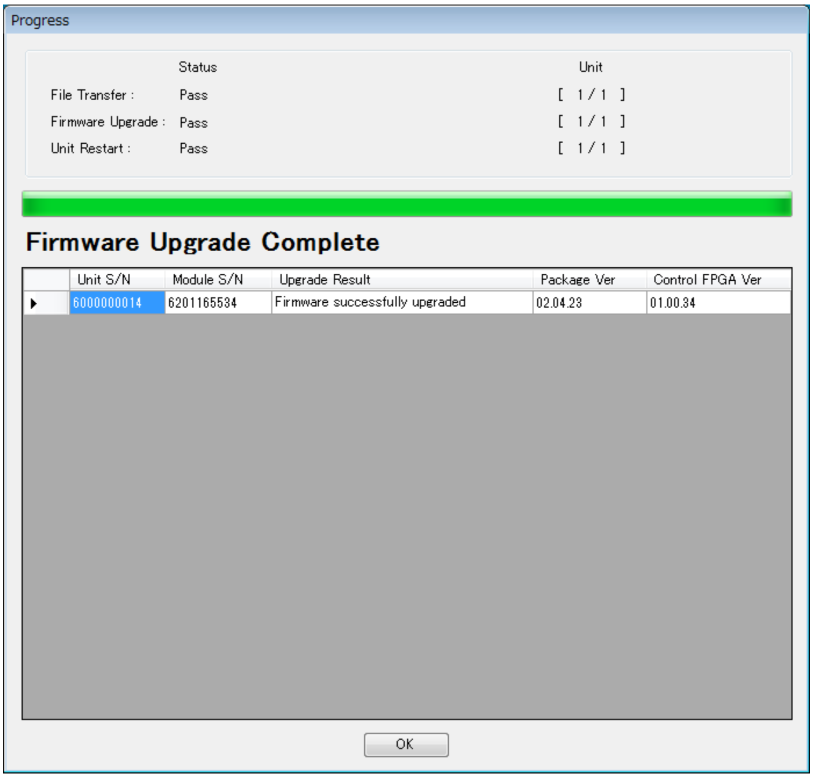


Figure 8.3.8-2 Progress Window after Upgrading

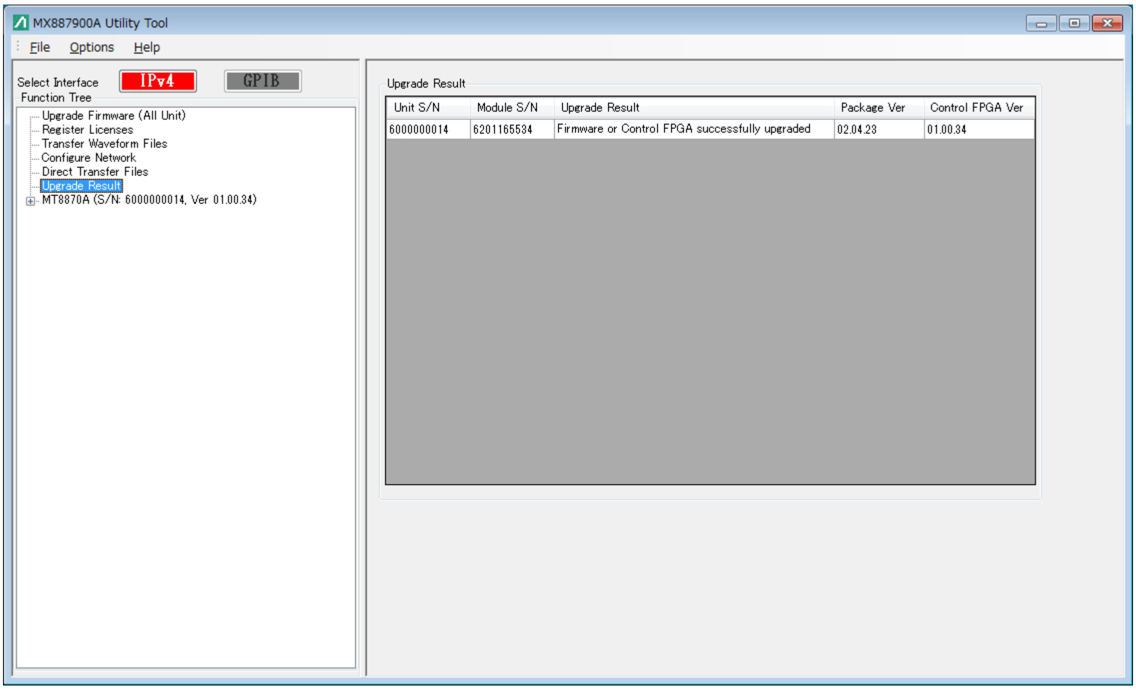


Figure 8.3.8-3 Upgrade Result Window

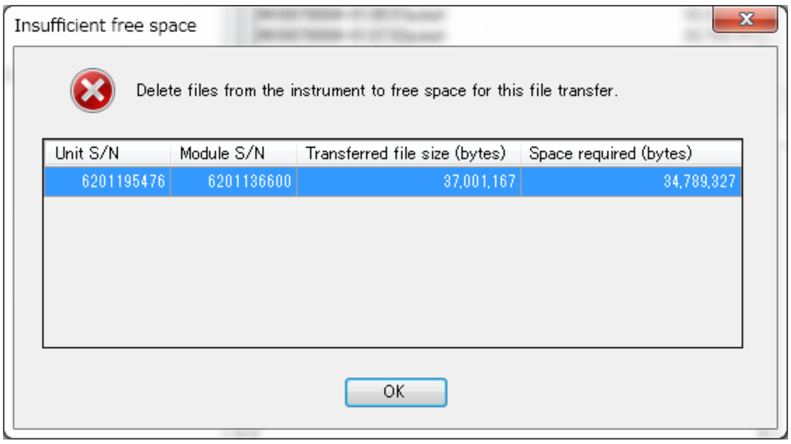


Figure 8.3.8-4 Insufficient Free Space Error Message

8

Utility Tool

8.3.9 Transferring files

Files can be transferred to and from storage media of modules when using IPv4 for control.

The tree view does not display **Transfer files** when using GPIB for control.

1. Click [+] in the tree view.
2. Click [+] for Module1 to Module4.
3. Click **Transfer files**. The file transfer window is displayed.

   PC: Displays files in computer

   Instrument: Displays files in storage media of modules
4. Click **Browse** of PC to display the folder selection window.
5. Select a folder to display the list of files.
6. Select the files to transfer, and click **OK**.

   The selected files are added to the list.
7. To transfer files from the PC to modules, click **->**.

   Note:

   The file is not transferred when the error message in Figure 8.3.8-4 is displayed. In this case, delete unnecessary files in the module memory.

   To transfer files from modules to the PC, click **<-**.

   If there is a file with the same name at the transfer destination, a file overwrite confirmation dialog is displayed.
8. To delete files from the storage media of modules, select the files from the Instrument list and click **Delete**.

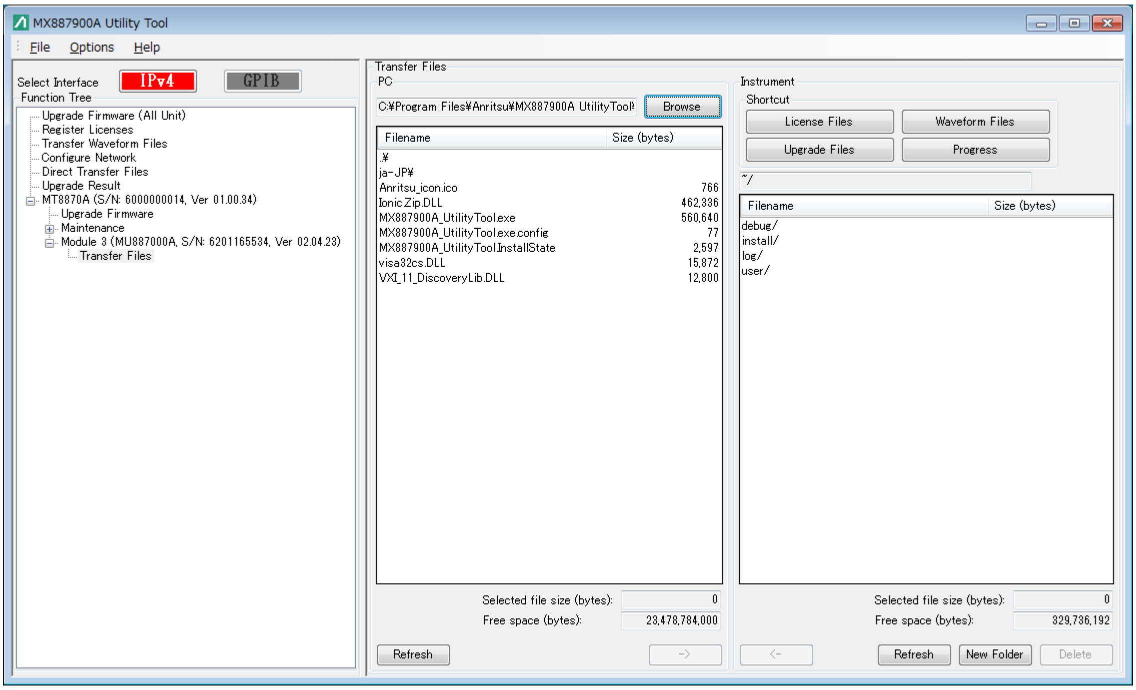


Figure 8.3.9-1 File Transfer Window

Table 8.3.9-1 Screen Description

Name	Description
PC	Displays PC files
Browse	Selects file folder
Refresh	Refreshes contents displaying
->	Transfers files from PC to module
Instrument	Displays files in modules
License files	Displays license folder
Upgrade files	Displays folders for firmware and FPGA data
Waveform Files	Displays waveform file folder
Progress	Displays log file folder
New Folder	Creates new folder
<-	Transfers files from modules to PC
Delete	Deletes files for selected module
Refresh	Updates the displayed contents.

8.3.10 Batch upgrading

The firmware of multiple MT8870A units can be upgraded as a batch when using IPv4 for control.

The tree view does not display **Upgrade Firmware (All Unit)** when using GPIB for control.

To check the version of the built-in program, click **Upgrade Result** of the tree view. The upgrading results are displayed for checking versions.

1. Click **Update Firmware (All Unit)** in the tree view to display the Batch Upgrade window.
2. Click **Browse** to display the folder selection window.
3. Select a folder and click **OK** to display the list of files.
4. Select the file to upgrade by clicking the file name.
5. Set a checkmark in the checkbox for the MT8870A unit(s) to upgrade in the Select list.

   Select All: Checks all MT8870A checkboxes

   Clear All: Unchecks all MT8870A checkboxes
6. Click **Upgrade**. The upgrading status appears on the Progress window.

   Note:

   The batch upgrading is interrupted when the error message in Figure 8.3.8-4 is displayed. In this case, one of the MT8870A units has insufficient free space in the module memory. Delete unnecessary files in the module memory of the MT8870A unit in trouble.

   During upgrading, the status lamp 1 to 6 of the MU887000A blink. The blinking lamp varies depending on the file under upgrading. Refer to Appendix D “Status indication of lamps” for details.
7. When upgrading completes, the MT8870A reboots. It takes time for rebooting.
8. When rebooting of the MT8870A completes, the result is displayed on the Progress window. Click **OK** to close the Progress window.

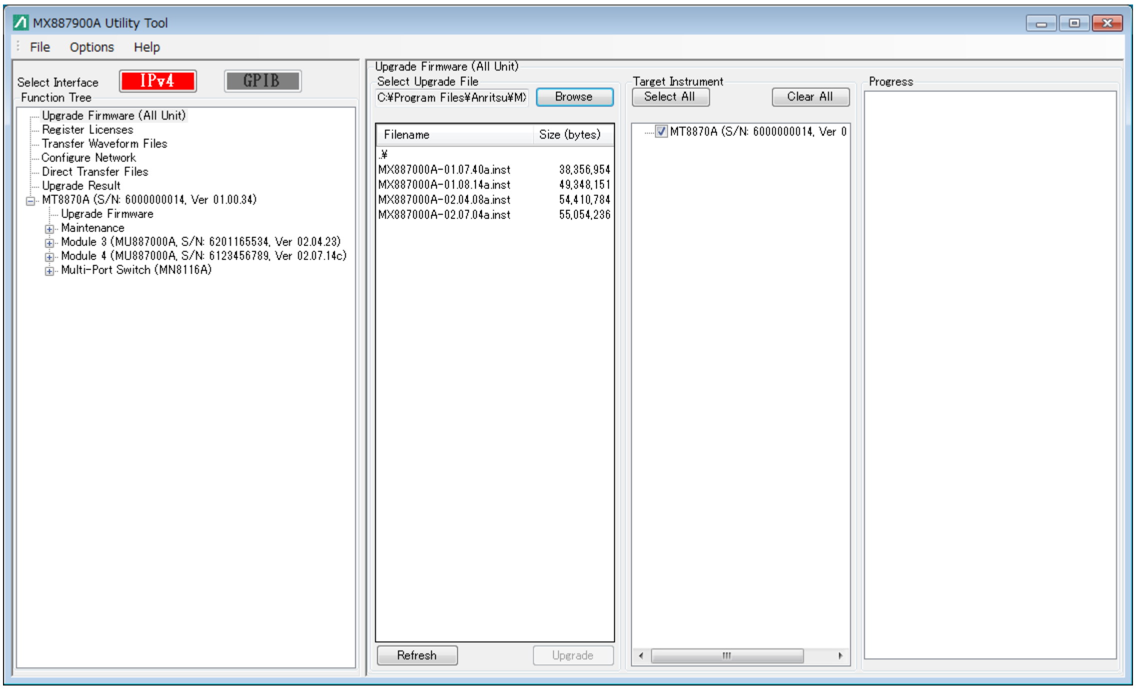


Figure 8.3.10-1 Batch Upgrade Window

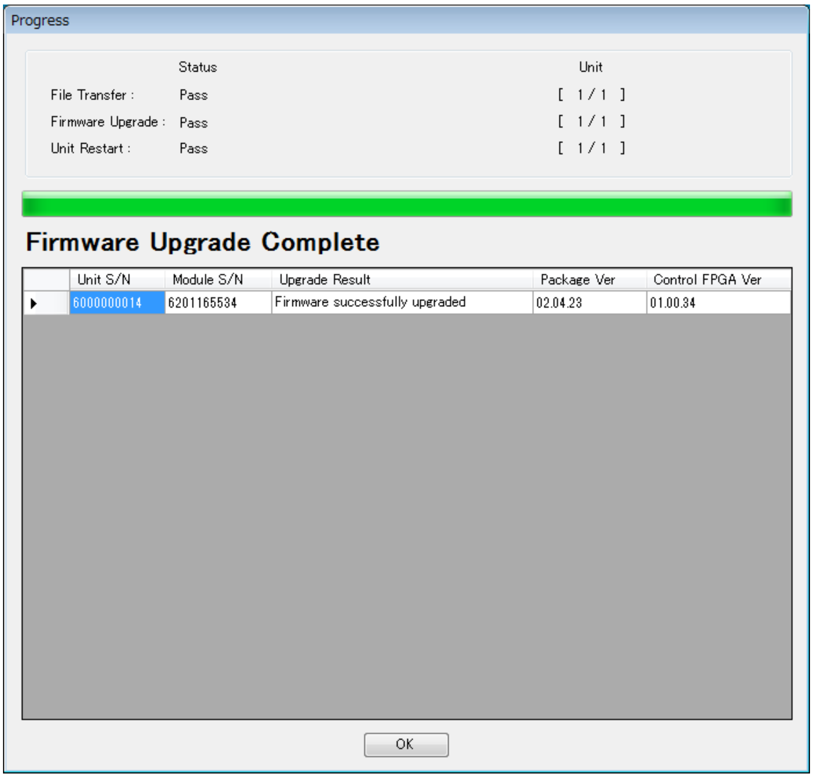


Figure 8.3.10-2 Progress Window after Upgrading Completes

8.3.11 Registering license

This section describes how to register the application software license and waveform license to MT8870A. The license key files are located on the supplied storage media.

The tree view does not display **Register Licenses** when using GPIB for control.

1. Click **Register Licenses** in the tree view. The license registration window is displayed.
2. Click **Browse** to display the folder selection window.
3. Select a folder and click **OK** to display the list of files.
4. Select file names to select license files.
5. Click **Register**.

   Note:

   The license registration is interrupted when the error message in Figure 8.3.8-4 is displayed.

   In this case, one of the MT8870A units for license registration has insufficient free space in the module memory. Delete unnecessary files in the module memory of the MT8870A unit in trouble.

   A message appears during license registration.
   To stop registering licenses, click **Stop**.
6. After license registration, click the interface button to stop communication with the MT8870A.
7. Click the interface button to start communication with the MT8870A.
8. Click Information in the tree view when the tree view is updated.

   Make sure the application software license and waveform license are changed.

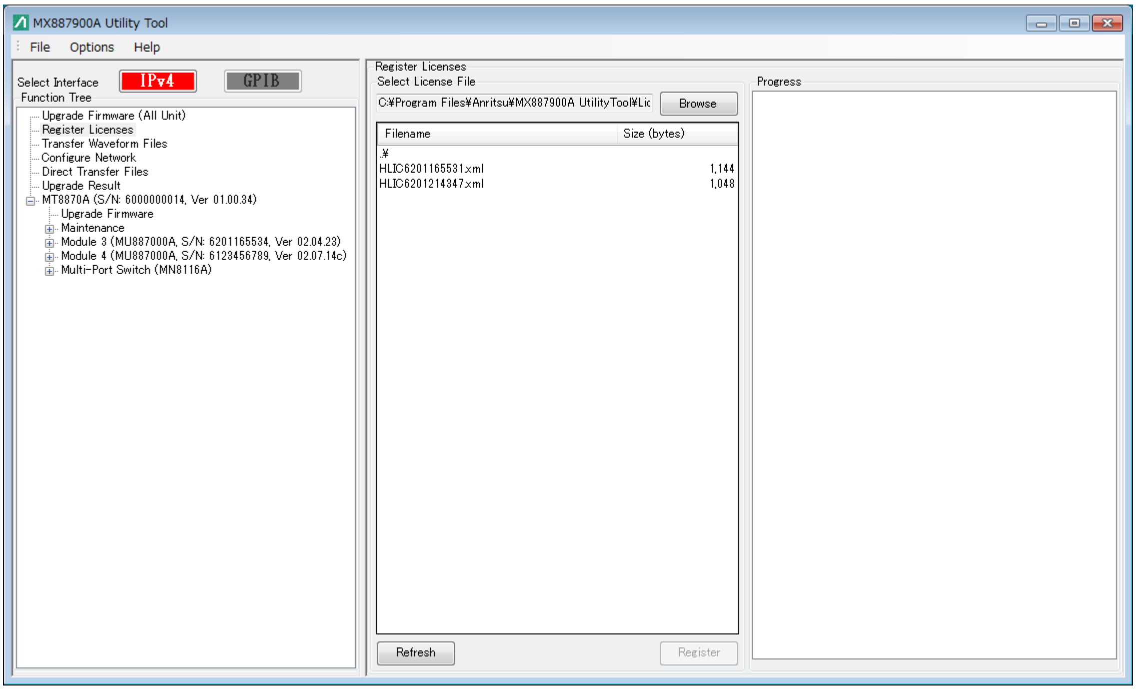


Figure 8.3.11-1 Register License Window

8.3.12 Transferring waveform files

The VSG waveform files can be transferred to modules when IPv4 is used for control. The tree view does not display **Transfer Waveform Files** when using GPIB for control.

1. Click **Transfer Waveform Files** in the tree view to display the waveform file transfer window.
2. Click **Browse** at the Select Waveform File tab to display the folder selection window.
3. Select a folder and click **OK** to display the list of files.
4. Click the file name to select waveform files.

   The waveform files are displayed only when a files with extension “wvd” and a file with extension “xml” both exist. The extension of the waveform file is not displayed on the folder selection window.
5. Set a checkmark at Overwrite Processing for handling transfer of files with the same name at the transfer destination.

   Don't Confirm: Confirmation dialog not displayed

   Always Confirm: Confirmation dialog displayed
6. Set the checkboxes for modules under Target Instrument receiving transferred waveform files.
7. Click **Transfer**.

   Note:

   The waveform file transfer is interrupted when the error message in Figure 8.3.8-4 is displayed.

   In this case, one of the MT8870A units of transfer destination has insufficient free space in the module memory. Delete unnecessary files in the module memory of the MT8870A unit in trouble.

A message is displayed during waveform file transfer. To stop transfer, click **Stop**.

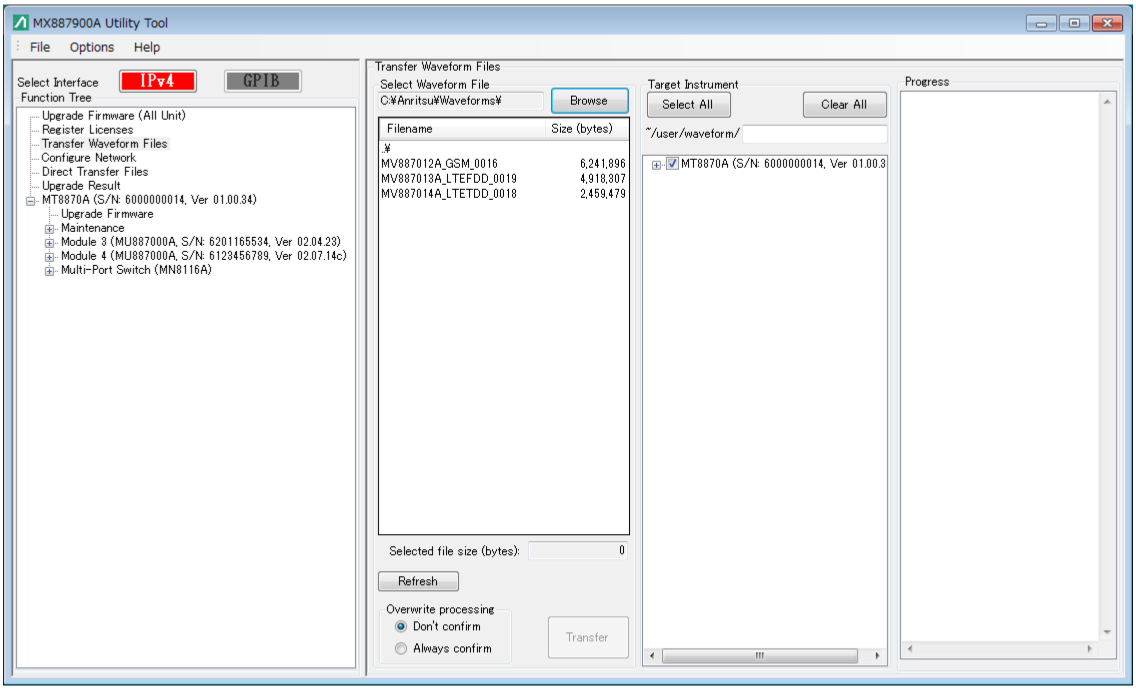


Figure 8.3.12-1 Transfer Waveform Files Window

8.3.13 Transferring files directly

Files can be transferred directly by inputting the destination IP address and using the file transfer function when the Utility Tool cannot detect the MT8870A.

This function is supported only when using IPv4 for control.
While controlling with GPIB, **Direct Transfer Files** are displayed, but Ethernet cable is required to transfer the file.

1. Click **Direct Transfer Files** in the tree view.

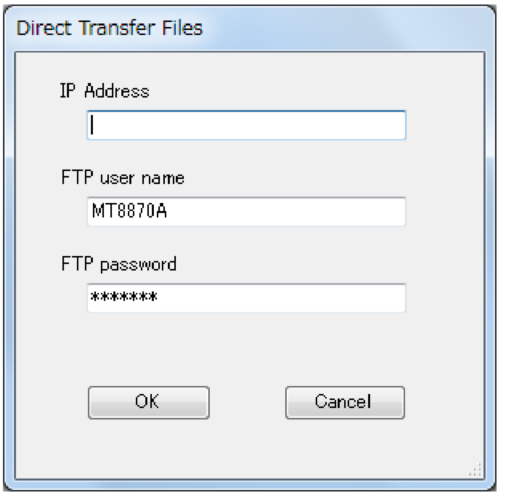


Figure 8.3.13-1 Direct Transfer Files Dialogue

2. Input the transfer destination IP address, FTP user name and password and click **OK**.
3. The File Transfer window shown in Figure 8.3.9-1 File Transfer Window is displayed. Refer to section 8.3.9 “Transferring files”.

8.4 Operating MN8116A

This section explains how to operate MN8116A using the Utility Tool.

This function is available for the Utility Tool of Ver 1.3.0 or later and for the MT8870A (Chassis) with FPGA of Ver 1.0.34 or later.

8.4.1 Setting MN8116A

1. Click [+] of **MT8870A** in the tree view.
2. Click [+] of **Multi-Port Switch (MN8116A)**.

Note:

Multi-Port Switch (MN8116A) is displayed when the firmware version of any module(s) inserted to the MT8870A is Ver 2.6.0 or later.

3. Click on **Settings (MN8116A)**.The MN8116A Settings Window is displayed.

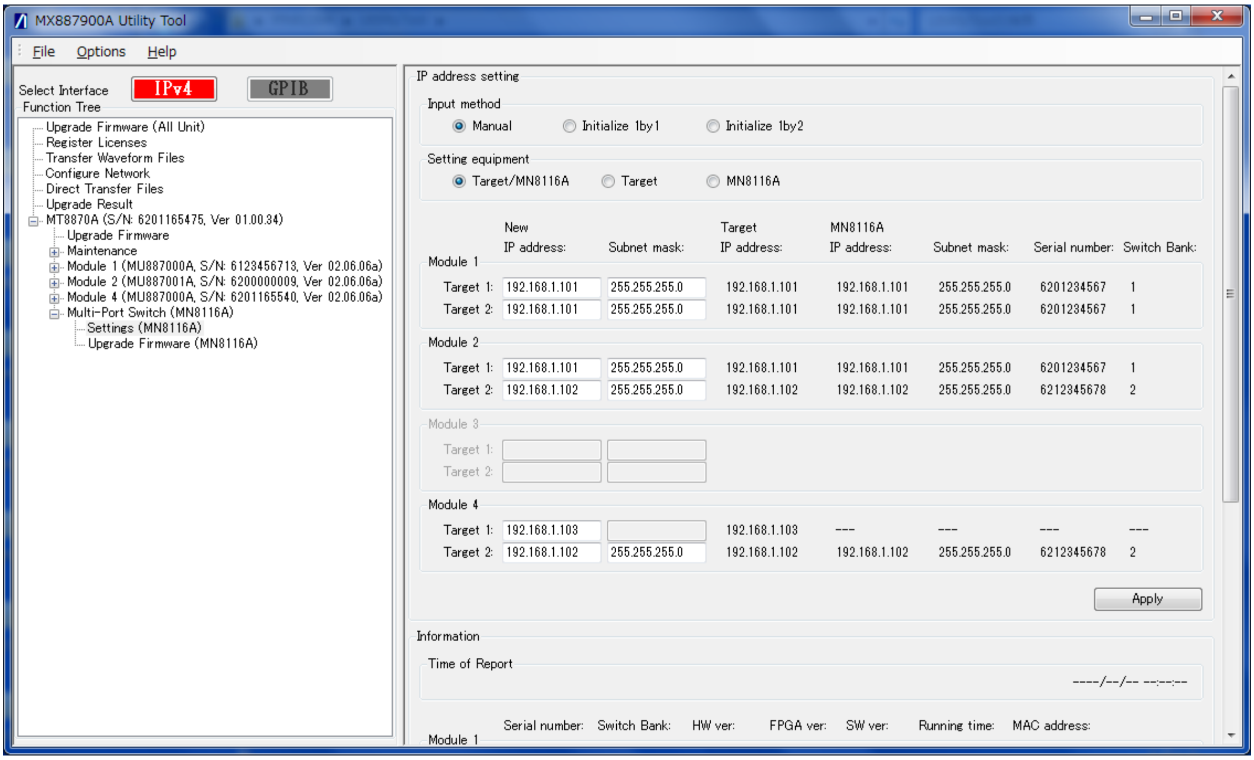


Figure 8.4.1-1 MN8116A Settings Window

Table 8.4.1-1 Screen Details (1/2)

Name	Description
IP address setting	
Input method	Select a desired IP address setting method.
Manual	To specify IP address directly.
Initialize 1by1	To use IP RESET of MN8116A. For the relationship of IP RESET switch number and IP address, refer to Table 8.4.1-2.
Initialize 1by2	To use IP RESET of MN8116A. For the relationship of IP RESET switch number and IP address, refer to Table 8.4.1-3.
Setting equipment	Select a desired IP address setting destination.
Target/MN8116 A	To set the New IP address for both Switch Bank(Target) and MN8116A. Refer to “Apply” on Table 8.4.1-1 (2/2) for details.
Target	To set the value of New IP address as Switch Bank (Target) which is a communication target of MU887000A. Refer to “Apply” on Table 8.4.1-1 (2/2) for details.
MN8116A	To set the value of New IP address to MN8116A. Refer to “Apply” on Table 8.4.1-1 (2/2) for details.
Module 1 to Module 4	Displays the setting information of Slot 1 to 4 of the selected MT8870A.
New	
IP address	Input IP addresses. When selecting **Initialize 1by1** or **Initialize 1by2**, the IP addresses are automatically initialized.
Subnet mask	Input subnet mask values. Subnet mask is unavailable if MN8116A is not connected.
Target	
IP address	Displays the set MN8116A IP address to be connected to MU887000A.
MN8116A	When MN8116A is not connected, “ - ” is displayed for the following items.
IP address	Displays the IP address of MN8116A.
Subnet mask	Displays the subnet mask of MN8116A.
Serial number	Displays the serial number of MN8116A.
Switch Bank	Displays the switch bank of MN8116A.

Table 8.4.1-1 Screen Details (2/2)

Name	Description
Apply	Sets an IP address of the equipment selected by Setting equipment for the communication between MU887000A and MN8116A. When the "Target" is selected, searches MN8116A after setting New IP address for Switch Bank(Target) as the communication target of MU887000A. When "MN8116A" is selected, changes IP address of the connected MN8116A. When "Target/MN8116A" is selected, the IP addresses of Switch Bank (Target) and MN8116A that are communication targets of MU887000A are changed simultaneously.
Information	
Time of Report	Displays the time when the information on the screen was obtained.
Serial number	Displays the serial number of MN8116A.
Switch Bank	Displays the switch bank number of MN8116A. For the switch bank number, refer to the MN8116A operation manual.
HW ver	Displays the MN8116A hardware version.
FPGA ver	Displays the MN8116A FPGA version.
SW ver	Displays the MN8116A software version.
Running time	Displays the MN8116A running time.
MAC address	Displays the MN8116A MAC address.
Reload	Reloads information from MN8116A.
Export	Exports the displayed information to an external file in text.

Table 8.4.1-2 Relationship of MN8116A IP Reset Switch and Initialize 1by1

MT8870A		MT8870A Input Method: Initialize 1by1	
		MN8116A	
Module No.	Target No.	Default IP Address	Switch Number and Switch Bank
1	1	192.168.1.101	Switch Bank 1 when set to RESET 1
1	2	Blank	
2	1	192.168.1.102	Switch Bank 2 when set to RESET 1
2	2	Blank	
3	1	192.168.1.103	Switch Bank 1 when set to RESET 2
3	2	Blank	
4	1	192.168.1.104	Switch Bank 2 when set to RESET 2
4	2	Blank	

Table 8.4.1-3 Relationship of MN8116A IP Reset Switch and Initialize 1by2

MT8870A		MT8870A Input Method: Initialize 1by2	
		MN8116A	
Module No.	Target No.	Default IP Address	Switch Number and Switch Bank
1	1	192.168.1.101	Switch Bank 1 when set to RESET 3
1	2	192.178.1.105	Switch Bank 2 when set to RESET 3
2	1	192.168.1.102	Switch Bank 1 when set to RESET 4
2	2	192.178.1.106	Switch Bank 2 when set to RESET 4
3	1	Blank	
3	2	Blank	
4	1	Blank	
4	2	Blank	

8.4.2 Upgrading MN8116A

The MN8116A firmware can be upgraded when using IPv4 for control.

The tree view does not display **Upgrade Firmware (MN8116A)** when using GPIB for control.

1. Click [+] of **MT8870A** in the tree view.
2. Click [+] of **Multi-Port Switch (MN8116A)**.

Note:

Multi-Port Switch (MN8116A) is displayed when the firmware version of the module(s) inserted to the selected MT8870A is 2.6.0 or later.

3. Click **Upgrade Firmware (MN8116A)**.The MN8116A Upgrade window is displayed.
4. Click Browse to display the folder selection window.
5. Select a folder containing the file(s) to upgrade, and click **OK**. The selected file(s) is added to the list.
6. Select the checkbox for the instrument(s) to upgrade displayed under **Target MN8116A**.
7. Click the file(s) to upgrade in the list, and click **Upgrade**.
8. The upgrading status appears on the Progress window.
9. When upgrading completes, the upgrading result window appears. Turn On the MN8116A and click **Close** to close the upgrade result window.

Note:

Do not turn Off the MN8116A during upgrading.

Because the MN8116A does not reboot automatically after upgrading, turn it On again.

The new firmware is used when turning On the MN8116A again.

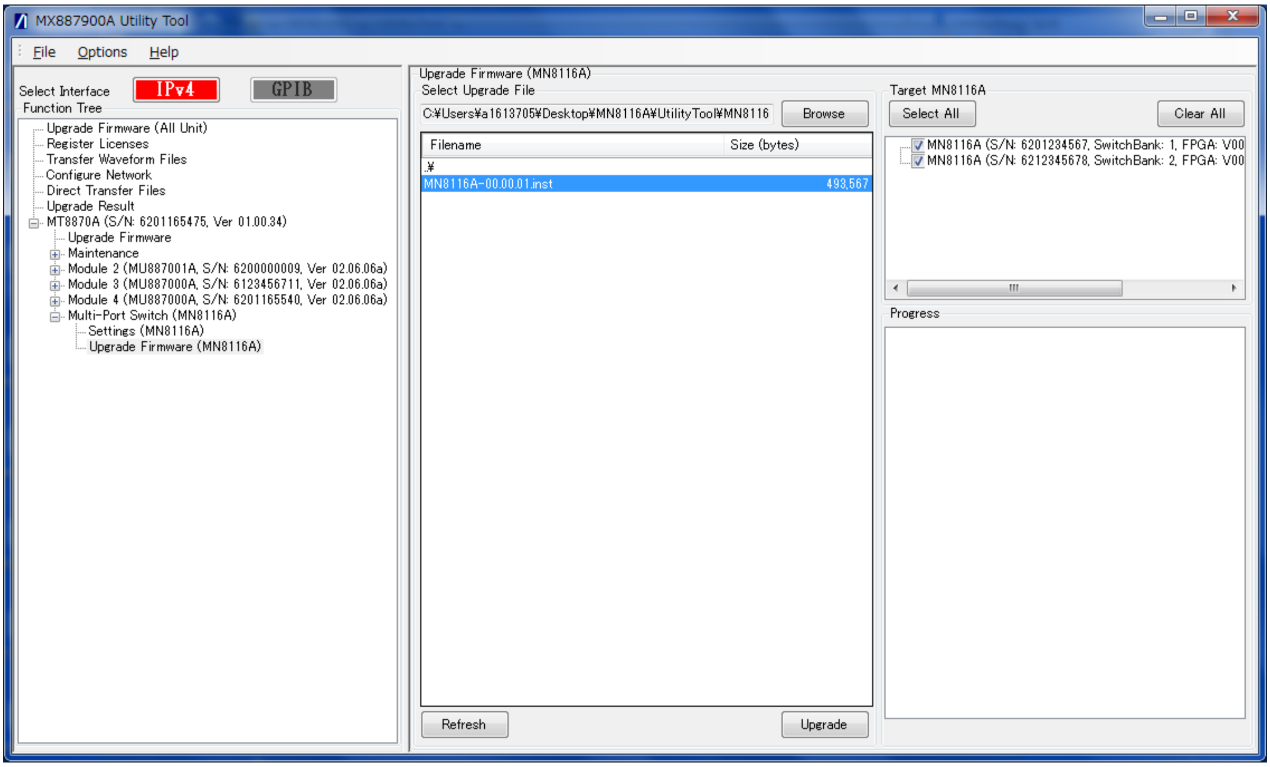


Figure 8.4.2-1 MN8116A Upgrade Window

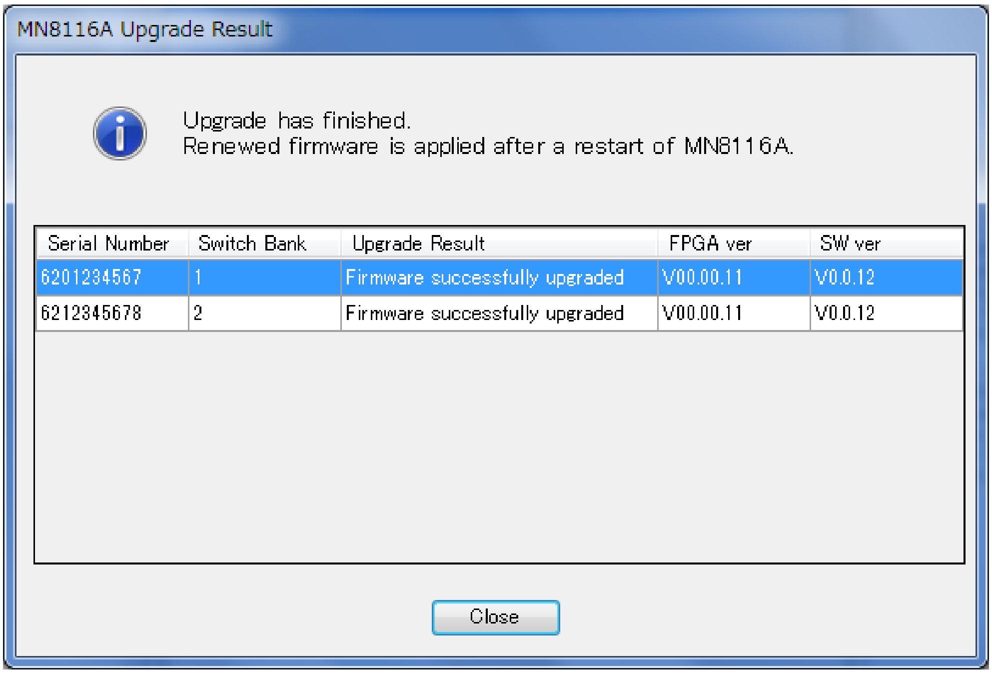


Figure 8.4.2-2 MN8116A Upgrade Result Window

Appendix A Specifications

This appendix describes the specifications. For the product configurations, options, and application parts, refer to Section 1.3 "Product Configuration".

Table A-1 Input/Output Terminal

<table>
<tr><th>Item</th><th>Specifications</th></tr>
<tr><td>RF Test ports
Connector

Impedance
VSWR</td><td>4
Test ports 1 and 2: N (female)
Test ports 3 and 4: N (female)
50 Ω, Nominal
Test ports 1 and 2:
<table>
<tr><th>Frequency (MHz)</th><th>Specifications</th></tr>
<tr><td>10 ≤ f <400</td><td><1.5</td></tr>
<tr><td>400 ≤ f ≤ 2700</td><td><1.2</td></tr>
<tr><td>2700 < f ≤ 3800</td><td><1.3</td></tr>
<tr><td>3800 < f ≤ 6000</td><td><1.5</td></tr>
</table>
Test ports 3 and 4:
<table>
<tr><th>Frequency (MHz)</th><th>Specifications</th></tr>
<tr><td>10 ≤ f < 30</td><td><1.8</td></tr>
<tr><td>30 ≤ f ≤ 3800</td><td><1.5</td></tr>
<tr><td>3800 < f ≤ 6000</td><td><1.6</td></tr>
</table>
</td></tr>
<tr><td>Input level max.</td><td>Test ports 1 and 2: +35 dBm
Test ports 3 and 4: +25 dBm</td></tr>
<tr><td>Audio Ports
Connector</td><td>
Analog: Input L/R Channel — BNC (female)
Output L/R Channel — BNC (female)
Digital: Input/Output — RJ-45</td></tr>
</table>

Appendix

Appendix A

Table A-2 Signal Generator

Item	Specifications
Frequency	
Frequency setting range	10 to 3800 MHz 10 to 6000 MHz (with MU887000A-001/101)
Frequency resolution	1 Hz
Frequency accuracy	Depends on MT8870A reference frequency accuracy
Amplitude	
Level setting range	Test ports 1 and 2: –130 to –10 dBm (Frequency ≤3800 MHz) –130 to –18 dBm (Frequency >3800 MHz) Test ports 3 and 4: –120 to 0 dBm (Frequency ≤3800 MHz) –120 to –8 dBm (Frequency >3800 MHz)
Level setting resolution	0.1 dB
Level accuracy*1	(see table below)
Spurious response	
Harmonic wave distortion	<–25 dBc
Vector modulation	
Modulation bandwidth	≤160 MHz

Level accuracy*1:

Frequency (MHz)	Test ports 1 and 2*2	Test ports 3 and 4*3
10≤ f <400	±1.3 dB*4	±1.3 dB
400≤ f ≤3800	±1.0 dB and ±0.7 dB*5	±1.0 dB and ±0.7 dB*5
3800< f ≤6000*6	±1.3 dB and ±1.0 dB*5	±1.3 dB and ±0.7 dB*5

*1: CW, 10 to 40 °C

*2: Output level ≥–120 dBm, (Frequency ≤3800 MHz),
Output level ≥–100 dBm (Frequency >3800 MHz), after CAL

*3: Output level ≥–110 dBm, after CAL

*4: Signal Analyzer input level +15 dBm

*5: Typical value

*6: With MU887000A-001/101

Table A-3 Signal Analysis

<table>
<tr><th>Item</th><th>Specifications</th></tr>
<tr><td>Frequency
Frequency setting range
Frequency resolution</td><td>
10 to 3800 MHz
10 to 6000 MHz (with MU887000A-001/101)
1 Hz</td></tr>
<tr><td>Amplitude
Level setting range*2</td><td>
<table>
<tr><th>Frequency (MHz)</th><th>Test ports 1 and 2</th><th>Test ports 3 and 4</th></tr>
<tr><td>10 ≤ f < 350</td><td>–65 to +15 dBm</td><td>–65 to +15 dBm</td></tr>
<tr><td>350 ≤ f ≤ 3800</td><td>–65 to +35 dBm</td><td>–65 to +25 dBm</td></tr>
<tr><td>3800 ≤ f ≤ 6000*1</td><td>–65 to +35 dBm</td><td>–65 to +25 dBm</td></tr>
</table>
</td></tr>
<tr><td>Level setting resolution</td><td>0.1 dB</td></tr>
<tr><td>Level accuracy*3</td><td>Test ports 1 and 2:
<table>
<tr><th>Frequency (MHz) / Level (dBm)</th><th>10 ≤ f < 400</th><th>400 ≤ f ≤ 3800</th><th>3800 ≤ f ≤ 6000 *1</th></tr>
<tr><td>–30 ≤ level ≤ +35</td><td>–</td><td>±0.3 dB*4
and
±0.5 dB</td><td>±0.7 dB*5</td></tr>
<tr><td>–30 ≤ level ≤ +15</td><td>±0.7 dB</td><td>–</td><td>–</td></tr>
<tr><td>–55 ≤ level < –30</td><td>±0.9 dB</td><td>±0.7 dB</td><td>±0.9 dB*5</td></tr>
<tr><td>–65 ≤ level < –55</td><td>±1.1 dB</td><td>±0.9 dB</td><td>±1.1 dB*5</td></tr>
</table>
Test ports 3 and 4:
<table>
<tr><th>Frequency (MHz) / Level (dBm)</th><th>10 ≤ f < 400</th><th>400 ≤ f ≤ 3800</th><th>3800 < f ≤ 6000 *1</th></tr>
<tr><td>–30 ≤ level ≤ +25</td><td>–</td><td>±0.7 dB</td><td>±0.7 dB*5</td></tr>
<tr><td>–30 ≤ level ≤ +15</td><td>±0.7 dB</td><td>–</td><td>–</td></tr>
<tr><td>–55 ≤ level < –30</td><td>±0.9 dB</td><td>±0.9 dB</td><td>±0.9 dB*5</td></tr>
<tr><td>–65 ≤ level < –55</td><td>±1.1 dB</td><td>±1.1 dB</td><td>±1.1 dB*5</td></tr>
</table>
</td></tr>
</table>

*1: With MU887000A-001/101

*2: CW

*3: CW, Measurement Bandwidth 300 kHz, RBW 100 kHz, 10 to 40 °C, after CAL

*4: Typical value

*5: 20 to 30 °C

Table A-3 Signal Analysis (Cont'd)

Item	Specifications		
Amplitude (Continued) Level linearity*6	Input level	Test ports 1 and 2	Test ports 3 and 4
	≥–55 dBm	±0.2 dB	±0.2 dB
	≥–65 dBm	±0.4 dB	±0.4 dB
Modulation analysis Maximum analysis bandwidth	f: Frequency setting $10 \leq f < 500$ MHz: 25 MHz $500 \leq f < 1900$ MHz: 80 MHz $1900 \leq f \leq 6000$ MHz: 160 MHz		

*6: CW, Measurement Bandwidth 300 kHz, RBW 100 kHz, –40 to 0 dB

Table A-4 Analog Audio

Item	Specifications
Audio signal generator	
Frequency setting range	20 Hz to 20 kHz
Output level range	0 (Off), 1 mV to 5 Vpeak*1
Impedance	1 Ω (Nominal), AC coupling
Audio analyzer	
Frequency range	20 Hz to 20 kHz
Input level range	1 mV to 5 Vpeak*2
Impedance	100 kΩ, AC coupling

*1: 100 kΩ termination

*2: 30 Vrms max.

Table A-5 Digital Audio

Item	Specifications
Audio signal generator	
Frequency setting range	20 Hz to 20 kHz*1 20 Hz to 14 kHz*2 20 Hz to 7 kHz*3
Bit resolution	16/24-bit
Audio analyzer	
Sampling rate	16 kHz, 32 kHz, 44.1 kHz, 44 kHz
Bit resolution	16/24-bit

*1: Sampling rate 44.1 kHz, 48 kHz

*2: Sampling rate 32 kHz

*3: Sampling rate 16 kHz

Table A-6 General Performance

Item	Specifications
Interface	
Trigger function	Trigger signals input/output at Trigger connectors on MT8870A rear panel
Remote control	Ethernet: via MT8870A interface GPIB: with MT8870A GPIB option (MT8870A-001/101) Interface function: SH1, AH1, T6, L4, SR1, RL1, PP0, DC1, DT0, C0, E2
Dimensions	90 (W) × 193.6 (H) × 325 (D) mm (excluding protrusions)
Mass	≤5 kg (including options)

Appendix B Software Licenses

This product includes the software listed in the following table.
For the software details, refer to the Anritsu Web site at
https://www.anritsu.com/
Package software in the table is not included our software licensing.

Table B-1 Packages and Corresponding Licenses

Package	License Documents	Remarks
acl	(*1), (*2)	GPLv2, LGPLv2.1
acpid	(*1)	GPLv2
attr	(*1), (*2)	GPLv2, LGPLv2.1
audit	(*1)	GPLv2
bash	(*3)	GPLv3
bc	(*1)	GPLv2
coreutils	(*3)	GPLv3
cpio	(*3)	GPLv3
cpufrequtils	(*5)	
cpuspeed	(*1)	GPLv2
cronie	(*6)	
db4	(*7)	
db4-utils	(*7)	
dhclient	(*8)	
dmidecode	(*1)	GPLv2
dosfstools	(*3)	GPLv3
DotNetZip	(*29)	Ms-PL
e2fsprogs	(*1)	GPLv2, BSD
ed	(*3)	GPLv3
file	(*9)	
ftp		BSD with advertising
glibc	(*1), (*2)	GPLv2, LGPLv2.1
groff	(*1)	GPLv2
initscripts	(*1)	GPLv2
iproute	(*1)	GPLv2
iptables	(*1)	GPLv2

Table B-1 Packages and Corresponding Licenses (Cont'd)

Package	License Documents	Remarks
iptables-ipv6	(*1)	GPLv2
iputils	(*1)	BSD with advertising and GPLv2+ and Rdisc
kbd	(*10)	
libedit	(*11)	
libgssglue	(*25)	MIT
libpcap	(*26)	BSD
libtirpc	(*12)	
libxml2	(*13)	MIT
linux-2.6.32	(*1)	GPLv2
linux-gpib	(*1)	GPLv2
lsof	(*14)	
man	(*1)	GPLv2
man-pages	(*15)	
ncurses	(*16)	
ntp		(MIT and BSD and BSD with advertising) and GPLv2
ntpdate		(MIT and BSD and BSD with advertising) and GPLv2
openssh-server	(*17)	
passwd	(*1)	BSD, GPLv2
patch	(*3)	GPLv3
pciutils	(*1)	GPLv2
policycoreutils	(*1)	GPLv2
postfix	(*18)	
procps	(*1)	GPLv2
pth	(*2)	LGPLv2.1
rhnsd	(*1)	GPLv2

Table B-1 Packages and Corresponding Licenses (Cont'd)

Package	License Documents	Remarks
rpcbind	(*19)	BSD
rpm	(*1), (*4)	GPLv2, LGPLv2
rsyslog	(*3)	GPLv3
selinux-policy-tar geted	(*1)	GPLv2
setserial	(*20)	GPL
shadow-utils	(*1)	BSD, GPLv2
sudo	(*21)	
tcpdump	(*22)	BSD
telnet	(*23)	BSD
TinyXML 2.5.3	(*27)	
unRAR	(*31)	
unzip	(*24)	
util-linux-ng	(*1), (*2), (*3)	GPLv3, GPLv2, LGPLv2.1, BSD, PublicDomain
vim-minimal	(*25)	
vsftpd	(*1)	GPLv2
VXI-11 1.10	(*28)	
xz	(*1), (*2), (*3)	GPLv3, GPLv2, LGPLv2.1
xz-lzma-compat	(*2)	LGPLv2.1
yum	(*1)	GPLv2
yum-rhn-plugin	(*1)	GPLv2
7-Zip	(*30)	LGPLv2.1

(*1) GPLv2:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample;

alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> <signature of Ty Coon>, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

(*2) LGPL2.1:

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

- [This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of

the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for

enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

(*3) GPLv3:

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty;
and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License,
in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part obit. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions;
the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.
Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.

(*4) LGPLv2.0:

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices. Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under

Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

(*5) Other:

This package was debianized by Mattia Dongili <malattia@debian.org> on Sun, 28 Nov 2004 17:26:30 +0100.

It was downloaded from http://kernel.org/pub/linux/utils/kernel/cpufreq/

Copyright: 2004-2006 Dominik Brodowski

Upstream Author: Dominik Brodowski <linux@dominikbrodowski.de>

License:

This package is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 dated June, 1991.

This package is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this package; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

(*6) Other:

```
/*
 * Copyright (c) 2004 by Internet Systems Consortium, Inc. ("ISC")
 * Copyright (c) 1997,2000 by Internet Software Consortium, Inc.
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND ISC DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS.  IN NO EVENT SHALL ISC BE LIABLE FOR
 * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
 * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
 * OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 */
/*
 * Copyright (c) 1988, 1993, 1994
 *      The Regents of the University of California.  All rights reserved.
 *
 * This code is derived from software written by Ken Arnold and
 * published in UNIX Review, Vol. 6, No. 8.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
```

```
 * SUCH DAMAGE.
 *
 */
/*
 * Copyright (c) 1989, 1993
 *         The Regents of the University of California.   All rights reserved.
 *
 * This code is derived from software contributed to Berkeley by
 * Paul Vixie.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *      notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *      notice, this list of conditions and the following disclaimer in the
 *      documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *      may be used to endorse or promote products derived from this software
 *      without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.   IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 *         @(#)bitstring.h 8.1 (Berkeley) 7/19/93
 */
```

(*7) Other:

```
/*
 * Copyright (c) 1990,2008 Oracle.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Redistributions in any form must be accompanied by information on
 *    how to obtain complete source code for the DB software and any
 *    accompanying software that uses the DB software.  The source code
 *    must either be included in the distribution or be available for no
 *    more than the cost of distribution plus a nominal fee, and must be
 *    freely redistributable under reasonable conditions.  For an
 *    executable file, complete source code means the source code for all
 *    modules it contains.  It does not include source code for modules or
 *    files that typically accompany the major components of the operating
 *    system on which the executable file runs.
 *
 * THIS SOFTWARE IS PROVIDED BY ORACLE ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR
 * NON-INFRINGEMENT, ARE DISCLAIMED.  IN NO EVENT SHALL ORACLE BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
 * BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
 * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
 * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
 * IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
/*
 * Copyright (c) 1990, 1993, 1994, 1995
 *      The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
```

```
 *      notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *      notice, this list of conditions and the following disclaimer in the
 *      documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *      may be used to endorse or promote products derived from this software
 *      without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
/*
 * Copyright (c) 1995, 1996
 *        The President and Fellows of Harvard University.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *      notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *      notice, this list of conditions and the following disclaimer in the
 *      documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *      may be used to endorse or promote products derived from this software
 *      without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY HARVARD AND ITS CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL HARVARD OR ITS CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
```

```
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
/***
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2005 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */
```

(*8) Other:

(*9) Other:

(*10) Other:

The file
kbdrate.c
is Copyright (C) 1992 Rickard E. Faith.

The files
dumpkeys.c loadkeys.y analyze.l ksyms.c ksyms.h
showkey.c keymaps.5 loadkeys.1 dumpkeys.1 showkey.1
are Copyright (C) 1993 Risto Kankkunen.

The files
setfont.c setfont.8 mapscrn.c mapscrn.8
(and changes to loadkeys.y)
are Copyright (C) 1993 Eugene G. Crosser.

The files
psfaddtable.1 psfgettable.1 psfstriptable.1
are Copyright (C) 1994 H. Peter Anvin.

The files
chvt.c clrunimap.c deallocvt.c fgconsole.c findfile.c
getfd.c getkeycodes.c getunimap.c kbd_mode.c loadunimap.c
psffontop.c psfxtable.c resizecons.c screendump.c setkeycodes.c
setleds.c setmetamode.c setvesablank.c showconsolefont.c
spawn_console.c spawn_login.c
chvt.1 deallocvt.1 kbd_mode.1 setleds.1 setmetamode.1
getkeycodes.8 loadunimap.8 resizecons.8 setkeycodes.8
kbd.FAQ.*
(and changes to earlier mentioned programs) are
Copyright (C) 1994-1999 Andries E. Brouwer.

All files in this package may be freely copied under the terms of the GNU General Public License (GPL), version 2, or at your option any later version - except possibly for the restrictions mentioned in the directory consolefonts.

(*11) Other:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(*12) Other:

Sun Industry Standards Source License 1.0

DEFINITIONS

1.1. "Commercial Use" means distribution or otherwise making the Original Code available to a third party.

1.2. "Contributor Version" means the combination of the Original Code, and the Modifications made by that particular Contributor.

1.3. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.4. "Executable" means Original Code in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by 2 (Exhibit A)

1.6. "Larger Work" means a work which combines Original Code or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. A Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications. .

1.10. "Original Code" means Source Code of computer
software code which is described in the Source Code notice required by Exhibit A as Original Code.

1.11. "Patent Claims" means any patent claims, now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means the preferred form of the Original Code for making modifications to it, including all modules it contains, plus any associated interface definition files, or scripts used to control compilation and installation of an Executable.

1.13. "Standards" means the standard identified in Exhibit B or a subsequent version of such standard.

1.14. "You" or "Your" means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition,
"control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2.0 SOURCE CODE LICENSE

2.1 The Initial Developer Grant:　The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use,　reproduce, modify, display, perform, sub license and distribute the Original Code (or portions thereof)with or without Modifications, and/or as part of a Larger Work; and

b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

d) Notwithstanding Section 2.1(b)above, no patent license is granted:　1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by:

i) the modification of the Original Code or

ii) the combination of the Original Code with other software or devices, including but not limited to Modifications.

3.0 DISTRIBUTION OBLIGATIONS

3.1 Application of License.　The Source Code version of Original Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute.

You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipient's rights hereunder.

Your license for shipment of the Contributor Version is conditioned upon your full compliance with this Section.

The Modifications which you create must comply with all requirements set out by the Standards body in effect 120 days before You ship the Contributor Version.

In the event that the Modifications do not meet such requirements, You agree to publish (i) any deviation from the Standards protocol resulting from implementation of your Modifications and (ii) a reference implementation of Your Modifications, and to make any such deviation and reference implementation available to all third parties under the same terms as the license on a royalty free basis within thirty (30) days of Your first customer shipment of Your Modifications.

3.2 Required Notices.　You must duplicate the notice in Exhibit A in each file of the Source Code.

If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice.

If You created one or more Modifications) You may add your name as a Contributor to the notice described in Exhibit A.

You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Initial Code.

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Your version of the Code.　However, You may do so only on Your own behalf, and not on behalf of the Initial Developer.

You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer for any liability incurred by the Initial Developer as a result of warranty, support, indemnity or liability terms You offer.

3.3 Distribution of Executable Versions. You may distribute Original Code in Executable and Source form only if the requirements of Section 3.1 and 3.2 have been met for that Original Code, and if You include a notice stating that the Source Code version of the Original Code is available under the terms of this License.

The notice must be conspicuously included in any notice in an Executable or Source versions, related documentation or collateral in which You describe recipients' rights relating to the Original Code.

You may distribute the Executable and Source versions of Your version of the Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License.

If You distribute the Executable and Source versions under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer .

You hereby agree to indemnify the Initial Developer for any liability incurred by the Initial Developer as a result of any such terms You offer .

3.4 Larger Works. You may create a Larger Work by combining Original Code with other code not governed by the terms of this License and distribute the Larger Work as a single product.

In such a case, You must make sure the requirements of this License are fulfilled for the Original Code.

4.0 INABILITY TO COMPLY DUE TO STATUTE OR REGULATION

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Original Code due to statute, judicial order, or regulation then You must:

a) comply with the terms of this License to the maximum extent possible; and

b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.2 and must be included with all distributions of the Source Code.

Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5.0 APPLICATION OF THIS LICENSE

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Modifications as set out in Section 3.1.

6.0 VERSIONS OF THE LICENSE

6.1 New Versions. Sun Microsystems, Inc. Sun may publish revised and/or new versions of the License from time to time.

Each version will be given a distinguishing version number .

6.2 Effect of New Versions. Once Original Code has been published under a particular version of the License, You may always continue to use it under the terms of that version.

You may also choose to use such Original Code under the terms of any subsequent version of the License published by Sun.

No one other than Sun has the right to modify the terms applicable to Original Code.

7. DISCLAIMER OF WARRANTY.

ORIGINAL CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE ORIGINAL CODE IS

FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING.
THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE ORIGINAL CODE IS WITH YOU.
SHOULD ANY ORIGINAL CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER)ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION.
THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE.
NO USE OF ANY ORIGINAL CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8.0 TERMINATION

8.1 This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach.
All sublicenses to the Original Code which are properly granted shall survive any termination of this License.
Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2 .In the event of termination under Section 8.1 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9.0 LIMIT OF LIABILITY

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE) ,CONTRACT,
OR OTHER WISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF ORIGINAL CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR

(*13) Other:

Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar licence but with different Copyright notices) all the files are:

Copyright (C) 1998-2003 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FIT-NESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,

ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

(*14) Other:

```
* Copyright 2002 Purdue Research Foundation, West Lafayette,
* Indiana 47907.  All rights reserved.
*
* Written by Victor A. Abell
*
* This software is not subject to any license of the American
* Telephone and Telegraph Company or the Regents of the
* University of California.
*
* Permission is granted to anyone to use this software for
* any purpose on any computer system, and to alter it and
* redistribute it freely, subject to the following
* restrictions:
*
* 1. Neither the authors nor Purdue University are responsible
*    for any consequences of the use of this software.
*
* 2. The origin of this software must not be misrepresented,
*    either by explicit claim or by omission.  Credit to the
*    authors and Purdue University must appear in documentation
*    and sources.
*
* 3. Altered versions must be plainly marked as such, and must
*    not be misrepresented as being the original software.
*
* 4. This notice may not be removed or altered.
```

(*15) Other:

(*16) Other:

'---

Appendix

Appendix B

(*17) Other:

1)

* Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
* All rights reserved
*
* As far as I am concerned, the code I have written for this software
* can be used freely for any purpose. Any derived versions of this
* software must be clearly marked as such, and if the derived work is
* incompatible with the protocol description in the RFC file, it must be
* called by a name other than "ssh" or "Secure Shell".

[Tatu continues]

* However, I am not implying to give any licenses to any patents or
* copyrights held by third parties, and the software includes parts that
* are not under my direct control. As far as I know, all included
* source code is used in accordance with the relevant license agreements
* and can be used freely for any purpose (the GNU license being the most
* restrictive); see below for details.

[However, none of that term is relevant at this point in time. All of these restrictively licenced software components which he talks about have been removed from OpenSSH, i.e.,

- RSA is no longer included, found in the OpenSSL library
- IDEA is no longer included, its use is deprecated
- DES is now external, in the OpenSSL library
- GMP is no longer used, and instead we call BN code from OpenSSL
- Zlib is now external, in a library
- The make-ssh-known-hosts script is no longer included
- TSS has been removed
- MD5 is now external, in the OpenSSL library
- RC4 support has been replaced with ARC4 support from OpenSSL
- Blowfish is now external, in the OpenSSL library

[The licence continues]

Note that any information and cryptographic algorithms used in this software are publicly available on the Internet and at any major bookstore, scientific library, and patent office worldwide. More information can be found e.g. at "http://www.cs.hut.fi/crypto".

The legal status of this program is some combination of all these

permissions and restrictions. Use only at your own responsibility.
You will be responsible for any legal consequences yourself; I am not
making any claims whether possessing or using this is legal or not in
your country, and I am not taking any responsibility on your behalf.

NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

2)

The 32-bit CRC compensation attack detector in deattack.c was
contributed by CORE SDI S.A. under a BSD-style license.

* Cryptographic attack detector for ssh - source code
*
* Copyright (c) 1998 CORE SDI S.A., Buenos Aires, Argentina.
*
* All rights reserved. Redistribution and use in source and binary
* forms, with or without modification, are permitted provided that
* this copyright notice is retained.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
* WARRANTIES ARE DISCLAIMED. IN NO EVENT SHALL CORE SDI S.A. BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR
* CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OR MISUSE OF THIS

* SOFTWARE.
*
* Ariel Futoransky <futo@core-sdi.com>
* <http://www.core-sdi.com>

3)

ssh-keyscan was contributed by David Mazieres under a BSD-style
license.

* Copyright 1995, 1996 by David Mazieres <dm@lcs.mit.edu>.
*
* Modification and redistribution in source and binary forms is
* permitted provided that due credit is given to the author and the
* OpenBSD project by leaving this copyright notice intact.

4)

The Rijndael implementation by Vincent Rijmen, Antoon Bosselaers
and Paulo Barreto is in the public domain and distributed
with the following license:

* @version 3.0 (December 2000)
*
* Optimised ANSI C code for the Rijndael cipher (now AES)
*
* @author Vincent Rijmen <vincent.rijmen@esat.kuleuven.ac.be>
* @author Antoon Bosselaers <antoon.bosselaers@esat.kuleuven.ac.be>
* @author Paulo Barreto <paulo.barreto@terra.com.br>
*
* This code is hereby placed in the public domain.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHORS "AS IS" AND ANY EXPRESS
* OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.　IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
* BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
* OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5)

One component of the ssh source code is under a 3-clause BSD license,

held by the University of California, since we pulled these parts from
original Berkeley code.

6)

Remaining components of the software are provided under a standard
2-term BSD licence with the following names as copyright holders:

Markus Friedl
Theo de Raadt
Niels Provos
Dug Song
Aaron Campbell
Damien Miller
Kevin Steves
Daniel Kouril
Wesley Griffin
Per Allansson

Nils Nordman
Simon Wilkinson

Portable OpenSSH additionally includes code from the following copyright holders, also under the 2-term BSD license:

Ben Lindstrom
Tim Rice
Andre Lucas
Chris Adams
Corinna Vinschen
Cray Inc.
Denis Parker
Gert Doering
Jakob Schlyter
Jason Downs
Juha Yrj<F6>l<E4>
Michael Stone
Networks Associates Technology, Inc.
Solar Designer
Todd C. Miller
Wayne Schroeder
William Jones
Darren Tucker
Sun Microsystems
The SCO Group
Daniel Walsh

```
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
```

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

8) Portable OpenSSH contains the following additional licenses:

a) md5crypt.c, md5crypt.h

* "THE BEER-WARE LICENSE" (Revision 42):
* <phk@login.dknet.dk> wrote this file. As long as you retain this
* notice you can do whatever you want with this stuff. If we meet
* some day, and you think this stuff is worth it, you can buy me a
* beer in return. Poul-Henning Kamp

b) snprintf replacement

* Copyright Patrick Powell 1995
* This code is based on code written by Patrick Powell
* (papowell@astart.com) It may be used for any purpose as long as this
* notice remains intact on all source code distributions

c) Compatibility code (openbsd-compat)

Apart from the previously mentioned licenses, various pieces of code
in the openbsd-compat/ subdirectory are licensed as follows:

Some code is licensed under a 3-term BSD license, to the following
copyright holders:

Todd C. Miller
Theo de Raadt
Damien Miller
Eric P. Allman
The Regents of the University of California
Constantin S. Svintsoff

* Redistribution and use in source and binary forms, with or without

(*18) Other:

IBM PUBLIC LICENSE VERSION 1.0 - SECURE MAILER

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS IBM PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of International Business Machines Corporation ("IBM"), the Original Program, and

b) in the case of each Contributor,

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor.

A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.

Contributions do not include additions to the Program which:

(i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and

(ii) are not derivative works of the Program.

"Contributor" means IBM and any other entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Original Program" means the original version of the software accompanying this Agreement as released by IBM, including source code, object code and documentation, if any.

"Program" means the Original Program and Contributions.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form.

This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents.

The patent license shall not apply to any other combinations which include the Contribution.

No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity.

Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Each Contributor must include the following in a conspicuous location in the Program:

Copyright (c) 1997,1998,1999, International Business Machines
Corporation and others. All Rights Reserved.

In addition, each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like.

While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim,

and

b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone.

Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed.

In addition, If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under
this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

IBM may publish new versions (including revisions) of this Agreement from time to time.

Each new version of the Agreement will be given a distinguishing version number.

The Program (including Contributions) may always be distributed subject to the version of the Agreement under
which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. No one other than IBM has the right to modify this Agreement. Except as expressly stated in Sections
2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise.

All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights
to a jury trial in any resulting litigation.

(*19) Other:

*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

(*20) Other:

* Copyright (C) 1992 by Theodore Ts'o.
*
* Redistribution of this file is permitted under the terms of the GNU
* Public License (GPL)

(*21) Other:

Sudo is distributed under the following ISC-style license:

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F39502-99-1-0512.

Additionally, fnmatch.c, fnmatch.h, getcwd.c, glob.c, glob.h and snprintf.cbear the following UCB license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

nonunix.h and vasgroups.c bear the following license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of Quest Software, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(*22) Other:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION,
THE IMPLIEDWARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

(*23) Other:

4.4BSD-Lite2

(*24) Other:

(*25) Other:

VIM LICENSE

I) There are no restrictions on distributing unmodified copies of Vim except that they must include this license text. You can also distribute unmodified parts of Vim, likewise unrestricted except that they must include this license text. You are also allowed to include executables that you made from the unmodified Vim sources, plus your own usage examples and Vim scripts.

II) It is allowed to distribute a modified (or extended) version of Vim, including executables and/or source code, when the following four conditions are met:

1) This license text must be included unmodified.

2) The modified Vim must be distributed in one of the following five ways:

a) If you make changes to Vim yourself, you must clearly describe in the distribution how to contact you. When the maintainer asks you (in any way) for a copy of the modified Vim you distributed, you must make your changes, including source code, available to the maintainer without fee. The maintainer reserves the right to include your changes in the official version of Vim. What the maintainer will do with your changes and under what license they will be distributed is negotiable. If there has been no negotiation then this license, or a later version, also applies to your changes.

The current maintainer is Bram Moolenaar <Bram@vim.org>. If this changes it will be announced in appropriate places (most likely vim.sf.net, www.vim.org and/or comp.editors). When it is completely impossible to contact the maintainer, the obligation to send him your changes ceases. Once the maintainer has confirmed that he has received your changes they will not have to be sent again.

b) If you have received a modified Vim that was distributed as mentioned under a) you are allowed to further distribute it unmodified, as mentioned at I). If you make additional changes the text under a) applies to those changes.

c) Provide all the changes, including source code, with every copy of the modified Vim you distribute. This may be done in the form of a context diff. You can choose what license to use for new code you add. The changes and their license must not restrict others from making their own changes to the official version of Vim.

d) When you have a modified Vim which includes changes as mentioned under c), you can distribute it without the source code for the changes if the following three conditions are met:

- The license that applies to the changes permits you to distribute the changes to the Vim maintainer without fee or restriction, and permits the Vim maintainer to include the changes in the official version of Vim without fee or restriction.

- You keep the changes for at least three years after last distributing the corresponding modified Vim. When the maintainer or someone who you distributed the modified Vim to asks you (in any way) for the changes within this period, you must make them available to him.

- You clearly describe in the distribution how to contact you. This contact information must remain valid for at least three years after last distributing the corresponding modified Vim, or as long as possible.

e) When the GNU General Public License (GPL) applies to the changes, you can distribute the modified Vim under the GNU GPL version 2 or any later version.

3) A message must be added, at least in the output of the ":version" command and in the intro screen, such that the user of the modified Vim is able to see that it was modified. When distributing as mentioned under 2)e) adding the message is only required for as far as this does not conflict with the license used for the changes.

4) The contact information as required under 2)a) and 2)d) must not be removed or changed, except that the person himself can make corrections.

III) If you distribute a modified version of Vim, you are encouraged to use the Vim license for your changes and make them available to the maintainer, including the source code. The preferred way to do this is by e-mail or by uploading the files to a server and e-mailing the URL. If the number of changes is small (e.g., a modified Makefile) e-mailing a context diff will do. The e-mail address to be used is <maintainer@vim.org>

IV) It is not allowed to remove this license from the distribution of the Vim sources, parts of it or from a modified version. You may use this license for previous Vim releases instead of the license that they came with, at your

(*26) Other:

License: BSD

(*27) Other:

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

(*28) Other:

```
******************************************************************************
* Copyright © 2006 <University of Chicago and other copyright holders>. All
* rights reserved.
******************************************************************************
*
******************************************************************************
* vxi11.x is distributed subject to the following license conditions:
* SOFTWARE LICENSE AGREEMENT
* Software: vxi11.x
*
* 1. The "Software", below, refers to vxi11.x (in either source code, or
*     binary form and accompanying documentation). Each licensee is addressed
*     as "you" or "Licensee."
*
* 2. The copyright holders shown above and their third-party licensors hereby
*     grant Licensee a royalty-free nonexclusive license, subject to the
*     limitations stated herein and U.S. Government license rights.
*
* 3. You may modify and make a copy or copies of the Software for use within
*     your organization, if you meet the following conditions:
*         1. Copies in source code must include the copyright notice and this
*             Software License Agreement.
*         2. Copies in binary form must include the copyright notice and this
*              Software License Agreement in the documentation and/or other
*              materials provided with the copy.
*
* 4. You may modify a copy or copies of the Software or any portion of it,
*     thus forming a work based on the Software, and distribute copies of such
*       work outside your organization, if you meet all of the following
*       conditions:
*         1. Copies in source code must include the copyright notice and this
*             Software License Agreement;
*         2. Copies in binary form must include the copyright notice and this
*             Software License Agreement in the documentation and/or other
*             materials provided with the copy;
*         3. Modified copies and works based on the Software must carry
*             prominent notices stating that you changed specified portions of
*             the Software.
*
* 5. Portions of the Software resulted from work developed under a U.S.
*     Government contract and are subject to the following license: the
*     Government is granted for itself and others acting on its behalf a
```

*   paid-up, nonexclusive, irrevocable worldwide license in this computer
*   software to reproduce, prepare derivative works, and perform publicly
*   and display publicly.
*
* 6. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF
*   ANY KIND. THE COPYRIGHT HOLDERS, THEIR THIRD PARTY LICENSORS, THE UNITED
*   STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND THEIR EMPLOYEES: (1)
*   DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
*   ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
*   PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY
*   OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE
*   SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE
*   PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION
*   UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.
*
* 7. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT HOLDERS, THEIR
*   THIRD PARTY LICENSORS, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF
*   ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT, INCIDENTAL,
*   CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF ANY KIND OR NATURE,
*   INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS OR LOSS OF DATA, FOR ANY
*   REASON WHATSOEVER, WHETHER SUCH LIABILITY IS ASSERTED ON THE BASIS OF
*   CONTRACT, TORT (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE,
*   EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE POSSIBILITY OF SUCH
*   LOSS OR DAMAGES.

(*29) Other:

Microsoft Public License (Ms-PL)

This license governs use of the accompanying software, the DotNetZip library ("the software"). If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

3. Conditions and Limitations

(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

(*30) Other:

7-Zip source code
~~~~~~~~~~~~~~~~~
License for use and distribution
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

7-Zip Copyright (C) 1999-2010 Igor Pavlov.

Licenses for files are:

1) CPP/7zip/Compress/Rar* files:　GNU LGPL + unRAR restriction
2) All other files:　GNU LGPL

The GNU LGPL + unRAR restriction means that you must follow both
GNU LGPL rules and unRAR restriction rules.

GNU LGPL information

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.　See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA　02111-1307　USA

unRAR restriction

The decompression engine for RAR archives was developed using source
code of unRAR program.
All copyrights to original unRAR code are owned by Alexander Roshal.

The license for original unRAR code has the following restriction:

The unRAR sources cannot be used to re-create the RAR compression algorithm,
which is proprietary. Distribution of modified unRAR sources in separate form
or as a part of other software is permitted, provided that it is clearly
stated in the documentation and source comments that the code may
not be used to develop a RAR (WinRAR) compatible archiver.

--

Igor Pavlov

(*31) Other:

```
******    *****   ******   unRAR - free utility for RAR archives
 **   **  **   **  **   **   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 ******   *******  ******    License for use and distribution of
 **   **  **   **  **   **   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 **   **  **   **  **   **         FREE portable version
                                   ~~~~~~~~~~~~~~~~~~~~~
```

The source code of unRAR utility is freeware. This means:

1. All copyrights to RAR and the utility unRAR are exclusively owned by the author - Alexander Roshal.

2. The unRAR sources may be used in any software to handle RAR archives without limitations free of charge, but cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver.

3. The unRAR utility may be freely distributed. No person or company may charge a fee for the distribution of unRAR without written permission from the copyright holder.

4. THE RAR ARCHIVER AND THE UNRAR UTILITY ARE DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.

5. Installing and using the unRAR utility signifies acceptance of these terms and conditions of the license.

6. If you don't agree with terms of the license you must remove unRAR files from your storage devices and cease to use the utility.

Thank you for your interest in RAR and unRAR.

Alexander L. Roshal

Appendix C Sample format for performance test result sheet

Use the following test result sheet when performing the MU887000A performance tests. Duplicate this sheet as necessary for tests.

Appendix

Appendix C

C.1 Test Condition

Test location	________________	Document No.	________________
	________________	Date	________________
	________________	Person-in-charge	________________
	________________	Ambient temperature	________________ °C
	________________	Relative humidity	________________ %

Model Name ________________________________

Serial number ________________

Option ________________

Model Name ________________________________

Serial number ________________

Option ________________

Model Name ________________________________

Serial number ________________

Option ________________

Model Name ________________________________

Serial number ________________

Option ________________

Model Name ________________________________

Serial number ________________

Option ________________

Remarks ________________________________

____

____

C.2 VSWR

Performance test result sheets of VSWR are common to Port1 and Port2, Port3 and Port4.
When using the result sheet, circle the number of port to measure.

The table number of result sheet to use by measurement condition is shown in the next table.

Table C.2-1 Measurement condition and the table number of result sheet

Measurement condition	Table number
Port1	Table C.2-2
Port2	Table C.2-2
Port3 (Output port setting)	Table C.2-3
Port3 (Input port setting)	Table C.2-3
Port4 (Output port setting)	Table C.2-3
Port4 (Input port setting)	Table C.2-3

Table C.2-2 VSWR (Port1, Port2)

Test port: Port 1 2

Frequency (MHz) f	Worst VSWR Frequency (MHz)	Worst VSWR	Spec.
$10 \le f < 400$			<1.5
$400 \le f \le 2700$			<1.2
$2700 < f \le 3800$			<1.3
$3800 < f \le 6000$			<1.5

Table C.2-3 VSWR (Port3, Port4)

Test port: Port 3 4

Frequency (MHz) f	Worst VSWR Frequency (MHz)	Worst VSWR	Spec.
$10 \le f < 30$			<1.8
$30 \le f \le 3800$			<1.5
$3800 < f \le 6000$			<1.8

C.3 SG Level Accuracy

Performance test result sheets of SG Level Accuracy are common to Port1 and Port2, Port3 and Port4. When using the result sheet, circle the number of port to measure.

The table number of result sheet to use by port and frequency is shown in the next table.

Table C.3-1 Measurement condition and the table number of result sheet

Frequency*1 (MHz)	MU887000A Setting Frequency*2 (MHz)	Port1	Port2	Port3	Port4
10	10.1	Table C.3-2	Table C.3-2	Table C.3-5	Table C.3-5
350	350.1	Table C.3-2	Table C.3-2	Table C.3-5	Table C.3-5
800	800.1	Table C.3-3	Table C.3-3	Table C.3-6	Table C.3-6
1000	1000.1	Table C.3-3	Table C.3-3	Table C.3-6	Table C.3-6
1500	1500.1	Table C.3-3	Table C.3-3	Table C.3-6	Table C.3-6
1800	1800.1	Table C.3-3	Table C.3-3	Table C.3-6	Table C.3-6
2000	2000.1	Table C.3-3	Table C.3-3	Table C.3-6	Table C.3-6
2200	2200.1	Table C.3-3	Table C.3-3	Table C.3-6	Table C.3-6
3000	3000.1	Table C.3-3	Table C.3-3	Table C.3-6	Table C.3-6
3800	3799.9	Table C.3-3	Table C.3-3	Table C.3-6	Table C.3-6
4000	4000.1	Table C.3-4	Table C.3-4	Table C.3-7	Table C.3-7
5000	5000.1	Table C.3-4	Table C.3-4	Table C.3-7	Table C.3-7
6000	5999.9	Table C.3-4	Table C.3-4	Table C.3-7	Table C.3-7

*1: The frequency to write in the leftmost column (Frequency) in the performance test result sheet

*2: The frequency to write in the second column from left (MU887000A Settings - Frequency) in the performance test result sheet

Table C.3-2 SG Level Accuracy

<u>Test port: Port 1 2</u>

Frequency (MHz)	MU887000A Settings: Frequency (MHz)	MU887000A Settings: Output Level (dBm)	Power Meter: Level (dBm)	Signal Analyzer: Setting	Signal Analyzer: Center Frequency	Signal Analyzer: #x→#1 offset (dB)	Signal Analyzer: Level (dBm)	Signal Analyzer: Converted Level to #1(dBm)	MU887000A: Accuracy(dB)	MU887000A: Spec. (dB)
		−10 = P1		#1	F0			B1	= (B1 - A + C) - P1	±1.3
		−15 = P2						= B2	= (B2 - A + C) - P2	±1.3
		−20 = P3	= C					= A = B3	= C - P3	±1.3
		−25 = P4						= B4	= (B4 - A + C) - P4	±1.3
		−30 = P5						= B5	= (B5 - A + C) - P5	±1.3
		−35 = P6						= B6	= (B6 - A + C) - P6	±1.3
		−40 = P7						= B7	= (B7 - A + C) - P7	±1.3
		−45 = P8						= B8	= (B8 - A + C) - P8	±1.3
		−50 = P9		#2			= G9	= B9 = G9 + D	= (B9 - A + C) - P9	±1.3
		−55 = P10					= G10	= B10 = G10 + D	= (B10 - A + C) - P10	±1.3
	= F0	−60 = P11				= D	= G11	= B11 = G11 + D	= (B11 - A + C) - P11	±1.3

Table C.3-2 SG Level Accuracy (Cont'd)

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		−65 = P12		#2	F0		= G12	= B12 = G12 + D	= (B12 - A + C) - P12	±1.3
		−70 = P13					= G13	= B13 = G13 + D	= (B13 - A + C) - P13	±1.3
		−75 = P14					= G14	= B14 = G14 + D	= (B14 - A + C) - P14	±1.3
		−80 = P15		#3			= G15	= B15 = G15 + D	= (B15 - A + C) - P15	±1.3
		−85 = P16					= G16	= B16 = G16 + D	= (B16 - A + C) - P16	±1.3
		−90 = P17					= G17	= B17 = G17 + D	= (B17 - A + C) - P17	±1.3
		−95 = P18					= G18	= B18 = G18 + D	= (B18 - A + C) - P18	±1.3
		−100 = P19					= G19	= B19 = G19 + D	= (B19 - A + C) - P19	±1.3
		−105 = P20		#4	F0 + 100 Hz		= G20	= B20 = G20 + D	= (B20 - A + C) - P20	±1.3
		−110 = P21					= G21	= B21 = G21 + D	= (B21 - A + C) - P21	±1.3
		−115 = P22		#5			= G22	= B22 = G22 + D	= (B22 - A + C) - P22	±1.3
	= F0	−120 = P23				= D	= G23	= B23 = G23 + D	= (B23 - A + C) - P23	±1.3

Table C.3-3 SG Level Accuracy

<u>Test port: Port 1 2</u>

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		−10 = P1		#1	F0			B1	= (B1 - A + C) - P1	±1.0
		−15 = P2						= B2	= (B2 - A + C) - P2	±1.0
		−20 = P3	= C					= A = B3	= C - P3	±1.0
		−25 = P4						= B4	= (B4 - A + C) - P4	±1.0
		−30 = P5						= B5	= (B5 - A + C) - P5	±1.0
		−35 = P6						= B6	= (B6 - A + C) - P6	±1.0
		−40 = P7						= B7	= (B7 - A + C) - P7	±1.0
		−45 = P8						= B8	= (B8 - A + C) - P8	±1.0
		−50 = P9		#2			= G9	= B9 = G9 + D	= (B9 - A + C) - P9	±1.0
		−55 = P10					= G10	= B10 = G10 + D	= (B10 - A + C) - P10	±1.0
	= F0	−60 = P11				= D	= G11	= B11 = G11 + D	= (B11 - A + C) - P11	±1.0

Table C.3-3 SG Level Accuracy (Cont'd)

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		−65 = P12		#2	F0		= G12	= B12 = G12 + D	= (B12 - A + C) - P12	±1.0
		−70 = P13					= G13	= B13 = G13 + D	= (B13 - A + C) - P13	±1.0
		−75 = P14					= G14	= B14 = G14 + D	= (B14 - A + C) - P14	±1.0
		−80 = P15		#3			= G15	= B15 = G15 + D	= (B15 - A + C) - P15	±1.0
		−85 = P16					= G16	= B16 = G16 + D	= (B16 - A + C) - P16	±1.0
		−90 = P17					= G17	= B17 = G17 + D	= (B17 - A + C) - P17	±1.0
		−95 = P18					= G18	= B18 = G18 + D	= (B18 - A + C) - P18	±1.0
		−100 = P19					= G19	= B19 = G19 + D	= (B19 - A + C) - P19	±1.0
		−105 = P20		#4	F0 + 100 Hz		= G20	= B20 = G20 + D	= (B20 - A + C) - P20	±1.0
		−110 = P21					= G21	= B21 = G21 + D	= (B21 - A + C) - P21	±1.0
		−115 = P22		#5			= G22	= B22 = G22 + D	= (B22 - A + C) - P22	±1.0
	= F0	−120 = P23				= D	= G23	= B23 = G23 + D	= (B23 - A + C) - P23	±1.0

Table C.3-4 SG Level Accuracy

Test port: Port 1 2

Frequency (MHz)	MU887000A Settings: Frequency (MHz)	MU887000A Settings: Output Level (dBm)	Power Meter: Level (dBm)	Signal Analyzer: Setting	Signal Analyzer: Center Frequency	Signal Analyzer: #x→#1 offset (dB)	Signal Analyzer: Level (dBm)	Signal Analyzer: Converted Level to #1(dBm)	MU887000A: Accuracy(dB)	MU887000A: Spec. (dB)
		−18 = P1		#1	F0			B1	= (B1 - A + C) - P1	±1.3
		−23 = P2						= B2	= (B2 - A + C) - P2	±1.3
		−28 = P3	= C					= A = B3	= C - P3	±1.3
		−33 = P4						= B4	= (B4 - A + C) - P4	±1.3
		−38 = P5						= B5	= (B5 - A + C) - P5	±1.3
		−43 = P6						= B6	= (B6 - A + C) - P6	±1.3
		−48 = P7						= B7	= (B7 - A + C) - P7	±1.3
		−53 = P8		#2		= D	= G8	= B8 = G8 + D	= (B8 - A + C) - P8	±1.3
		−58 = P9					= G9	= B9 = G9 + D	= (B9 - A + C) - P9	±1.3
		−63 = P10					= G10	= B10 = G10 + D	= (B10 - A + C) - P10	±1.3
	= F0	−68 = P11					= G11	= B11 = G11 + D	= (B11 - A + C) - P11	±1.3

Table C.3-4 SG Level Accuracy (Cont'd)

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		−73 = P12		#2	F0		= G12	= B12 = G12 + D	= (B12 - A + C) - P12	±1.3
		−78 = P13					= G13	= B13 = G13 + D	= (B13 - A + C) - P13	±1.3
		−83 = P14		#3			= G14	= B14 = G14 + D	= (B14 - A + C) - P14	±1.3
		−88 = P15					= G15	= B15 = G15 + D	= (B15 - A + C) - P15	±1.3
		−93 = P16					= G16	= B16 = G16 + D	= (B16 - A + C) - P16	±1.3
		−98 = P17					= G17	= B17 = G17 + D	= (B17 - A + C) - P17	±1.3
		−100 = P18				= D	= G18	= B18 = G18 + D	= (B18 - A + C) - P18	±1.3
	= F0									

Table C.3-5 SG Level Accuracy

Test port: Port 3 4

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		0 = P1		#1	F0			B1	= (B1 - A + C) - P1	±1.3
		–5 = P2						= B2	= (B2 - A + C) - P2	±1.3
		–10 = P3	= C					= A = B3	= C - P3	±1.3
		–15 = P4						= B4	= (B4 - A + C) - P4	±1.3
		–20 = P5						= B5	= (B5 - A + C) - P5	±1.3
		–25 = P6						= B6	= (B6 - A + C) - P6	±1.3
		–30 = P7						= B7	= (B7 - A + C) - P7	±1.3
		–35 = P8						= B8	= (B8 - A + C) - P8	±1.3
		–40 = P9						= B9	= (B9 - A + C) - P9	±1.3
		–45 = P10						= B10	= (B10 - A + C) - P10	±1.3
	= F0	–50 = P11		#2		= D	= G11	= B11 = G11 + D	= (B11 - A + C) - P11	±1.3

Table C.3-5 SG Level Accuracy (Cont'd)

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		−55 = P12		#2	F0		= G12	= B12 = G12 + D	= (B12 - A + C) - P12	±1.3
		−60 = P13					= G13	= B13 = G13 + D	= (B13 - A + C) - P13	±1.3
		−65 = P14					= G14	= B14 = G14 + D	= (B14 - A + C) - P14	±1.3
		−70 = P15					= G15	= B15 = G15 + D	= (B15 - A + C) - P15	±1.3
		−75 = P16					= G16	= B16 = G16 + D	= (B16 - A + C) - P16	±1.3
		−80 = P17		#3			= G17	= B17 = G17 + D	= (B17 - A + C) - P17	±1.3
		−85 = P18					= G18	= B18 = G18 + D	= (B18 - A + C) - P18	±1.3
		−90 = P19					= G19	= B19 = G19 + D	= (B19 - A + C) - P19	±1.3
		−95 = P20					= G20	= B20 = G20 + D	= (B20 - A + C) - P20	±1.3
		−100 = P21					= G21	= B21 = G21 + D	= (B21 - A + C) - P21	±1.3
		−105 = P22		#4	F0 + 100 Hz		= G22	= B22 = G22 + D	= (B22 - A + C) - P22	±1.3
	= F0	−110 = P23				= D	= G23	= B23 = G23 + D	= (B23 - A + C) - P23	±1.3

Table C.3-6 SG Level Accuracy

Test port: Port 3 4

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		0 = P1		#1	F0		B1		= (B1 - A + C) - P1	±1.0
		–5 = P2					= B2		= (B2 - A + C) - P2	±1.0
		–10 = P3	= C				= A = B3		= C - P3	±1.0
		–15 = P4					= B4		= (B4 - A + C) - P4	±1.0
		–20 = P5					= B5		= (B5 - A + C) - P5	±1.0
		–25 = P6					= B6		= (B6 - A + C) - P6	±1.0
		–30 = P7					= B7		= (B7 - A + C) - P7	±1.0
		–35 = P8					= B8		= (B8 - A + C) - P8	±1.0
		–40 = P9					= B9		= (B9 - A + C) - P9	±1.0
		–45 = P10					= B10		= (B10 - A + C) - P10	±1.0
	= F0	–50 = P11		#2		= D	= G11	= B11 = G11 + D	= (B11 - A + C) - P11	±1.0

Table C.3-6 SG Level Accuracy (Cont'd)

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		−55 = P12		#2	F0		= G12	= B12 = G12 + D	= (B12 - A + C) - P12	±1.0
		−60 = P13					= G13	= B13 = G13 + D	= (B13 - A + C) - P13	±1.0
		−65 = P14					= G14	= B14 = G14 + D	= (B14 - A + C) - P14	±1.0
		−70 = P15					= G15	= B15 = G15 + D	= (B15 - A + C) - P15	±1.0
		−75 = P16					= G16	= B16 = G16 + D	= (B16 - A + C) - P16	±1.0
		−80 = P17		#3			= G17	= B17 = G17 + D	= (B17 - A + C) - P17	±1.0
		−85 = P18					= G18	= B18 = G18 + D	= (B18 - A + C) - P18	±1.0
		−90 = P19					= G19	= B19 = G19 + D	= (B19 - A + C) - P19	±1.0
		−95 = P20					= G20	= B20 = G20 + D	= (B20 - A + C) - P20	±1.0
		−100 = P21					= G21	= B21 = G21 + D	= (B21 - A + C) - P21	±1.0
		−105 = P22		#4	F0 + 100 Hz		= G22	= B22 = G22 + D	= (B22 - A + C) - P22	±1.0
	= F0	−110 = P23				= D	= G23	= B23 = G23 + D	= (B23 - A + C) - P23	±1.0

Table C.3-7 SG Level Accuracy

Test port: Port 3 4

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		−8 = P1						B1	= (B1 - A + C) - P1	±1.3
		−13 = P2						= B2	= (B2 - A + C) - P2	±1.3
		−18 = P3	= C					= A = B3	= C - P3	±1.3
		−23 = P4		#1				= B4	= (B4 - A + C) - P4	±1.3
		−28 = P5						= B5	= (B5 - A + C) - P5	±1.3
		−33 = P6			F0			= B6	= (B6 - A + C) - P6	±1.3
		−38 = P7						= B7	= (B7 - A + C) - P7	±1.3
		−43 = P8						= B8	= (B8 - A + C) - P8	±1.3
		−48 = P9						= B9	= (B9 - A + C) - P9	±1.3
		−53 = P10		#2			= G10	= B10 = G10 + D	= (B10 - A + C) - P10	±1.3
	= F0	−58 = P11				= D	= G11	= B11 = G11 + D	= (B11 - A + C) - P11	±1.3

Table C.3-7 SSG Level Accuracy (Cont'd)

Frequency (MHz)	MU887000A Settings		Power Meter	Signal Analyzer					MU887000A	
	Frequency (MHz)	Output Level (dBm)	Level (dBm)	Setting	Center Frequency	#x→#1 offset (dB)	Level (dBm)	Converted Level to #1(dBm)	Accuracy(dB)	Spec. (dB)
		−63 = P12		#2	F0		= G12	= B12 = G12 + D	= (B12 - A + C) - P12	±1.3
		−68 = P13					= G13	= B13 = G13 + D	= (B13 - A + C) - P13	±1.3
		−73 = P14					= G14	= B14 = G14 + D	= (B14 - A + C) - P14	±1.3
		−78 = P15					= G15	= B15 = G15 + D	= (B15 - A + C) - P15	±1.3
		−83 = P16		#3			= G16	= B16 = G16 + D	= (B16 - A + C) - P16	±1.3
		−88 = P17					= G17	= B17 = G17 + D	= (B17 - A + C) - P17	±1.3
		−93 = P18					= G18	= B18 = G18 + D	= (B18 - A + C) - P18	±1.3
		−98 = P19					= G19	= B19 = G19 + D	= (B19 - A + C) - P19	±1.3
		−103 = P20		#4	F0 + 100 Hz		= G20	= B20 = G20 + D	= (B20 - A + C) - P20	±1.3
		−108 = P21					= G21	= B21 = G21 + D	= (B21 - A + C) - P21	±1.3
		−110 = P22				= D	= G22	= B22 = G22 + D	= (B22 - A + C) - P22	±1.3
	= F0									

C.4 SG Spurious

Performance test result sheets of SG Spurious are common to Port1 and Port3.
When using the result sheet, circle the number of port to measure.

The value to write in column of “Output Level” is 0 for Port1, –10 for Port3.

Table C.4-1 SG Spurious

Test port: Port 1 3

MU887000A Settings		Signal Analyzer					Spec. (dBc)
Output Level (dBm)	Frequency (MHz)	Fundamental Level (dBm) A	HD2 Level (dBm) B	HD3 Level (dBm) C	HD2 (dBc) = B – A	HD3 (dBc) = C – A	
	10						< –25
	100						< –25
	200						< –25
	300						< –25
	400						< –25
	500						< –25
	600						< –25
	700						< –25
	800						< –25
	900						< –25
	1000						< –25
	1100						< –25
	1200						< –25
	1300						< –25
	1400						< –25
	1500						< –25
	1600						< –25
	1700						< –25
	1800						< –25
	1900						< –25
	2000						< –25
	2100						< –25
	2200						< –25
	2300						< –25
	2400						< –25
	2500						< –25
	2600						< –25
	2700						< –25
	2800						< –25
	2900						< –25
	3000						< –25

C.5 SA Level Accuracy

Performance test result sheets of SA Level Accuracy are common to Port1 and Port2.
When using the result sheet, circle the number of port to measure.

The table number of result sheet to use by port and frequency is shown in the next table.

Table C.5-1 Measurement condition and the table number of result sheet

MU887000A Setting Frequency (MHz)	MG3700A Setting Frequency (MHz)	Port1	Port2
1000	999.9	Table C.5-2	Table C.5-2
2000	1999.9	Table C.5-2	Table C.5-2
3000	2999.9	Table C.5-2	Table C.5-2
3800	3799.9	Table C.5-2	Table C.5-2

Table C.5-2 SA Level Accuracy

Test port: Port 1 2

MU887000A Settings		Signal Generator Settings		Power Meter	Signal Analyzer	MU887000A		
Frequency (MHz)	Input Level (dBm)	Frequency (MHz)	Level (dBm)	Level (dBm)	Level (dBm)	Level (dBm)	Accuracy (dB)	Spec. (dB)
	+35					= A1	= A1 - C0	±0.5
	+30					= A2	= A2 - C0	±0.5
	+25					= A3	= A3 - C0	±0.5
	+20					= A4	= A4 - C0	±0.5
	+15					= A5	= A5 - C0	±0.5
	+10					= A6	= A6 - C0	±0.5
	+5					= A7	= A7 - C0	±0.5
	0		= P0	= C0 = approx. 0dBm		= A8	= A8 - C0	±0.5
	–5					= A9	= A9 - C30	±0.5
	–10					= A10	= A10 - C30	±0.5
	–15					= A11	= A11 - C30	±0.5
	–20					= A12	= A12 - C30	±0.5
	–25					= A13	= A13 - C30	±0.5
	–30		= P30	= C30 =approx. -30dBm		= A14	= A14 - C30	±0.5
			= P25	= C25 =approx. -25dBm	= S			
	–35					= A15	= A15 - (C25 + S1 - S)	±0.7
	–40					= A16	= A16 - (C25 + S1 - S)	±0.7
	–45					= A17	= A17 - (C25 + S1 - S)	±0.7
	–50		P25 – 10			= A18	= A18 - (C25 + S1 - S)	±0.7
	–55					= A19	= A19 - (C25 + S1 - S)	±0.7
	–60					= A20	= A20 - (C25 + S1 - S)	±0.9
	–65				= S1	= A21	= A21 - (C25 + S1 - S)	±0.9

C.6 SA Linearity

Performance test result sheets of SA Linearity are common to Port1 and Port2.
When using the result sheet, circle the number of port to measure.

The table number of result sheet to use by port and frequency is shown in the next table.

Table C.6-1 Measurement condition and the table number of result sheet

MU887000A Setting Frequency (MHz)	Input Level (dBm)	Port1	Port2
1000	0	Table C.6-2	Table C.6-2
2000	0	Table C.6-2	Table C.6-2
3000	0	Table C.6-2	Table C.6-2
1000	–25	Table C.6-3	Table C.6-3
2000	–25	Table C.6-3	Table C.6-3
3000	–25	Table C.6-3	Table C.6-3

Table C.6-2 SA Linearity (0 dBm)

<u>Test port: Port 1 2</u>

MU887000A Settings		Signal Generator Settings		Power Meter	Signal Analyzer	MU887000A		
Frequency (MHz)	Input Level (dBm)	Frequency (MHz)	Level (dBm)	Level (dBm)	Level (dBm)	Level (dBm)	Accuracy (dB)	Spec. (dB)
	0		= P	= approx. 0dBm	= S	= A	–	–
			P – 10		= S1	= B1	= (B1 - A) - (S1 - S)	±0.2
			P – 20		= S2	= B2	= (B2 - A) - (S2 - S)	±0.2
			P – 30		= S3	= B3	= (B3 - A) - (S3 - S)	±0.2
			P – 40		= S4	= B4	= (B4 - A) - (S4 - S)	±0.2

Table C.6-3 SA Linearity (–25 dBm)

<u>Test port: Port 1 2</u>

MU887000A Settings		Signal Generator Settings		Power Meter	Signal Analyzer	MU887000A		
Frequency (MHz)	Input Level (dBm)	Frequency (MHz)	Level (dBm)	Level (dBm)	Level (dBm)	Level (dBm)	Accuracy (dB)	Spec. (dB)
	–25		= P	= approx. –25dBm	= S	= A	–	–
			P – 10		= S1	= B1	= (B1 - A) - (S1 - S)	±0.2
			P – 20		= S2	= B2	= (B2 - A) - (S2 - S)	±0.2
			P – 30		= S3	= B3	= (B3 - A) - (S3 - S)	±0.2
			P – 40		= S4	= B4	= (B4 - A) - (S4 - S)	±0.4

Appendix D Status indication of lamps

This appendix explains the status indication of lamps. For the position of lamps, refer to Figure 2.1-1 “Front Panel Part Names”.

Table D-1 Status indication of lamps

Name	Description	
Status	Red:	Self-test error or temperature error occurred.
	Red blink:	Error occurred when executing level calibration.
	Orange blink:	Unexpected error occurred.*1
	Green:	Normal*2
	Green blink:	Error in the module software occurred at the power-on process*1, at the power-off process.
Remote	Green:	Remote controlled
	Green blink:	Error occurred by the remote controlling*3
	Off:	Not remote controlled
1	Green blink:	Updating software*1
	Off:	No error
2	Red:	Error occurred when loading Waveform file.
	Green:	RF signal output is on.
	Green blink:	Updating software*1, Loading the waveform file
	Off:	RF signal output is off.
3	Red:	Error occurred when executing the measurement or analysis by the application software*4
	Green:	Executing the measurement or analysis by the application software.*4
	Green blink:	Updating software*1
	Off:	No error
4	Orange:	Output signal from Audio Port is deformed.
	Green:	Signal output from Audio Port is On.
	Green blink:	Updating software*1
	Off:	Signal output from Audio Port is Off.
5	Red:	Reception error of Audio Port occurred.
	Green:	Audio Port is receiving signal.
	Green blink:	Updating software*1
	Off:	Audio Port stops receiving signal.
6	Green blink:	Updating software*1
	Off:	No error
Input lamp	On at signal input via connector All input lamps are on during level calibration execution.	
Output lamp	On at signal output via connector All output lamps are on during level calibration execution.	

*1: The module cannot be controlled via remote interface.

*2: Remote control is enabled when the green light is lit after the power is turned On. The machine makes a peep-peep sound at that time

*3: If buzzer setting is on, one short audible signal is emitted when the Remote lamp is lit.
Error information can be obtained by next commands.
:SYSTem:ERRor?,SYSERR?

*4: Refer to Table 1.3.3-1 “Application Software” for the model of application software.

Note:

When the power is supplied, all the lamps are lit orange.

When the power is turned off, all the lamps are lit orange. When the power is turned off, sometimes status lamp blinks at orange. This is not error.

Action when the error is indicated

When the Status lamp is red, orange blink or green blink, turn off the power and turn on the power again. If phenomena do not change, contact an Anritsu Service and Sales office. Contact information is available in a separate file (for the PDF version), and on the last page of this manual (for the printed version).

When the status indication lamp 2 is lit red, query the cause by using following command.
SCPI command mode:

```
:SOURce:GPRF:GENerator:ARB:FILE:LOAD? <file_name>
```

Native command mode:

```
SOUR:GPRF:GEN:ARB:FILE:LOAD? <file_name>
```

When the measurement error occurs, the status indication lamp 3 of MU887000A is lit in red.
In that case, query the cause by using command in the next table.
For the command explanation, refer to the operation manual of the application software.

Table D-2 Commands to query the cause

Model of Application Software	Command
MX887010A, MX887011A, MX887012A, MX887013A, MX887014A, MX887015A, MX887016A, MX887017A, MX887065A, MX887067A	`MSTAT?` `:FETCh:CELLular:MEASurement:STATe?`
MX887030A, MX887031A, MX887032A, MX887033A, MX887040A, MX887050A	`STAT:SRW:MEAS?` `:STATus:SRWireless:MEASurement?`
MX887060A	`STAT:LRWP:MEAS?` `:STATus:LRWPan:MEASurement?`
MX887061A	`STAT:ZWAV:MEAS?` `:STATus:ZWAVe:MEASurement?`
MX887070A	`FETC:FMA:MEAS:STAT?` `:FETCh:FMAudio:MEASurement:STATe?`

When the status indication lamp 4 is lit orange, adjust output signal level of Audio Port.

Index

References are page numbers.

U

V

W