MU887002A
TRX Test Module
Operation Manual

Eighth Edition

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

ANRITSU CORPORATION

Document No.: M-W4048AE-8.0

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.

MU887002A
TRX Test Module
Operation Manual

20 August 2020 (First Edition)
14 February 2023 (Eighth Edition)

Copyright © 2020-2023, ANRITSU CORPORATION.
All rights reserved. No part of this manual may be reproduced without the prior written permission of the publisher.
The operational instructions of this manual may be changed without prior notice.
Printed in Japan

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.

⚠ CAUTION

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, and software bug fixes will be performed in accordance with the separate Software End-User License Agreement, provide, however, that 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 or flood, 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 damage or financial loss of the customer due to the use of or a failure to use this equipment, unless the damage or loss is caused due to Anritsu Corporation’s intentional or gross negligence.

Note:
For the purpose of this Warranty, "unusual environments" 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, please 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.
Also, the Anritsu website provides you with contact information and a contact form.

Notes On Export Management

- We assume no responsibility if this product is re-exported and used without our approval since it may not comply with the safety standards, etc., of specific countries.
- 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.
  For more details, visit
  https://www.anritsu.com/support/export-procedures.
  When you dispose of the products or manuals subject to export control, they must be physically destroyed or 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.

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 carefully read and accept this Software End-User License Agreement (hereafter this EULA) before using (includes executing, copying, installing, registering, etc.) this Software (includes programs, databases, scenarios, etc., used to operate, set, etc., Anritsu electronic equipment, etc.). By using this Software, you shall be deemed to have agreed to be bound by the terms of this EULA, 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.

Article 1. Grant of License and Limitations

1. You may not sell, transfer, rent, lease, lend, disclose, sublicense, or otherwise distribute this Software to third parties, whether or not paid therefor.
2. You may make one copy of this Software for backup purposes only.
3. You are not permitted to reverse engineer, disassemble, decompile, modify or create derivative works of this Software.
4. This EULA allows you to install one copy of this Software on one piece of Equipment.

Article 2. Disclaimers

To the extent not prohibited by law, in no event shall Anritsu be liable for direct, 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, and damages claimed by third parties, arising out of or related to your use or inability to use this Software, unless the damages are caused due to Anritsu's intentional or gross negligence.

Article 3. Limitation of Liability

1. If a fault (bug) is discovered in this Software, making this Software unable to operate as described in the operation manual or specifications even though you have used this Software as described in the manual, Anritsu shall at its own discretion, fix the bug, or replace the software, or suggest a workaround, free-of-charge, provided, however, that the faults caused by the following items and any of your lost or damaged data whatsoever shall be excluded from repair and the warranty.
   i) If this Software is deemed to be used for purposes not described in the operation manual or specifications.
   ii) If this Software has been used in conjunction with other non-Anritsu-approved software.
   iii) If this Software or the Equipment has been modified, repaired, or otherwise altered without Anritsu's prior approval.
   iv) For any other reasons out of Anritsu's direct control and responsibility, such as but not limited to, natural disasters, software virus infections, or any devices other than this Equipment, etc.
2. Expenses incurred for transport, hotel, daily allowance, etc., for on-site repairs or replacement by Anritsu engineers necessitated by the above faults shall be borne by you.
3. The warranty period for faults listed in Section 1 of this Article shall be either 6 months from the date of purchase of this Software or 30 days after the date of repair or replacement, whichever is longer.

Article 4. Export Restrictions

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

Article 5. Change of Terms

Anritsu may change without your approval the terms of this EULA if the changes are for the benefit of general customers, or are reasonable in light of the purpose of this EULA and circumstances of the changes. At the time of change, Anritsu will inform you of those changes and its effective date, as a general rule 45 days, in advance on its website, or in writing or by e-mail.

Article 6. Termination

1. Anritsu may terminate this EULA immediately if you violate any conditions described herein. This EULA shall also be terminated immediately by Anritsu if there is any good reason that it is deemed difficult to continue this EULA, such as your violation of Anritsu copyrights, patents, etc. or any laws and ordinances, or if it turns out that you belong to an antisocial organization or has a socially inappropriate relationship with members of such organization.
2. You and Anritsu may terminate this EULA by a written notice to the other party 30 days in advance.

Article 7. Damages

If Anritsu suffers any damages or 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.

Article 8. Responsibility after Termination

Upon termination of this EULA in accordance with Article 6, you shall cease all uses 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.

Article 9. Negotiation for Dispute Resolution

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

Article 10. Governing Law and Court of Jurisdiction

This EULA shall be governed by and interpreted in accordance with the laws of Japan without regard to the principles of the conflict of laws thereof, and any disputes arising from or in relation to this EULA that cannot be resolved by negotiation described in Article 9 shall be subject to and be settled by the exclusive agreed jurisdiction of the Tokyo District Court of Japan.

Revision History:

February 29th, 2020
December 17th, 2021

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 flash drive 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 flash drive 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 flash drive and CF memory card).
- Connecting to network
  Connect your computer to the network that provides adequate protection against computer viruses.
- Protection against malware (malicious software such as viruses).
  To connect your computer to network, the following is advised.
  - Activate Firewall.
  - Install important updates of Windows.
  - Use antivirus software.

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 in accordance with the Decision 768/2008/EC to indicate that it conforms to the EMC, LVD, and RoHS directive of the European Union (EU).

CE marking



1. Product Model

Plug-in Units: MU887002A TRX Test Module

2. Applied Directive and Standards

When the MU887002A TRX Test Module is installed in the MT8870A or MT8872A, the applied directive and standards of this unit conform to those of the MT8870A or MT8872A main frame.

PS: About main frame
Please contact Anritsu for the latest information on the main frame types that MU887002A can be used with.

UKCA Marking

Anritsu affixes the UKCA marking on the following product in accordance with the guidance to indicate that it conforms to the EMC, LVD, and RoHS regulations in the United Kingdom.

UKCA marking



1. Product Model

Model: MU887002A TRX Test Module

2. Applied Regulations and Standards

When the MU887002A TRX Test Module is installed in the MT8870A or MT8872A, the applied regulations and standards of this unit conform to those of the MT8870A or MT8872A main frame.

PS: About main frame
Please contact Anritsu for the latest information on the main frame types that MU887002A can be used with.

RCM Conformity Marking

Anritsu affixes the RCM mark on the following product in accordance with the regulation to indicate that it conforms to the EMC framework of Australia/New Zealand.

RCM marking



1. Product Model

Plug-in Units: MU887002A TRX Test Module

2. Applied Standards

When the MU887002A TRX Test Module is installed in the MT8870A or MT8872A, the applied directive and standards of this unit conform to those of the MT8870A or MT8872A main frame.

PS: About main frame

Please contact Anritsu for the latest information on the main frame types that MU887002A can be used with.

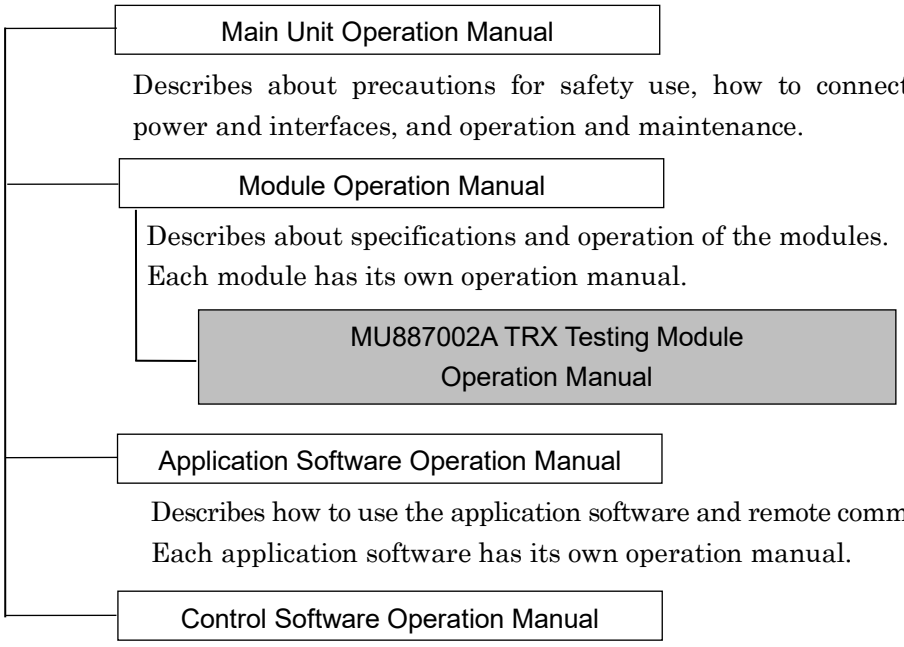
About This Manual

This manual mainly describes the use, panels, and specifications of the MU887002A 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 MT8870A
- Control software installed in a PC to control the MT8870A remotely

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

Convention Used in This Manual

Except where there is a good reason to do otherwise, this manual assumes the use of MT8870A.
Note that the descriptions in this manual also apply to MT8872A.

Additionally, the Anritsu management code of the options described in this manual is "0" unless there is a reason.

The option number: The option number is indicated by three digits.

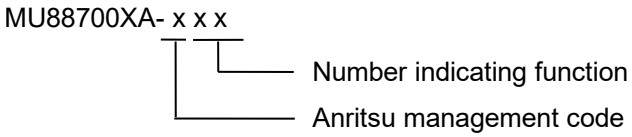


Table of Contents

For Safety iii

About This Manual I

Chapter 1 Outline 1-1

1.1 Overview 1-2
1.2 Features 1-3
1.3 Product Configuration 1-4
1.4 Abbreviations 1-9

Chapter 2 Before Use 2-1

2.1 Part Names and Functions 2-2
2.2 Installing Module 2-5
2.3 Confirming Connection 2-7
2.4 Electrostatic Countermeasures 2-9
2.5 Connecting Mobile 2-12

Chapter 3 Fundamental Operation 3-1

3.1 Command Language Modes 3-2
3.2 Configuring Communications 3-3
3.3 Transferring Files 3-8
3.4 System Settings 3-9
3.5 Preparing for Measurement 3-14
3.6 Status Registers 3-21

Chapter 4 Vector Signal Generator 4-1

4.1 Vector Signal Generator (VSG) Introduction 4-2
4.2 Operation Mode 4-3
4.3 Normal Mode 4-4
4.4 Sequence Mode 4-17

Chapter 5 SCPI Command Reference 5-1

5.1 List of Commands 5-2
5.2 Details of Commands 5-27

Chapter 6 Native Command Reference..... 6-1

6.1 List of Commands.....6-2
6.2 Details of Commands.....6-12

Chapter 7 Performance Tests 7-1

7.1 Outline.....7-2
7.2 Instruments for Performance Tests.....7-3
7.3 Performance Test for Each Measurement 7-8
7.4 Servicing.....7-46

Chapter 8 Utility Tool 8-1

8.1 Overview 8-2
8.2 Installing 8-9
8.3 Operating.....8-16

Appendix A Specifications A-1

Appendix B Software Licenses..... B-1

Appendix C Sample format for performance test result sheet C-1

Appendix D Status indication of lamps D-1

Appendix E MX887092A.....E-1

Index Index-1

Chapter 1 Outline

This chapter reviews the MU887002A TRX Test Module functions and product configurations. For performance specifications and functions, refer to Appendix A “Specifications”.

1.1 Overview 1-2
1.2 Features 1-3
1.3 Product Configuration 1-4
1.3.1 Standard configuration 1-4
1.3.2 Options 1-5
1.3.3 Application software 1-6
1.3.4 Optional accessories 1-8
1.3.5 Warranty service 1-8
1.4 Abbreviations 1-9

1.1 Overview

The MU887002A TRX Test Module (module hereafter) is installed in the MT8870A Universal Wireless Testing System (MT8870A hereafter).

- Slot size: 2 slots size 2Tx/2Rx
  Usable independently as 1 TRX/1 slot.
- Number of measurement port: 12 Port × 2 slot
- TRX frequency: 0.4 to 6 GHz
- Signal bandwidth: 200 MHz for both Downlink/Uplink
- Simultaneous output: 12 Port/1 TRX
- Installation: Can be installed simultaneously with the MU887000A/01A.

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).

By purchasing application software sold separately, you can send signals that conform to communication standards. Also, signal analyzer function was added for analyzing received signals.
For the application software sold separately, refer to Section 1.3.3.

The module is controlled by commands sent via Ethernet.
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

(3) Capable of simultaneously outputting DL signals from any test ports (TRX 1/2, up to 12 ports for each) at the calibrated level

(4) Test system and test efficiency improved by high level (0 dBm) output and 200 MHz bandwidth.

(5) High compatibility with MU887000A/01A eliminates the need to create evaluation programs and re-evaluation.

(6) 2-slot configuration to be installed on MT8870A that can use the existing production equipment and production line design as they are.

Various communication standards can be tested by switching the application software installed on the module.

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	MU887002A	TRX Test Module	1
Standard Accessories	W4048AE	MU887002A TRX Test Module Operation Manual	1
	KUWM-32-M4-16-OR 	Wing knob This tool is a screw for MU887002A to remove from slot of MT8870A or MT8872A.	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 MU887002A Options" list the options for the MU887002A. The model names of the added options can be found on the label placed on the front panel.

Additionally, the Anritsu management code of the options described in this manual is "0" unless there is a reason.

Table 1.3.2-1 MU887002A Options

Model/Code	Name	Remarks
MU887002A-007	7GHz Extension Function	Install at initial factory shipment
MU887002A-107	7GHz Extension Function Retrofit	Retrofit at factory in Japan*
MU887002A-207	7GHz Extension Function Retrofit	Retrofit at local service center outside Japan*
MU887002A-097	7GHz Extension Hardware	Install at initial factory shipment
MU887002A-197	7GHz Extension Hardware Retrofit	Retrofit at factory in Japan*
MU887002A-297	7GHz Extension Hardware Retrofit	Retrofit at local service center outside Japan*

*: The installation work differs depending on the serial number. Contact an Anritsu Service and Sales office for details.

The option number: The option number is indicated by three digits.

MU887002A- x x x

- Number indicating function
- Anritsu management code
  - 0: Install at initial factory shipment.
  - 1: Retrofit at factory in Japan.
  - 2: Retrofit at local service center outside Japan.

1.3.3 Application software

Application software and waveform files for the MU887002A 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
MX887018A	NR FDD sub-6GHz Uplink TX Measurement
MX887019A	NR TDD sub-6GHz Uplink TX Measurement
MX887021A	W-CDMA/HSPA Downlink TX Measurement
MX887023A	LTE FDD Downlink 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
MX887034A	WLAN 802.11be 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
MX887068A	LTE-V2X TX Measurement
MX887090A	Multi-DUT Measurement Scheduler
MX887092A	Pathloss Measurement Function for MU887002A

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
MV887018A	NR FDD sub-6GHz Downlink Waveforms
MV887019A	NR TDD sub-6GHz Downlink Waveforms
MV887021A	W-CDMA/HSPA Uplink Waveforms
MV887023A	LTE FDD Uplink Waveforms
MV887030A	WLAN 802.11b/g/a/n Waveforms
MV887031A	WLAN 802.11ac Waveforms
MV887032A	WLAN 802.11p Waveforms
MV887033A	WLAN 802.11ax Waveforms
MV887034A	WLAN 802.11be 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
MV887068A	LTE-V2X Waveforms
MV887100A	GPS Waveforms
MV887101A	Galileo Waveforms
MV887102A	GLONASS Waveforms
MV887103A	BeiDou Waveforms
MV887104A	QZSS Waveforms
MV887110A	DVB-H Waveforms
MV887111A	ISDB-T 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
MX887018A-001	NR FDD Contiguous ENDC Tx Measurement
MX887019A-001	NR TDD Contiguous ENDC Tx Measurement
MX887040A-001	DLE TX Measurement
MX887040A-002	2LE TX Measurement
MX887040A-003	BLR TX Measurement
MX887040A-004	BLE AoA/AoD TX Measurement
MX887068A-001	LTE-V2X PSCCH TX Measurement

Table 1.3.3-4 Waveforms Option

Model/Code	Name
MV887040A-001	DLE Waveforms
MV887040A-002	2LE Waveforms
MV887040A-003	BLR Waveforms
MV887040A-004	BLE AoA/AoD Waveforms
MV887100A-002	GPS L5 Waveforms

1.3.4 Optional accessories

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

Table 1.3.4-1 Optional Accessories

Model/Code	Name	Remarks
W4048AE	MU887002A 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

1.3.5 Warranty service

Table 1.3.5-1 MU887002A Warranty Service

Model/Code	Name	Remarks
MU887002A-ES210	2 Years Extended Warranty Service	
MU887002A-ES310	3 Years Extended Warranty Service	
MU887002A-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
IP	Internet Protocol
L	Left
MAC	Media Access Control
Max.	Maximum
NR	New Radio
R	Right
rms	Root Mean Square
SA	Signal Analyzer
SG	Signal Generator
TRX	Transceiver
VISA	Virtual Interface Software Architecture
VSG	Vector Signal Generator
V2X	Vehicle-to-Everything
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.

⚠ CAUTION

When using the MU887002A for the first time after purchase, or if it has not been used for a long time, warm it up sufficiently before use because the level accuracy may be affected.

2.1 Part Names and Functions 2-2
2.2 Installing Module 2-5
2.2.1 Installing module 2-5
2.2.2 Removing module 2-6
2.3 Confirming Connection 2-7
2.3.1 Ethernet 2-7
2.4 Electrostatic Countermeasures 2-9
2.5 Connecting Mobile 2-12

2.1 Part Names and Functions

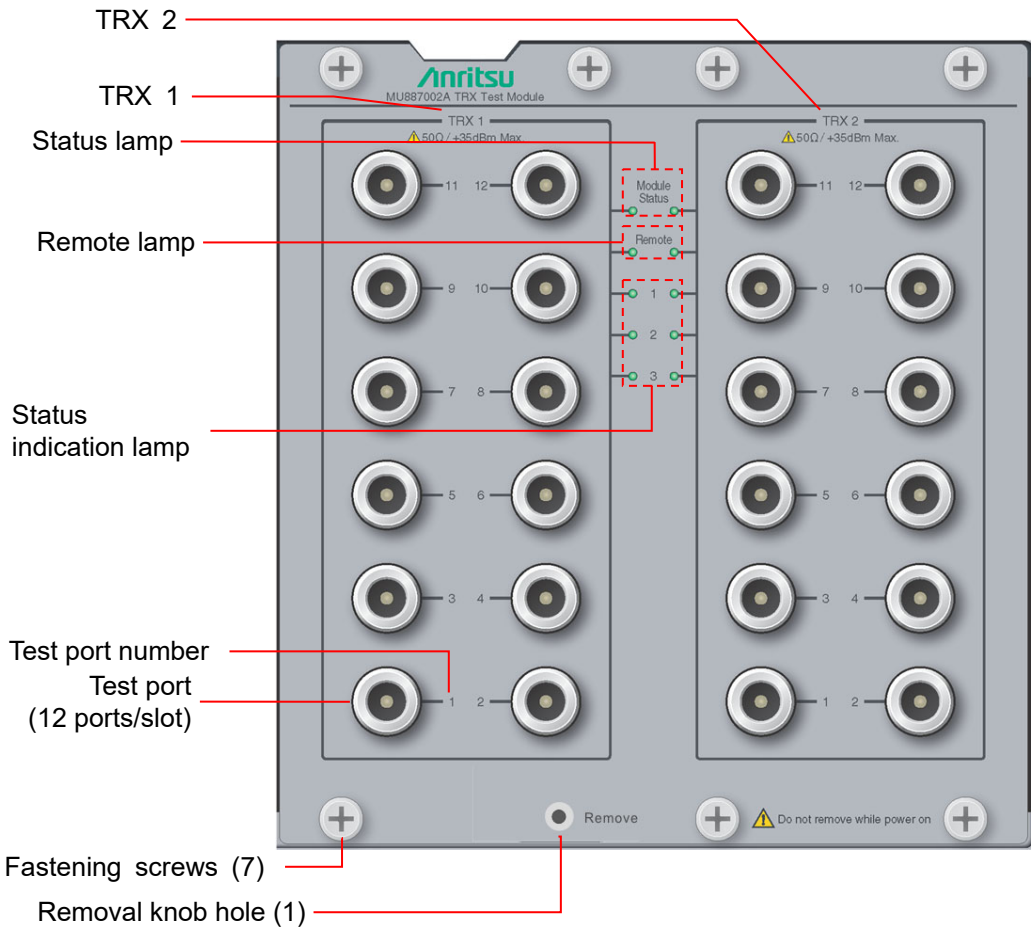


Figure 2.1-1 Front Panel Part Names

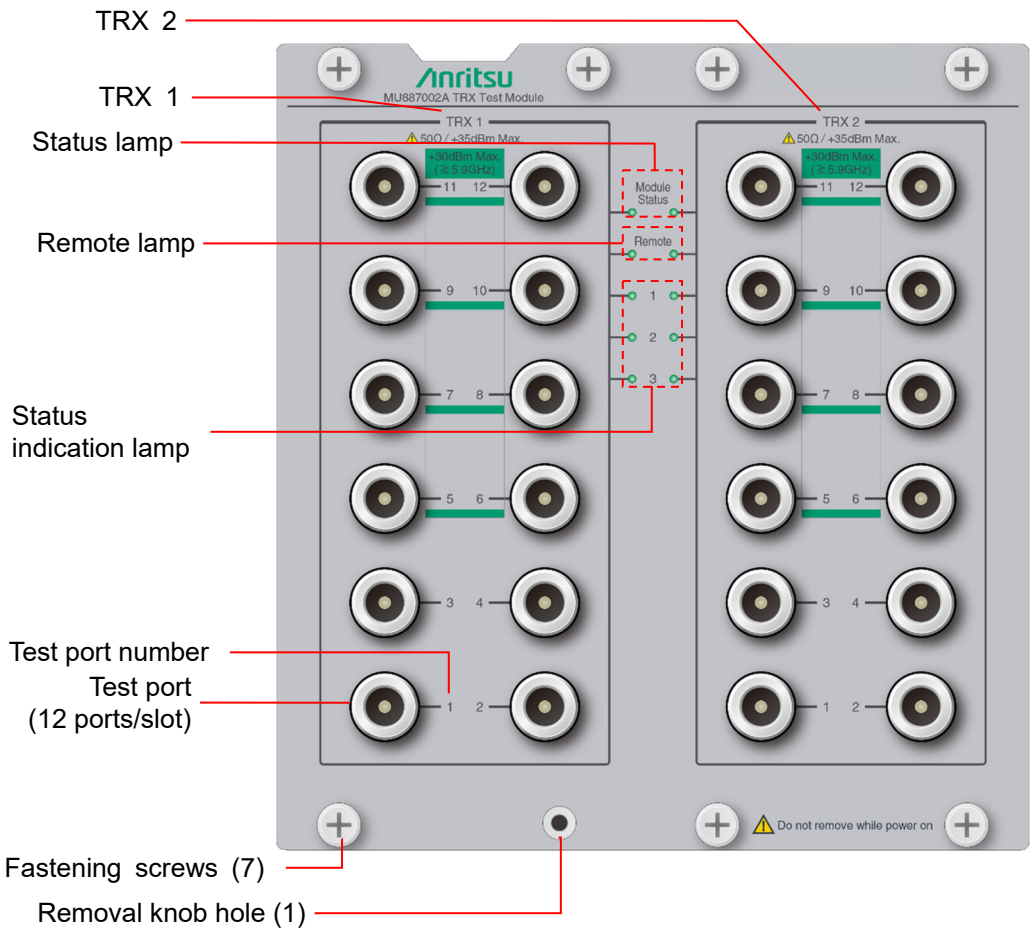


Figure 2.1-2 Front Panel (with MU887002A-007) Part Names

Table 2.1-1 Functions of Front-Panel Parts

Name	Function
Fastening screws (7)	Secures module in MT8870A or MT8872A.
Removal knob hole (1)	Insert the wing knob into this hole to remove the module.
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 3)*	Indicates the status of module or application software. Refer to Appendix D "Status indication of lamps" for details.
Test ports 1 to 12	Connectors for high-frequency signals. Both simultaneous signal input and output supported.
With MU887002A-007 Test ports 1 to 4	Connectors for high-frequency signals. The ports can input and output signals of 6.0 GHz or lower simultaneously.
With MU887002A-007 Test ports 5 to 12	Connectors for high-frequency signals. Both simultaneous signal input and output supported. However, for signals of 5.9 GHz or higher: ● The ports cannot input and output the signals simultaneously. ● Max. input level: +30 dBm

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

⚠ CAUTION

Always tighten the fastening screws before using the MT8870A. The MU887002A 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 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 screws on the blank panel using a Phillips screwdriver.
   If the screws are not turned easily, hold the other screws and lift the module, then tighten the screws.

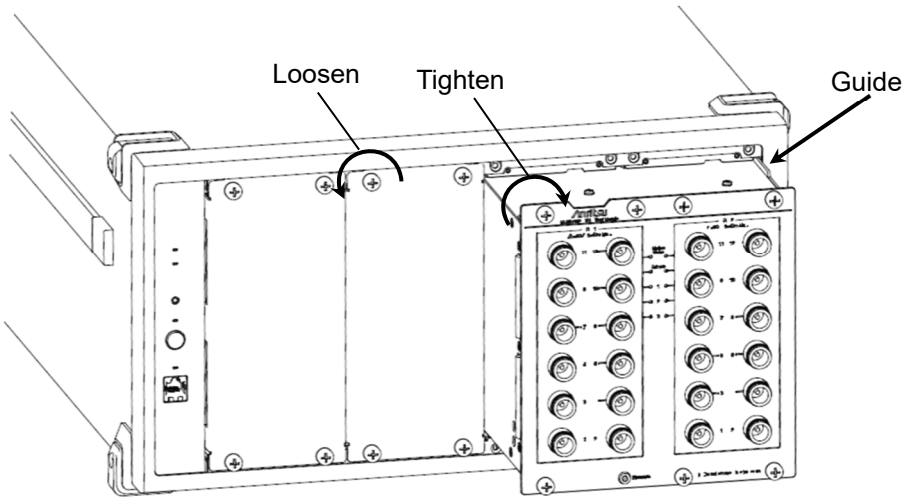


Figure 2.2.1-1 Installing Module

⚠ 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.

2.2.2 Removing module

1. Loosen the 7 screws on the panel using a Phillips screwdriver.
2. Insert the wing knob into the removal knob hole, and turn it clockwise. Keep turning it clockwise until the module comes out toward you.

Note:

Do not pull the wing knob with your finger.

3. Turn the wing knob counterclockwise, remove the wing knob from the module.

Note:

Do not insert the module to be installed before removing the wing knob. Doing so may damage the module.

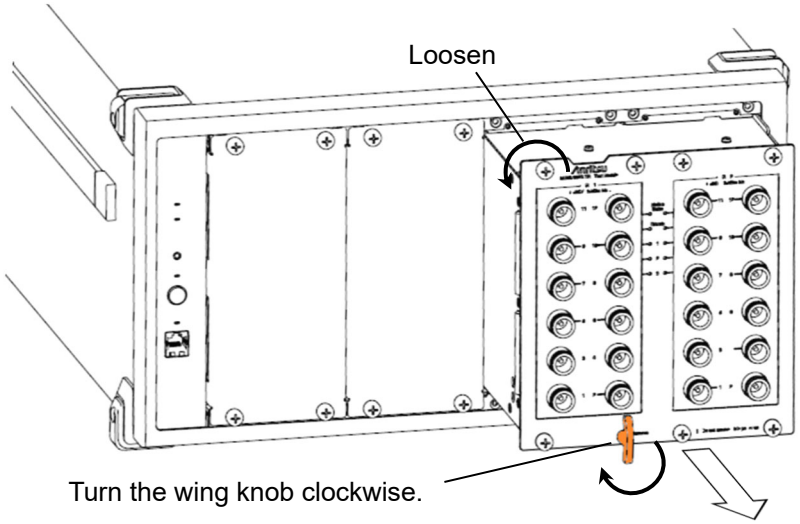


Figure 2.2.2-1 Removing Module

4. Hold the module to pull the module out.
5. Replace the blank panel in the slot.
6. Tighten the screws on the panel using a Phillips screwdriver.

⚠ 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 so as not to apply excessive force to one part or risk dropping the module.

2.3 Confirming Connection

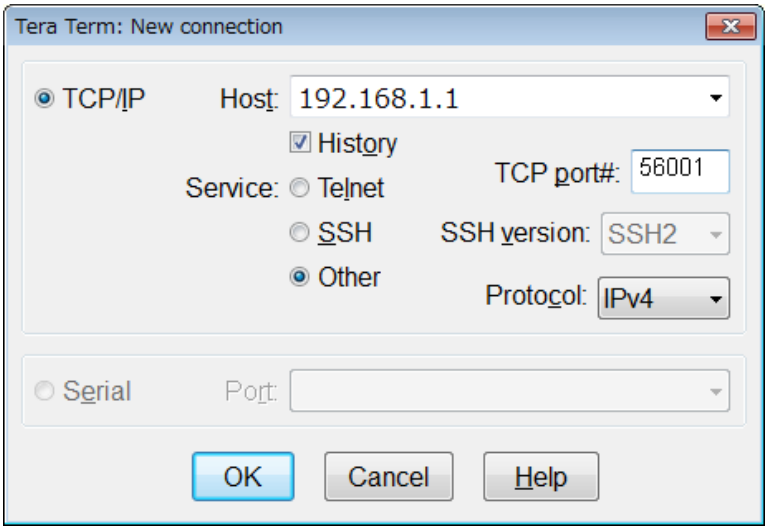
Confirm that the external PC controller recognizes the MT8870A via Ethernet.
This description assumes that the module is installed in slot 1 of the MT8870A and the interface settings are as follows:
IPv4 address: Slot 1: 192.168.1.1, Slot 2: 192.168.1.2
Port number: 56001

2.3.1 Ethernet

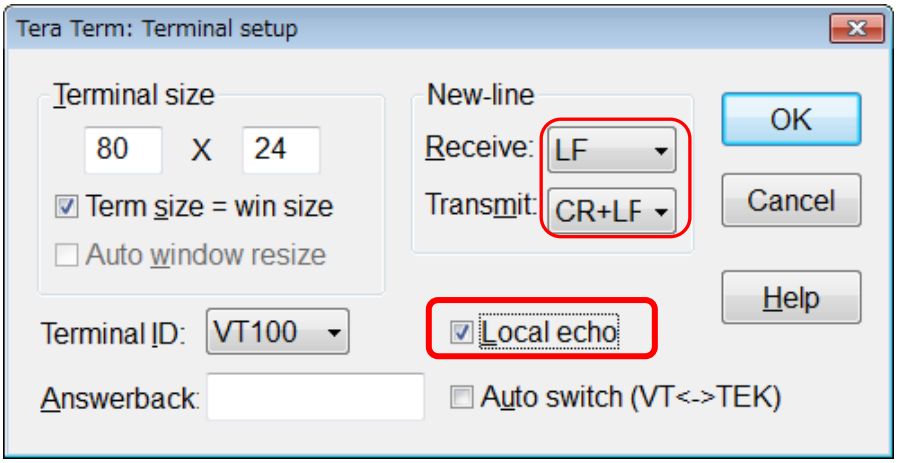
This section describes how to use the Tera Term freeware.

Start a separate Tera Term for each IPv4 address and perform settings.

1. On starting Tera Term, the New connection dialog box 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.

2.4 Electrostatic Countermeasures

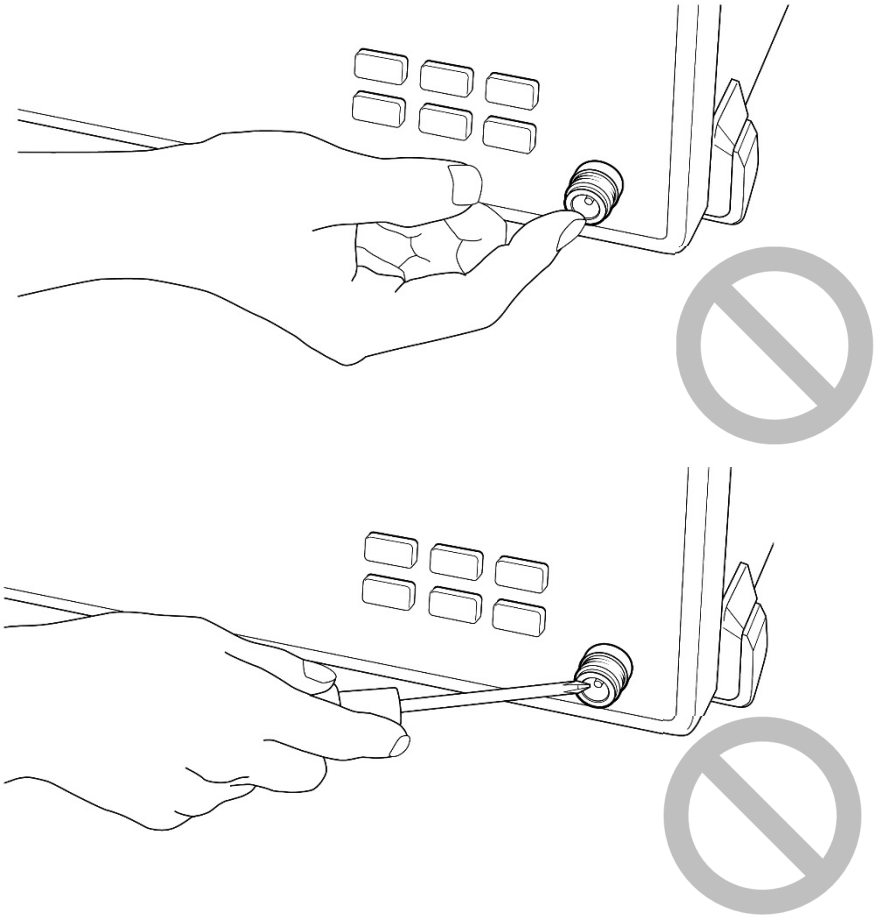
This section describes the general precautions against electrostatic discharges.

⚠ 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.

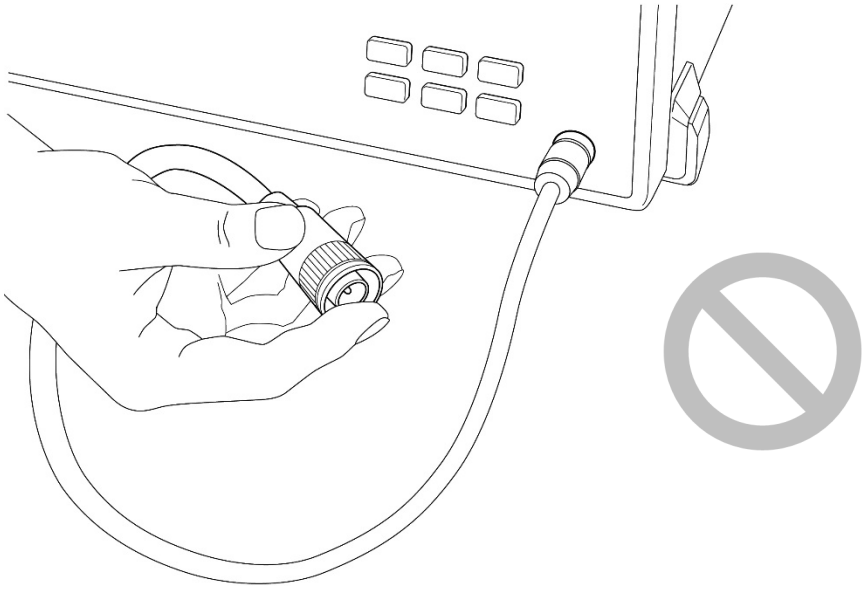
  NEVER connect the MT8870A and DUT without grounding, otherwise an electrostatic discharge may damage the MT8870A input circuits.

- 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.



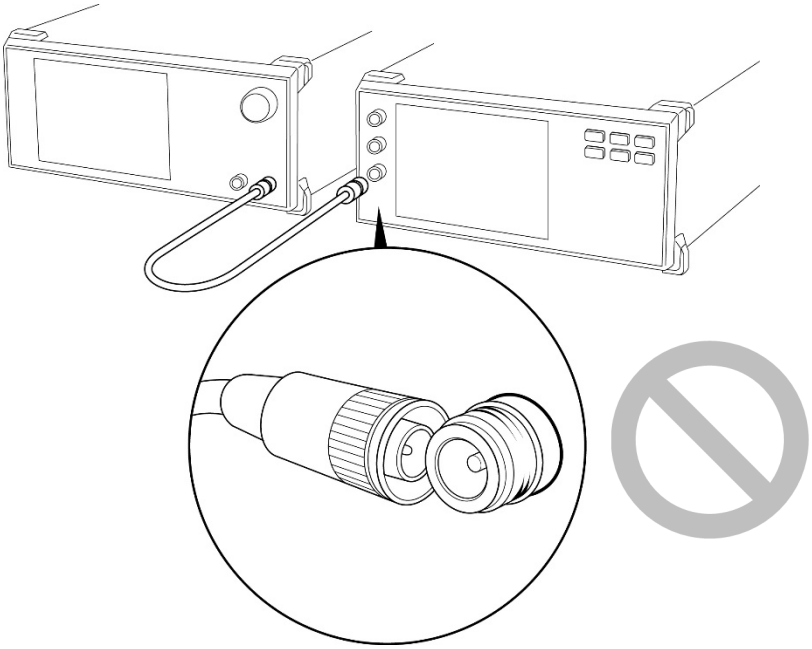
⚠ CAUTION

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.

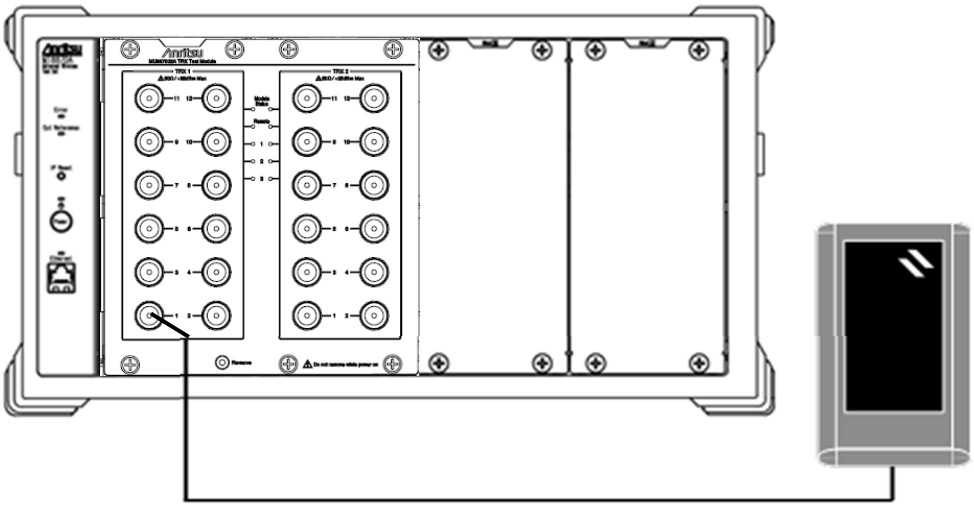


Figure 2.5-1 Mobile Connection Example

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 3-2
3.2 Configuring Communications 3-3
3.2.1 Configuring Ethernet 3-3
3.2.2 Line termination 3-7
3.2.3 Updating settings 3-7
3.3 Transferring Files 3-8
3.4 System Settings 3-9
3.4.1 General settings 3-9
3.4.2 Viewing product information 3-11
3.5 Preparing for Measurement 3-14
3.5.1 Specifying frequencies 3-14
3.5.2 Level calibration 3-15
3.5.3 Correcting cable losses 3-17
3.5.4 Configuring ports 3-20
3.5.5 Configuring trigger connector 3-20
3.6 Status Registers 3-21
3.6.1 Configuring registers 3-21
3.6.2 Status byte register 3-24
3.6.3 Standard event status register 3-26
3.6.4 Event status register 3-28
3.6.5 Operation status register 3-29
3.6.6 Questionable status register 3-33

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 status 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.
One or both of the Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) can be used.

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 router, 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 MU887002A 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 MU887002A via Ethernet, MU887002A 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 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
```

3.2.3 Updating settings

When settings such as Ethernet 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 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.

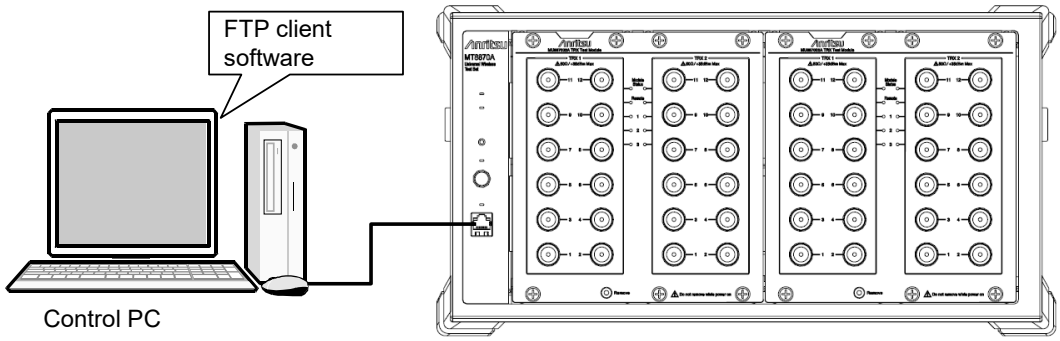


Figure 3.3-1 File Transfer Setup

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
- Switching the application software

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`
- Switching the application software
  `SYSSEL`
  `:INSTrument[:SELect]`
  `STDSEL`
  `:CONFigure:CELLular:MEASurement:STANdard`
  Refer to the operation manuals of the application software to be used for the description of the application software switching commands.

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 MU887002A TRX Test module
  Model, serial number, TRX1/2, hardware options, FPGA version, power-on count, run time, insertion/removal count, internal temperature, 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 MU887002A TRX Test module:

- Model, product name, hardware options, and FPGA version
  ```
  SYSINFO
  :SYSTem:INFormation
  ```
- Serial number
  ```
  SYSSERIAL
  :SYSTem:INFormation:DEVice:ID
  ```
- TRX 1/2
  ```
  SYSTRX
  :SYSTem:INFormation:MAINframe:TRX
  ```
- 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 MU887002A
- Application software installed on MU887002A

Use the following commands to view the information about the MU887002A 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.
Full calibration is executed in all the RF I/O bandwidths of the module. Band calibration is executed in the communication system bandwidth supported by the application software. 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 temperature fluctuates and 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 MU887002A 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.

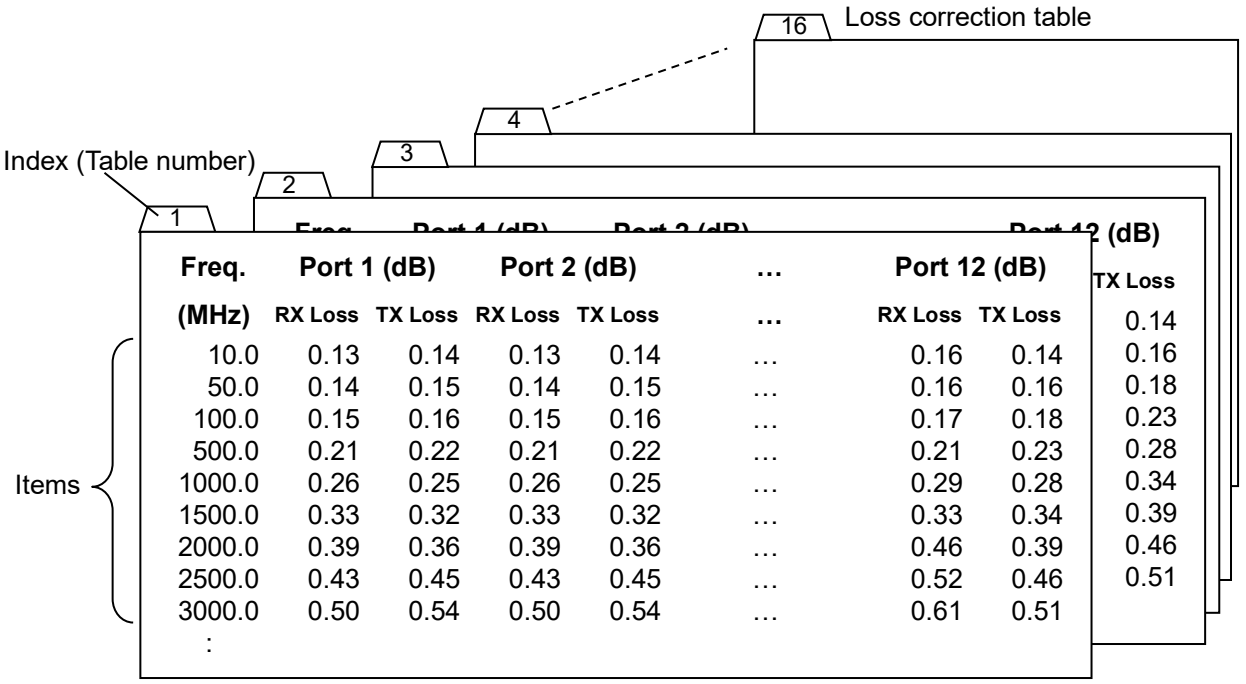


Figure 3.5.3-1 Loss Correction Data Layout

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

- For signals received by MU887002A (uplink) (Reception loss)
- For signals sent by MU887002A (downlink) (Transmission loss)

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

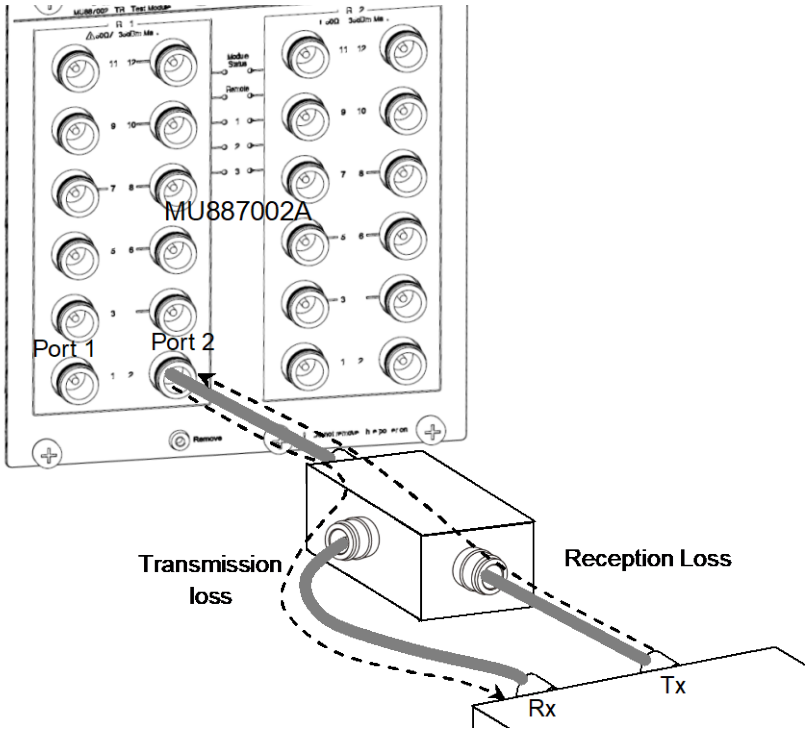


Figure 3.5.3-2 Loss Correction Data Structure

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

To set external loss of Port 1 to Port 4 of the specified frequency:

```
LOSSTBLVAL
:CALCulate:EXTLoss:TABLe:VALue
```

To set the external loss of Port1 to Port12 of the specified frequency:

```
LOSSTBLVALALL
:CALCulate:EXTLoss:TABLe:VALue:ALL
```

To delete external loss data:

```
DELLOSSTBL
:CALCulate:EXTLoss:TABLe:DELete
```

To query the number of external loss data:

```
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
```

When outputting signals from multiple test ports, the signals will be output below the specified output level if the difference in values in the loss correction table between the test ports exceeds 12 dB.
In this case, the status indication lamp 2 will be lit orange.
However, even if it exceeds 8 dB, the error in output level may become large.

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
```

- Input: Sets the test port for input signals to the MU887002A.
- Output: Sets the test port for output signals from the MU887002A.
- Off: Turn Off the inputs of all test ports and turn Off the output of each test port.

Only one test port per TRX can be set for input constantly.
Up to 12 test ports per TRX can be set for output constantly.
All test ports can be set for either input or output, but only one port can be set for both simultaneously for each TRX.

However, with MU887002A-007:

- Test ports 1 to 4 cannot be set for RF signals above 6 GHz.
- Test ports 5 to 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

When outputting signals from multiple test ports, the signals will be output below the specified output level if the difference in values in the loss correction table between the test ports exceeds 12 dB. In this case, the status indication lamp 2 will be lit orange.
However, even if it exceeds 8 dB, the error in output level may become large.

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 and each TRX.

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.

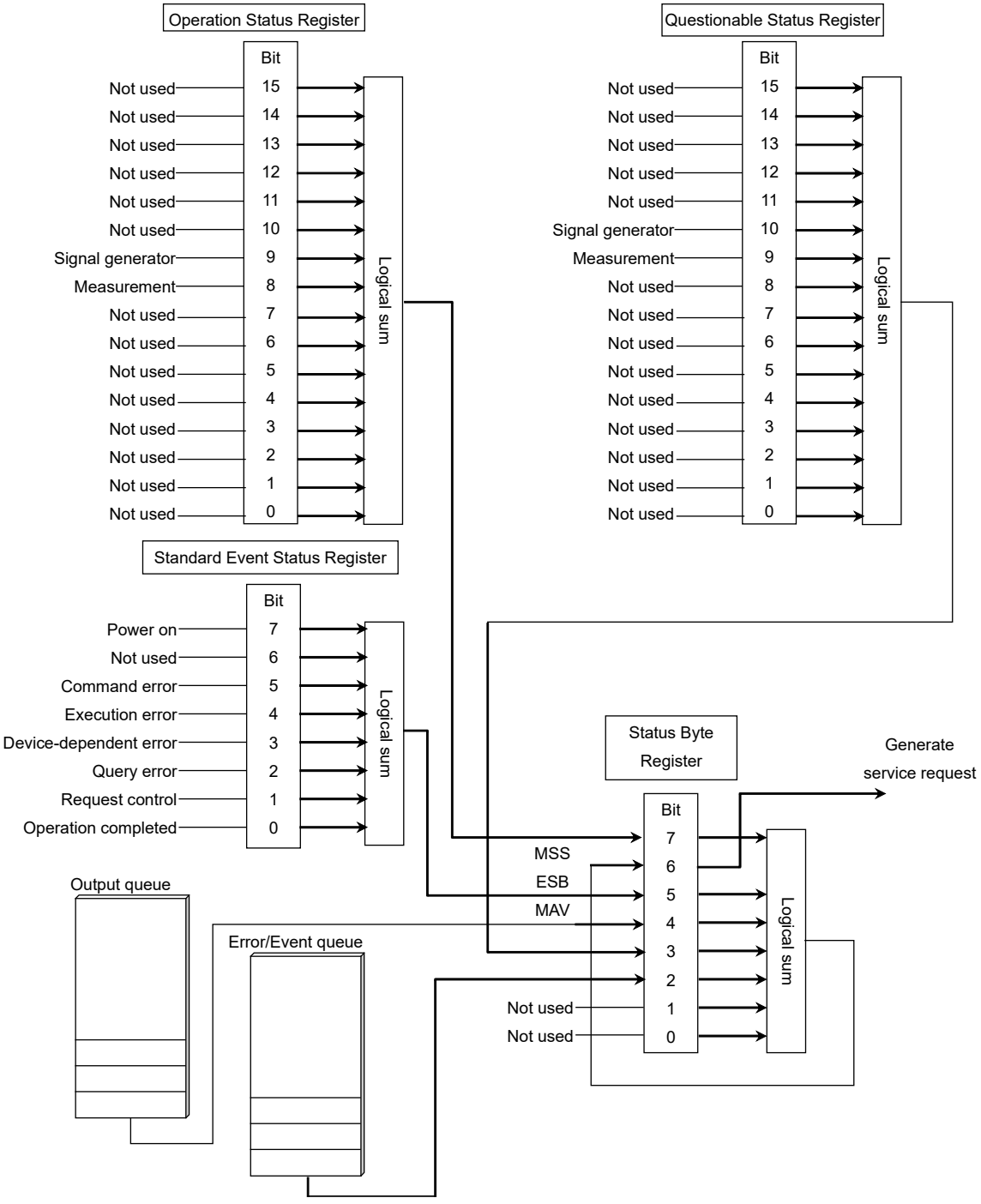


Figure 3.6.1-1 Register Configurations (SCPI Command Mode)

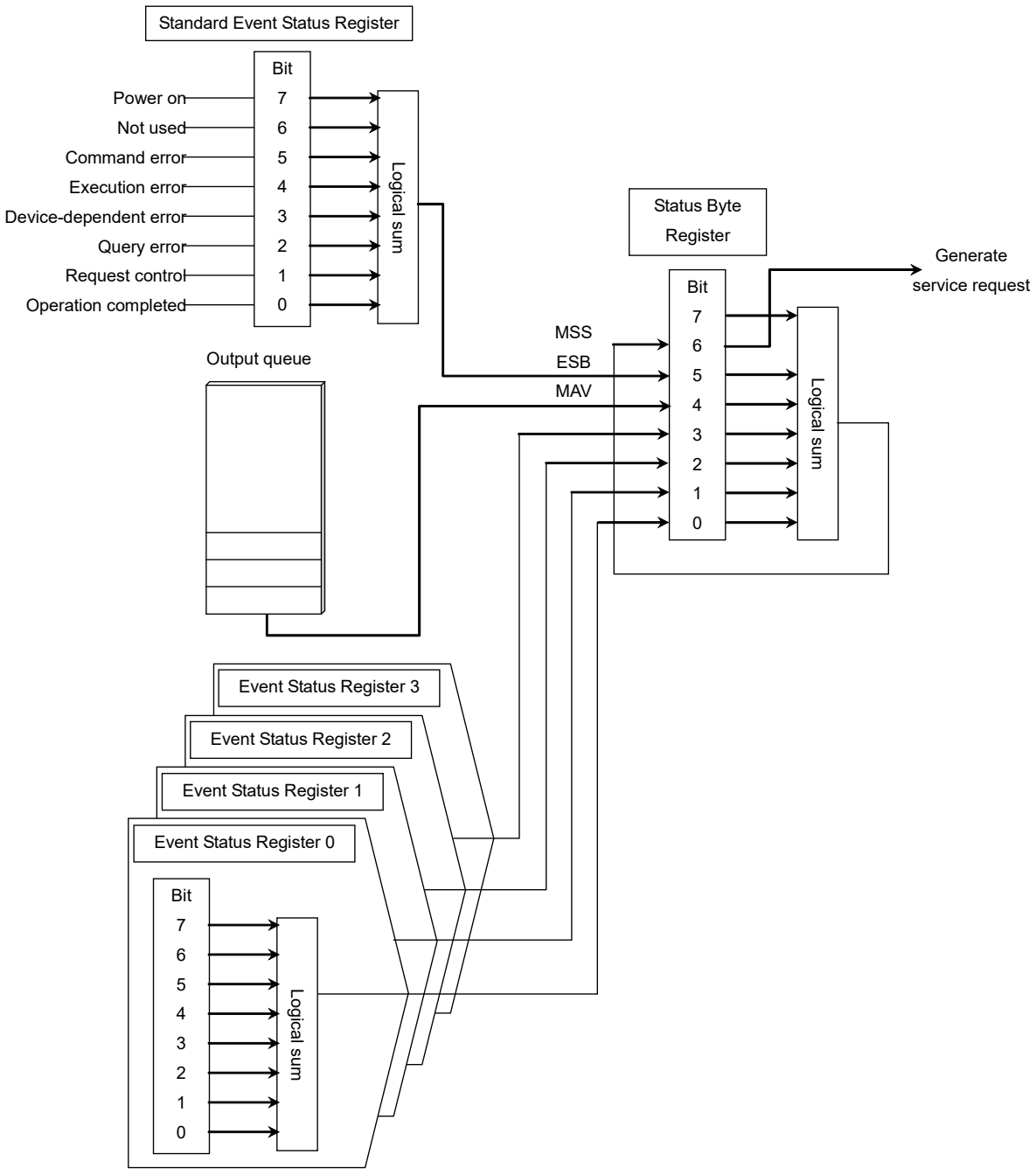


Figure 3.6.1-2 Register Configurations (Native Command Mode)

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 control PC 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.

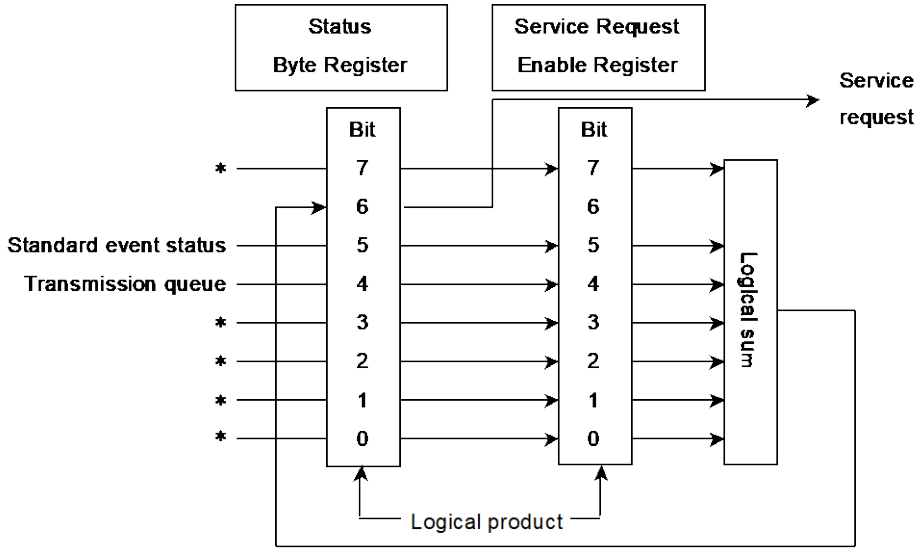


Figure 3.6.2-1 Configuration of Status Byte Register and Service Request Enable Register

Note:

A service request cannot be used over Ethernet.

To read the status byte register:

- Use the common command `*STB?`

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 Status 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 status register
4	Message Available summary (MAV) Changed to 1 if response message exists in output queue of measuring instrument
3	SCPI command mode: questionable status register Native command mode: event status register 3
2	SCPI command mode: changed to 1 if data exists in error event queue of measuring instrument Native command mode: event status register 2
1	SCPI command mode: not used and always set to 0 Native command mode: event status register 1
0	SCPI command mode: not used and always set to 0 Native command mode: event status 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 control PC. A service request is generated when MSS is changed from 0 to 1.

Bit 5 of the status byte register indicates the standard event status 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.

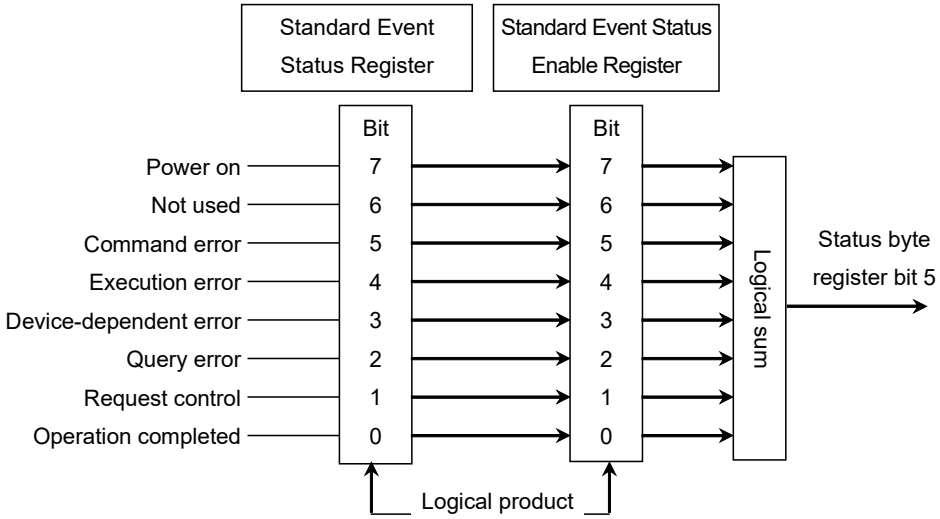


Figure 3.6.3-1 Standard Event Status Register Configuration

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

Table 3.6.3-1 Standard Event Status 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 status 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.

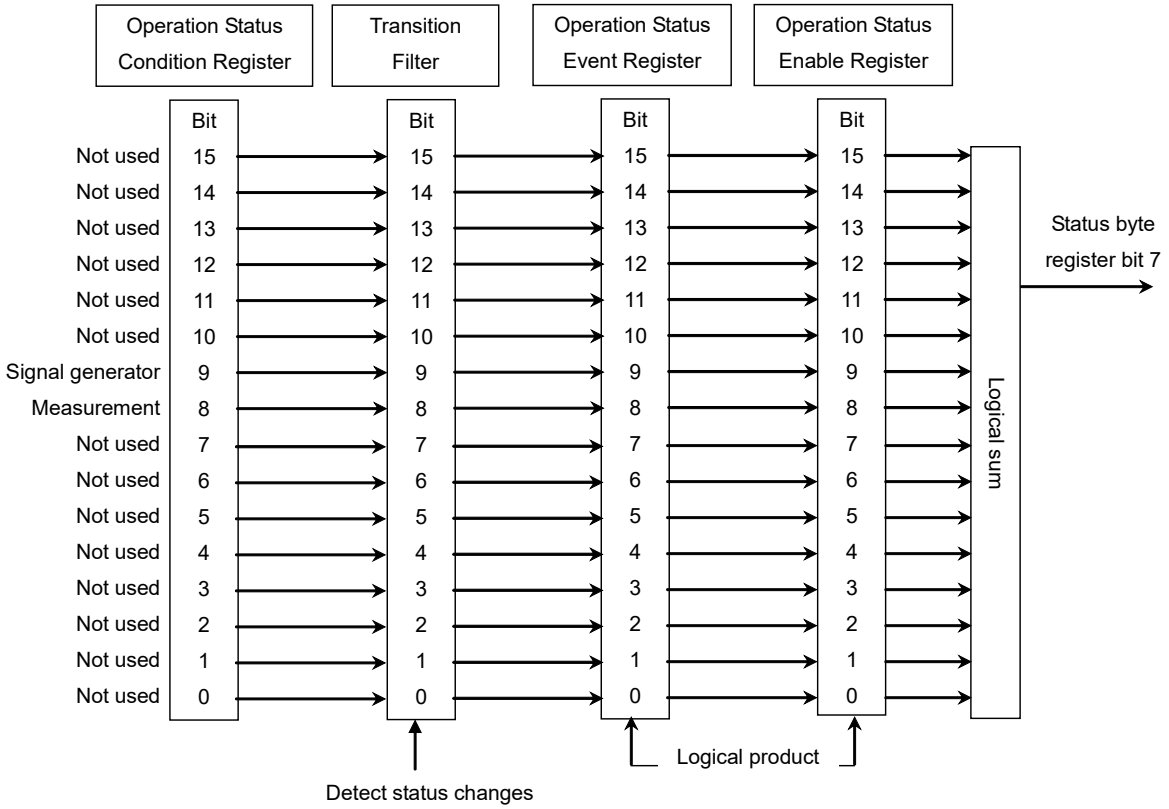


Figure 3.6.5-1 Operation Status Register Configurations

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 operation status registers
8	Logical sum of measurement operation 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 operation status register and the measurement operation status register with the same configuration as the operation status register.

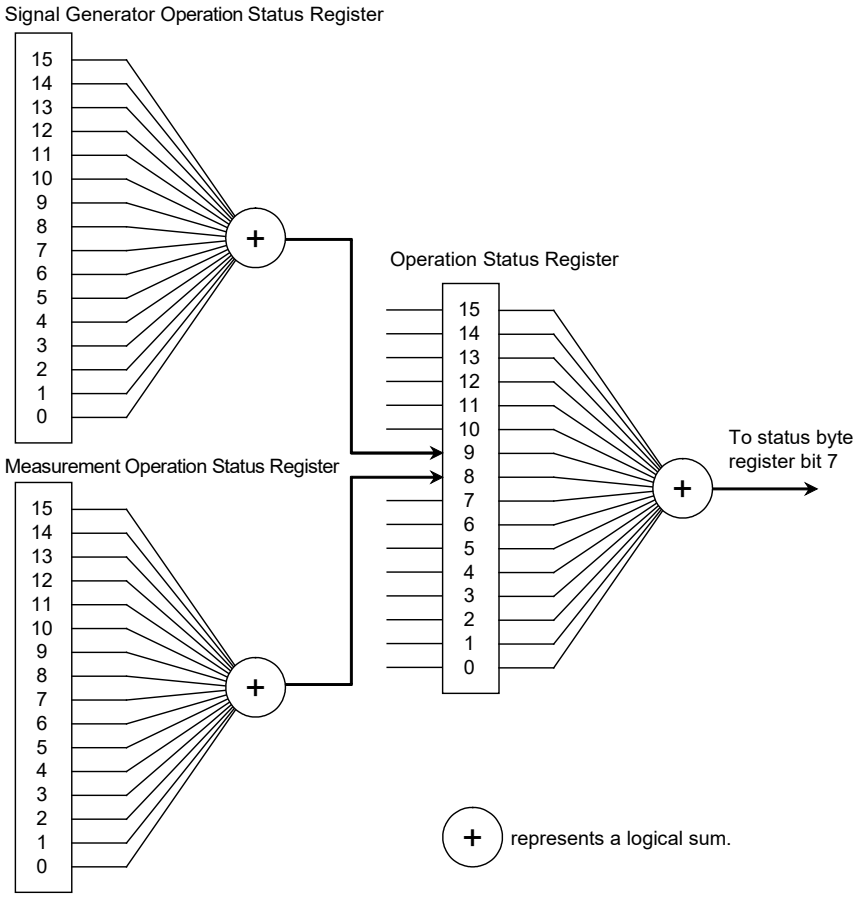


Figure 3.6.5-2 Operation Status Register Connections

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

Table 3.6.5-2 Bit Definitions of Signal Generator Operation 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 Operation 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 operation 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 operation 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 status register

The questionable status 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 status register consists of the following registers:

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

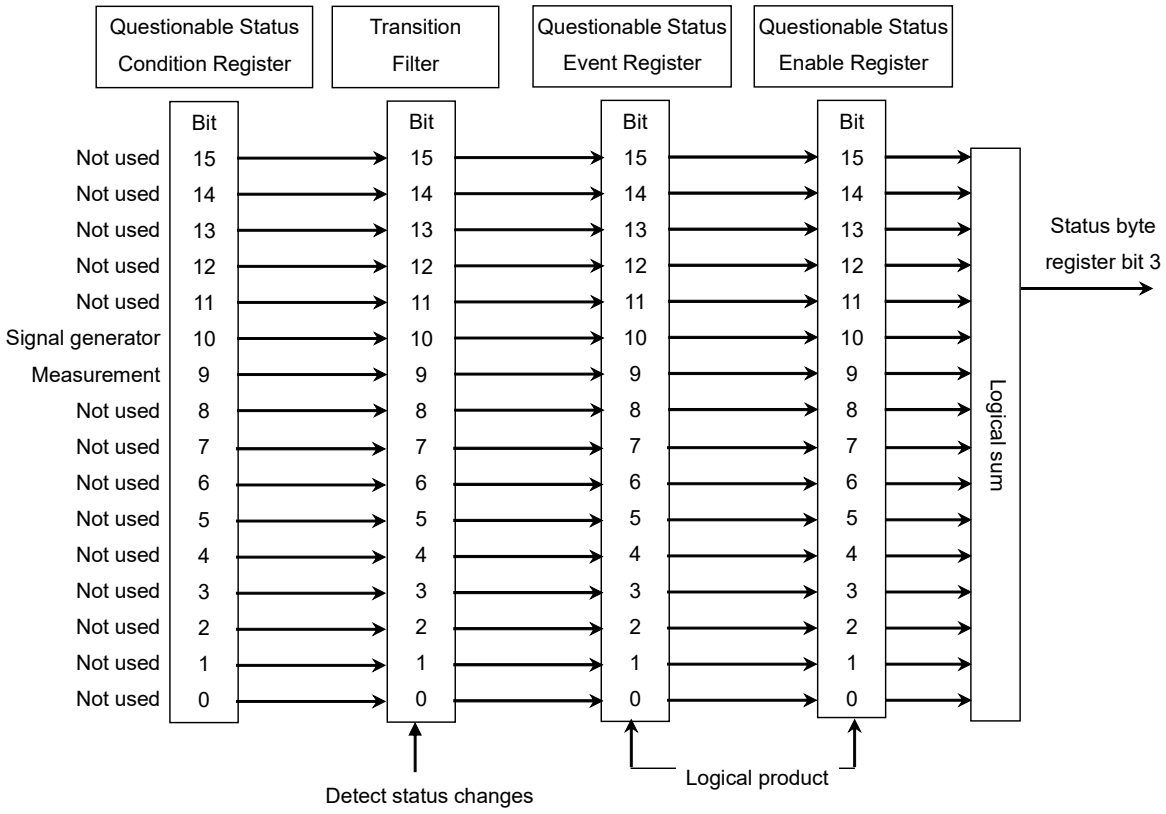


Figure 3.6.6-1 Questionable Status Register Configurations

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

Table 3.6.6-1 Bit Definitions of Questionable Status Condition Register

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

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

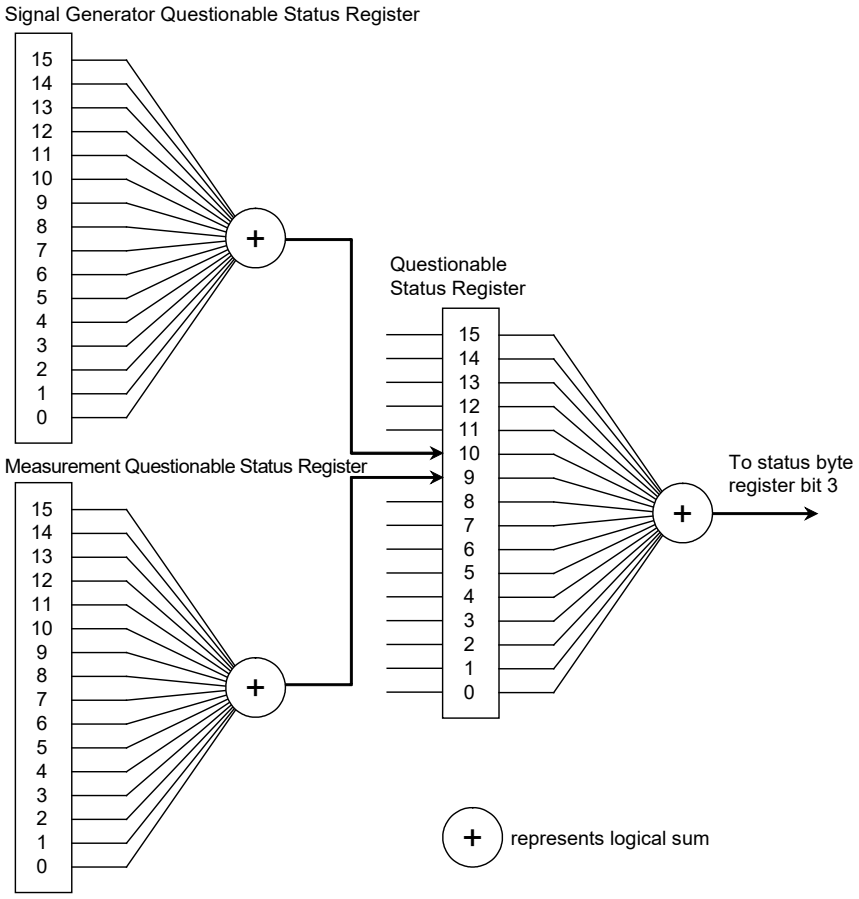


Figure 3.6.6-2 Questionable Status Register Connections

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

Table 3.6.6-2 Bit Definitions of Signal Generator Questionable Status 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 Status 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 status 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 status 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 status 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 MU887002A 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 4-2
4.2 Operation Mode 4-3
4.3 Normal Mode 4-4
4.3.1 Fundamental setting 4-4
4.3.2 Modulation 4-6
4.3.3 Waveform pattern synchronization among TRXs 4-11
4.4 Sequence Mode 4-17
4.4.1 Operation details 4-17
4.4.2 Sequence table configuration 4-18
4.4.3 File operations 4-24
4.4.4 Setting item error check 4-25
4.4.5 Sequence execution 4-26
4.4.6 Waveform pattern synchronization among TRXs 4-27

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 MU887002A TRX Test Module performs phase modulation by regenerating the baseband signal saved in the waveform file.

Refer to the followings for the waveform file specifications.
Waveform File for Cellular Application Operation Manual
Waveform File for Short Range Wireless Operation Manual
Waveform File for GNSS & Broadcast Operation Manual

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.

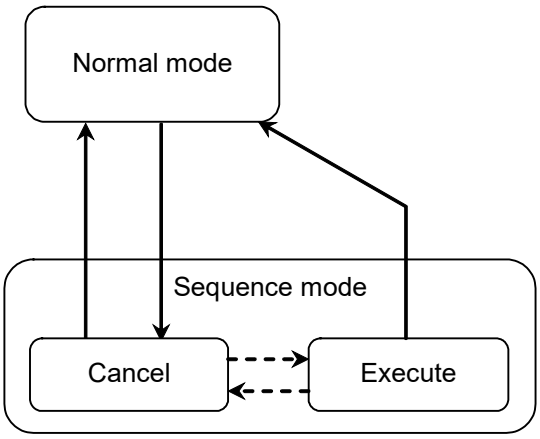


Figure 4.2-1 Operation Mode Setting Changes and Directions

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 MU887002A modulates signals using a waveform file describing IQ data.

The waveform file is saved in the MU887002A so the waveform memory must be queried to use it. The operation to read the waveform file into the waveform memory is called “loading.”

Note:

The waveform file for TRX 1/2 is stored in a same file storage medium. Therefore, if you delete the waveform file for TRX 1 from the file storage medium, the waveform file for TRX 2 will also be deleted.

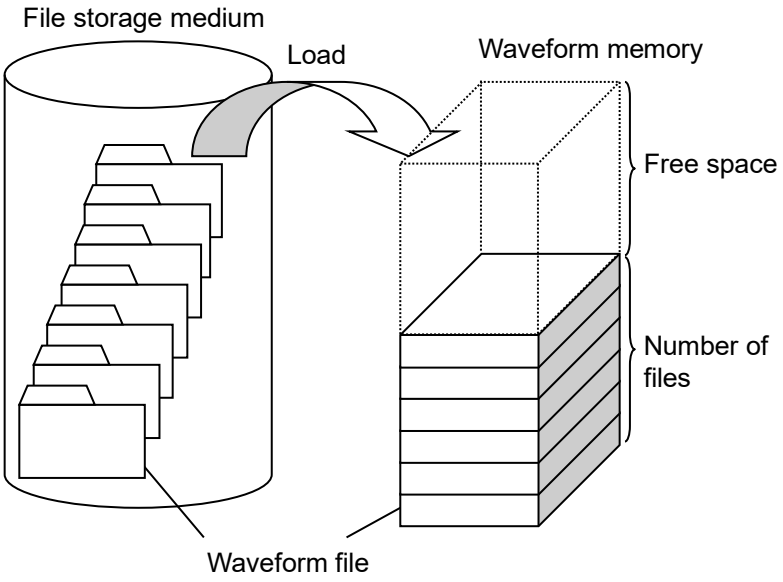


Figure 4.3.2-1 Loading Waveform File

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:

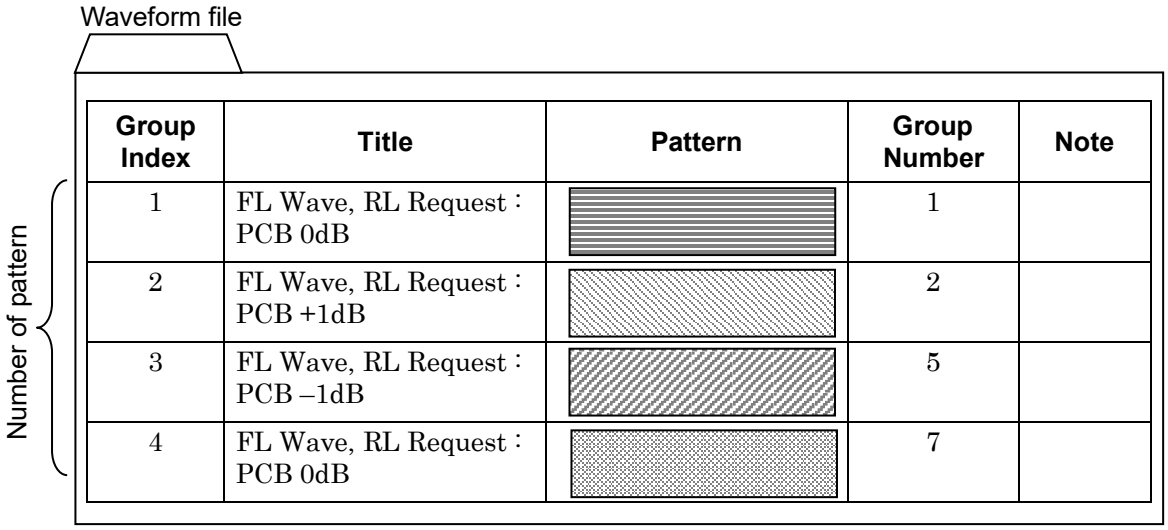
When the waveform file for modulation is not specified, signals are not output from the test port on the panel even after setting the modulation to ARB.

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.



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.

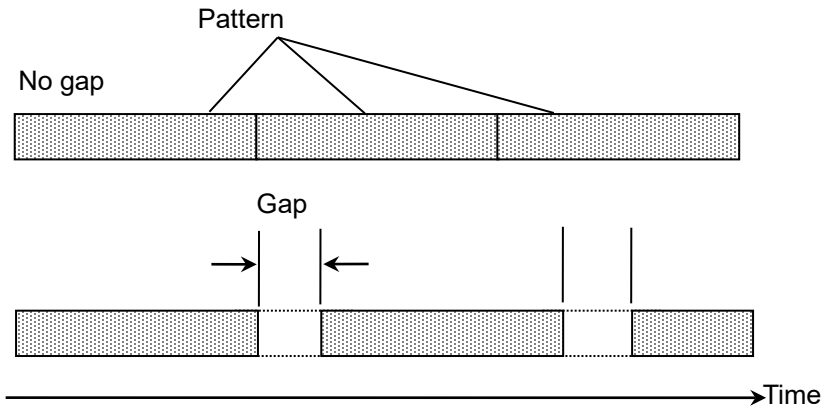


Figure 4.3.2-3 Pattern Gap

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 TRXs

The MU887002A can output waveform patterns of multiple TRXs 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.

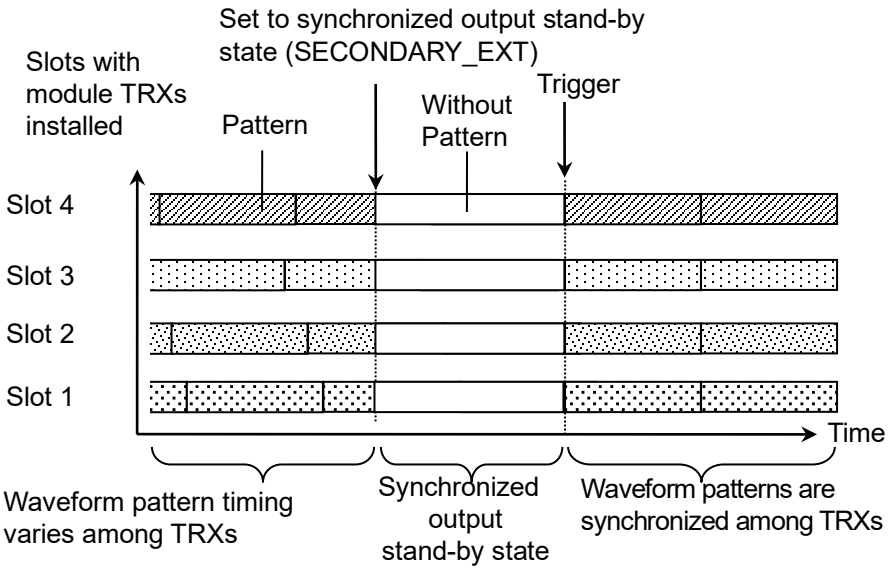


Figure 4.3.3-1 Synchronization with the Timing of External Trigger

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

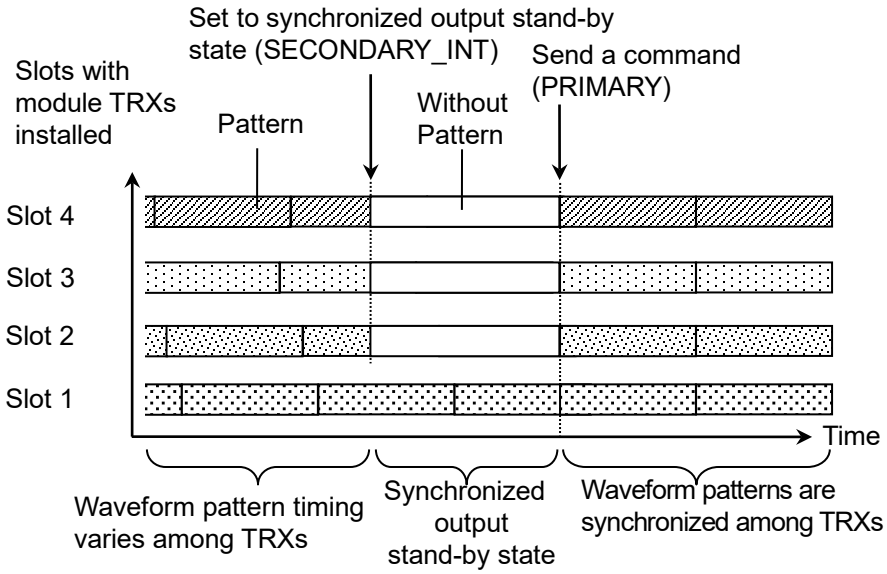


Figure 4.3.3-2 Synchronization with Waveform Pattern of Slot 1 TRX

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

- 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 TRX.

PRIMARY: Reference for waveform pattern timing of other TRXs

SECONDARY_INT: Synchronizes the waveform pattern timing with that of the Slot 1 TRX.

SECONDARY_EXT: Synchronizes the waveform pattern timing with external trigger.

Available synchronization modes differ among the slots.

Table 4.3.3-1 Available Synchronization Modes

Slot / Synchronization Mode	Slot 1	Slot 2	Slot 3	Slot 4
PRIMARY	✓	–	–	–
SECONDARY_INT	–	✓	✓	✓
SECONDARY_EXT	✓	✓	✓	✓

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

- A TRX is not installed to Slot 1.
- The synchronization mode for Slot 1 is set to SECONDARY_EXT.

When set to SECONDARY_INT or SECONDARY_EXT, a TRX 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 TRX is set to PRIMARY, the TRXs set to SECONDARY_INT synchronize with Slot 1 and output waveform patterns. The TRXs are released from synchronized output stand-by state.

When the TRX is set to SECONDARY_EXT, 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 TRXs 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 TRX, 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	PRIMARY	SECONDARY _INT	SECONDARY _EXT
: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 SECONDARY_EXT and Slot 1 are changed to INPUT.

Set the two units of the MT8870A as the figure below. When setting the Slot 1 TRX of the MT8870A-1 to PRIMARY, the waveform patterns of the 8 TRXs synchronize.

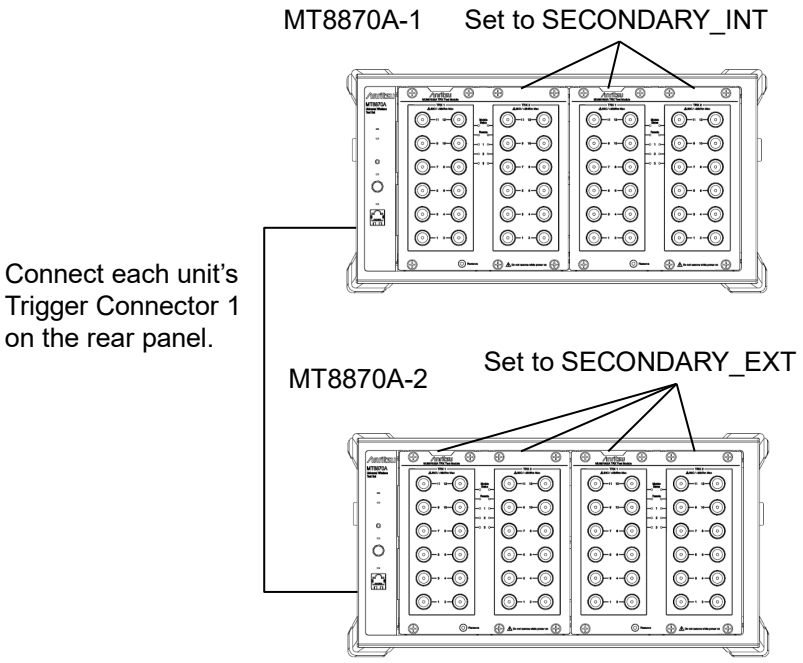


Figure 4.3.3-3 Waveform Pattern Synchronization Among 8 TRXs

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

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

13. Set the TRXs from Slot 2 to 4 of the MT8870A-1 as below.
    Synchronization mode: Synchronizes with Slot 1 pattern (SECONDARY_INT)
    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
    SECONDARY_INT,"MV887012A_GSM_0002",1,1
    ```

14. Query all the TRXs of the MT8870A-2 and the Slot 2 to 4 TRXs 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 TRX of the MT8870A-1 as below.
    Synchronization mode: Primary of pattern (PRIMARY)
    Waveform pattern: Pattern 1 of "MV887012A_GSM_0002"

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

    ```
    :SOURce:GPRF:GENerator:ARB:WAVeform:SYNChronize:STARt
    :PATTern:SELect PRIMARY,"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.

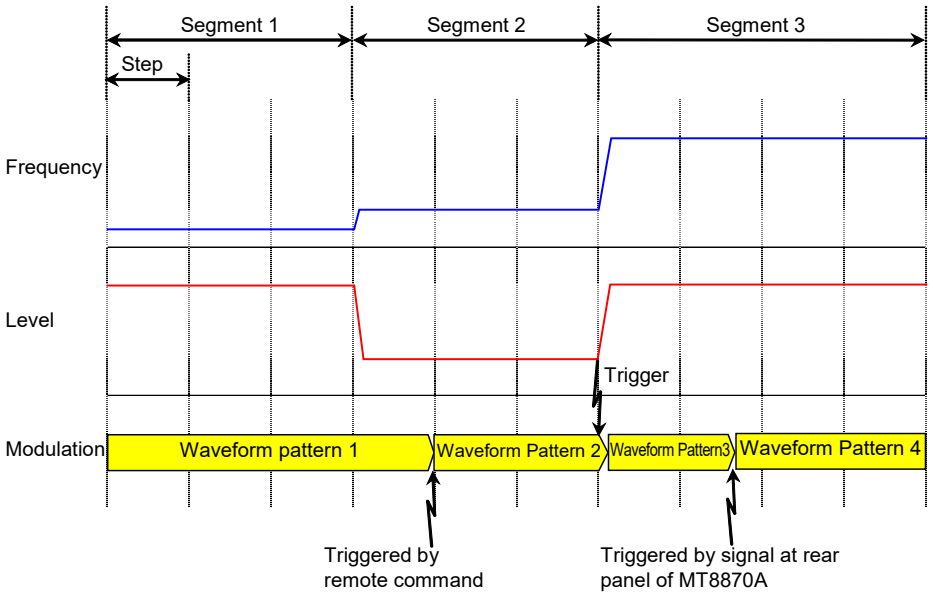


Figure 4.4.1-1 Sequence Mode Execution

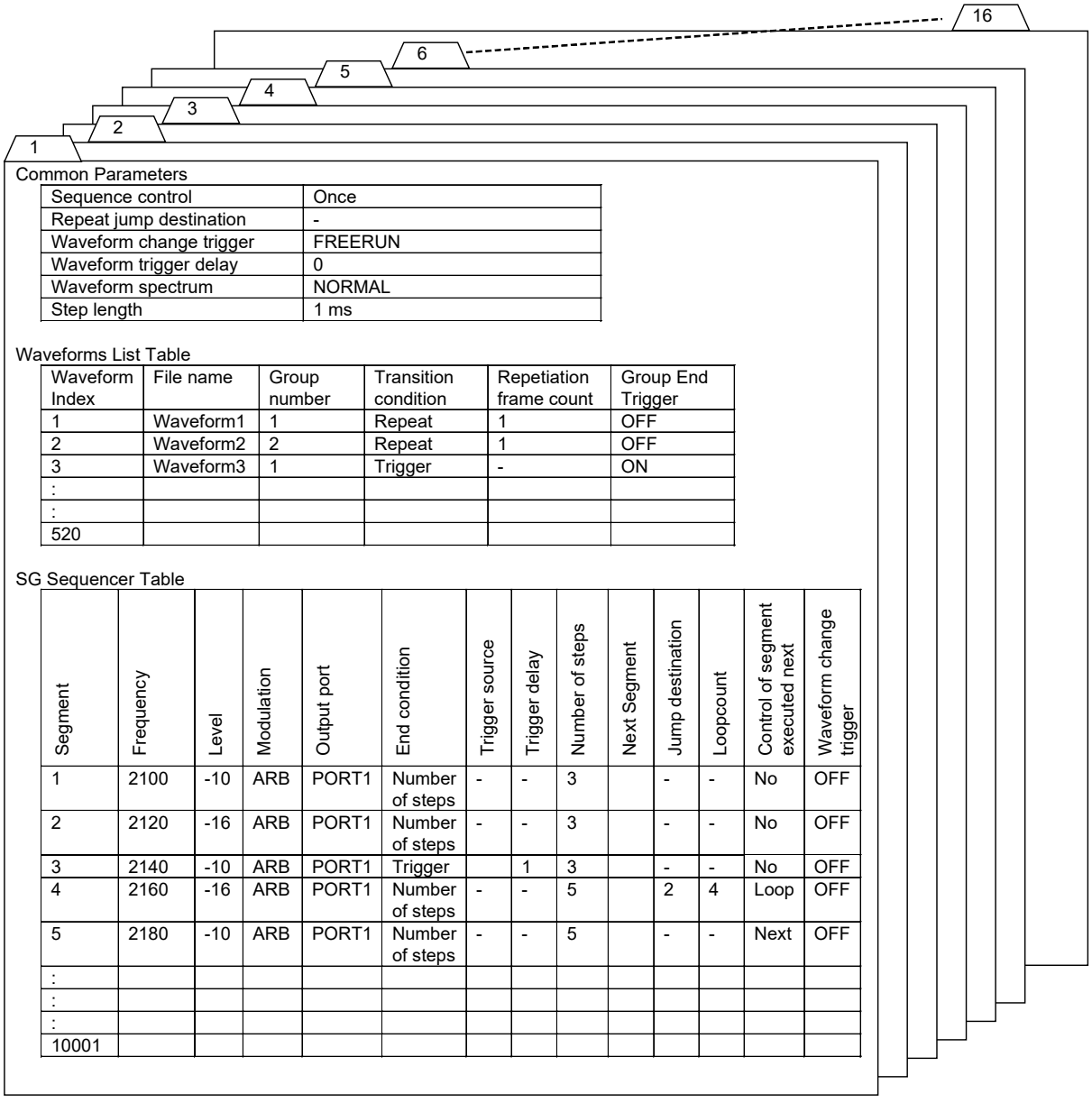
⚠ CAUTION

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	Repetiation frame count	Group End Trigger
1	Waveform1	1	Repeat	1	OFF
2	Waveform2	2	Repeat	1	OFF
3	Waveform3	1	Trigger	-	ON
:					
:					
520					

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		-	-	No	OFF
2	2120	-16	ARB	PORT1	Number of steps	-	-	3		-	-	No	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

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 12)
- 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 TRXs

The MU887002A can output waveform patterns of multiple TRXs 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 TRX

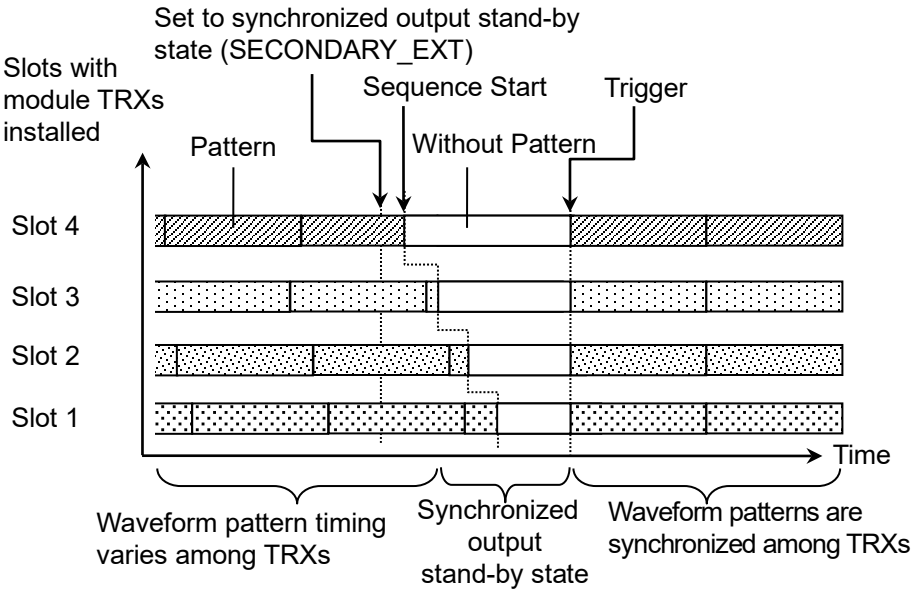


Figure 4.4.6-1 Synchronization with External Trigger Timing

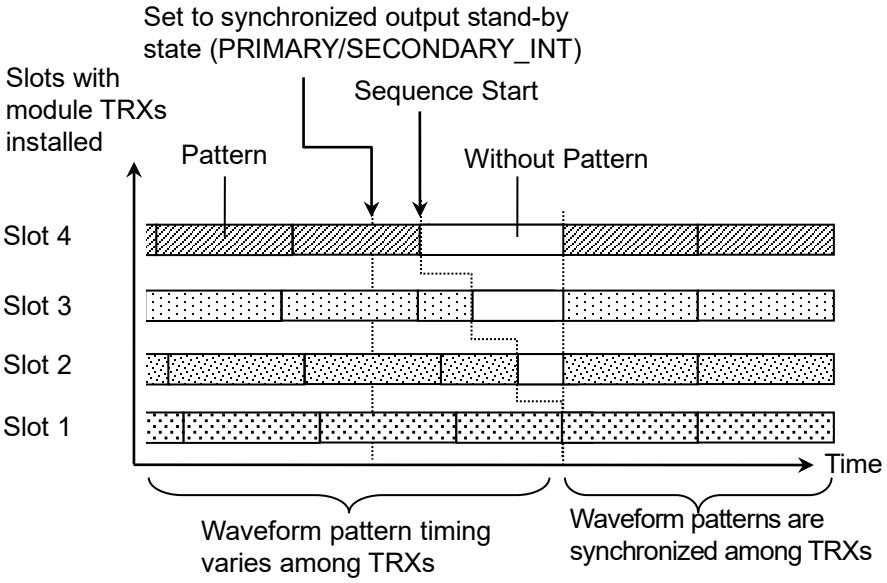


Figure 4.4.6-2 Synchronization with Waveform Pattern of Slot 1 TRX

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

- 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 TRXs.

NONE: No synchronization of waveform patterns among TRXs.

PRIMARY: Reference for waveform pattern timing of other TRXs.

SECONDARY_INT: Sets the waveform pattern timing to the Slot 1 TRX.

SECONDARY_EXT: Sets the waveform pattern timing to external trigger.

Available synchronization modes differ among the slots.

Table 4.4.6-1 Available Synchronization Modes

Slot / Synchronization Mode	Slot 1	Slot 2	Slot 3	Slot 4
NONE	✓	✓	✓	✓
PRIMARY	✓	–	–	–
SECONDARY_INT	–	✓	✓	✓
SECONDARY_EXT	✓	✓	✓	✓

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

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

When set to SECONDARY_INT or SECONDARY_EXT, the TRX 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 TRX set to SECONDARY_INT outputs waveform pattern synchronized with the Slot 1 TRX.
The TRX set to SECONDARY_EXT 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 TRX, the trigger setting on the rear panel of the MT8870A is changed.

Table 4.4.6-2 Trigger Setting at Pattern Synchronization

Synchronization Mode / Trigger Setting Command	NONE	PRIMARY	SECONDARY_INT	SECONDARY_EXT
: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 SECONDARY_EXT and Slot 1 are changed to INPUT.

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

1. Set the operation mode of all the TRXs to Sequence mode.
   `:SOURce:GPRF:GENerator:MODE SEQUENCE`
2. Set the sequence table and order of all the TRXs by the commands below.
   `:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern`
3. Set the TRXs from Slot 2 to 4 as below.
   Synchronization mode: Synchronizes with Slot 1 pattern (SECONDARY_INT)
   `:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize SECONDARY_INT`
4. Execute the sequence of the TRXs from Slot 2 to 4 as below. The waveform pattern is not output at this timing.
   `:SOURce:GPRF:GENerator:SEQuence:EXECute`
5. Query the TRXs 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 TRX of Slot 1 as below.
   Synchronization mode: Primary of pattern (PRIMARY)
   The waveform pattern is not synchronized among TRXs at this timing.
   `:SOURce:GPRF:GENerator:SEQuence:COMBination:PATTern:SYNChronize PRIMARY`
7. Execute the sequence of the Slot 1 TRX as below. When the Slot 1 TRX starts sequence, the waveform patterns of all the TRXs 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.....5-2
5.1.1 Fundamental operation commands 5-2
5.1.2 VSG commands.....5-15
5.2 Details of Commands.....5-27
5.2.1 Fundamental operation commands 5-28
5.2.2 VSG commands.....5-152

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 IPv4 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 IPv4 Subnet Mask	-----	:SYSTem:COMMunicate:NET:IPVFour:CURRent:SMASk?	<mask>
IPv4 DNS Server Auto	:SYSTem:COMMunicate:NET:IPVFour:DNS:AUTO:ENABle <switch>	:SYSTem:COMMunicate:NET:IPVFour:DNS:AUTO:ENABle?	<switch>
IPv4 DNS Primary Address	:SYSTem:COMMunicate:NET:IPVFour:DNS:PRIMary <ip4addr>	:SYSTem:COMMunicate:NET:IPVFour:DNS:PRIMary?	<ip4addr>
IPv4 DNS Secondary Address	:SYSTem:COMMunicate:NET:IPVFour:DNS:SECondary <ip4addr>	:SYSTem:COMMunicate:NET:IPVFour:DNS:SECondary?	<ip4addr>
IPv4 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>
IPv4 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 IPv6 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 IPv6 Link Local Address	-----	:SYSTem:COMMunicate:NET:IPVSix:CURRent:LOCaladdress?	<ip6addr>
IPv6 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
IPv6 DNS Primary Address	:SYSTem:COMMunicate:NET:IPVSix:DNS:PRIMary <ip6addr>	:SYSTem:COMMunicate:NET:IPVSix:DNS:PRIMary?	<ip6addr>
IPv6 DNS Secondary Address	:SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary <ip6addr>	:SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary?	<ip6addr>
IPv6 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 No.	-----	:SYSTem:INFormation:DEVice:ID?	<serial>
Module Power ON Count	-----	:SYSTem:INFormation:POWeron:COUNt?	<num>
Main Frame Information	-----	:SYSTem:INFormation:MAINframe? <index>	-----
Main Frame Serial No.	-----	: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>
TRX 1/2	-----	:SYSTem:INFormation:MAINframe:TRX?	<trx>
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:LAST time?	<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>[,<output>,…,<output>]	:ROUTe:PORT:CONNect:DIRection?	<input>,<output>[,<output>,…,<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 into 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>[,<Port5out>,<Port5in>,<Port6out>,<Port6in>,<Port7out>,<Port7in>,<Port8out>,<Port8in>,<Port9out>,<Port9in>,<Port10out>,<Port10in>,<Port11out>,<Port11in>,<Port12out>,<Port12in>]	:CALCulate:EXTLoss:TABLe:VALue:ALL? <n>	<frequency>,<Port1out>,<Port1in>,<Port2out>,<Port2in>,<Port3out>,<Port3in>,<Port4out>,<Port4in>[,<Port5out>,<Port5in>,<Port6out>,<Port6in>,<Port7out>,<Port7in>,<Port8out>,<Port8in>,<Port9out>,<Port9in>,<Port10out>,<Port10in>,<Port11out>,<Port11in>,<Port12out>,<Port12in>]
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 Gap Length	: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>

Baseband ARB (Cont'd)

Function	Command	Query	Response
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>
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.

■ Conventions used in 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
```

Details

With MU887002A-007, the calibration takes longer than the time without it.

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 for Port 1 to Port 4 in the loss correction table lists.
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 Port 1 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 7300 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

Details

When outputting signals from multiple ports, the signals will be output below the specified output level if the difference in values in the loss correction table between the ports exceeds 12 dB. In this case, the status indication lamp 2 will be lit orange.
However, even if it exceeds 8 dB, the error in output level may become large.

Example of Use

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

No	Frequency	Port 1 (dB)		Port 2 (dB)		Port 3 (dB)		Port 4 (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 Port 1 to Port 12.

Command

```
:CALCulate:EXTLoss:TABLe:VALue:ALL
<frequency>,<Port1out>,<Port1in>,<Port2out>,<Port2in>,<Port3out>,<Port3in
>,<Port4out>,<Port4in>[,<Port5out>,<Port5in>,<Port6out>,<Port6in>,<Port7o
ut>,<Port7in>,<Port8out>,<Port8in>,<Port9out>,<Port9in>,<Port10out>,<Port
10in>,<Port11out>,<Port11in>,<Port12out>,<Port12in>]
```

Query

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

Response

<frequency>,<Port1out>,<Port1in>,<Port2out>,<Port2in>,<Port3out>,<Port3in>,<Port4out>,<Port4in>[,<Port5out>,<Port5in>,<Port6out>,<Port6in>,<Port7out>,<Port7in>,<Port8out>,<Port8in>,<Port9out>,<Port9in>,<Port10out>,<Port10in>,<Port11out>,<Port11in>,<Port12out>,<Port12in>]

Parameters

<frequency>	Frequency point to be set
Range	10 MHz to 7300 MHz
Resolution	1 Hz
Suffix code	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
<Port1out>	Port 1 DL loss value
…	…
<Port12out>	Port 12 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
<Port1in>	Port 1 UL loss value
…	…
<Port12in>	Port 12 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

Details

The settings for Port 1 to 4 are mandatory. Set 0 dB for a port which does not need loss values.
The settings of Ports 5 to 12 can be omitted only when no value is set for all these ports.
When a value is set for one of them, all the ports need to be set. Set 0 dB for a port which does not need loss values.
When outputting signals from multiple ports, the signals will be output below the specified output level if the difference in values in the loss correction table between the ports exceeds 12 dB. In this case, the status indication lamp 2 will be lit orange.
However, even if it exceeds 8 dB, the error in output level may become large.

Example of Use

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

No	Frequency	Port 1 (dB)		Port 2 (dB)		Port 3 (dB)		Port 4 (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
		Port 5 (dB)		Port 6 (dB)		Port 7 (dB)		Port 8 (dB)	
		out	in	out	in	out	in	out	in
		18.00	19.00	20.00	21.00	22.00	23.00	24.00	25.00
		Port 9 (dB)		Port 10 (dB)		Port 11 (dB)		Port 12 (dB)	
		out	in	out	in	out	in	out	in
		26.00	27.00	28.00	29.00	30.00	31.00	32.00	33.00

:CALCulate:EXTLoss:TABLe:VALue:ALL 1GHZ,10,11,12,13,14,15,16,17,18,19,20,
21,22,23,24,25,26,27,28,29,30,31,32,33
:CALCulate:EXTLoss:TABLe:VALue:ALL? 1
> 1000000000,10.00,11.00,12.00,13.00,14.00,15.00,16.00,17.00,18.00,19.00,20.00,
21.00,22.00,23.00,24.00,25.00,26.00,27.00,28.00,29.00,30.00,31.00,32.00,33.00

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?
> MU887002A

: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

Details

When outputting signals from multiple ports, the signals will be output below the specified output level if the difference in values in the loss correction table between the ports exceeds 12 dB. In this case, the status indication lamp 2 will be lit orange.
However, even if it exceeds 8 dB, the error in output level may become large.

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 test ports for RF signals input/output.

Command

```
:ROUTe:PORT:CONNect:DIRection <input>,<output>[,<output>,…,<output>]
```

Query

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

Response

<input>,<output>[,<output>,…,<output>]

Parameters

<input>	Test Port number
Range	OFF, Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
OFF	No any Input Test Ports
Default	PORT1
<output>	Test Port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
Default	PORT1

Details

Test Port 1 to Test Port 12 can be set for both input and output.
With MU887002A-007, there are the following restrictions. The command can be executed, but an error occurs.

- Test Port 1 to Test Port 4 cannot be set for RF signals above 6 GHz.
- Test Port 5 to Test Port 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

Example of Use

To set Test Port 1 for RF input signals and Test Port 2 for RF output signals:
:ROUTe:PORT:CONNect:DIRection PORT1,PORT2
:ROUTe:PORT:CONNect:DIRection?
> PORT1,PORT2

Set Test Port 4, Test Port 6, and Test Port 10 as RF signal output test ports, and set No any Input Test Ports.
:ROUT:PORT:CONN:DIR OFF,PORT10,PORT4,PORT6

:ROUT:PORT:CONN:DIR?
> OFF,PORT4,PORT6,PORT10

: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 TRX to control installed.
TRGIN1	Trigger input to Trigger Connector 1.
SLOT1	Trigger signal output from the Slot 1 module TRX.
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>	Trigger connector input/output setting
INPUT	Input
OUTPUT	Output
Default	INPUT

Details

The trigger connector input/output can be set for each module TRX.

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 operation 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 operation 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 operation 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 operation 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 operation 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 operation 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 operation 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 operation 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 operation 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 operation 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 operation 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 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 measurement operation status register bits 0, 1 to 15 becomes the response.

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 operation 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 operation 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 operation 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 operation 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 $2^8 = 256$ 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

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

Command

```
:STATus:PRESet
```

Details

The operation status register, questionable status register, measurement operation status register, measurement questionable status register, signal generator operation status register, signal generator questionable status 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 status register and condition register is not initialized.
Status byte and Standard 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 status 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 status 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 status 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 status 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 status 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 status 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 status 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 status 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 status 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 status 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 questionable status register:
:STAT:QUES:MEAS?
> 1

:STATus:QUEStionable:MEASure:NTRansition

Measurement Questionable Status Negative Transition Filter

Function

Sets or queries measurement questionable status 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 status 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 status 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 status 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 status 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 status 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 status 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 status 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 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>		Factory default reference frequency correction value
	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
5	Normal file but cannot prepare for TRX 2 updates

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.

For MU887002A, the firmware package to be installed must be V3.00.00 or later.
For MU887002A with MU887002A-007, the firmware package to be installed must be V4.00.00 or later.

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

For MU887002A, the GPIB address can be set, but GPIB cannot be used.
A GPIB address can be set for each module TRX 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 TRXs.

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 Slot 1
Range	1 to 30
Default	1
<address2>	GPIB address of Slot 2
Range	1 to 30
Default	1
<address3>	GPIB address of Slot 3
Range	1 to 30
Default	1
<address4>	GPIB address of Slot 4
Range	1 to 30
Default	1

Details

For MU887002A, the GPIB address can be set, but GPIB cannot be used.
This command can set the GPIB addresses of all the modules TRXs.
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>

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>

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 MU887002A 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	V4

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 IPv4 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 IPv4 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

IPv4 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>

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

IPv4 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 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 :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

IPv4 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 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 :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:IPVF: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

IPv4 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 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 :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

IPv4 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>

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 IPv6 Default Router Address

Function

Queries current default router address.

Query

```
:SYSTem:COMMunicate:NET:IPVSix:CURRent:DEFRouter?
```

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

: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>,<ip6addr>,…

Parameter

<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 IPv6 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

IPv6 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>

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

IPv6 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>

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 :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"
:SYST:COMM:NET:REST

:SYST:COMM:NET:IPVS:DNS:PRIM?
> fe80::215:c5ff:febb:42cf
```

:SYSTem:COMMunicate:NET:IPVSix:DNS:SECondary

IPv6 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>

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 :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

IPv6 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 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 :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 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 :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>

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>	Minute
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>.....

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"
<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.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 0 to 10

Example of Use

:SYST:INF? 1
> TRX Test Module
:SYST:INF? 2
> MU887002A
:SYST:INF? 3
> 3.01.10.5.11
:SYST:INF? 4
> CentOS Linux release 7.4.1708 (Core) /Linux 3.10.0-693.el7.x86_64
:SYST:INF? 5
> 5,"Carrier Board"/00, "BB Board"/01, "RF Board"/01,"Divider1 Board"/01,"Divider2 Board"/01
:SYST:INF? 6
> 4,"BB FPGA"/00.00.92,"RF FPGA"/07.00.00,"Divider1 FPGA"/16.00.00,"Divider2 FPGA"/16.00.00
:SYST:INF? 7
> 1, 007/"7GHz Frequency Extension"/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>	Module serial number 10-digit number Example 6200103056

Example of Use

:SYST:INF:DEV:ID?
> 6123456789

Remarks

For MU887002A, the same module serial number is returned, for TRX1 and TRX2.

: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 (0 to 999)

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:MAINframe:TRX?

TRX 1/2

Function

Queries TRX 1/2.

Query

```
:SYSTem:INFormation:MAINframe:TRX?
```

Response

<trx>

Parameter

<trx>	TRX 1/2
1	TRX 1
2	TRX 2

Example of Use

:SYST:INF:MAIN:TRX?

> 1

: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 the slot number where module installed.

Query

```
:SYSTem:INFormation:SLOT?
```

Response

<slot>

Parameter

<slot>	Slot number
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:

"Chassis EEPROM: xx","Module EEPROM: xx","RAM: xx","Correction ROM: xx","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?
> "Chassis EEPROM: Pass","Module EEPROM:Pass","RAM: Pass","Correction ROM: Pass","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>	day (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?
> 01.00.00

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
9	MU887002A does not support the set waveform file

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 *Waveform File for Short Range Wireless 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.

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

: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.

: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 250 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>,<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 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 ≤ f ≤ 250 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:SYNC:STAR:CANC

: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:
PRIMARY	Starts pattern synchronized output as a primary. (Only Slot 1 module can be set to this.)
SECONDARY_INT	Prepares pattern synchronized output as a secondary. (Synchronizes with the MT8870A signal.)
SECONDARY_EXT	Prepares pattern synchronized output as a secondary. (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 PRIMARY.
:SOUR:GPRF:GEN:ARB:WAV:SYNC:STAR:PATT:SEL PRIMARY,"MV887012A_GSM_0002"

To query the information of the pattern currently being played:
:SOUR:GPRF:GEN:ARB:WAV:SYNC:STAR:PATT:SEL?
> "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
PRIMARY	✓	–	–	–
SECONDARY_INT	–	✓	✓	✓
SECONDARY_EXT	✓	✓	✓	✓

✓: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	PRIMARY	SECONDARY_INT	SECONDARY_EXT
: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 SECONDARY_EXT, 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	Release of synchronized output start stand-by 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 the radio frequency.

Command

```
:SOURce:GPRF:GENerator:RFSettings:FREQuency <freq>
```

Query

```
:SOURce:GPRF:GENerator:RFSettings:FREQuency?
```

Response

<freq>

Unit Hz

Parameter

<freq>	Frequency
Range	400 MHz to 6.0 GHz 400 MHz to 6.0 GHz (with MU887002A-007, Test Port 1 to 4) 400 MHz to 7.3 GHz (with MU887002A-007, Test Port 5 to 12)
Resolution	1 Hz
Suffix Code	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
Default	1 GHz

Details

With MU887002A-007, there are the following restrictions. The command can be executed, but an error occurs.

- Test Port 1 to Test Port 4 cannot be set for RF signals above 6 GHz.
- Test Port 5 to Test Port 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

Example of Use

To set the radio frequency to 3 GHz:
:SOUR:GPRF:GEN:RFS:FREQ 3GHZ

To query the radio frequency setting value:
:SOUR:GPRF:GEN:RFS:FREQ?
> 3000000000

Remarks

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	(–130.0 dBm – CableLoss) to (0.0 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 running

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.
PRIMARY	Starts waveform list table as a primary. (Available only for the module installed to Slot 1.)
SECONDARY_INT	Starts waveform list table as a secondary. (Synchronizes with the MT8870A signal.)
SECONDARY_EXT	Starts waveform list table as a secondary. (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 SECONDARY_EXT

Remarks

Available Synchronization modes vary according to the slot to which the module is installed.

Slot No. / Synchronization Mode	Slot1	Slot2	Slot3	Slot4
NONE	✓	✓	✓	✓
PRIMARY	✓	–	–	–
SECONDARY_INT	–	✓	✓	✓
SECONDARY_EXT	✓	✓	✓	✓

✓: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.

<mode> / Command	NONE	PRIMARY	SECONDARY_INT	SECONDARY_EXT
: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 SECONDARY_EXT, 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 running

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 running

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

: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>	Baseband mode of Segment number 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

Details

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.

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

An execution occurs if the operation mode is:

- Sequence mode running

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 data.

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 running

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>		End condition of Segment number 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

Details

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.

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

An execution error 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: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	400 MHz to 6.0 GHz 400 MHz to 6.0 GHz (with MU887002A-007, Test Port 1 to 4) 400 MHz to 7.3 GHz (with MU887002A-007, Test Port 5 to 12)
Resolution	1 Hz
Suffix Code	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
Default	1 GHz

Details

With MU887002A-007, there are the following restrictions. The command can be executed, but an error occurs.

- Test Port 1 to Test Port 4 cannot be set for RF signals above 6 GHz.
- Test Port 5 to Test Port 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

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

An execution error 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: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>	Frequency of Segment number n
<freq_1>	Frequency of Segment 1
<freq_2>	Frequency of Segment 2
…	
<freq_10001>	Frequency of Segment 10001
Range	400 MHz to 6.0 GHz 400 MHz to 6.0 GHz (with MU887002A-007, Test Port 1 to 4) 400 MHz to 7.3 GHz (with MU887002A-007, Test Port 5 to 12)
Resolution	1 Hz
Suffix	HZ, Z, KHZ, KZ, MHZ, MZ, GHZ, GZ, (uses HZ when omitted)
Default	1 GHz

Details

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.
With MU887002A-007, there are the following restrictions. The command can be executed, but an error occurs.

- Test Port 1 to Test Port 4 cannot be set for RF signals above 6 GHz.
- Test Port 5 to Test Port 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

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

Remarks

An execution error occurs if the operation mode:

- Sequence mode running

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 running

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 running

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 running

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 running

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 running

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 100.0 dBm*
Resolution	0.1 dB
Suffix Code	DBM (uses DBM when omitted)
Default	–120.0 dBm

*: Settable range with combination of test ports and cable loss values

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

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: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 of Segment number n
<level_1> Level of Segment 1
<level_2> Level of Segment 2
…
<level_10001> Level of Segment 10001
Range –230.0 to 100.0 dBm*
Resolution 0.1 dB
Suffix Code DBM (uses DBM when omitted)
Default –120.0 dBm

*: Settable range settable with combination of test ports and cable loss values

Details

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.

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

An execution error occurs when 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: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 running

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>	Repetition control at Segment number 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

Details

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.

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

An execution error occurs when 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: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*
Default	0

*: Repetition control is not executed when the repetition count is 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

An execution error occurs when 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: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>	Repetition count of Segment number 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*
Default	0

*: If the repetition count is set to 0, repetition control is not performed.

Details

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 controls for Segment 11 and subsequent segments are not changed.

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

An execution error occurs when 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: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

Details

A parameter error occurs if the jump destination segment number is bigger than the set segment number.

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

An execution error occurs when 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: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 destination segment number of Segment number 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*
	Default	1

*: A parameter error occurs if the jump destination segment number is bigger than the set segment number.

Details

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.

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

An execution error occurs when 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: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>[,<port>,...,<port>]
```

Query

```
:SOURce:GPRF:GENerator:SEQuence:RX:OUTPut:STATe?
<list_table_no>,<segment_no>
```

Response

<port>[,<port>,...,<port>]

Parameters

<list_table_no>	Sequence table number
Range	1 to 16
<segment_no>	Segment number
Range	1 to 10001
<port>	Test Port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
Default	PORT1

Details

The ports on the front panel that output RF signals are set.
With MU887002A-007, there are the following restrictions. The command can be executed, but an error occurs.

- Test Port 1 to Test Port 4 cannot be set for RF signals above 6 GHz.
- Test Port 5 to Test Port 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

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

Set the output port of Segment 10 of Sequence table #1 to PORT1, PORT7, and PORT11.
:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT 1,10,PORT1,PORT11,PORT7

Queries the output port of Segment 10 of Sequence table #1.
:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT? 1,10
> PORT1,PORT7,PORT11

Remarks

An execution error occurs when 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:RX:OUTPut:STATe:ALL

Output Port Group of All Segments in Sequence

Function

Sets or queries output port of all segments 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 number n
<port_1>	Port setting for Segment 1
<port_2>	Port setting for Segment 2
….	
<port_10001>	Port setting for Segment 10001
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
Default	PORT1

Details

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.
To set multiple ports for one segment, connect them with "&".

With MU887002A-007, there are the following restrictions. The command can be executed, but an error occurs.

- Test Port 1 to Test Port 4 cannot be set for RF signals above 6 GHz.
- Test Port 5 to Test Port 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

Example of Use

To set the output port of all segments of Sequence table #10 collectively:
:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT:ALL 10,PORT2,PORT3,PORT3

To query the output port of all segments of Sequence table #10:
:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT:ALL? 10
> PORT2,PORT3,PORT3,PORT1,PORT1,...,PORT1

Set the output ports of all segments of Sequence table #10 collectively.

Settings	Segment	1	2	3	4	5	6 to 10001
	Output Port	2, 4	3, 5, 11	3	4, 5	11	1

:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT:ALL
10,PORT2&PORT4,PORT3&PORT5&PORT11,PORT3,PORT4&PORT5,PORT11

To query the output port of all segments of Sequence table #10:
:SOUR:GPRF:GEN:SEQ:RX:OUTP:STAT:ALL? 10
> PORT2&PORT4,PORT3&PORT11&PORT5,PORT3,PORT4&PORT5,PORT11,
PORT1,PORT1,...,PORT1

Remarks

An execution error occurs when 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: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 running

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>	Continuous step count of Segment number 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

Details

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.

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

An execution error occurs when 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: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>

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 running

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 time of Segment number 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

Details

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.

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

An execution error occurs when 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: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 running

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 source of Segment number 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

Details

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.

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

An execution error occurs when 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: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 running

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 of Segment number n
<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>	Waveform change trigger setting of Segment number n
<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

Details

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.

Example of Use

To set the waveform change trigger of each segment of Sequence table #3 collectively:
: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

An execution error occurs when 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: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 running

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 520
<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 running

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 running

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 running

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 520
<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 running

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 520
<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 running

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 520

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 running

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 520
<file_name>	Waveform file name (character string)
<group_number>	Group number
Range	Range of group number defined by waveform file

Details

The file name is a character string without the extension (xml).
When the group number is omitted, 1 is set.

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

An execution error occurs when the operation mode is:

- Sequence mode running

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

Details

With MU887002A-007, there are the following restrictions. The command can be executed, but an error occurs.

- Test Port 1 to Test Port 4 cannot be set for RF signals above 6 GHz.
- Test Port 5 to Test Port 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

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 running

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 6-2
6.1.1 IEEE488.2 common commands 6-2
6.1.2 Basic operation commands 6-3
6.1.3 VSG commands 6-11
6.2 Details of Commands 6-12
6.2.1 IEEE488.2 common commands 6-13
6.2.2 Commands in alphabetical order 6-23

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:MU887002A],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

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 IPv4 Default Gateway	-----	IPV4_CURRENT_DEFAULTGATEWAY?	gateway
Get Current IPv4 Subnet Mask	-----	IPV4_CURRENT_SUBNETMASK?	mask
IPv4 DNS Server Auto	IPV4_DNS_AUTO_SW switch	IPV4_DNS_AUTO_SW?	switch
IPv4 DNS Primary Address	IPV4_DNS_PRIMARY ip4addr	IPV4_DNS_PRIMARY?	ip4addr
IPv4 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
IPv4 Default Gateway	IPV4_STATIC_DEFAULTGATEWAY gateway	IPV4_STATIC_DEFAULTGATEWAY?	gateway
IPv4 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 IPv6 Default Router Address	-----	IPV6_CURRENT_DEFAULTROUTER?	ip6addr
Get IPv6 Link Local Address	-----	IPV6_CURRENT_LINK_LOCAL_ADDR?	ip6addr
IPv6 DNS Server Auto	IPV6_DNS_AUTO_SW switch	IPV6_DNS_AUTO_SW?	switch
IPv6 DNS Primary Address	IPV6_DNS_PRIMARY ip6addr	IPV6_DNS_PRIMARY?	ip6addr
IPv6 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
IPv6 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
TRX 1/2		SYSTRX?	trx
Package Version	-----	SYSVER?	ver
Chassis Power-on Count	-----	MFPOWERONCNT?	num
Count of Module Insertion	-----	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
Lower Limit of Firmware Update	-----	UPDATELOWERLIMIT?	lowerlimit

Interface Setting

Function	Command	Query	Response
Set In/Out of Trigger Connector	MFTRIGPORT trigger	MFTRIGPORT?	trigger
Set Direction of RF Connector	PORT input,output[,output,…,output]	PORT?	input,output[,output,…,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 into 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[,Port5out,Port5in,Port6out,Port6in,Port7out,Port7in,Port8out,Port8in,Port9out,Port9in,Port10out,Port10in,Port11out,Port11in,Port12out,Port12in]	LOSSTBLVALALL? n	frequency,Port1out,Port1in,Port2out,Port2in,Port3out,Port3in,Port4out,Port3in,Port4in[,Port5out,Port5in,Port6out,Port6in,Port7out,Port7in,Port8out,Port8in,Port9out,Port9in,Port10out,Port10in,Port11out,Port11in,Port12out,Port12in]

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.

■ Conventions used in 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
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 1 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:MU887002A,serial1:serial2,firm1:firm2

ANRITSU	Company name (Anritsu Corporation)
MT8870A	Model name
slot	Number of slot where module installed
MU887002A	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,MU887002A,serial,firm

ANRITSU	Company name (Anritsu Corporation)
MU887002A	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?

Remarks

For MU887002A, the same module serial number is returned, for TRX1 and TRX2.

*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	Minute
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 if 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
```

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

Details

When outputting signals from multiple ports, the signals will be output below the specified output level if the difference in values in the loss correction table between the ports exceeds 12 dB. In this case, the status indication lamp 2 will be lit orange.
However, even if it exceeds 8 dB, the error in output level may become large.

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
5	Normal file but cannot prepare for TRX 2 updates

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.
For MU887002A, the firmware package to be installed must be V3.00.00 or later.
For MU887002A with MU887002A-007, the firmware package to be installed must be V4.00.00 or later.

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
```

Details

With MU887002A-007, the calibration takes longer than the time without it.

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

For MU887002A, the GPIB address can be set, but GPIB cannot be used.
A GPIB address can be set for each module TRX 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 TRXs.

Command

```
GPIBADDRALL address1,address2,address3,address4
```

Query

```
GPIBADDRALL?
```

Response

address1,address2,address3,address4

Parameter

address1	GPIB address of Slot 1
Range	1 to 30
Default	1
address2	GPIB address of Slot 2
Range	1 to 30
Default	1
address3	GPIB address of Slot 3
Range	1 to 30
Default	1
address4	GPIB address of Slot 4
Range	1 to 30
Default	1

Details

For MU887002A, the GPIB address can be set, but GPIB cannot be used.
This command can set the GPIB addresses of all the modules TRXs.
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 following 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 IPv4 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 IPv4 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

IPv4 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	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

IPV4_DNS_AUTO_SW ON
NETRESTART

IPV4_DNS_AUTO_SW?
>ON

IPV4_DNS_PRIMARY

IPv4 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 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

IPv4 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 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

IPv4 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

IPv4 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 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 IPv6 Address and Prefix

Function

Queries current IPv6 address/prefix length.

Query

```
IPV6_CURRENT_ADDR?
```

Response

num,ipaddr,…

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 IPv6 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 IPv6 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

IPv6 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

IPv6 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

IPv6 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

IPV6_STATIC_DEFAULTROUTER

IPv6 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 sets to On.
V6	Only IPv6 sets 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 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 Port 1 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 7300 MHz
Resolution	1 Hz
Suffix Code	HZ, Z,KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
Port1_2/out	Port 1/2 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port1_2/in	Port 1/2 UL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port3_4/out	Port 3/4 DL loss value
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port3_4/in	Port 3/4 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

Details

When outputting signals from multiple ports, the signals will be output below the specified output level if the difference in values in the loss correction table between the ports exceeds 12 dB. In this case, the status indication lamp 2 will be lit orange.
However, even if it exceeds 8 dB, the error in output level may become large.

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 Port 1 to Port 12.

Command

```
LOSSTBLVALALL
frequency,Port1out,Port1in,Port2out,Port2in,Port3out,Port3in,Port4out,Port4in[,Port5out,Port5in,Port6out,Port6in,Port7out,Port7in,Port8out,Port8in,Port9out,Port9in,Port10out,Port10in,Port11out,Port11in,Port12out,Port12in]
```

Query

```
LOSSTBLVALALL? n
```

Response

frequency,Port1out,Port1in,Port2out,Port2in,Port3out,Port3in,Port4out,Port4in[,Port5out,Port5in,Port6out,Port6in,Port7out,Port7in,Port8out,Port8in,Port9out,Port9in,Port10out,Port10in,Port11out,Port11in,Port12out,Port12in]

Parameters

frequency	Frequency point to be set
Range	10 MHz to 7300 MHz
Resolution	1 Hz
Suffix Code	HZ, Z,KHZ, KZ, MHZ, MZ, GHZ, GZ (uses HZ when omitted)
Port1out	Port 1 DL loss value
…	…
Port12out	DL loss value of Port 12
Range	–100.00 to 100.00 dB
Resolution	0.01 dB
Suffix code	DB (uses DB when omitted)
Port1in	Port 1 UL loss value
…	…
Port12in	UL loss value of Port 12
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

Details

The settings for Port 1 to 4 are mandatory. Set 0 dB for a port which does not need loss values.
The settings of Ports 5 to 12 can be omitted only when no value is set for all these ports.
When a value is set for one of them, all the ports need to be set. Set 0 dB for a port which does not need loss values.
When outputting signals from multiple ports, the signals will be output below the specified output level if the difference in values in the loss correction table between the ports exceeds 12 dB. In this case, the status indication lamp 2 will be lit orange.
However, even if it exceeds 8 dB, the error in output level may become large.

Example of Use

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

No	Frequency	Port 1 (dB)		Port 2 (dB)		Port 3 (dB)		Port 4 (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
		Port 5 (dB)		Port 6 (dB)		Port 7 (dB)		Port 8 (dB)	
		out	in	out	in	out	in	out	in
		18.00	19.00	20.00	21.00	22.00	23.00	24.00	25.00
		Port 9 (dB)		Port 10 (dB)		Port 11 (dB)		Port 12 (dB)	
		out	in	out	in	out	in	out	in
		26.00	27.00	28.00	29.00	30.00	31.00	32.00	33.00

LOSSTBLVALALL 1GHZ,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,
29,30,31,32,33
LOSSTBLVALALL? 1
> 1000000000,10.00,11.00,12.00,13.00,14.00,15.00,16.00,17.00,18.00,19.00,20.00,
21.00,22.00,23.00,24.00,25.00,26.00,27.00,28.00,29.00,30.00,31.00,32.00,33.00

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 (0 to 999)

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 TRX.

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 REBoot.

PORT

Set Direction of RF Connector

Function

Sets or queries test ports for RF signal input/output.

Command

```
PORT input,output[,output,…,output]
```

Query

```
PORT?
```

Response

input,output[,output,…,output]

Parameters

input	Test Port number
Range	OFF, Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
OFF	No any Input Test Ports
Default	PORT1
output	Test Port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
Default	PORT1

Details

Test Port 1 to Test Port 12 can be set for both input and output.
With MU887002A-007, there are the following restrictions. The command can be executed, but an error occurs.

- Test Port 1 to Test Port 4 cannot be set for RF signals above 6 GHz.
- Test Port 5 to Test Port 12 cannot be set as both the input and the output ports for RF signals of 5.9 GHz or higher.

Example of Use

To set Test Port 1 for RF input signals and Test Port 2 for RF output signals:
PORT PORT1,PORT2
PORT?
> PORT1,PORT2

To set Test Port 4, Test Port 6, and Test Port 10 as RF signal output test ports, and set No any Input Test Ports.
:ROUT:PORT:CONN:DIR OFF,PORT10,PORT4,PORT6

:ROUT:PORT:CONN:DIR?
> OFF,PORT4,PORT6,PORT10

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 (≤ 15 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.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	0 to 10

Example of Use

```
SYSINFO? 1
> TRX Test Module
SYSINFO? 2
> MU887002A
SYSINFO? 3
> 3.01.10.5.11
SYSINFO? 4
> CentOS Linux release 7.4.1708 (Core) /Linux 3.10.0-693.el7.x86_64
SYSINFO? 5
> 5,"Carrier Board"/00, "BB Board"/01, "RF Board"/01,"Divider1 Board"/01,"Divider2 Board"/01
SYSINFO? 6
> 4,"BB FPGA"/00.00.92,"RF FPGA"/07.00.00,"Divider1 FPGA"/16.00.00,"Divider2
FPGA"/16.00.00
SYSINFO? 7
> 1, 007/"7GHz Frequency Extension"/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?
> MU887002A

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	Module serial number
	10-digit number Example 6200103056

Example of Use

SYSSERIAL?
> 6123456789

Remarks

For MU887002A, the same module serial number is returned, for TRX1 and TRX2.

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:

"Chassis EEPROM: xx","Module EEPROM: xx","RAM: xx","Correction ROM: xx","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	hour (0 to 23)
min	minute (0 to 59)

Example of Use

SYSTEST?
> "Chassis EEPROM: Pass","Module EEPROM: Pass","RAM: Pass","Correction ROM: Pass","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	day (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 TRX to control installed.
TRGIN1	Trigger input to Trigger Connector 1.
SLOT1	Trigger signal output from the Slot 1 module TRX.
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

SYSTRX?

TRX 1/2

Function

Queries TRX 1/2.

Query

```
SYSTRX?
```

Response

trx

Parameter

trx	TRX 1/2
1	TRX 1
2	TRX 2

Example of Use

SYSRTX?
> 1

SYSVER?

Package Version

Function

Queries package version.

Query

```
SYSVER?
```

Response

ver

Parameter

ver	Package version (xx.xx.xx)

Example of Use

SYSVER?
> 01.00.00

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.

UPDATELOWERLIMIT?

Lower Limit of Firmware Update

Function

Queries the earliest version of the compatible firmware updates.

Query

```
UPDATELOWERLIMIT?
```

Response

lowerlimit

Parameter

lowerlimit	The earliest version of the compatible firmware updates (indicated by "xx.xx.xx")

Details

For MU887002A, the firmware package to be installed must be V3.00.00 or later.
For MU887002A with MU887002A-007, the firmware package to be installed must be V4.00.00 or later.

Example of Use

UPDATELOWERLIMIT?
> 04.00.00

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 7-2
7.2 Instruments for Performance Tests 7-3
7.3 Performance Test for Each Measurement 7-8
7.3.1 VSWR 7-8
7.3.2 SG Level Accuracy 7-12
7.3.3 SG Spurious 7-22
7.3.4 SA Level Accuracy 7-25
7.3.5 SA Linearity 7-35
7.3.6 Pathloss Measurement Accuracy (MX887092A) 7-43
7.4 Servicing 7-46

7.1 Outline

The performance tests are performed to assure that the MU887002A performance does not deteriorate. Test the performance of the MU887002A 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
- Pathloss Measurement Accuracy (MX887092A)

We recommend testing the performance periodically once or twice a year. If the test results do not meet the specifications, please 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.

⚠ CAUTION

Warm-up the MU887002A 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 MU887002A 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.

*: When using the MU887002A for the first time after purchase, or if it has not been used for a long time, warm it up sufficiently before use because the level accuracy may be affected.

7.2 Instruments for Performance Tests

The following table lists the measuring instruments required for testing the MU887002A performance and the specifications for each instrument.

Table 7.2-1 Measuring Instruments for Performance Test (without MU887002A-007)

Instrument Name	Instrument Required Performance*1	Recommended Instrument
Signal Generator	• Frequency Range: 400 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 (MG3710E) High Frequency 6 GHz (MG3710E-036)
Power Meter	• Main Frame Accuracy: ±0.02 dB • Frequency Range: 400 to 6000 MHz • Display Resolution: 0.01 dB	Power Meter (ML2438A)
Power Sensor	• Frequency Range: 400 to 6000 MHz • Measured Power Range: –40 to +20 dBm • Input Connector: N type	Power Sensor (MA2442D)
Fixed Attenuator (6 dB)	• Frequency Range: 400 to 6000 MHz • Attenuation: 6 dB • Connector: N type	Fixed Attenuator (BW-N6W5+)
Fixed Attenuator (3 dB)	• Frequency Range: 400 to 6000 MHz • Attenuation: 3 dB • Connector: N type	Fixed Attenuator (BW-N3W5+)
Relay Adapter	• Frequency Range: 400 to 6000 MHz 100 to 8000 MHz*2 • Connector: N-J・N-J	(34NFNF50)
Coaxial cable for Pathloss measurement *2	• Frequency Range: 400 to 6000 MHz • Connector: N-P・N-J • Cable length: 1.5 m • Relative permittivity: 2.03	
OPEN/SHORT for Pathloss measurement *2	• Frequency Range: 400 to 6000 MHz • Connector: N-P	(22N50)
Signal Analyzer	• Frequency Range: 400 to 6000 MHz • Frequency Setting Resolution: 1 Hz • Measured Power Range: –140 to +20 dBm • External Reference Input: (10 MHz)	Signal Analyzer (MS269xA or MS2830A-041)

*1: The performance covers the test item measurement range.

*2: Required for performance test of Pathloss measurement accuracy.

Table 7.2-1 Measuring Instruments for Performance Test (without MU887002A-007) (Cont'd)

Instrument Name	Instrument Required Performance*1	Recommended Instrument
Network Analyzer	• Frequency Range: 400 to 6000 MHz • Input Connector: N type	Network Analyzer (MS2026C)
	• Frequency Range: 100 to 8000 MHz*2 • Input Connector: N type	Network Analyzer (S820E) 8GHz Site Master (S820E-0708) Vector Network Analyzer (S820E-0440)
Application Software		Cellular Standards Sequence Measurement (MX887010A)
		Pathloss Measurement Function for MU887002A*2 (MX887092A)

Table 7.2-2 Measuring Instruments for Performance Test (with MU887002A-007)

Instrument Name	Instrument Required Performance*1	Recommended Instrument
Signal Generator	• Frequency Range: 400 to 7300 MHz • Frequency Setting Resolution: 100 kHz • Output Level Range: –30 to 0 dBm • External Reference Input: (10 MHz) • Connector: K type	Vector Signal Generator (MG3692C) Analog down converter (8 MHz to 2 GHz) (MG3690C/5)
Power Meter	• Main Frame Accuracy: ±0.02 dB • Frequency Range: 400 to 7300 MHz • Display Resolution: 0.01 dB	Power Meter (ML2438A)
Power Sensor	• Frequency Range: 400 to 7300 MHz • Measured Power Range: –40 to +20 dBm • Input Connector: N type	Power Sensor (MA2442D)
Fixed Attenuator (6 dB)	• Frequency Range: 400 to 7300 MHz • Attenuation: 6 dB • Connector: N type	Fixed Attenuator (BW-N6W5+)
Fixed Attenuator (3 dB)	• Frequency Range: 400 to 7300 MHz • Attenuation: 3 dB • Connector: N type	Fixed Attenuator (BW-N3W5+)
Relay Adapter	• Frequency Range: 400 to 7300 MHz 100 to 8000 MHz*2 • Connector: N-J・N-J	(34NFNF50)
Conversion Adapter	• Frequency Range: 400 to 7300 MHz • Connector: K-P・N-J	(34RKNF50)
Coaxial cable for Pathloss measurement *2	• Frequency Range: 400 to 6000 MHz • Connector: N-P・N-J • Cable length: 1.5 m • Relative permittivity: 2.03	
OPEN/SHORT for Pathloss measurement *2	• Frequency Range: 400 to 6000 MHz • Connector: N-P	(22N50)
Signal Analyzer	• Frequency Range: 400 to 7300 MHz • Frequency Setting Resolution: 1 Hz • Measured Power Range: –140 to +20 dBm • External Reference Input: (10 MHz)	Signal Analyzer (MS2830A-044) Microwave Preselector Bypass (MS2830A-067) Microwave Preamplifier (MS2830A-068)

*1: The performance covers the test item measurement range

*2: Required for performance test of Pathloss measurement accuracy.

Table 7.2-2 Measuring Instruments for Performance Test (with MU887002A-007) (Cont'd)

Instrument Name	Instrument Required Performance*1	Recommended Instrument
Network Analyzer	• Frequency Range: 400 to 7400 MHz 100 to 8000 MHz*2 • Input Connector: N type	Network Analyzer (S820E) 8GHz Site Master (S820E-0708) Vector Network Analyzer (S820E-0440)
Application Software		Cellular Standards Sequence Measurement (MX887010A)
		Pathloss Measurement Function for MU887002A*2 (MX887092A)

Table 7.2-3 Measuring Instruments Item and Instrument to use

Instrument Name	Performance Test Item					
	VSWR	SG Level Accuracy	SG Spurious	SA Level Accuracy	SA Linearity	Pathloss Measurement Accuracy
Signal Generator		✓		✓	✓	
Power Meter		✓		✓	✓	
Power Sensor		✓		✓	✓	
Fixed Attenuator (6 dB)		✓				
Fixed Attenuator (3 dB)				✓	✓	
Relay Adapter		✓		✓	✓	✓
Conversion Adapter*		✓		✓	✓	
Coaxial cable for Pathloss measurement						✓
OPEN/SHORT for Pathloss measurement						✓
Signal Analyzer		✓	✓	✓	✓	
Network Analyzer	✓					✓
Application Software	✓	✓		✓	✓	✓

*: When MU887002A-007 is installed.

7.3 Performance Test for Each Measurement

7.3.1 VSWR

The following test related to the VSWR at signal input to the MU887002A connector is performed.

(1) Test target specification

Without MU887002A-007, TRX 1/2, Port 1 to Port 12, At 20 to 30°C
With MU887002A-007, TRX 1/2, Port 1 to Port 4, At 20 to 30°C

Frequency f	Specifications
400 MHz ≤ f < 450 MHz	< 1.4
450 MHz ≤ f ≤ 2700 MHz	< 1.3
2700 MHz < f ≤ 3800 MHz	< 1.4
3800 MHz < f ≤ 6000 MHz	< 1.4

With MU887002A-007, TRX 1/2, Port 5 to Port 12, At 20 to 30°C

Frequency f	Specifications
400 MHz ≤ f < 450 MHz*1	< 1.4
450 MHz ≤ f ≤ 2700 MHz*1	< 1.3
2700 MHz < f ≤ 3800 MHz*1	< 1.4
3800 MHz < f < 6000 MHz*1	< 1.4
5800 MHz ≤ f ≤ 7400 MHz*2	< 1.7

*1: Setting frequency < 5900 MHz
*2: Setting frequency ≥ 5900 MHz

(2) Measuring instruments

- Network analyzer: MS2026C (without MU887002A-007)
  S820E (with MU887002A-007)
- Calibration kit

(3) Setup

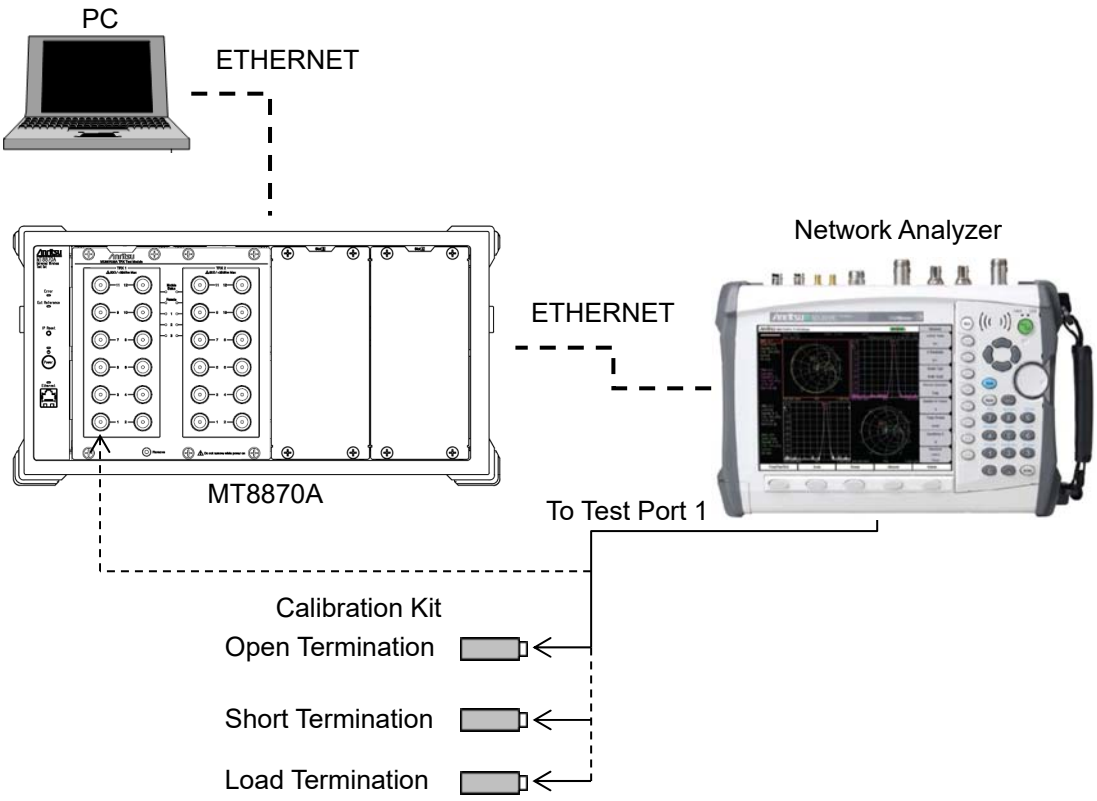


Figure 7.3.1-1 VSWR Test Setup

(4) a Procedure (without MU887002A-007, Port 1 to 12,
with MU887002A-007 Port 1 to 4,
or with MU887002A-007, Port 5 to 12
Setting frequency < 5900 MHz)

VSWR is measured at TRX 1/2, Port 1 to Port 12.
When testing other than TRX 1 Port 1, read "TRX 1 Port 1" in the following procedure as the port to be tested.

TRX 1 Port 1

1. Use an N-type cable to connect the network analyzer to the MU887002A.
2. Set the network analyzer as follows:
   • Start Frequency: 400 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 TRX 1 Port 1 of MU887002A.
5. Initialize the settings of MU887002A.
   `*RST`
6. Load a zero output waveform (zerowave.wvd, zerowave.xml). The zero output waveform is available from the Download Center on the Anritsu website.
7. Set the input and output ports to TRX 1 Port 1.
   `:ROUTe:PORT:CONNect:DIRection PORT1,PORT1`
8. Set the MU887002A input level to –65 dBm.
   `:CONFigure:CELLular:MEASurement:RFSettings:LEVel -65`
9. Set the MU887002A input frequency to 5000 MHz.
   `:CONFigure:CELLular:MEASurement:RFSettings:FREQuency 5000MHZ`
10 Set the VSG operation mode to Normal.
   `:SOURce:GPRF:GENerator:MODE NORMAL`
11 Set the VSG modulation to ARB.
   `:SOURce:GPRF:GENerator:BBMode ARB`
12. Set the VSG output level to –10 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`
13. Set the VSG output frequency to 5000 MHz.
   `:SOURce:GPRF:GENerator:RFSettings:FREQuency 5000MHZ`
14. Output the VSG signal.
   `:SOURce:GPRF:GENerator:STATe ON`
15. Measure S_{11} (VSWR) using the network analyzer.

(4) b Procedure (with MU887002A-007 Port 5 to 12
Setting frequency ≥ 5900 MHz)

VSWR is measured at TRX 1/2, Port 5 to Port 12.
When testing other than TRX 1 Port 5, read "TRX 1 Port 5" in the following procedure as the port to be tested.

TRX 1 Port 5

1. Use an N-type cable to connect the network analyzer to the MU887002A.
2. Set the network analyzer as follows:
   • Start Frequency: 400 MHz
   • Stop Frequency: 7400 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 TRX 1 Port 5 of MU887002A.
5. Initialize the settings of MU887002A.
   `*RST`
6. Load a zero output waveform (zerowave.wvd, zerowave.xml). The zero output waveform is available from the Download Center on the Anritsu website.
7. Set the input port to TRX 1 Port 5 and output port to TRX 1 Port 6. (When setting the input port to TRX 1 Port 6, set the output port to TRX 1 Port 5.)
   `:ROUTe:PORT:CONNect:DIRection PORT5,PORT6`
8. Set the MU887002A input level to –4 dBm.
   `:CONFigure:CELLular:MEASurement:RFSettings:LEVel -4`
9. Set the MU887002A input frequency to 6000 MHz.
   `:CONFigure:CELLular:MEASurement:RFSettings:FREQuency 6000MHZ`
10. Set the VSG operation mode to Normal.
    `:SOURce:GPRF:GENerator:MODE NORMAL`
11. Set the VSG modulation to ARB.
    `:SOURce:GPRF:GENerator:BBMode ARB`
12. Set the VSG output level to –10 dBm.
    `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`
13. Set the VSG output frequency to 6000 MHz.
    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 6000MHZ`
14. Stop outputting the VSG signal. .
    `:SOURce:GPRF:GENerator:STATe OFF`
15. Measure S_{11} (VSWR) using the network analyzer.

7.3.2 SG Level Accuracy

The following test related to the VSG level accuracy is performed.

(1) Test target specifications

Without MU887002A-007, TRX 1/2, Port 1 to Port 12
With MU887002A-007, TRX 1/2, Port 1 to Port 4,
CW, After calibration, At 20 to 30°C

Frequency f (MHz)	Output Level (dBm)	Specifications
400 ≤ f ≤ 3800	–120 ≤ Output Level ≤ –5	±1.0 dB
3800 < f ≤ 6000	–100 ≤ Output Level ≤ –8	±1.3 dB

With MU887002A-007, TRX 1/2, Port 5 to Port 12,
CW, After calibration, At 20 to 30°C

Frequency f (MHz)	Output Level (dBm)	Specifications
400 ≤ f ≤ 3800	–120 ≤ Output Level ≤ –5	±1.0 dB
3800 < f < 5900	–100 ≤ Output Level ≤ –8	±1.3 dB
5900 ≤ f ≤ 7300	–100 ≤ Output Level ≤ –10	±1.3 dB

(2) Measuring instruments

- Signal generator: MG3710E (without MU887002A-007)
  MG3692C (with MU887002A-007)
- Power meter: ML2438A
- Power sensor: MA2442D
- Signal Analyzer: MS269xA or MS2830A-041
  (without MU887002A-007)
  MS2830-044A (with MU887002A-007)
- Conversion Adapter: 34RKNF50 (with MU887002A-007)

(3) Setup

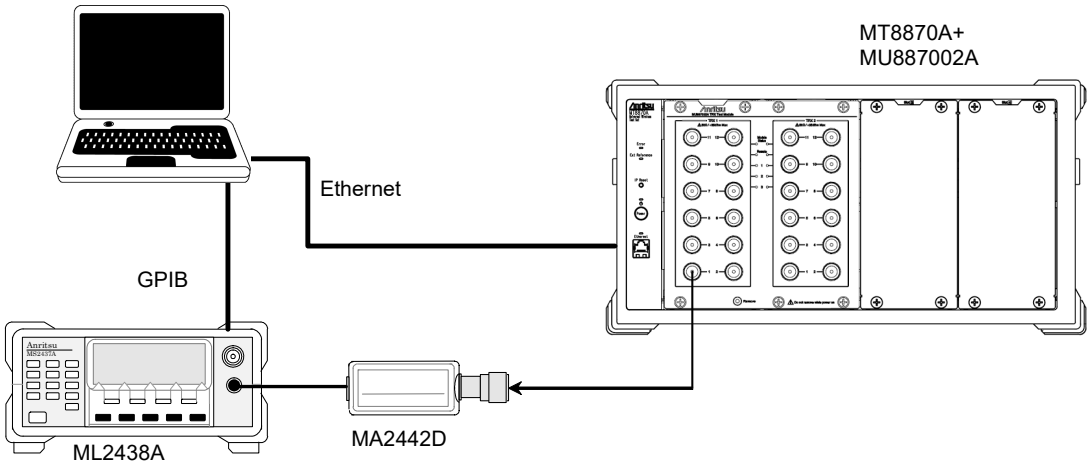


Figure 7.3.2-1 SG Level Accuracy Test Setup (Absolute level measurement)

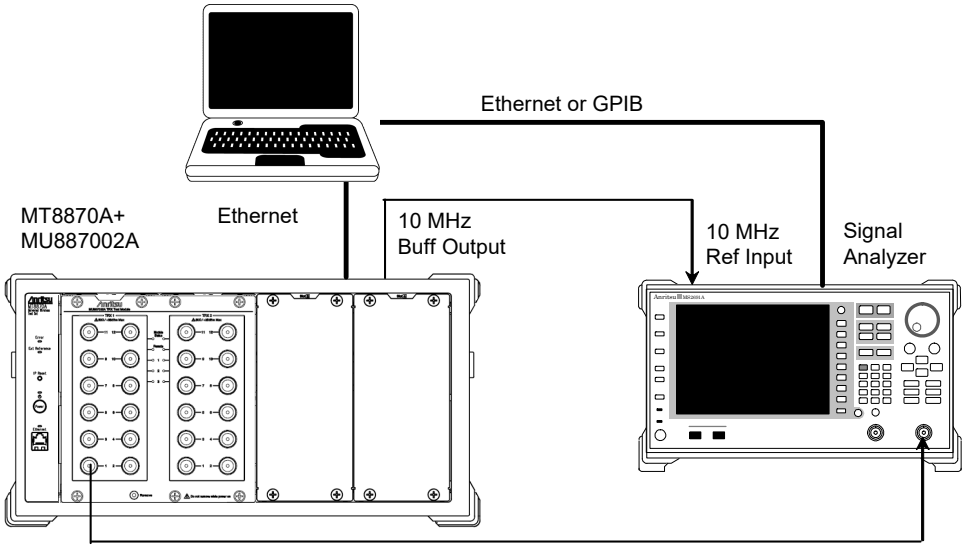


Figure 7.3.2-2 SG Level Accuracy Test Setup (Relative level measurement)

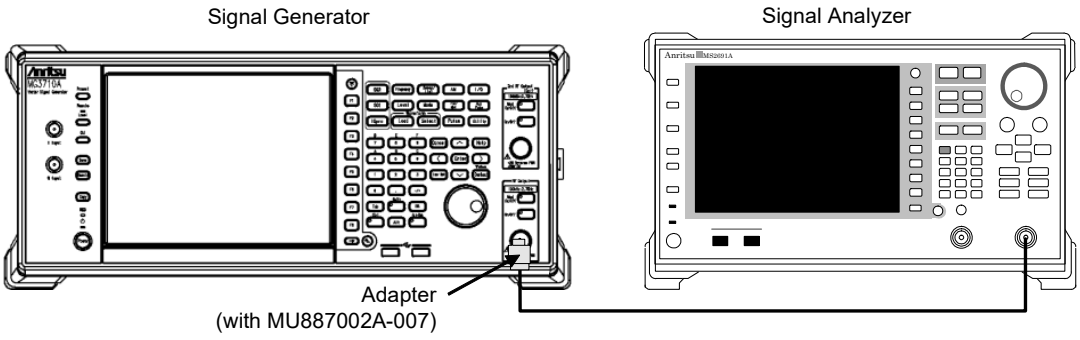


Figure 7.3.2-3 Measurement Level Calibration of Signal Analyzer

Preparation for SG Level Accuracy Measurement

1. Calibrate the power sensor (zero point and sensitivity).
2. Connect the signal generator and the signal analyzer as shown in "Figure 7.3.2-3 Measurement Level Calibration of Signal Analyzer".
3. Set the signal generator output level to –50 dBm.
4. Set the signal generator output frequency to 400.001 MHz.
5. Set the values of #1 in "Table 7.3.2-2 Settings of Signal Analyzer" to the signal analyzer.
6. #1 Measured Value is the level measured by the signal analyzer.
7. Set the values of #2 in "Table 7.3.2-2 Settings of Signal Analyzer" to the signal analyzer.
8. 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)
9. Set the signal generator output frequency in "Table 7.3.2-3 Frequency List (Signal Analyzer)" to signal generator and repeat steps 4 thru 8.

Table 7.3.2-1 Frequency List (Power Meter)

Frequency (MHz)	Frequency (MHz)	Frequency (MHz)
400.001*1,*2	4072.001*1,*2	5900.001*3
1318.001*1,*2	4990.001*1,*2	6818.001*3
2236.001*1,*2	5908.001*1	7299.999*3
3154.001*1,*2	5999.999*1	
3800.001*1,*2	–	

*1: Without MU887002A-007, TRX1/2, Port 1 to 12
or with MU887002A-007, TRX1/2, Port 1 to 4

*2: With MU887002A-007, TRX1/2, Port 5 to 12
Setting frequency < 5900 MHz

*3: With MU887002A-007, TRX1/2, Port 5 to 12
Setting frequency ≥ 5900 MHz

Table 7.3.2-2 Settings of Signal Analyzer

No.	MU887002A Output level setting (dBm)	Settings of Signal Analyzer (Microwave Preselector Bypass ON)							
		RBW (Hz)	Zone Width (Hz)	Span (kHz)	ATT (dB)	Preamp	Average	Average Count	Ref Lev (dBm)
#1	≤ 0 dBm ≥ –49 dBm	100	781.3	25	20	OFF	OFF	2	0
#2	< –49 dBm ≥ –79 dBm	100	781.3	25	0	ON	OFF	2	–40
#3	< –79 dBm ≥ –100 dBm	10	156.3	2.5	0	ON	OFF	2	–40
#4	< –100 dBm ≥ –110 dBm	10	15.63	1	0	ON	ON	2	–70
#5	< –110 dBm ≥ –120 dBm	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)	D = #1–#2 (dB)
400.001*1,*2	400.001*1,*2			
1318.001*1,*2	1318.001*1,*2			
2236.001*1,*2	2236.001*1,*2			
3154.001*1,*2	3154.001*1,*2			
3800.001*1,*2	3800.001*1,*2			
4072.001*1,*2	4072.001*1,*2			
4990.001*1,*2	4990.001*1,*2			
5908.001*1	5908.001*1			
5999.999*1 5900.001*3 6818.001*3 7299.999*3	5999.999*1 5900.001*3 6818.001*3 7299.999*3			

*1: Without MU887002A-007, TRX1/2, Port 1 to 12, or with MU887002A-007, TRX1/2, Port 1 to 4

*2: With MU887002A-007, TRX1/2, Port 5 to 12 Setting frequency < 5900 MHz

*3: With MU887002A-007, TRX1/2, Port 5 to 12 Setting frequency ≥ 5900 MHz

(4) a Procedure (without MU887002A-007, Port 1 to 12,
with MU887002A-007 Port 1 to 4,
or with MU887002A-007 Port 5 to 12 Setting frequency < 5900 MHz)

SG Level Accuracy is measured at TRX 1/2, Port 1 to Port 12.
When testing other than TRX 1 Port 1, read "TRX 1 Port 1" in the following procedure as the port to be tested.

TRX 1 Port 1

Absolute level measurement

1. To test TRX 1 Port 1, set-up the instruments as shown in Figure 7.3.2-1.
2. Initialize the settings of MU887002A.
   `*RST`
3. Set the input and output ports to TRX 1 Port 1.
   `:ROUTe:PORT:CONNect:DIRection PORT1,PORT1`
4. Set the MU887002A input level to –30 dBm.
   `:CONFigure:CELLular:MEASurement:RFSettings:LEVel -30`
5. Set the MU887002A 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 –22 dBm.
   `:SOURce:GPRF:GENerator:RFSettings:LEVel -22`
9. Output the VSG signal.
   `:SOURce:GPRF:GENerator:STATe ON`
10. Set the VSG output frequency to 400.001 MHz.
    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 400.001MHZ`
11. Measure the VSG output level at the power meter. This measured value is the absolute level reference of TRX 1 Port 1 at 400.001 MHz (C).
12. Set the VSG frequency in "Table 7.3.2-1 Frequency List (Power Meter)" and repeat steps 10 thru 11.

Relative level measurement

13. Set-up the instruments as shown in Figure 7.3.2-2.
14. Set the VSG output frequency to 400.001 MHz.

    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 400.001MHZ`
15. Set the VSG output level to –22 dBm.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -22`
16. Measure the VSG output level using the signal analyzer and make this measurement *A* dBm.
17. Set the VSG output level to –5 dBm and make this setting *P*.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -5`
18. Set the signal analyzer as shown in "Table 7.3.2-2 Settings of Signal Analyzer" according to the VSG output level.
19. Measure the VSG output level using the signal analyzer and make this measurement *B* dBm.
20. Calculate the level accuracy using the following equation.

    Level Accuracy = B – A + C – P
21. Decrease the VSG output level in 5 dB steps successively such as –10 dBm, –15 dBm, …, –115 dBm, –120 dBm and repeat steps 17 thru 20.

    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
22. Set the VSG frequency to ≤ 3800 MHz in "Table 7.3.2-3 Frequency List (Signal Analyzer)" and repeat steps 14 thru 21.
23. Set the VSG output frequency to 3800.001 MHz.

    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 3800.001MHZ`
24. Set the VSG output level to –22 dBm.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -22`
25. Measure the VSG output level using the signal analyzer and make this measurement *A* dBm.
26. Set the VSG output level to –8 dBm and make this setting *P*.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -8`
27. Set the signal analyzer as shown in "Table 7.3.2-2 Settings of Signal Analyzer" according to the VSG output level.
28. Measure the VSG output level using the signal analyzer and make this measurement *B* dBm.

29. Calculate the level accuracy using the following equation.

    Level Accuracy = B – A + C – P

30. Decrease the VSG output level in 5 dB steps successively such as –10 dBm, –15 dBm, …, –95 dBm, –100 dBm and repeat steps 28 thru 29.

    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

31. Set the VSG output frequency to each frequency in "Table 7.3.2-3 Frequency List (Signal Analyzer)" and repeat steps 23 thru 30.

(4) b Procedure (with MU887002A-007 Port 5 to 12, Setting frequency ≥ 5900 MHz)

SG Level Accuracy is measured at TRX 1/2, Port 5 to Port 12. When testing other than TRX 1 Port 5, read "TRX 1 Port 5" in the following procedure as the port to be tested.

TRX 1 Port 5

Absolute level measurement

1. To test TRX 1 Port 5, set-up the instruments as shown in Figure 7.3.2-1.
2. Initialize the settings of MU887002A.

   `*RST`
3. Set the input port to TRX 1 Port 6 and output port to TRX 1 Port 5. (When setting the input port to TRX 1 Port 5, set the output port to TRX 1 Port 6)

   `:ROUTe:PORT:CONNect:DIRection PORT6,PORT5`
4. Set the MU887002A input level to –30 dBm.

   `:CONFigure:CELLular:MEASurement:RFSettings:LEVel -30`
5. Set the MU887002A input frequency to 6000 MHz.

   `:CONFigure:CELLular:MEASurement:RFSettings:FREQuency 6000MHZ`
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 –22 dBm.

   `:SOURce:GPRF:GENerator:RFSettings:LEVel -22`
9. Output the VSG signal.

   `:SOURce:GPRF:GENerator:STATe ON`
10. Set the VSG output frequency to 5900.001 MHz.

    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 5900.001MHZ`
11. Measure the VSG output level at the power meter. This measured value is the absolute level reference of TRX 1 Port 5 at 5900.001 MHz (C).
12. Set the VSG frequency in "Table 7.3.2-1 Frequency List (Power Meter)" and repeat steps 10 thru 11.

Relative level measurement

13. Set-up the instruments as shown in Figure 7.3.2-2.
14. Set the VSG output frequency to 5900.001 MHz.

    `:SOURce:GPRF:GENerator:RFSettings:FREQuency 5900.001MHZ`
15. Set the VSG output level to –22 dBm.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -22`
16. Measure the VSG output level using the signal analyzer and make this measurement *A* dBm.
17. Set the VSG output level to –10 dBm and make this setting *P*.

    `:SOURce:GPRF:GENerator:RFSettings:LEVel -10`
18. Set the signal analyzer as shown in "Table 7.3.2-2 Settings of Signal Analyzer" according to the VSG output level.
19. Measure the VSG output level using the signal analyzer and make this measurement *B* dBm.
20. Calculate the level accuracy using the following equation.

    Level Accuracy = B – A + C – P
21. Decrease the VSG output level in 5 dB steps successively such as –15 dBm, …, –95 dBm, –100 dBm and repeat steps 17 thru 20.

    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
22. Set the VSG output frequency to each frequency in "Table 7.3.2-3 Frequency List (Signal Analyzer)" and repeat steps 14 thru 21.

7.3.3 SG Spurious

The following test related to the VSG spurious is performed.

(1) Test target specifications

Without MU887002A-007, TRX 1/2, Port 1 to Port 12.
With MU887002A-007, TRX 1/2, Port 1 to Port 4,
CW, –120 dBm ≤ Output Level ≤ –5 dBm, At 20 to 30°C

Harmonic Frequency f (MHz)	Specification (dBc)
800 ≤ f ≤ 6000	Harmonic Distortion < –25

With MU887002A-007, TRX 1/2, Port 5 to Port 12,
CW, –120 dBm ≤ Output Level ≤ –5 dBm, At 20 to 30°C

Harmonic Frequency f (MHz)	Specification (dBc)
800 ≤ f ≤ 7300	Harmonic Distortion < –25

(2) Measuring instrument

- Signal analyzer: MS269xA or MS2830A-041 (without MU887002A-007)
  MS2830A-044 (with MU887002A-007)

(3) Setup

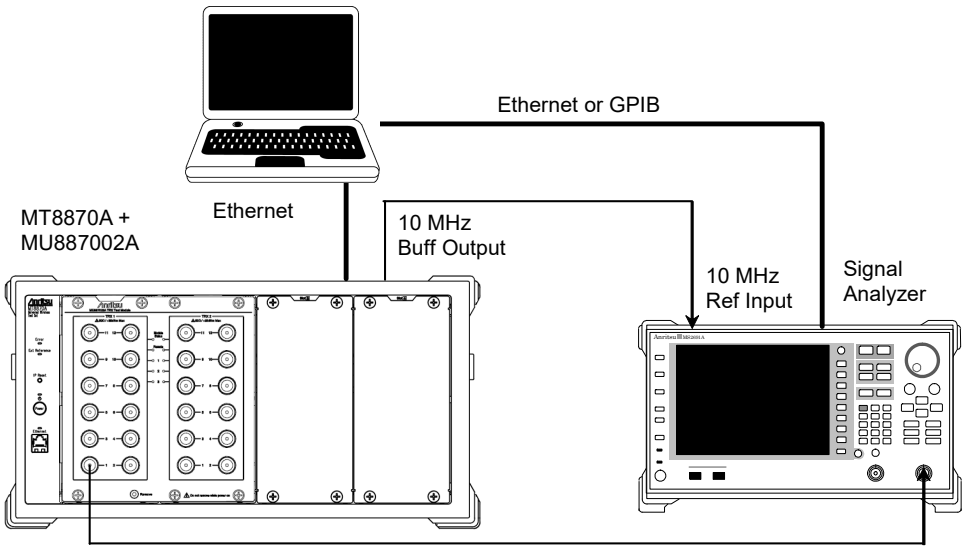


Figure 7.3.3-1 SG Spurious Measurement Setup

(4) Procedure

VSG Spurious is measured at TRX 1/2, Port 1 to Port 12.
When testing other than TRX 1 Port 1, read "TRX 1 Port 1" in the following procedure as the port to be tested.

TRX 1 Port 1

1. To test TRX 1 Port 1, set-up the instruments as shown in Figure 7.3.3-1.
2. Initialize the settings of MU887002A.

   `*RST`
3. Set the input and output ports to TRX 1 Port 1.

   `: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 –5 dBm.

   `:SOURce:GPRF:GENerator:RFSettings:LEVel -5`
7. Output the VSG signal.

   `:SOURce:GPRF:GENerator:STATe ON`
8. VSG output frequency to 400 MHz.

   `:SOURce:GPRF:GENerator:RFSettings:FREQuency 400MHZ`
9. Measure the 400 MHz signal level (Reference) at the signal analyzer. This value is set to *A* dBm.
10. Measure the 800 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.
    With MU887002A-007, you do not need to measure the signal levels of Port 5 to Port 12 at harmonics above 7.3 GHz.
11. Measure the 1200 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.
    With MU887002A-007, you do not need to measure the signal levels of Port 5 to Port 12 at harmonics above 7.3 GHz.
12. Calculate the harmonic distortion as following.

    Second harmonic distortion = B – A (dBc)

    Third harmonic distortion = C – A (dBc)
13. Change the VSG output frequency according to Table 7.3.3-1, and repeat steps 8 thru 12.

Table 7.3.3-1 Frequency point

No.	Without MU887002A-007, TRX 1/2 Test Port 1 to 12 With MU887002A-007, TRX 1/2 Test Port 1 to 4	With MU887002A-007, TRX 1/2 Test Port 5 to 12
	Output Frequency (MHz)	Output Frequency (MHz)
1	400	400
2	600	600
3	850	850
4	900	900
5	1000	1000
6	1500	1500
7	1800	1800
8	1950	1950
9	2200	2200
10	2450	2450
11	2650	2650
12	3000	3000
13	–	3300
14	–	3650

7.3.4 SA Level Accuracy

The following test related to the SA level accuracy is performed.

(1) Test target specifications

Without MU887002A-007, TRX 1/2, Port 1 to Port 12
With MU887002A-007, TRX 1/2, Port 1 to Port 4,
CW, After calibration, At 20 to 30°C,
Measurement bandwidth = 300 kHz, RBW = 100 kHz,
Measured signal with the same frequency as the set frequency,
Measured signal at the same level as the set level,

Frequency (MHz) / Level (dBm)	400 ≤ f ≤ 3800	3800 < f ≤ 6000
–30 ≤level ≤+35	±0.5 dB	±0.7 dB
–55 ≤level <–30	±0.7 dB	±0.9 dB
–65 ≤level <–55	±0.9 dB	±1.1 dB

With MU887002A-007, TRX 1/2, Port 5 to Port 12,
CW, After calibration, At 20 to 30°C,
Measurement bandwidth = 300 kHz, RBW = 100 kHz,
Measured signal with the same frequency as the set frequency,
Measured signal at the same level as the set level,

Frequency (MHz) / Level (dBm)	400 ≤ f ≤ 3800	3800 < f < 5900	5900 ≤ f ≤ 7300
–30 ≤level ≤+35	±0.5 dB	±0.7 dB	–
–30 ≤level ≤+30	–	–	±1.0 dB
–55 ≤level <–30	±0.7 dB	±0.9 dB	±1.2 dB
–65 ≤level <–55	±0.9 dB	±1.1 dB	±1.8 dB

(2) Measuring instruments

- Signal Generator: MG3710E (without MU887002A-007)
  MG3692C (with MU887002A-007)
- Power meter: ML2438A
- Power sensor: MA2442D
- Signal Analyzer: MS269xA or MS2830A-041 (without MU887002A-007)
  MS2830-044A (with MU887002A-007)
- N-type attenuator (6 dB): BW-N6W5+ 1 pce
- N-type attenuator (3 dB): BW-N3W5+ 1 pce
- Relay Adapter: 34NFNF50 1 pce
- Conversion Adapter: 34RKNF50 (with MU887002A-007)

(3) Setup

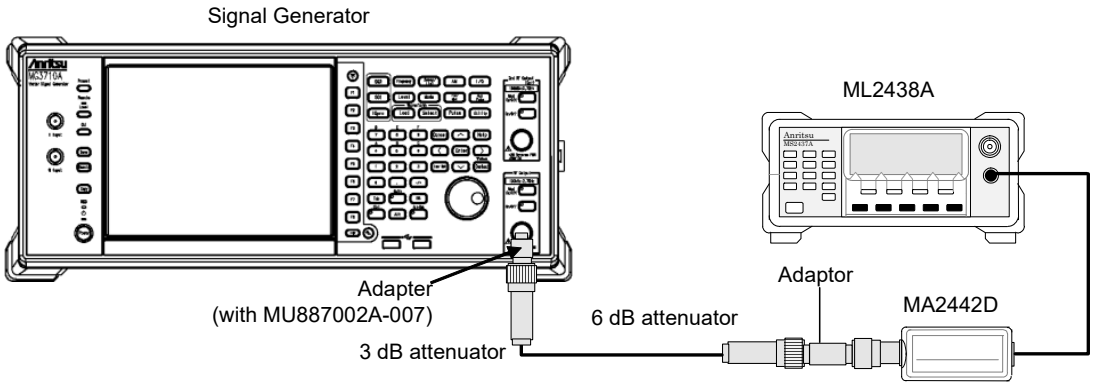


Figure 7.3.4-1 Output Level Setup (0, –25, –30 dBm)

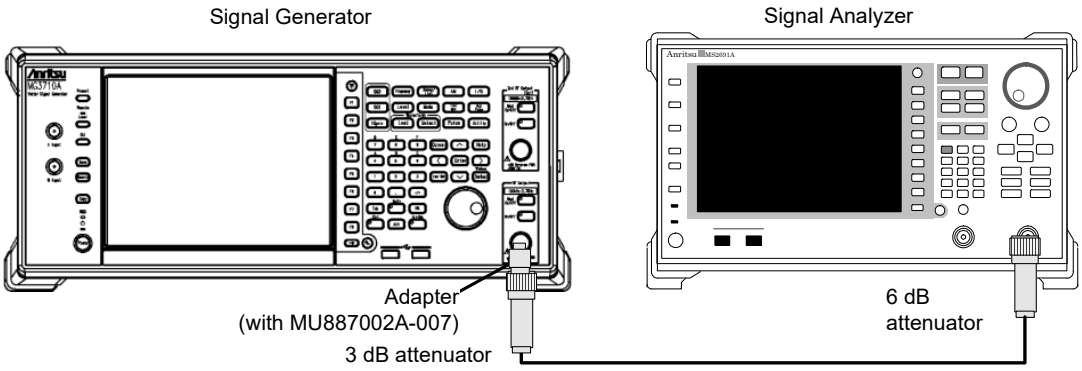


Figure 7.3.4-2 Output Level Setup (–35 dBm)

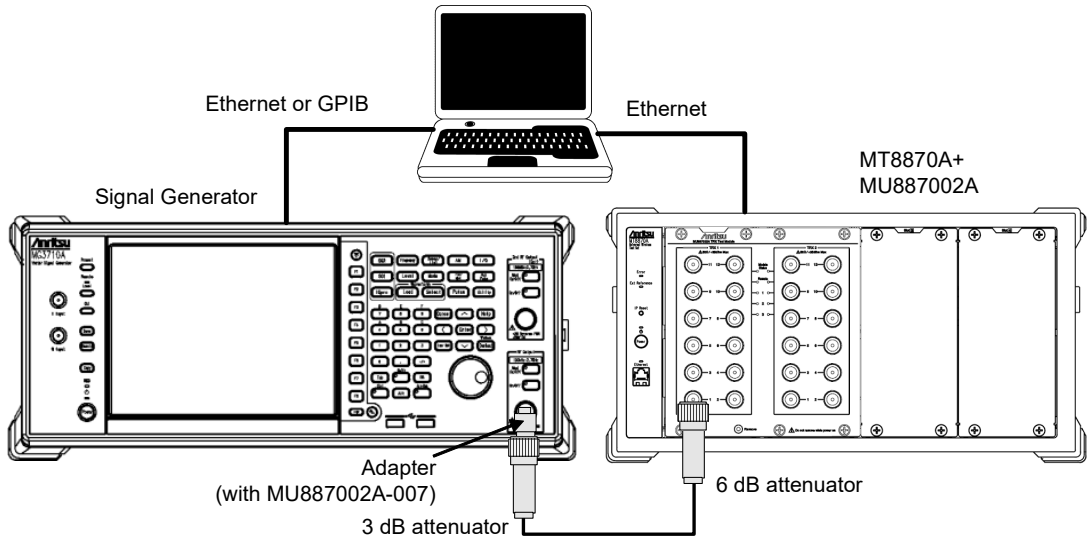


Figure 7.3.4-3 SA Level Accuracy Setup (TRX 1 Port 1)

Preparation for SA Level Accuracy Measurement

1. Calibrate the power sensor (zero point and sensitivity).
2. Connect the signal generator and the power meter using the cable with 3 dB and 6 dB attenuator.
3. Set the signal generator frequency to 1317.9 MHz.
4. Tune the signal generator output level so that the power meter displays the value nearest to 0 dBm.

   Make this setting value to the signal generator P0 and the measured value by the power meter C0.
5. Tune the signal generator output level so that the power meter displays the value nearest to –25 dBm.

   Make this setting value to the signal generator P25 and the measured value by the power meter C25.
6. Tune the signal generator output level so that the power meter displays the value nearest to –30 dBm.

   Make this setting value to the signal generator P30 and the measured value by the power meter C30.
7. Set the signal generator frequency to the value shown in "Table 7.3.4-1 Frequency List", and repeat steps 3 thru 6.
8. Connect the signal generator and the signal analyzer using the cable with 3 dB and 6 dB attenuator as shown in "Figure 7.3.4-2 Output Level Setup (–35 dBm)".
9. Set the signal generator frequency to 1317.9 MHz.
10. Set the signal generator 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 signal generator 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 signal generator frequency to the value shown in "Table 7.3.4-1 Frequency List", and repeat steps 9 thru 13.

Table 7.3.4-1 Frequency List

No.	MU887002A Input Frequency (MHz)	Signal Generator Setting Frequency (MHz)
1	1318*1,*2	1317.9*1,*2
2	2236*1,*2	2235.9*1,*2
3	3154*1,*2	3153.9*1,*2
4	3800*1,*2	3799.9*1,*2
5	4072*1,*2	4071.9*1,*2
6	4990*1,*2	4989.9*1,*2
7	6000*1	5999.9*1
8	5900*3	5900.1*3
9	6818*3	6817.9*3
10	7300*3	7299.9*3

*1: Without MU887002A-007, TRX1/2, Port 1 to 12
or with MU887002A-007, TRX1/2, Port 1 to 4

*2: With MU887002A-007, TRX1/2, Port 5 to 12,
Setting frequency < 5900 MHz

*3: With MU887002A-007, TRX1/2, Port 5 to 12
Setting frequency ≥ 5900 MHz

(4) a Procedure (without MU887002A-007, Port 1 to 12,
with MU887002A-007 Port 1 to 4,
or with MU887002A-007 Port 5 to 12 Setting frequency < 5900 MHz)

SA Level Accuracy is measured at TRX 1/2, Port 1 to Port 12.
When testing other than TRX 1 Port 1, read "TRX 1 Port 1" in the following procedure as the port to be tested.

TRX 1 Port 1

1. To test TRX 1 Port 1, set-up the instruments as shown in Figure 7.3.4-3.
2. Initialize the settings of MU887002A.
   ```
   *RST
   ```
3. Set the input and output ports to TRX 1 Port 1.
   ```
   :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 MU887002A 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 1318MHZ
   :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 signal generator output frequency to 1317.9 MHz (No.1 in "Table 7.3.4-1 Frequency List").

11. Set the signal generator output level to P0 obtained in "Preparation for SA Level Accuracy Measurement".
12. Set the input level of MU887002A to 0 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel 0
    ```
13. Measure the level at MU887002A 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 MU887002A input level in 5 dB steps successively such as +5 dBm, +10 dBm, …,+35 dBm and repeat steps 12 thru 14.
16. Set the signal generator output level to P30 obtained in "Preparation for SA Level Accuracy Measurement".
17. Set the input level of MU887002A to –30 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel -30
    ```
18. Measure the level at MU887002A 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 MU887002A input level in 5 dB steps successively such as –25 dBm, –20 dBm, …,–5 dBm and repeat steps 17 thru 19.
21. Set the signal generator output level to P25–10. P25 is obtained in "Preparation for SA Level Accuracy Measurement".
22. Set the input level of MU887002A to –35 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel -35
    ```
23. Measure the level at MU887002A 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 MU887002A 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.

(4) b Procedure (with MU887002A-007 Port 5 to 12, Setting frequency ≥ 5900 MHz)

SA Level Accuracy is measured at TRX 1/2, Port 5 to Port 12. When testing other than TRX 1 Port 5, read "TRX 1 Port 5" in the following procedure as the port to be tested.

TRX 1 Port 5

1. To test TRX 1 Port 5, set-up the instruments as shown in Figure 7.3.4-3.
2. Initialize the settings of MU887002A.

   ```
   *RST
   ```
3. Set the input port to TRX 1 Port 5 and output port to TRX 1 Port 6.

   (When setting the input port to TRX 1 Port 6, set the output port to TRX 1 Port 5)

   ```
   :ROUTe:PORT:CONNect:DIRection PORT5,PORT6
   ```
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 MU887002A 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.8 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:FREQuency 5900MHZ
   :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 signal generator output frequency to 5900.1 MHz (No.8 in "Table 7.3.4-1 Frequency List").
11. Set the signal generator output level to P0 obtained in "Preparation for SA Level Accuracy Measurement".
12. Set the input level of MU887002A to 0 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel 0
    ```
13. Measure the level at MU887002A 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 MU887002A input level in 5 dB steps successively such as +5 dBm, +10 dBm, …,+30 dBm and repeat steps 12 thru 14.
16. Set the signal generator output level to P30 obtained in "Preparation for SA Level Accuracy Measurement".
17. Set the input level of MU887002A to –30 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel -30
    ```
18. Measure the level at MU887002A 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 MU887002A input level in 5 dB steps successively such as –25 dBm, –20 dBm, …,–5 dBm and repeat steps 17 thru 19.
21. Set the signal generator output level to P25–10. P25 is obtained in "Preparation for SA Level Accuracy Measurement".
22. Set the input level of MU887002A to –35 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel -35
    ```
23. Measure the level at MU887002A 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 MU887002A input level in 5 dB steps successively such as –40 dBm, –45 dBm, …,–65 dBm and repeat steps 22 thru 24.
26. For 9 or later number in “Table 7.3.4-1 Frequency List”, repeat steps 9 thru 25.

7.3.5 SA Linearity

The following test related to the SA Linearity is performed.

(1) Test target specifications

TRX 1/2, Port 1 to Port 12
CW, At 20 to 30°C,
Measurement bandwidth = 300 kHz, RBW = 100 kHz,

Input Level	Input level Attenuation	Specifications
≥ –55 dBm	0 to –40 dB	±0.2 dB
≥ –65 dBm	0 to –40 dB	±0.4 dB

(2) Measuring instruments

- Signal Generator: MG3710E (without MU887002A-007)
  MG3692C (with MU887002A-007)
- Power meter: ML2438A
- Power sensor: MA2442D
- Signal Analyzer: MS269xA or MS2830A-041
  (without MU887002A-007)
  MS2830-044A (with MU887002A-007)
- N-type attenuator (6 dB): BW-N6W5+ 1 pce
- N-type attenuator (3 dB): BW-N3W5+ 1 pce
- Relay Adapter: 34NFNF50 1 pce
- Conversion Adapter: 34RKNF50 (with MU887002A-007)

(3) Setup

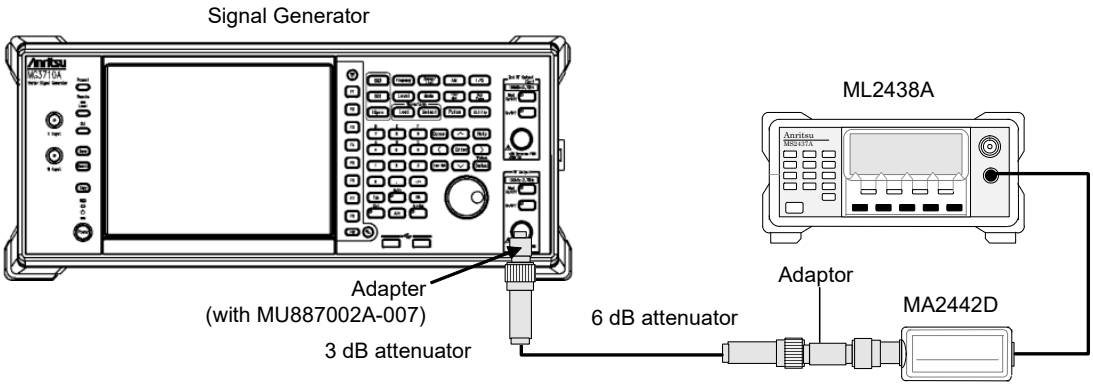


Figure 7.3.5-1 Output Level Setup (0, –25 dBm)

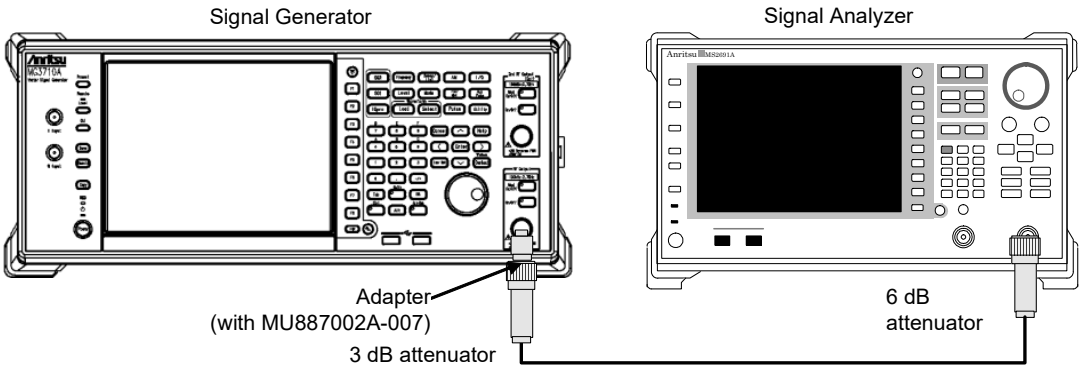


Figure 7.3.5-2 Output Level Setup (0 to –65 dBm)

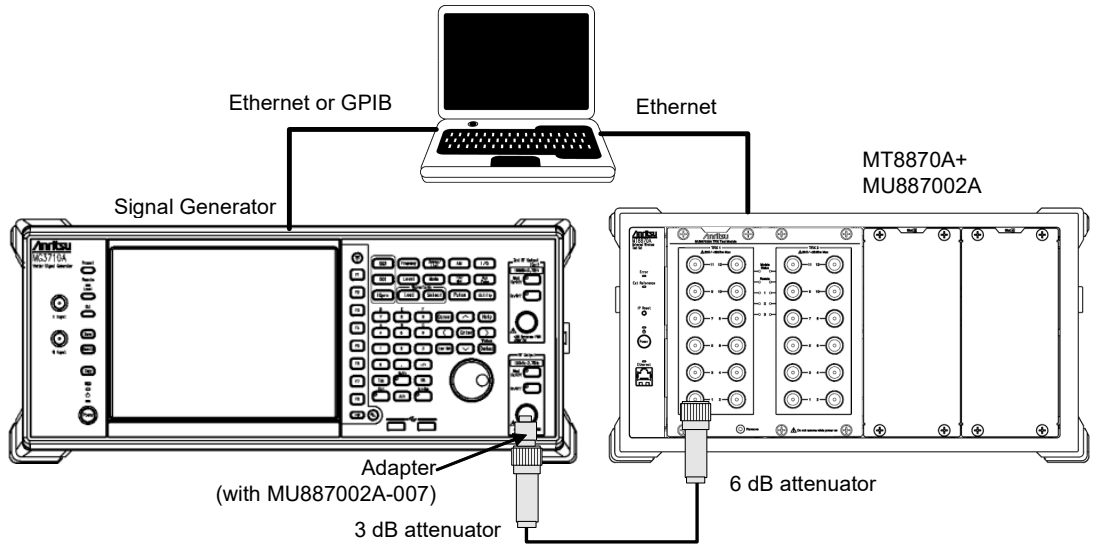


Figure 7.3.5-3 SA Linearity Setup (TRX 1 Port 1)

Preparation for SA Linearity Measurement

1. Calibrate the power sensor (zero point and sensitivity).
2. Connect the signal generator and the power meter using the cable with 3 dB and 6 dB attenuator.
3. Set the signal generator frequency to 1317.9 MHz.
4. Tune the signal generator output level so that the power meter displays the value nearest to 0 dBm.

   Make this setting value to the signal generator P0 and the measured value by the power meter C0.
5. Tune the signal generator output level so that the power meter displays the value nearest to –25 dBm.

   Make this setting value to the signal generator P25 and the measured value by the power meter C25.
6. Set the signal generator frequency to the value shown in "Table 7.3.5-1 Frequency List", and repeat steps 3 thru 5.
7. Connect the signal generator and the signal analyzer using the cable with 3 dB and 6 dB attenuator as shown in "Figure 7.3.5-2 Output Level Setup (0 to –65 dBm)".
8. Set the signal generator frequency to 1317.9 MHz.
9. Set the signal generator 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 signal generator 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 signal generator 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 signal generator 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 signal generator 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 signal generator output level to the value (P25) obtained in step 5, for outputting the –25 dBm level, and repeat steps 8 thru 19.
20. Set the signal generator frequency to the value shown in "Table 7.3.5-1 Frequency List", repeat steps 8 thru 19.

Table 7.3.5-1 Frequency List

No.	MU887002A Input Frequency (MHz)	Signal Generator Setting Frequency (MHz)
1	1318*1,*2	1317.9*1,*2
2	2236*1,*2	2235.9*1,*2
3	3154*1,*2	3153.9*1,*2
4	4072*1,*2	4071.9*1,*2
5	4990*1,*2	4989.9*1,*2
6	6000*1	5999.9*1
7	5900*3	5900.1*3
8	6818*3	6817.9*3
9	7300*3	7299.9*3

*1: Without MU887002A-007, TRX1/2, Port 1 to 12
or with MU887002A-007, TRX1/2, Port 1 to 4

*2: With MU887002A-007, TRX1/2, Port 5 to 12,
Setting frequency < 5900 MHz

*3: With MU887002A-007, TRX1/2, Port 5 to 12
Setting frequency ≥ 5900 MHz

(4) a Procedure (without MU887002A-007, Port 1 to 12,
with MU887002A-007 Port 1 to 4,
or with MU887002A-007 Port 5 to 12 Setting frequency < 5900 MHz)

SA Linearity is measured at TRX 1/2, Port 1 to Port 12.
When testing other than TRX 1 Port 1, read "TRX 1 Port 1" in the following procedure as the port to be tested.

TRX 1 Port 1

1. To test TRX 1 Port 1, set-up the instruments as shown in Figure 7.3.5-3.
2. Initialize the settings of MU887002A.
   ```
   *RST
   ```
3. Set the input and output ports to TRX 1 Port 1.
   ```
   :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 MU887002A 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 1318MHZ
   :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 signal generator output frequency to 1317.9 MHz (No.1 in "Table 7.3.5-1 Frequency List").

11. Set the signal generator output level to P0 obtained in "Preparation for SA Linearity Measurement".
12. Set the input level of MU887002A to 0 dBm.

    ```
    :CONFigure:CELLular:MEASurement:RFSettings:LEVel 0
    ```
13. Measure the level at MU887002A and make this measurement *A* dBm.

    ```
    :INITiate:CELLular:MEASurement:SINGle
    *WAI
    :FETCh:CELLular:COMMon:SPMonitor:POWer
    ```
14. Set the signal generator output level to the value which is 10 dB lower than the setting value in the step 11.
15. Measure the level at MU887002A and make this measurement *B1* dBm.
16. Calculate the level accuracy using the following equation.

    The following S, S1, S2, S3, S4 are the values obtained in "Preparation for SA Linearity Measurement".

    Linearity = (B1 – A) – (S1 – S)
17. Set the signal generator output level to the value which is 20 dB lower than the setting value in the step 11.
18. Measure the level at MU887002A and make this measurement *B2* dBm.
19. Calculate the level accuracy using the following equation.

    Linearity = (B2 – A) – (S2 – S)
20. Set the signal generator output level to the value which is 30 dB lower than the setting value in the step 11.
21. Measure the level at MU887002A and make this measurement *B3* dBm.
22. Calculate the level accuracy using the following equation.

    Linearity = (B3 – A) – (S3 – S)
23. Set the signal generator output level to the value which is 40 dB lower than the setting value in the step 11.
24. Measure the level at MU887002A and make this measurement *B4* dBm.
25. Calculate the level accuracy using the following equation.

    Linearity = (B4 – A) – (S4 – S)
26. Set the signal generator 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 each frequency in "Table 7.3.5-1 Frequency List", repeat steps 9 thru 26.

(4) b Procedure (with MU887002A-007 Port 5 to 12, Setting frequency ≥ 5900 MHz)

SA Linearity is measured at TRX 1/2, Port 5 to Port 12.
When testing other than TRX 1 Port 5, read "TRX 1 Port 5" in the following procedure as the port to be tested.

TRX 1 Port 5

1. To test TRX 1 Port 5, set-up the instruments as shown in Figure 7.3.5-3.
2. Initialize the settings of MU887002A.
   ```
   *RST
   ```
3. Set the input port to TRX 1 Port 5 and output port to TRX 1 Port 6. (When setting the input port to TRX 1 Port 6, set the output port to TRX 1 Port 5)
   ```
   :ROUTe:PORT:CONNect:DIRection PORT5,PORT6
   ```
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 MU887002A 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.7 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 5900MHZ
   :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 signal generator output frequency to 5900.1 MHz (No.7 in "Table 7.3.5-1 Frequency List").

11. Set the signal generator output level to P0 obtained in "Preparation for SA Linearity Measurement".
12. Set the input level of MU887002A to 0 dBm.

    `:CONFigure:CELLular:MEASurement:RFSettings:LEVel 0`
13. Measure the level at MU887002A and make this measurement *A* dBm.

    `:INITiate:CELLular:MEASurement:SINGle`
    `*WAI`
    `:FETCh:CELLular:COMMon:SPMonitor:POWer`
14. Set the signal generator output level to the value which is 10 dB lower than the setting value in the step 11.
15. Measure the level at MU887002A and make this measurement *B1* dBm.
16. Calculate the level accuracy using the following equation. The following S, S1, S2, S3, S4 are the values obtained in "Preparation for SA Linearity Measurement".

    Linearity = (B1 – A) – (S1 – S)
17. Set the signal generator output level to the value which is 20 dB lower than the setting value in the step 11.
18. Measure the level at MU887002A and make this measurement *B2* dBm.
19. Calculate the level accuracy using the following equation.

    Linearity = (B2 – A) – (S2 – S)
20. Set the signal generator output level to the value which is 30 dB lower than the setting value in the step 11.
21. Measure the level at MU887002A and make this measurement *B3* dBm.
22. Calculate the level accuracy using the following equation.

    Linearity = (B3 – A) – (S3 – S)
23. Set the signal generator output level to the value which is 40 dB lower than the setting value in the step 11.
24. Measure the level at MU887002A and make this measurement *B4* dBm.
25. Calculate the level accuracy using the following equation.

    Linearity = (B4 – A) – (S4 – S)
26. Set the signal generator 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 each frequency in "Table 7.3.5-1 Frequency List", repeat steps 9 thru 26.

7.3.6 Pathloss Measurement Accuracy (MX887092A)

The following test related to the measurement accuracy of MX887092A Pathloss Measurement Function is performed.

(1) Test target specifications

TRX 1/2, Port 1 to 12, 20 to 30°C, with moving average of 11 intervals,
Path length 1.5 m, a coaxial cable with a relative permittivity of 2.03

When Port 11 of TRX 1 or TRX 2 satisfy the specifications, the specifications of other ports of the same TRX are guaranteed by design.

Frequency f (MHz)	Specifications
400 ≤ f ≤ 2700	±0.5 dB
3300 ≤ f ≤ 6000	±0.9 dB

(2) Measuring instruments

- Network Analyzer: S820E
- Calibration kit
- Coaxial cable for Pathloss measurement: 1 pce
- OPEN/SHORT for Pathloss measurement: 22N50 1 pce
- Relay Adapter: 34NFNF50 1 pce

(3) Setup

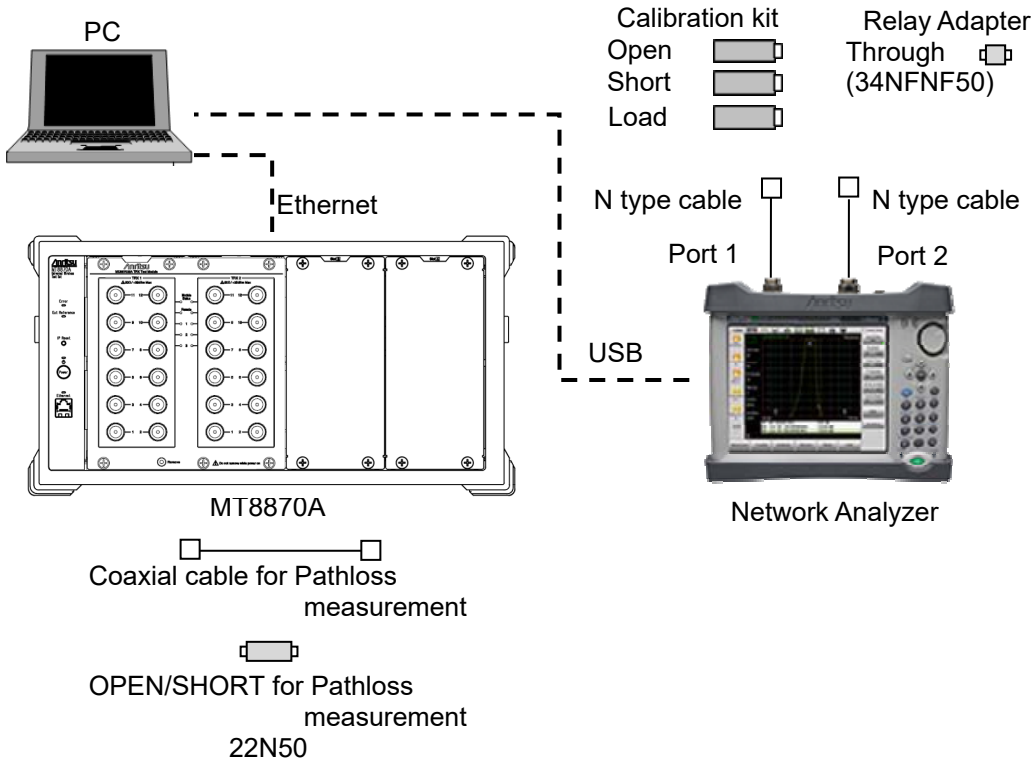


Figure 7.3.6-1 Pathloss Measurement Accuracy Setup

(4) Procedure

Pathloss Measurement Accuracy is measured at Port 11 of TRX 1/2. When Port 11 of TRX 1 or TRX 2 satisfy the specifications, the specifications of other ports of the same TRX are guaranteed by design.

1. Connect an N-type cable to the Network Analyzer.
2. Set the Network Analyzer as follows:

   Start Frequency: 100 MHz
   Stop Frequency: 8000 MHz
   Smoothing: 5%

3. Calibrate the Full 2 Port by attaching a calibration kit (Open, Short, and Load terminators) and the relay (Through) adapter to the end of the N-type cable.
4. Measure S_{21} of the coaxial cable for Path loss measurement.
5. Connect the coaxial cable for Pathloss measurement to Port 11 of TRX 1.
6. Set the number of intervals for moving average used for pathloss calculation to 11, and measure the pathloss at the frequencies in "Table 7.3.6-1 Frequency List" according to the procedure in Appendix E.
7. Calculate the Pathloss measurement accuracy from the pathloss value measured in step 6 and S_{21} measured in step 4.

   Pathloss measurement accuracy = (Pathloss value measured in step 6) + (S_{21} measured in step 4)
8. Connect the coaxial cable for Pathloss measurement to Port 11 of TRX 2, and repeat steps 6 to 7.

Table 7.3.6-1 Frequency List

No.	Frequency (MHz)
1	400
2	1852.5
3	2700
4	3300
5	4920
6	6000

7.4 Servicing

If any unit is found to be broken or does not operate as described in the specifications, please 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.

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 8-2
8.1.1 Functions 8-2
8.1.2 Operating environment 8-8
8.2 Installing 8-9
8.2.1 .NET Framework 8-9
8.2.2 NI-VISA 8-11
8.2.3 Utility Tool 8-13
8.3 Operating 8-16
8.3.1 Starting and quitting 8-16
8.3.2 Utility Tool screen elements 8-17
8.3.3 Searching for devices 8-19
8.3.4 Viewing information on devices 8-20
8.3.5 Network settings 8-23
8.3.6 Setting modules 8-25
8.3.7 Log Operation 8-27
8.3.8 Upgrading 8-28
8.3.9 Transferring files 8-31
8.3.10 Batch upgrading 8-33
8.3.11 Registering license 8-35
8.3.12 Transferring waveform files 8-37
8.3.13 Transferring files directly 8-39

8.1 Overview

8.1.1 Functions

The MX887900A Utility Tool has the following functions:

Searching for MT8870A

Searches for MT8870A connected via Ethernet and displays search results.

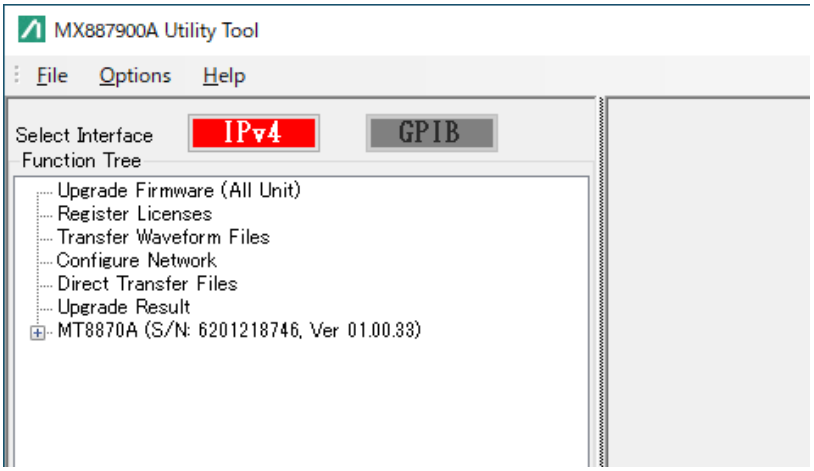


Figure 8.1.1-1 Search Results Display

Configuring network

Specifies network settings, such as module IP address and subnet mask. Module 1/2/3/4 can support two MU887002As.

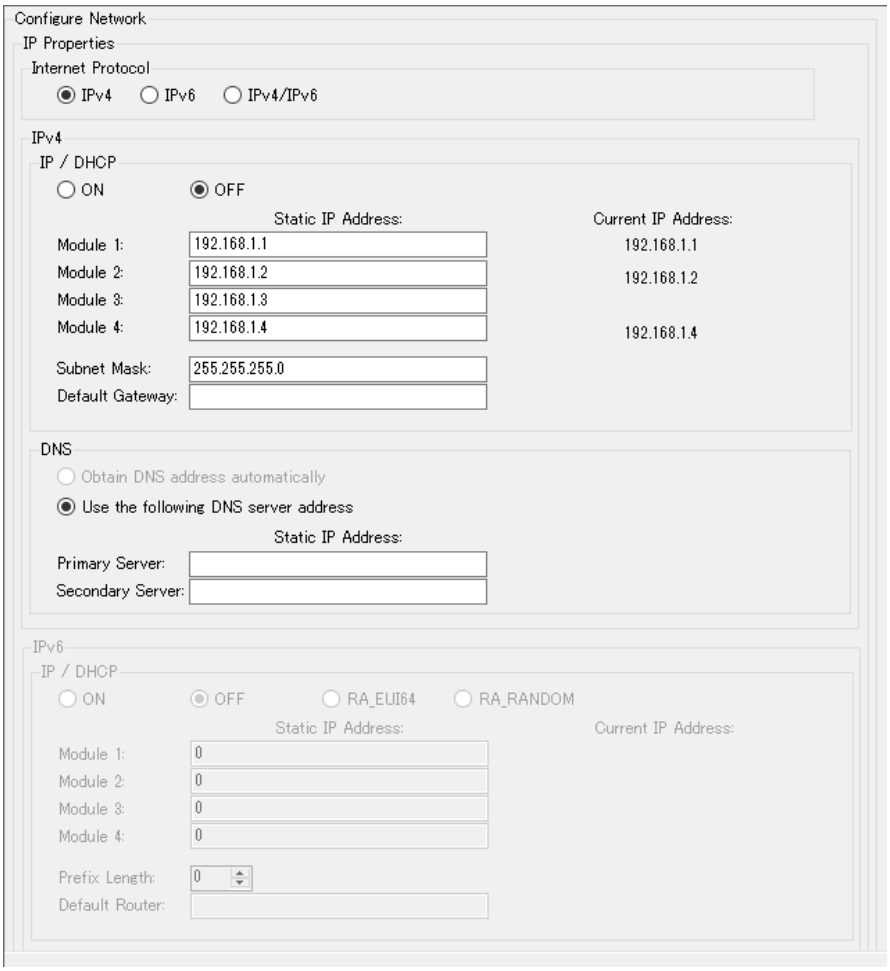


Figure 8.1.1-2 Network Settings Window

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)

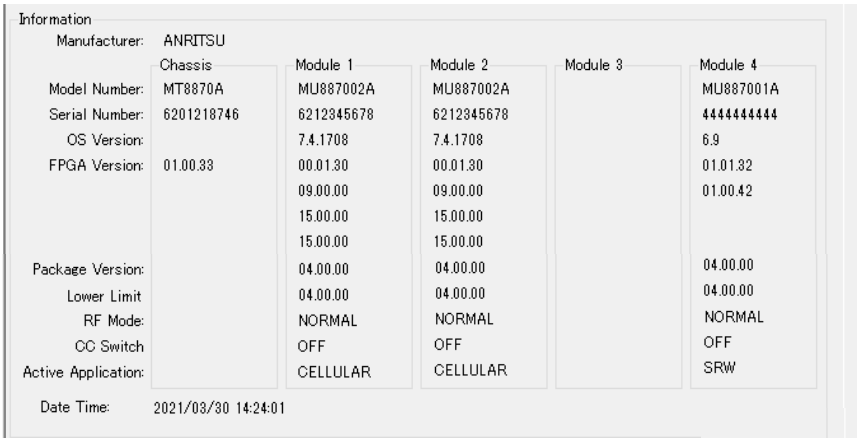


Figure 8.1.1-4 Maintenance Information Window (Hardware)

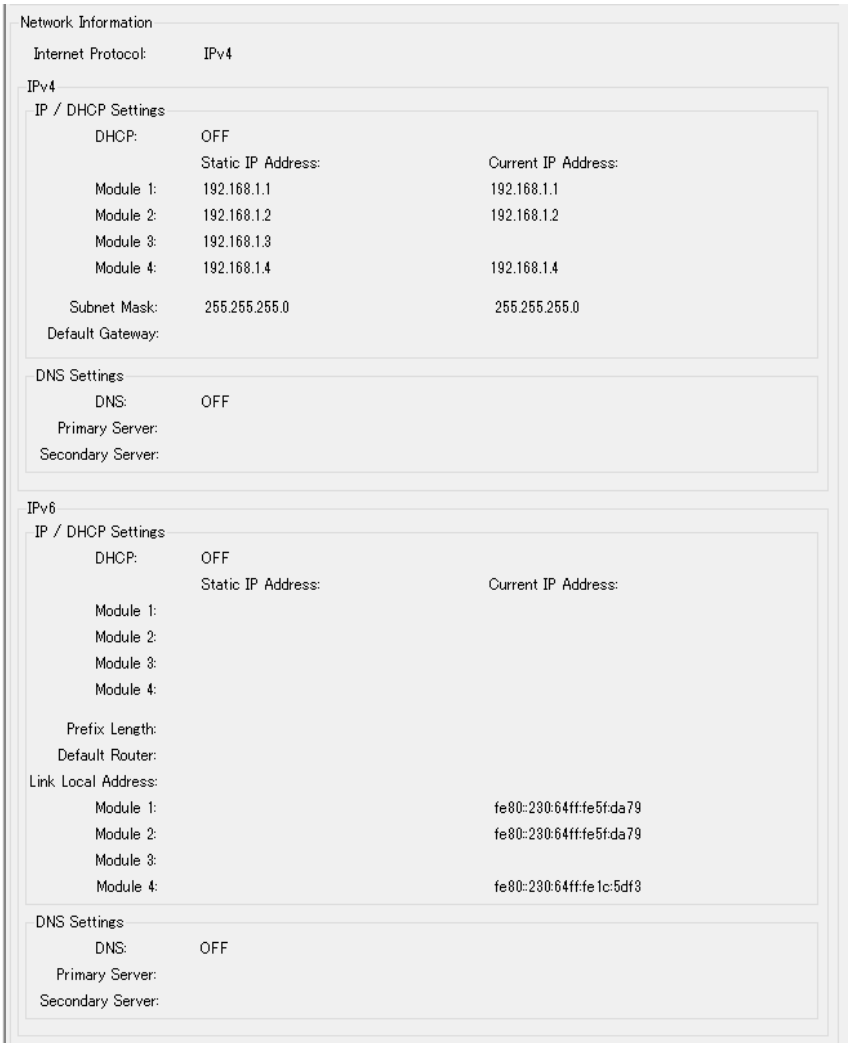


Figure 8.1.1-5 Maintenance Information Window (Communication Settings 1)

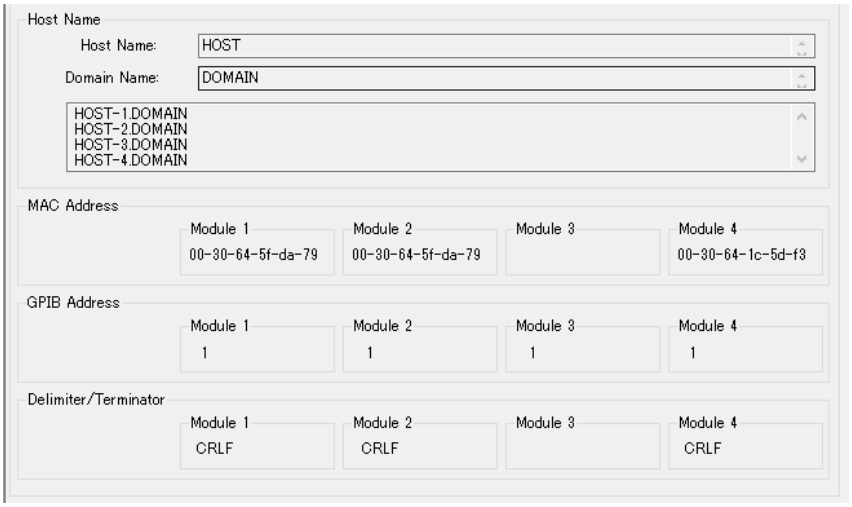


Figure 8.1.1-6 Maintenance Information Window (Communication Settings 2)

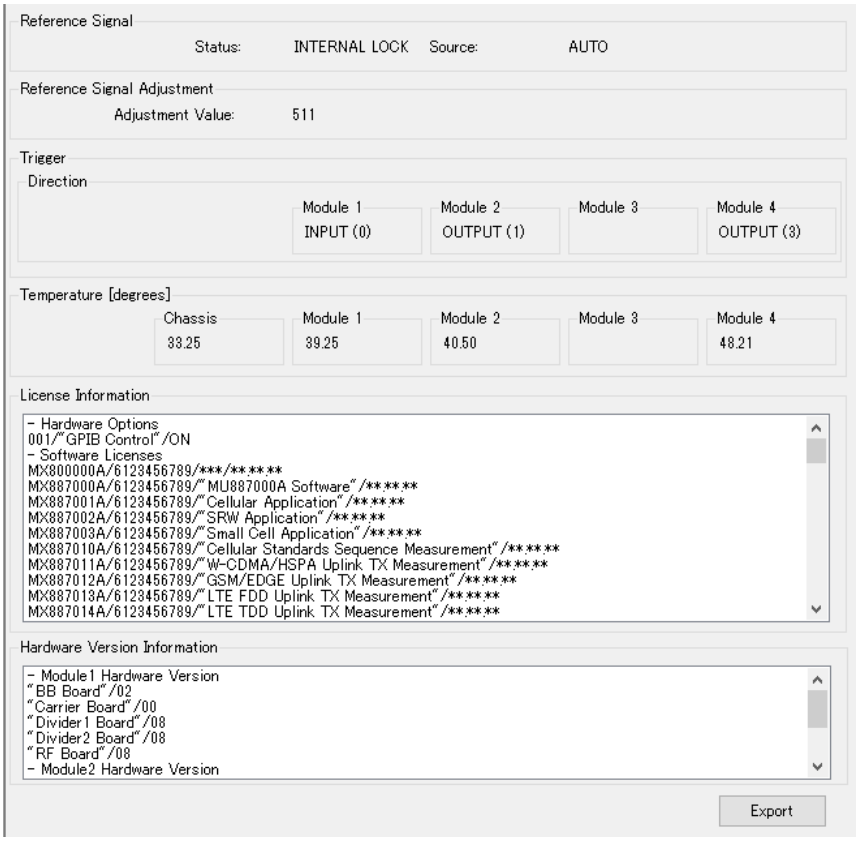


Figure 8.1.1-7 Maintenance Information Window (Trigger Settings, License, and Hardware Version)

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 must be Ver 2.0.0 or later.
The Utility Tool operates in the following environment:

Table 8.1.2-1 Utility Tool Operating Environment

Item	Specification
OS	Windows 7 Enterprise Service Pack 1 Japanese/English versions Windows 10 Japanese/English versions
Display	1024 × 768 or better
Memory	1 GB or more
HDD Free space	200 MB or more
VISA	NI-VISA*
Interface	Ethernet

*: For the version compatibility of .NET Framework and NI-VISA, refer to Table 8.1.2-2.

Table 8.1.2-2 Compatibility Between .NET Framework and NI-VISA

NET Framework	NI-VISA 5.0.3 to 5.2	NI-VISA 5.3 or later
.NET Framework 4.0	✓	✓
.NET Framework 4.5	–	✓

✓: Compatible, –: Incompatible

Table 8.1.2-3 NI-VISA Version

Utility Tool Version	NI-VISA Version
Ver 2.0.0 or later	Version 5.03 or later, Version 5.4.1 or earlier, Version 14.0 or later

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 8.1.2-4 Compatible Versions

Utility Tool Version	Module Version
Ver 2.0.0 or later	Version 3.0.0 or later

8.2 Installing

8.2.1 .NET Framework

If you are using Windows 7, you need to install NET Framework 4.5.

8.2.1.1 .NET Framework 4.5

Installing .NET Framework 4.5

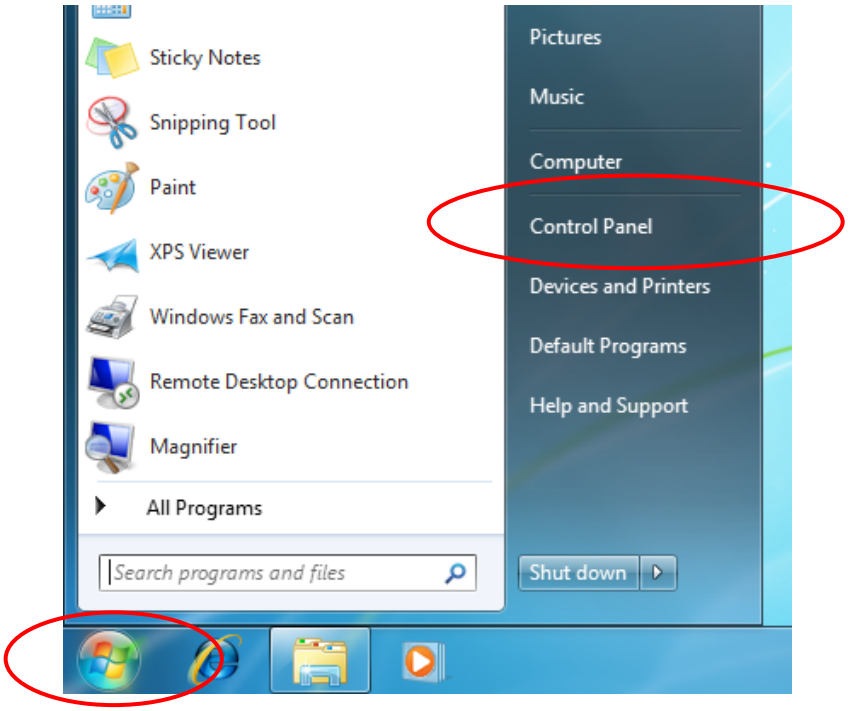
1. Download .NET Framework 4.5 from the Microsoft web site.
   https://www.microsoft.com/en-us/download/details.aspx?id=30653
2. Double-click the downloaded execution file. Start installation following the on-screen instructions.

Note:

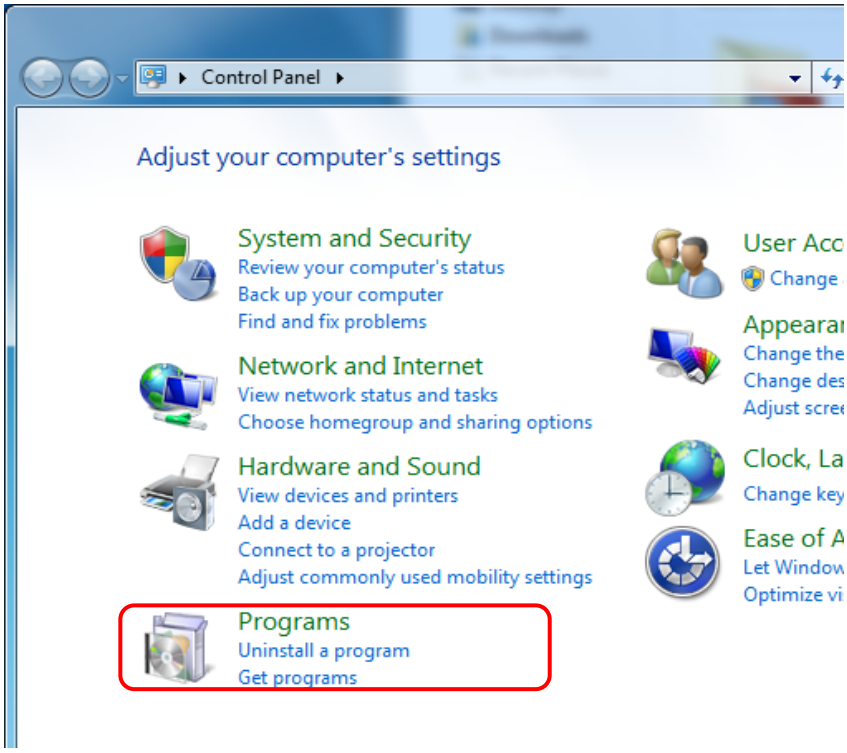
.NET Framework 4.0 is removed when .NET Framework 4.5 is installed.

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.



8.2.2 NI-VISA

When controlling this measuring equipment remotely using the Ethernet port, a VISA driver must be installed in the PC controller.
We recommend using NI-VISA™ from National Instruments™ (NI hereafter) as the VISA driver.
A license is generally required to use NI-VISA™
Be sure to comply with the NI license agreement for the usage and license scope.

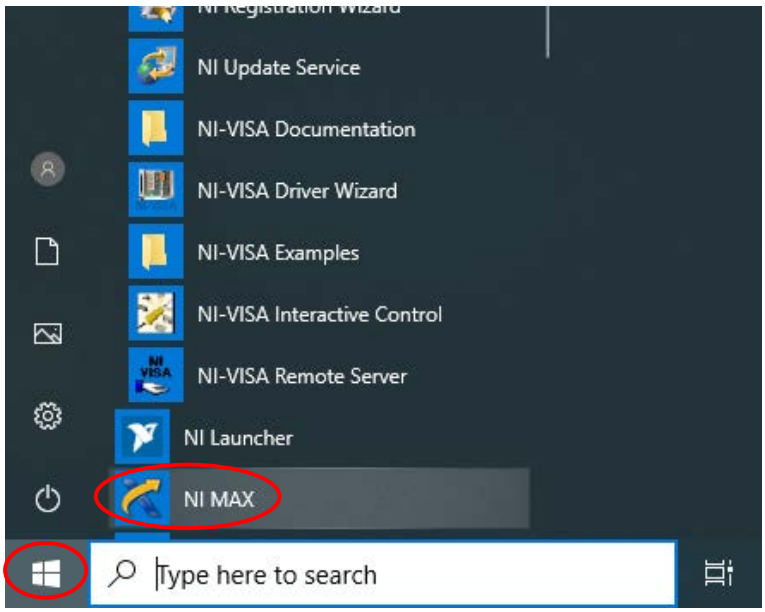
- Although the NI-VISA™ driver itself can be downloaded free-of-charge from the web, an implementation license is required for legal reasons when some requirements are not met. (Check the NI web page for the detailed requirements.)
- If these requirements are not met, permission is not granted to use NI hardware and software and an NI implementation license must be purchased.

Installing NI-VISA

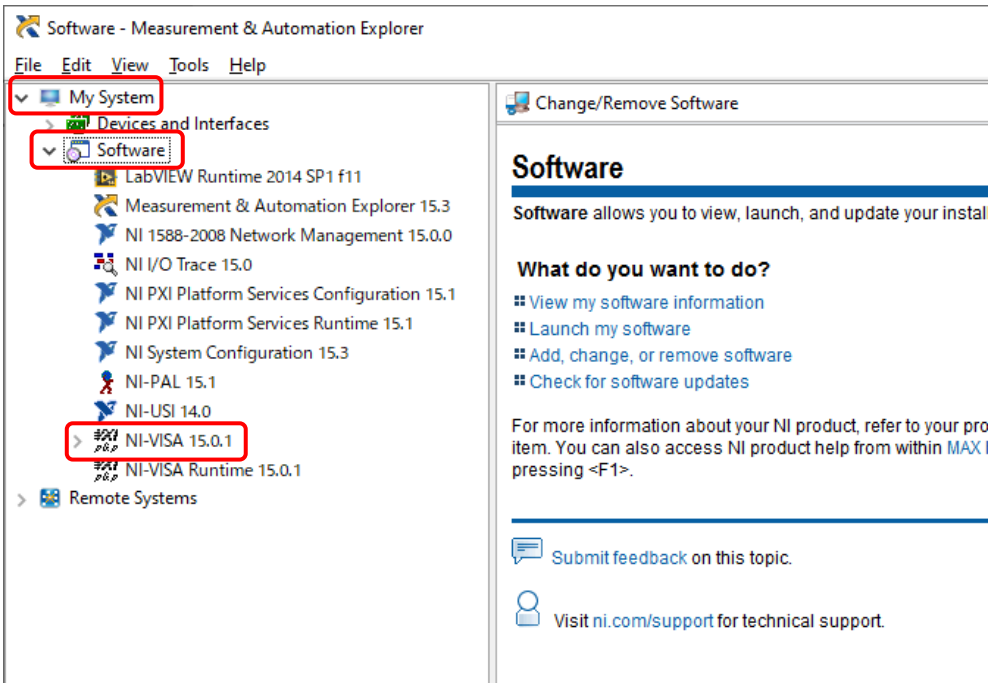
1. Download NI-VISA from the National Instruments web site.
2. Double-click the downloaded file and start the install following the on-screen instructions.
3. Restart the PC when the NI-VISA is successfully installed.

Confirming the software version of NI-VISA

1. Click **Start**, and then **NI MAX** in index N.



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.

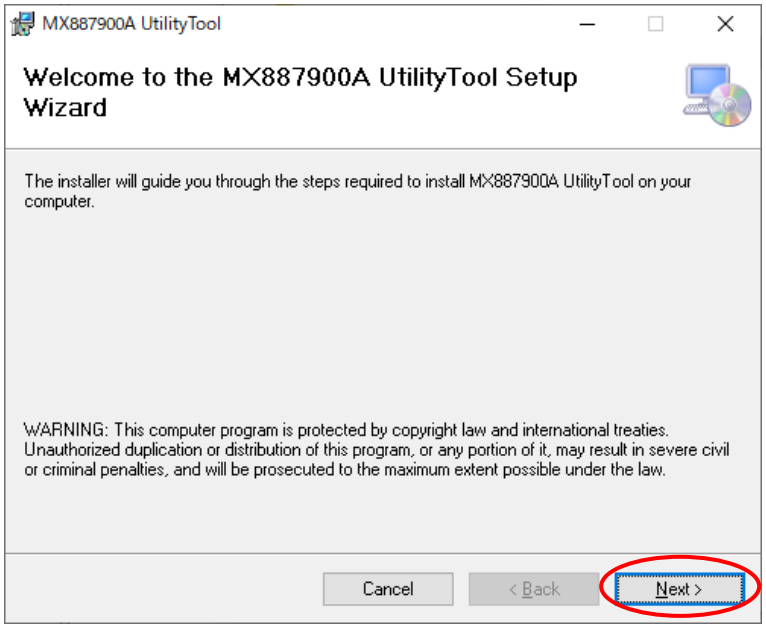
8.2.3 Utility Tool

Installing Utility Tool

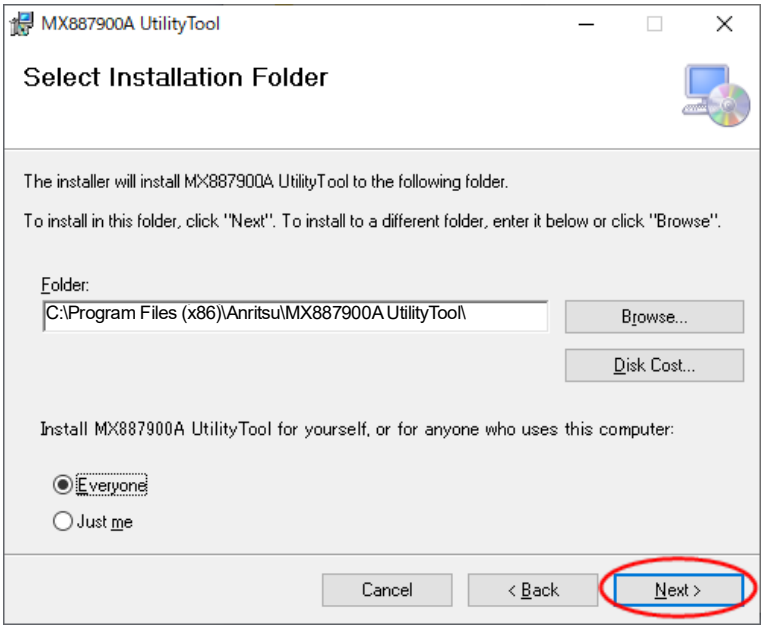
1. Double-click the setup program “setup.exe” on the storage media supplied with the MT8870A.

   The setup program is stored in \Installer\MT8870A_Utility Tool.

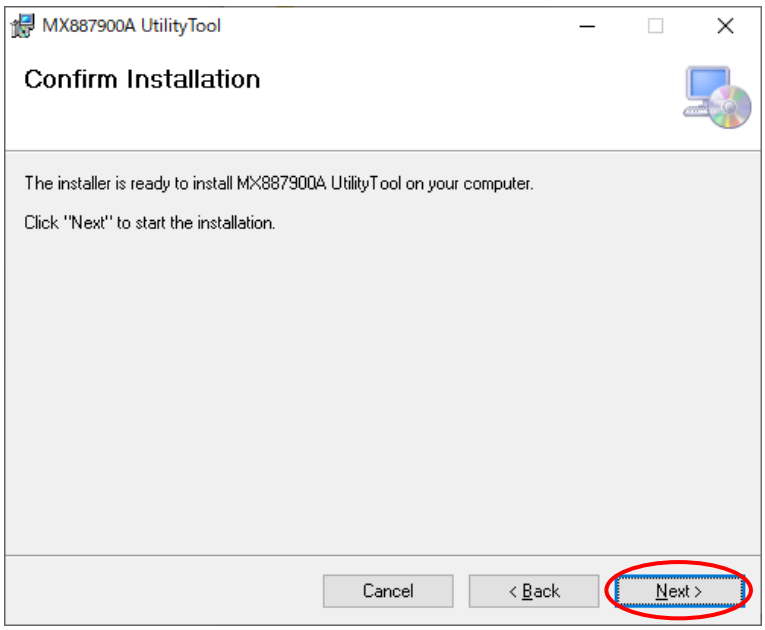
2. Click **Next** in the wizard.



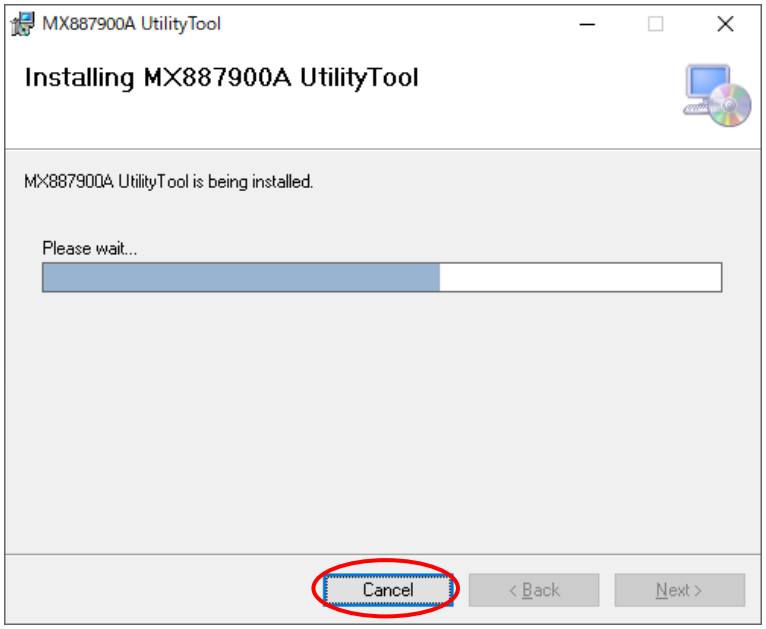
3. Specify the full path to the installation folder, and then click **Next**.



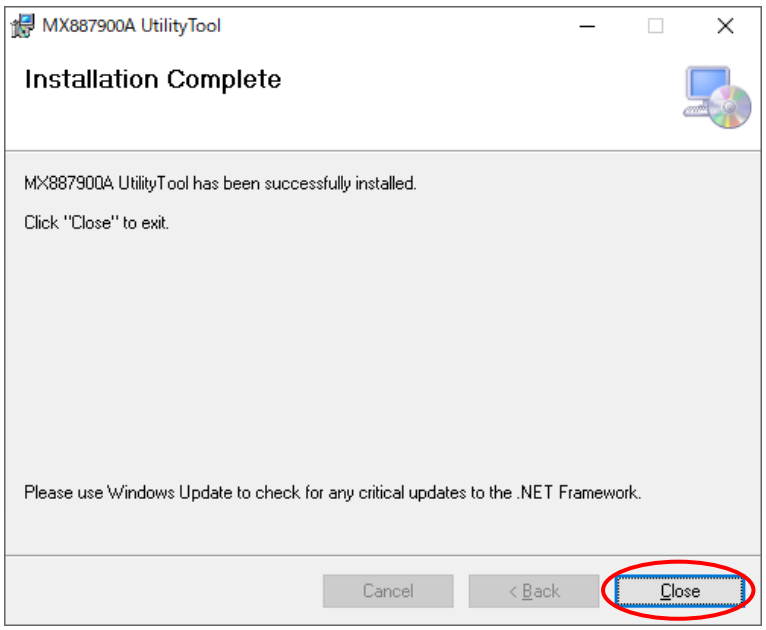
4. Click **Next** when the following message appears.



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.

8.3 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**) used to communicate with the MT8870A. The module search message is displayed.
3. The tree view display is updated when the module is detected.

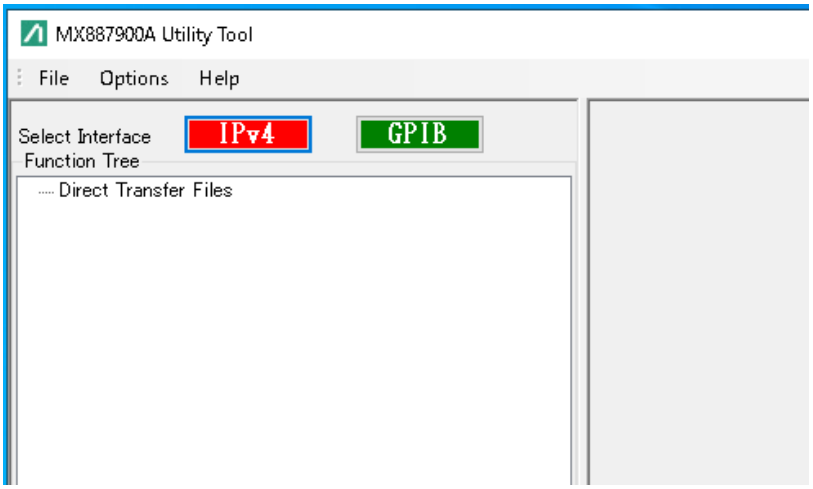


Figure 8.3.1-1 Utility Tool Initial Screen

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.

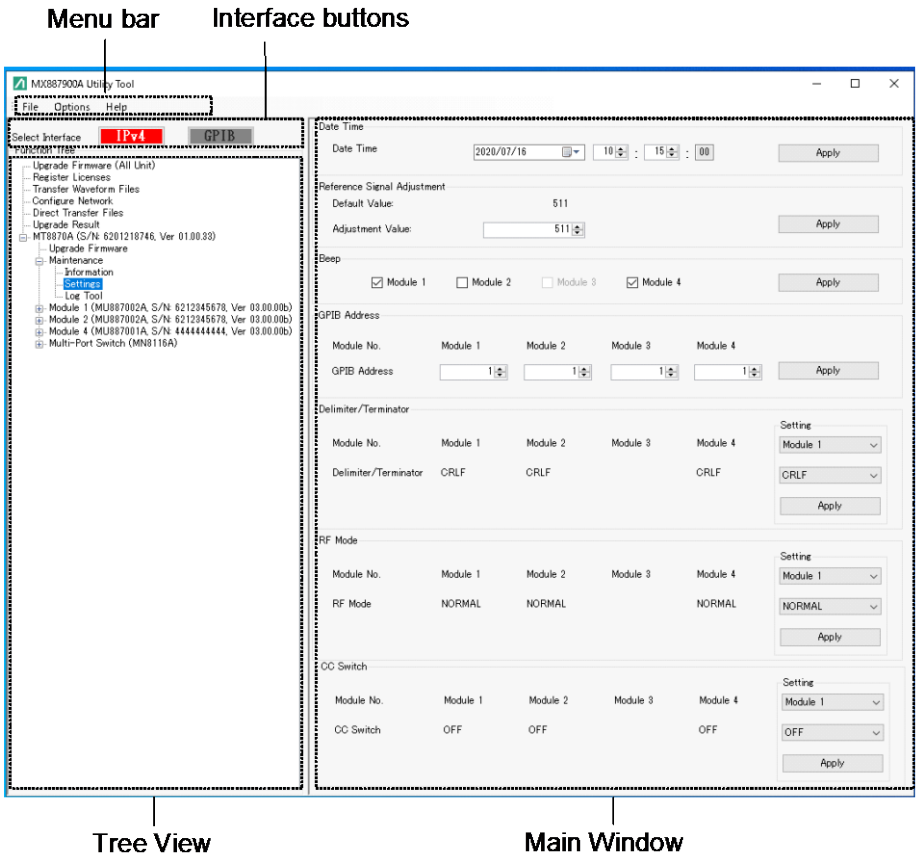


Figure 8.3.2-1 Utility Tool Screen Elements

Table 8.3.2-1 Screen Description

Name	Description	
Menu bar	File Exit: Quits Utility Tool.	
	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.	
	Help: Displays version information.	
Interface buttons	IPv4	Selects interface for communications with MT8870A. GPIB cannot be selected for MU887002A.
	GPIB	
Tree view	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.	
Main window	Displays the information and status of MT8870A/module and contains command buttons.	

8.3.3 Searching for devices

This section describes how to search for the MT8870A.

1. Click the **IPv4** 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.

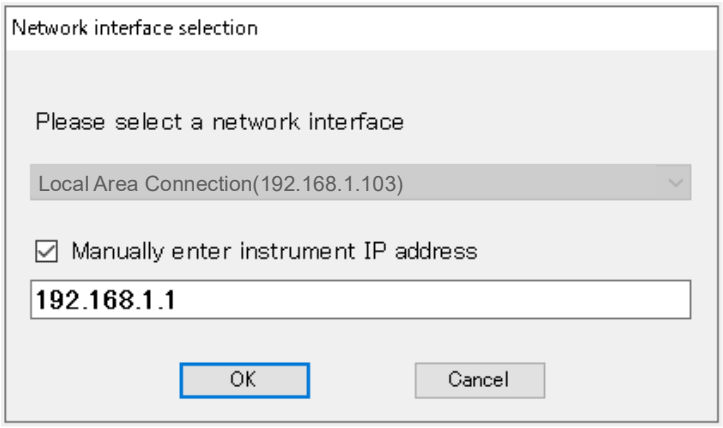


Figure 8.3.3-1 Network Interface Selection Dialog Box

2. When the MT8870A is detected, the model and serial number are displayed in the tree view.

   For the IPv4 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

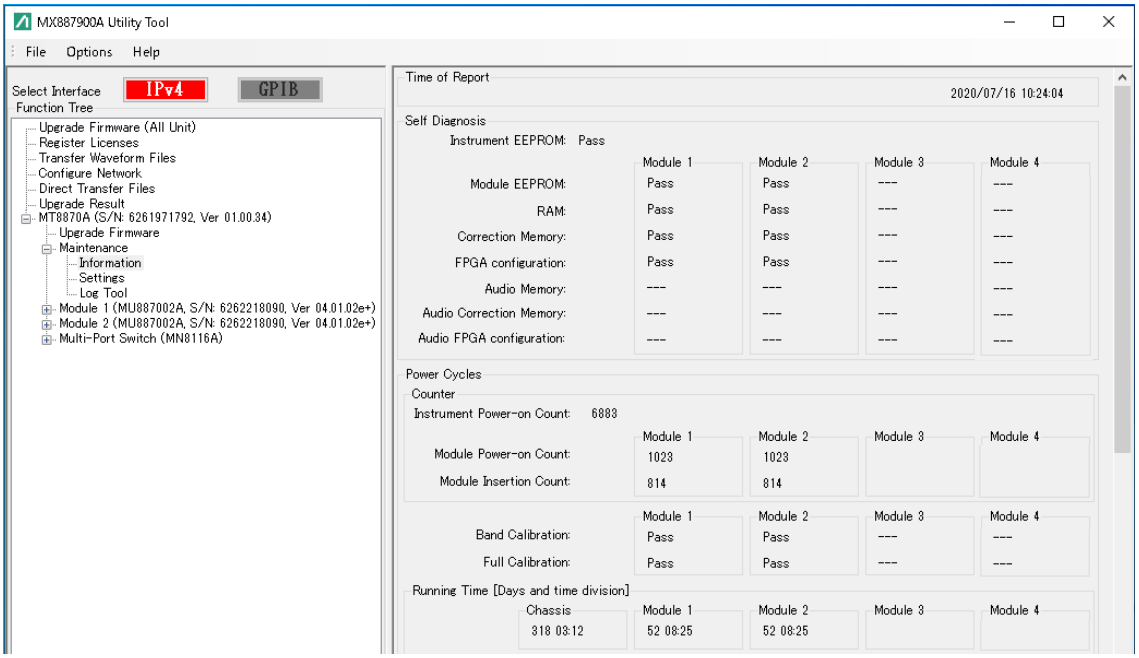
This section describes how to display the Information on the MT8870A and modules.

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

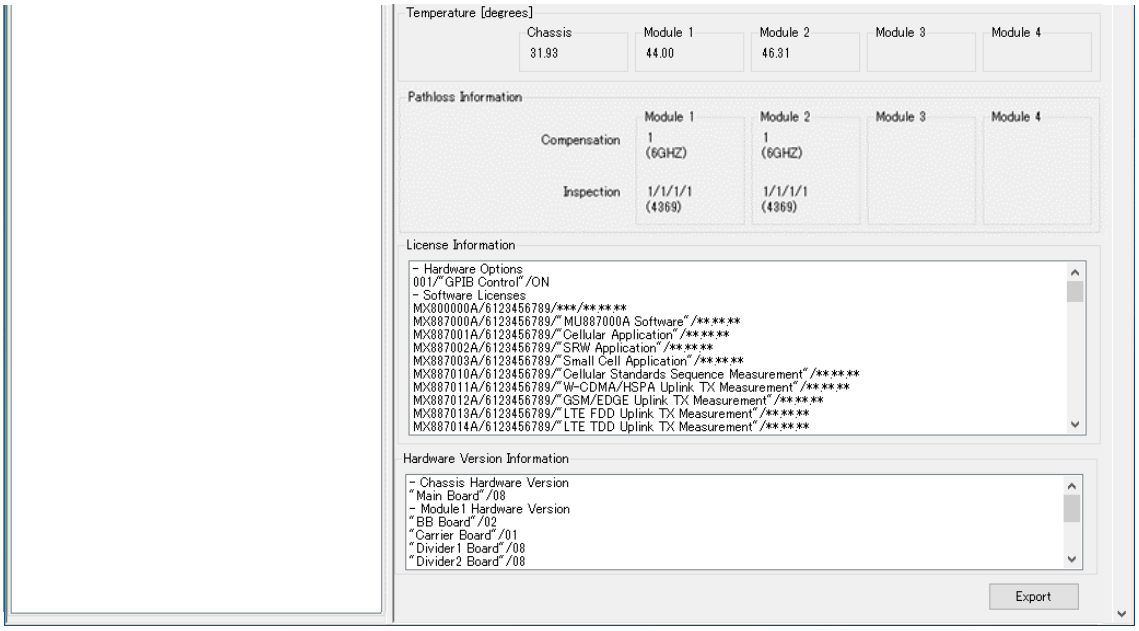


Figure 8.3.4-1 Maintenance Information Window

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)
Pathloss Information	By module (Slot 1 to 4) Compensation (Supported Freq, Range) Inspection
License Information	License information
Hardware Version Information	Hardware version information

8.3.5 Network settings

This section describes how to configure network communication settings.

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.

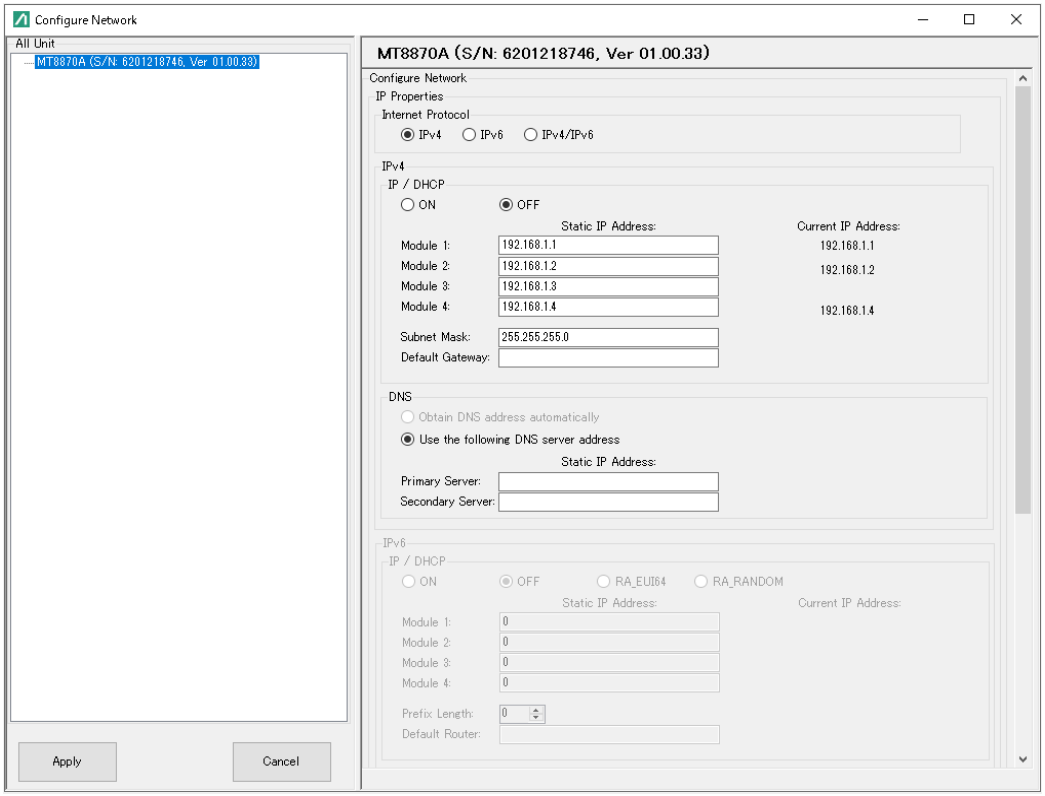


Figure 8.3.5-1 Network Settings Window 1

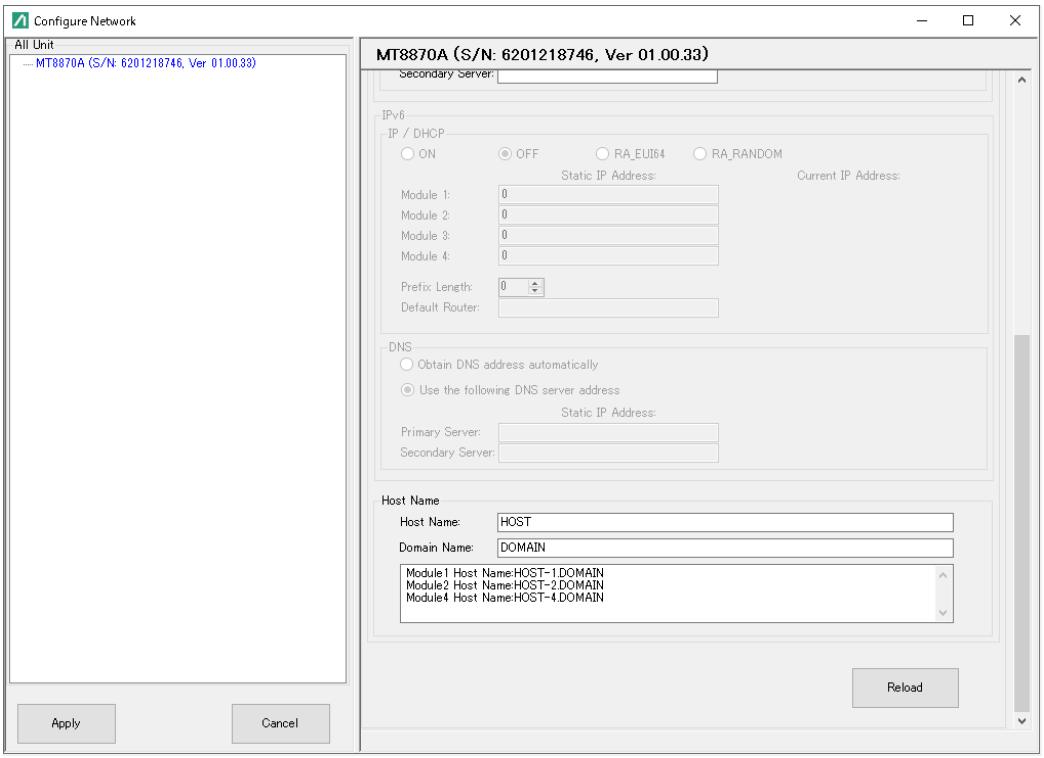


Figure 8.3.5-2 Network Settings Window 2

Table 8.3.5-1 Network Setting Items

Item	Description
IP Properties	
Internet Protocol	IP version
IPv4 IP /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 IP /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.
8. MU887002A does not support GPIB. GPIB Address is not required.
9. To set the delimiter/terminator, select a module and set its CRLF/LF/NONE, then click **Apply**. You can set only a single module at a time.
10. To set the CC Switch, select a module and set its ON/OFF, then click **Apply**. You can set only a single module at a time.

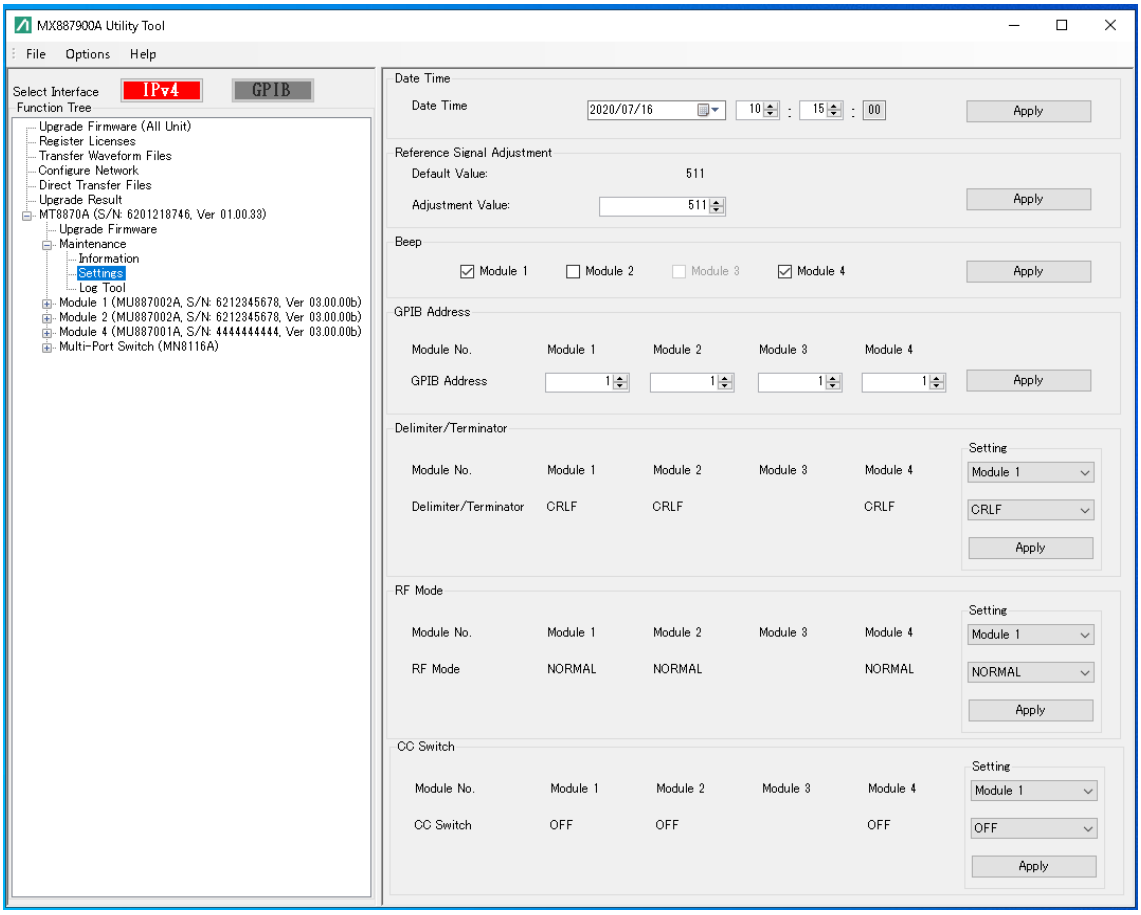


Figure 8.3.6-1 Module Settings Window

8.3.7 Log Operation

Retrieve and clear log of each module.

1. Click [+] in the tree view.
2. Click **Maintenance** – [+].
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.

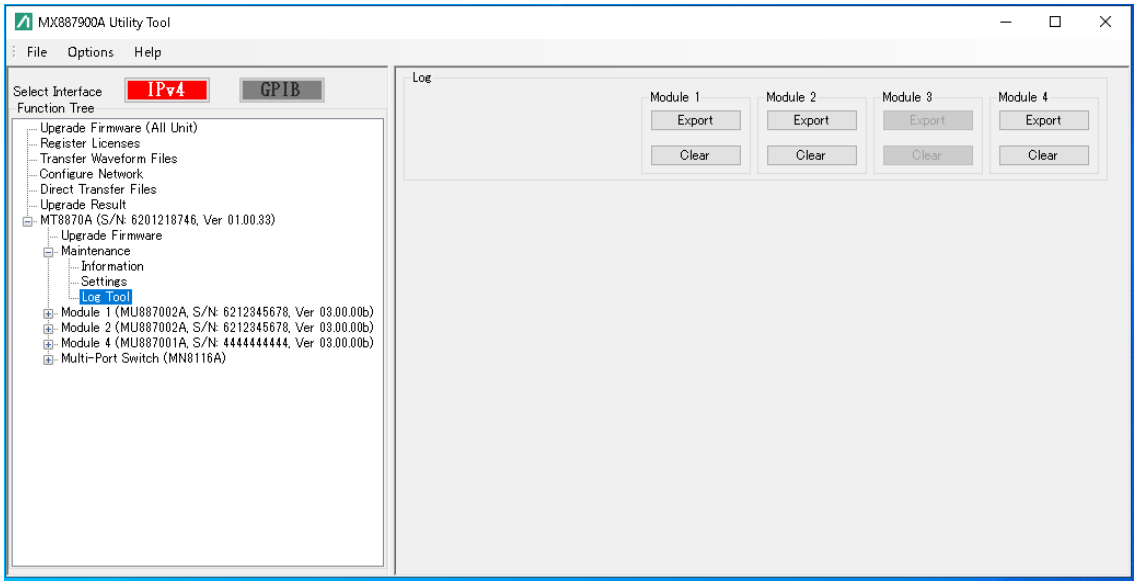


Figure 8.3.7-1 Log Operation Window

8.3.8 Upgrading

The MT8870A firmware can be upgraded when using IPv4 for control. 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**.

   Notes:

   - When the error message in Figure 8.3.8-4 is displayed.
     The upgrading is interrupted. 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.
   - When the error message in Figure 8.3.8-5 is displayed.
     The upgrading is interrupted.
     For MU887002A, the firmware package to be installed must be V3.00.00 or later.
     For MU887002A with MU887002A-007, the firmware package to be installed must be V4.00.00 or later.

   The upgrading status appears on the Progress window.

   During upgrading, the status lamp 1 to 3 of the MU887002A 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.

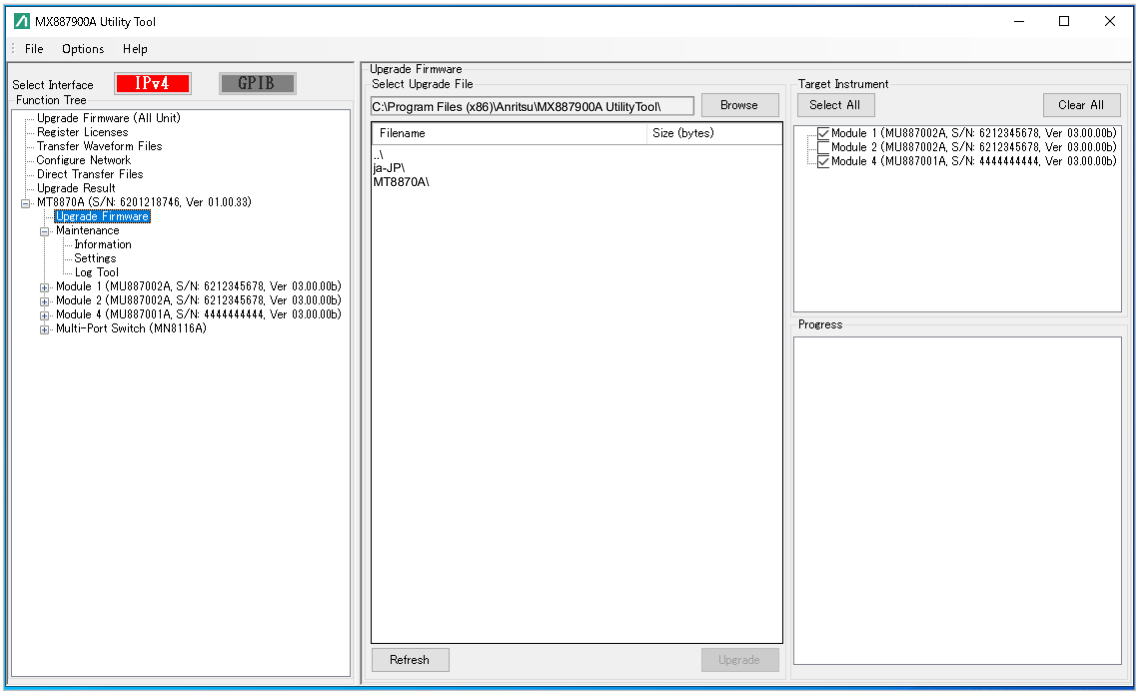


Figure 8.3.8-1 Upgrade Window

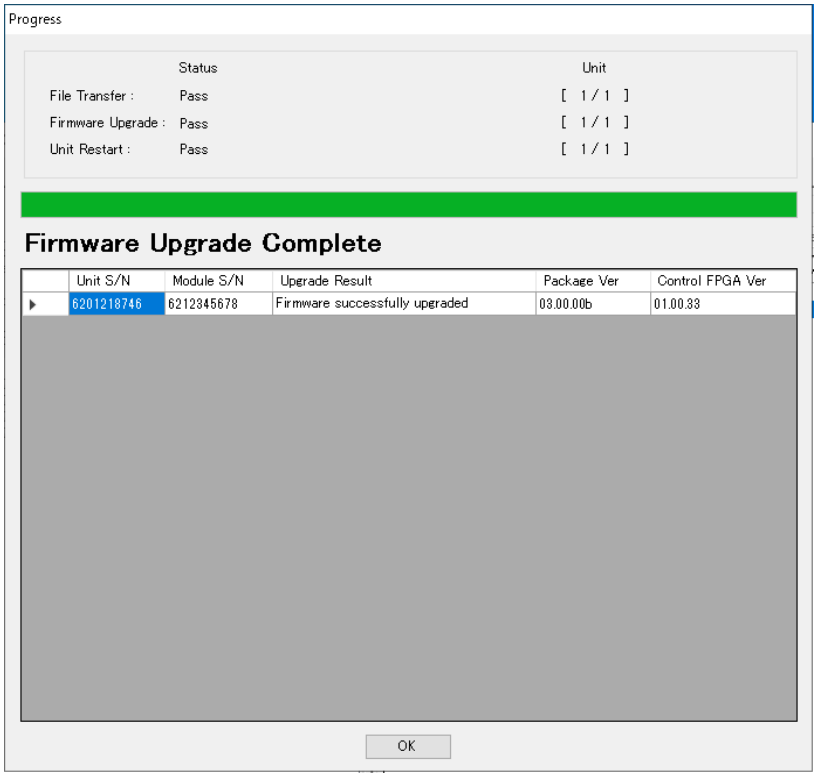


Figure 8.3.8-2 Progress Window after Upgrading

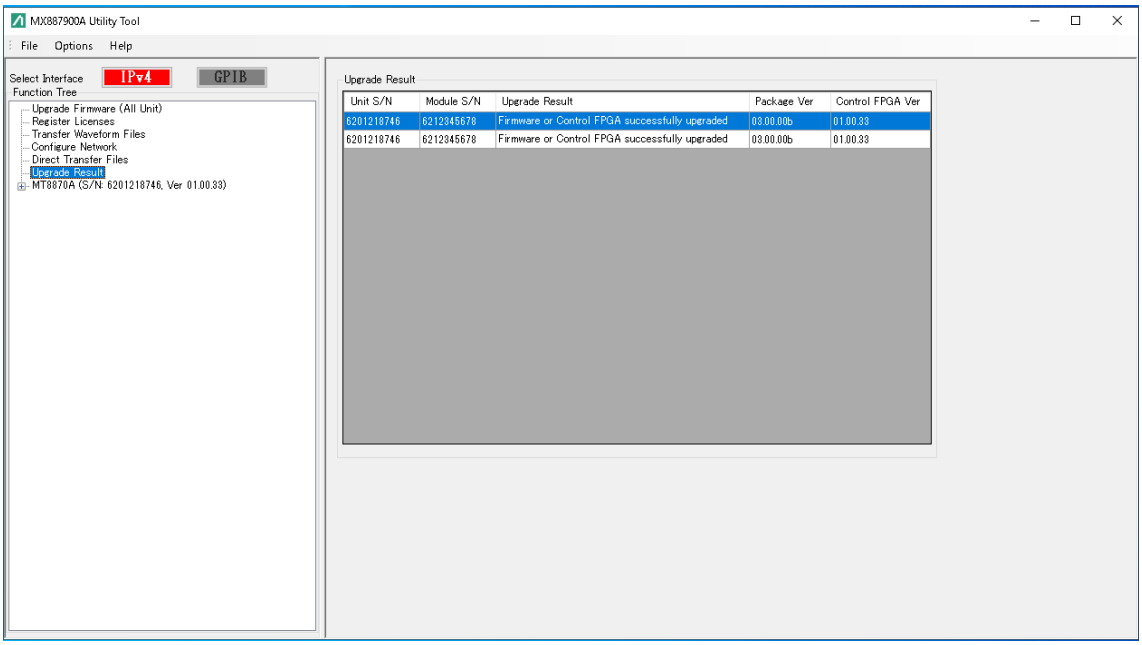


Figure 8.3.8-3 Upgrade Result Window

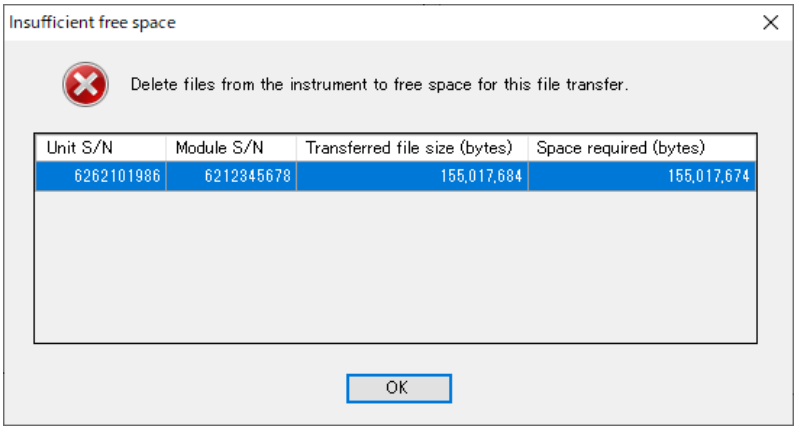


Figure 8.3.8-4 Insufficient Free Space Error Message

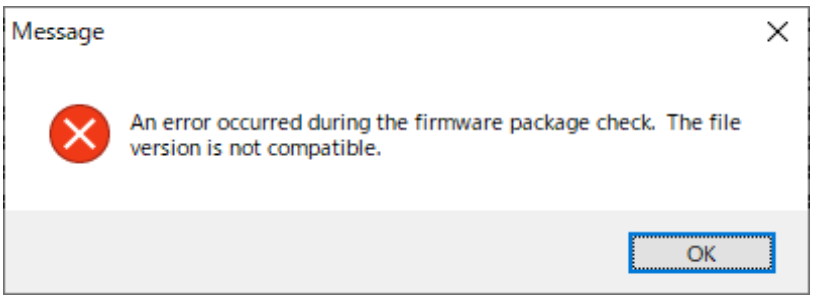


Figure 8.3.8-5 Error Message Shown When Firmware Is Incompatible Version

8.3.9 Transferring files

Files can be transferred to and from storage media of modules when using IPv4 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**.

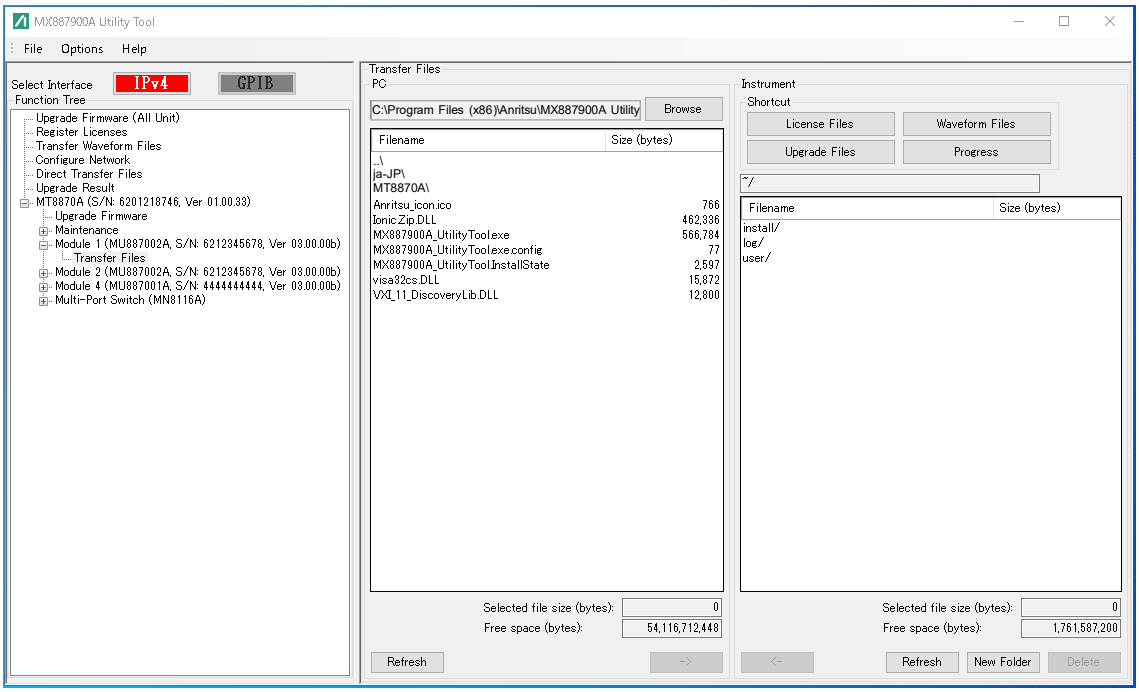


Figure 8.3.9-1 File Transfer Window

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.

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 3 of the MU887002A 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.

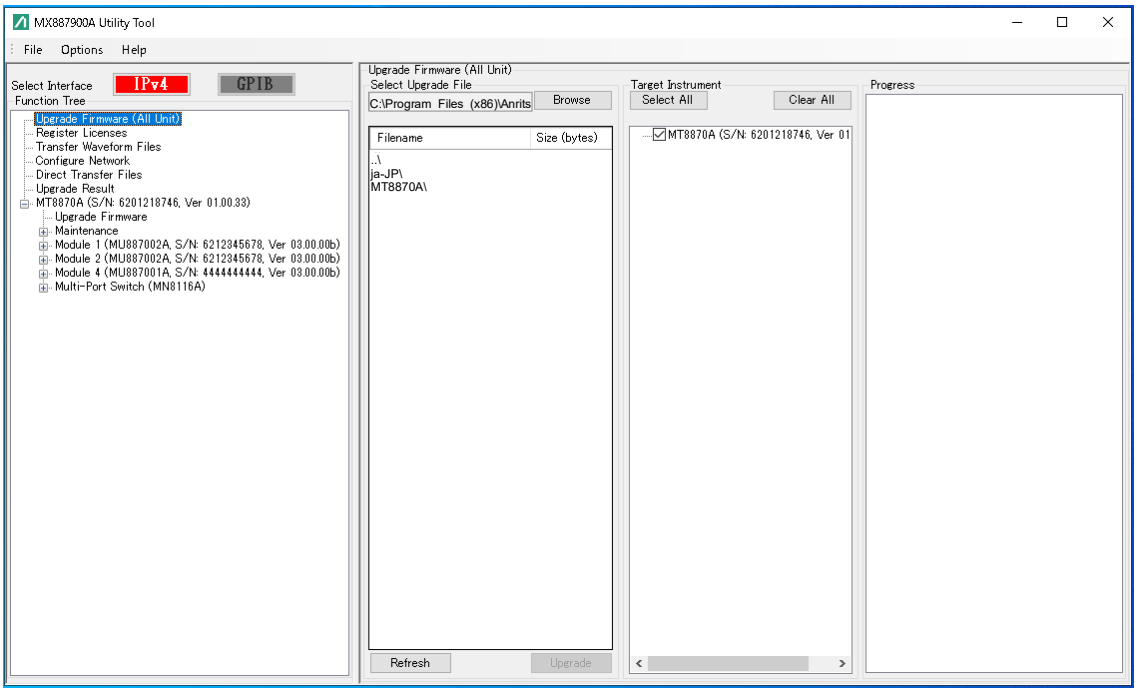


Figure 8.3.10-1 Batch Upgrade Window

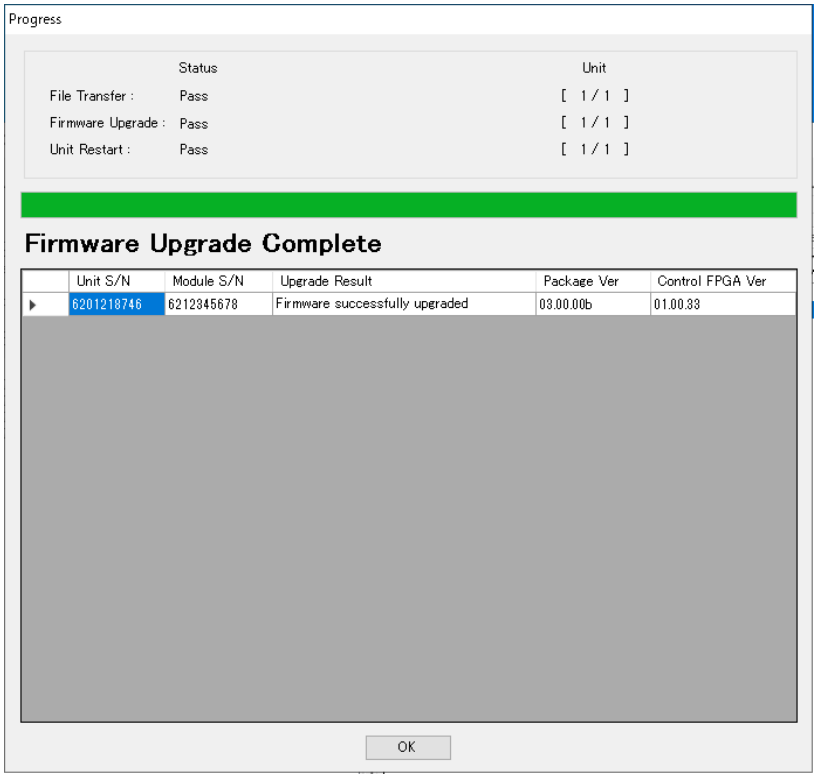


Figure 8.3.10-2 Progress Window after Upgrading Completes

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.

This function is supported only when using IPv4 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.

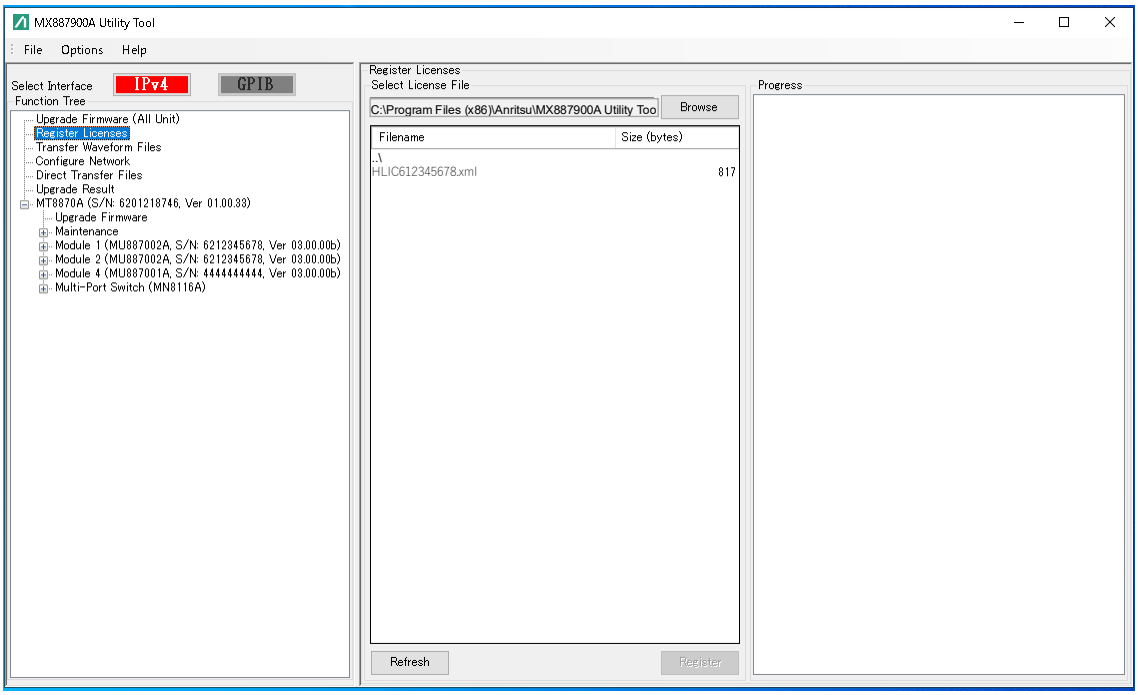


Figure 8.3.11-1 Register License Window

8.3.12 Transferring waveform files

The VSG waveform files can be transferred to modules when IPv4 is used for control.

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**.

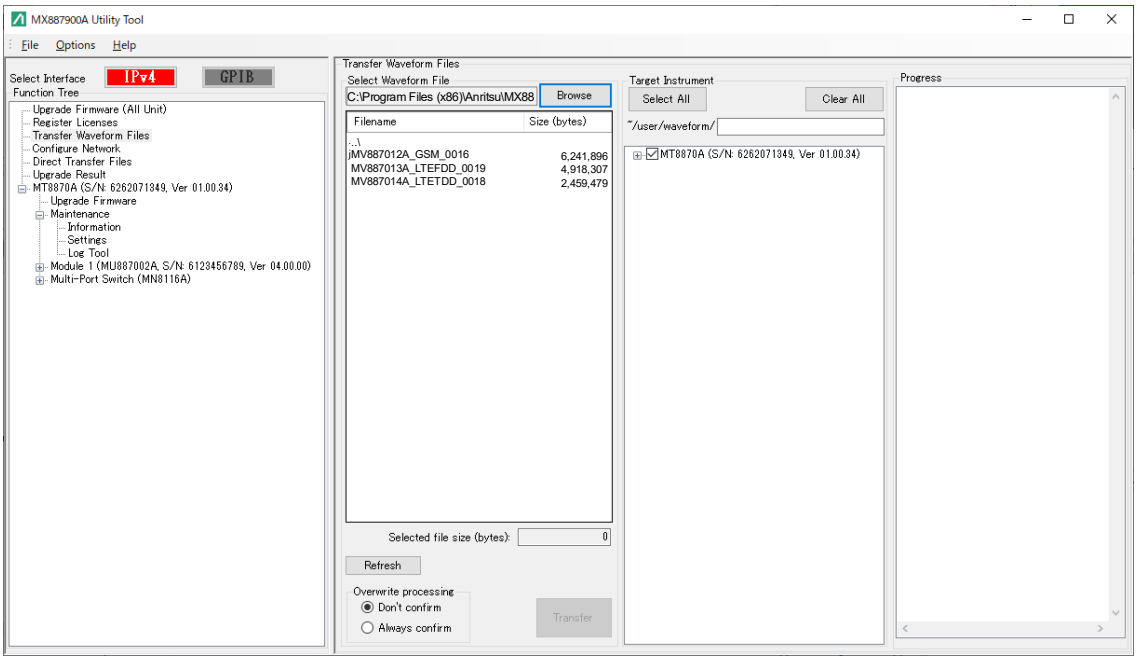


Figure 8.3.12-1 Transfer Waveform Files Window

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.

1. Click **Direct Transfer Files** in the tree view.

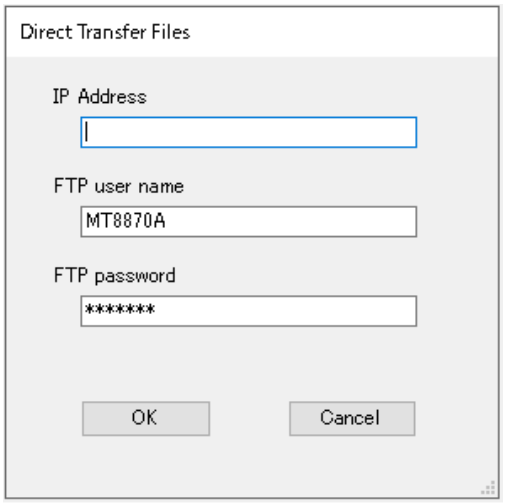


Figure 8.3.13-1 Direct Transfer Files Dialog Box

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”.

Appendix A Specifications

This appendix describes the specifications. For the product configurations, options, and optional accessories, refer to Section 1.3 “Product Configuration”.

(Typ.): Typical indicates the reference data at 20 to 30°C and is not a guaranteed value.

(Nominal): Nominal indicates the design value and is not guaranteed.

A.1 MU887002A..... A-2
A.2 MU887002A-007..... A-6

A.1 MU887002A

This section describes the specifications for MU887002A.

The following specification values are those under the conditions after 30-min warm-up at stable ambient temperature.

Table A.1-1 Input/Output Terminal

Item	Specifications
RF Test ports	TRX 1: Test Port 1 to 12 TRX 2: Test Port 1 to 12
Connector	N (female)
Impedance	50 Ω, (Nominal)
VSWR	< 1.4 (20 to 30°C) 400 MHz ≤ frequency < 450 MHz < 1.3 (20 to 30°C) 450 MHz ≤ frequency ≤ 2700 MHz < 1.4 (20 to 30°C) 2700 MHz < frequency ≤ 3800 MHz < 1.4 (20 to 30°C) 3800 MHz < frequency ≤ 6000 MHz
Input level max.	+35 dBm

Table A.1-2 Signal Generator

Item	Specifications
TRX 1/2, Test Port 1 to 12	
Frequency	
Frequency setting range	400 to 6000 MHz
Frequency resolution	1 Hz
Frequency accuracy	Depends on MT8870A reference frequency accuracy
Amplitude	
Level setting range	–130 to 0 dBm
Upper limits of guaranteed performance	–5 dBm 400 MHz ≤ frequency ≤ 3800 MHz –8 dBm 3800 MHz < frequency ≤ 6000 MHz However, for modulation waves, it is the RMS value.
Level setting resolution	0.1 dB
Level accuracy	CW, After calibration 400 MHz ≤ frequency ≤ 3800 MHz, –120 dBm ≤ output level ≤ –5 dBm ±0.7 dB (Typ.) ±1.0 dB (20 to 30°C) 3800 MHz < frequency ≤ 6000 MHz, –100 dBm ≤ output level ≤ –8 dBm ±1.0 dB (Typ.) ±1.3 dB (20 to 30°C)
Output Level Linearity with Cable Loss Setting	When broadcasting and when each test port has a different cable loss value, at 0 dB of reference cable loss ±0.2 dB (Typ.) Difference in the loss correction table value between the test ports: 0 to 8 dB
Output level deviation	When broadcasting, ≤ 0.6 dB (Nominal)
Spurious response	
Harmonic wave distortion	800 MHz ≤ Harmonic frequency ≤ 6000 MHz, –120 dBm ≤ Output level ≤ –5 dBm, CW, < –25 dBc (20 to 30°C)
Vector modulation	
Modulation bandwidth	Maximum 200 MHz

Table A.1-3 Signal Analysis

Item	Specifications
TRX 1/2, Test Port 1 to 12	
Frequency	
Frequency setting range	400 to 6000 MHz
Frequency resolution	1 Hz
Amplitude	
Level setting range	–65 to +35 dBm (CW)
Level setting resolution	0.1 dB
Level accuracy	CW, measurement bandwidth = 300 kHz, RBW = 100 kHz, after calibration, Measured signal with the same frequency as the set frequency, Measured signal at the same level as the set level 400 MHz ≤ frequency ≤ 3800 MHz ±0.3 dB (Typ.) –30 dBm ≤ level ≤ +35 dBm ±0.5 dB (20 to 30°C) –30 dBm ≤ level ≤ +35 dBm ±0.7 dB (20 to 30°C) –55 dBm ≤ level < –30 dBm ±0.9 dB (20 to 30°C) –65 dBm ≤ level < –55 dBm 3800 MHz < frequency ≤ 6000 MHz ±0.7 dB (20 to 30°C) –30 dBm ≤ level ≤ +35 dBm ±0.9 dB (20 to 30°C) –55 dBm ≤ level < –30 dBm ±1.1 dB (20 to 30°C) –65 dBm ≤ level < –55 dBm
Level linearity	CW, Measurement bandwidth = 300 kHz, RBW = 100 kHz, ±0.2 dB (20 to 30°C) 0 to –40 dB, Input level ≥ –55 dBm ±0.4 dB (20 to 30°C) 0 to –40 dB, Input level ≥ –65 dBm
Modulation analysis	
Maximum analysis bandwidth	25 MHz 400 MHz ≤ setting frequency < 500 MHz 80 MHz 500 MHz ≤ setting frequency < 600 MHz 200 MHz 600 MHz ≤ setting frequency ≤ 6000 MHz However, the results measured in the following condition are not guaranteed: (setting frequency – analysis bandwidth / 2) < 400 MHz, 6000 MHz < (setting frequency + analysis bandwidth / 2)

Table A.1-4 General Performance

Item	Specifications
Interface Trigger function Remote control	 Trigger signals input/output at Trigger connectors on MT8870A rear panel Ethernet: via MT8870A interface
Dimensions	181 (W) × 193.6 (H) × 325 (D) mm (excluding protrusions)
Mass	≤ 12.5 kg
Operating temperature	+5 to +40°C
Storage temperature	–20 to +60°C

A.2 MU887002A-007

This section describes the specifications for MU887002A with the MU887002A-007 7GHz Extension Function.

The following specification values are those under the conditions after 30-min warm-up at stable ambient temperature.

Table A.2-1 Input/Output Terminal

Item	Specifications
RF Test ports	TRX 1: Test Port 1 to 12 TRX 2: Test Port 1 to 12
Connector	N (female)
Impedance	50 Ω, (Nominal)
VSWR	TRX 1/2 Test Port 1 to 4 < 1.4 (20 to 30°C) 400 MHz ≤ frequency < 450 MHz < 1.3 (20 to 30°C) 450 MHz ≤ frequency ≤ 2700 MHz < 1.4 (20 to 30°C) 2700 MHz < frequency ≤ 3800 MHz < 1.4 (20 to 30°C) 3800 MHz < frequency ≤ 6000 MHz TRX 1/2 Test Port 5 to 12 setting frequency < 5900 MHz < 1.4 (20 to 30°C) 400 MHz ≤ frequency < 450 MHz < 1.3 (20 to 30°C) 450 MHz ≤ frequency ≤ 2700 MHz < 1.4 (20 to 30°C) 2700 MHz < frequency ≤ 3800 MHz < 1.4 (20 to 30°C) 3800 MHz < frequency < 6000 MHz 5900 MHz ≤ setting frequency < 1.7 (20 to 30°C) 5800 MHz ≤ frequency ≤ 7400 MHz
Input level max.	TRX 1/2 Test Port 1 to 4 +35 dBm 400 MHz ≤ setting frequency ≤ 6000 MHz TRX 1/2 Test Port 5 to 12 +35 dBm 400 MHz ≤ setting frequency < 5900 MHz +30 dBm 5900 MHz ≤ setting frequency ≤ 7300 MHz

Table A.2-2 Signal Generator

Item	Specifications
TRX 1/2, Test Port 1 to 12	
Frequency	
Frequency setting range	TRX 1/2, Test Port 1 to 4 400 to 6000 MHz TRX 1/2, Test Port 5 to 12 400 to 7300 MHz
Frequency resolution	1 Hz
Frequency accuracy	Depends on MT8870A reference frequency accuracy
Amplitude	
Level setting range	–130 to 0 dBm
Upper limits of guaranteed performance	TRX 1/2, Test Port 1 to 4 –5 dBm 400 MHz ≤ frequency ≤ 3800 MHz –8 dBm 3800 MHz < frequency ≤ 6000 MHz However, for modulation waves, it is the RMS value. TRX 1/2, Test Port 5 to 12 –5 dBm 400 MHz ≤ frequency ≤ 3800 MHz –8 dBm 3800 MHz < frequency < 5900 MHz –10 dBm 5900 MHz ≤ frequency ≤ 7300 MHz However, for modulation waves, it is the RMS value.
Level setting resolution	0.1 dB
Level accuracy	CW, After calibration TRX 1/2, Test Port 1 to 4 400 MHz ≤ frequency ≤ 3800 MHz, –120 dBm ≤ output level ≤ –5 dBm ±0.7 dB (Typ.) ±1.0 dB (20 to 30°C) –5 dBm < output level ≤ 0 dBm ±0.7 dB (Typ.) 3800 MHz < frequency ≤ 6000 MHz, –100 dBm ≤ output level ≤ –8 dBm ±1.0 dB (Typ.) ±1.3 dB (20 to 30°C) –8 dBm < output level ≤ 0 dBm ±1.0 dB (Typ.)

Table A.2-2 Signal Generator (Cont'd)

Item	Specifications
Amplitude Level accuracy (Cont'd)	TRX 1/2, Test Port 5 to 12 400 MHz ≤ frequency ≤ 3800 MHz, –120 dBm ≤ output level ≤ –5 dBm ±0.7 dB (Typ.) ±1.0 dB (20 to 30°C) –5 dBm < output level ≤ 0 dBm ±0.7 dB (Typ.) 3800 MHz < frequency < 5900 MHz, –100 dBm ≤ output level ≤ –8 dBm ±1.0 dB (Typ.) ±1.3 dB (20 to 30°C) –8 dBm < output level ≤ 0 dBm ±1.0 dB (Typ.) 5900 MHz ≤ frequency ≤ 7300 MHz, –100 dBm ≤ output level ≤ –10 dBm ±1.0 dB (Typ.) ±1.3 dB (20 to 30°C) –10 dBm < output level ≤ 0 dBm ±1.0 dB (Typ.)
Output Level Linearity with Cable Loss Setting	TRX 1/2, Test Port 1 to 4, 400 MHz ≤ frequency ≤ 6000 MHz, When broadcasting and when each test port has a different cable loss value, at 0 dB of reference cable loss ±0.2 dB (Typ.) Difference in the loss correction table value between the test ports: 0 to 8 dB TRX 1/2, Test Port 5 to 12, 400 MHz ≤ frequency ≤ 7300 MHz, When broadcasting and when each test port has a different cable loss value, at 0 dB of reference cable loss ±0.2 dB (Typ.) Difference in the loss correction table value between the test ports: 0 to 8 dB
Output level deviation	TRX 1/2, Test Port 1 to 12, When broadcasting ≤ 0.6 dB (Nominal)
Spurious response Harmonic wave distortion	TRX 1/2, Test Port 1 to 4, 800 MHz ≤ Harmonic frequency ≤ 6000 MHz, –120 dBm ≤ Output level ≤ –5 dBm, CW, < –25 dBc (20 to 30°C) TRX 1/2, Test Port 5 to 12, 800 MHz ≤ Harmonic frequency ≤ 7300 MHz, –120 dBm ≤ Output level ≤ –5 dBm, CW, < –25 dBc (20 to 30°C)
Vector modulation Modulation bandwidth	Maximum 200 MHz

Table A.2-3 Signal Analysis

Item	Specifications
TRX 1/2, Test Port 1 to 12	
Frequency	
Frequency setting range	TRX 1/2, Test Port 1 to 4 400 to 6000 MHz TRX 1/2, Test Port 5 to 12 400 to 7300 MHz
Frequency resolution	1 Hz
Amplitude	
Level setting range	TRX 1/2, Test Port 1 to 4 –65 to +35 dBm (CW) 400 MHz ≤ setting frequency ≤ 6000 MHz TRX 1/2, Test Port 5 to 12 –65 to +35 dBm (CW) 400 MHz ≤ setting frequency < 5900 MHz –65 to +30 dBm (CW) 5900 MHz ≤ setting frequency ≤ 7300 MHz
Level setting resolution	0.1 dB
Level accuracy	CW, measurement bandwidth = 300 kHz, RBW = 100 kHz, after calibration, Measured signal with the same frequency as the set frequency, Measured signal at the same level as the set level TRX 1/2, Test Port 1 to 4 400 MHz ≤ frequency ≤ 3800 MHz ±0.3 dB (Typ.) –30 dBm ≤ level ≤ +35 dBm ±0.5 dB (20 to 30°C) –30 dBm ≤ level ≤ +35 dBm ±0.7 dB (20 to 30°C) –55 dBm ≤ level < –30 dBm ±0.9 dB (20 to 30°C) –65 dBm ≤ level < –55 dBm 3800 MHz < frequency ≤ 6000 MHz ±0.7 dB (20 to 30°C) –30 dBm ≤ level ≤ +35 dBm ±0.9 dB (20 to 30°C) –55 dBm ≤ level < –30 dBm ±1.1 dB (20 to 30°C) –65 dBm ≤ level < –55 dBm TRX 1/2, Test Port 5 to 12 400 MHz ≤ frequency ≤ 3800 MHz ±0.3 dB (Typ.) –30 dBm ≤ level ≤ +35 dBm ±0.5 dB (20 to 30°C) –30 dBm ≤ level ≤ +35 dBm ±0.7 dB (20 to 30°C) –55 dBm ≤ level < –30 dBm ±0.9 dB (20 to 30°C) –65 dBm ≤ level < –55 dBm 3800 MHz < frequency < 5900 MHz ±0.7 dB (20 to 30°C) –30 dBm ≤ level ≤ +35 dBm ±0.9 dB (20 to 30°C) –55 dBm ≤ level < –30 dBm ±1.1 dB (20 to 30°C) –65 dBm ≤ level < –55 dBm 5900 MHz ≤ frequency ≤ 7300 MHz ±1.0 dB (20 to 30°C) –30 dBm ≤ level ≤ +30 dBm ±1.2 dB (20 to 30°C) –55 dBm ≤ level < –30 dBm ±1.8 dB (20 to 30°C) –65 dBm ≤ level < –55 dBm

Table A.2-3 Signal Analysis (Cont'd)

Item	Specifications
Amplitude Level linearity	CW, Measurement bandwidth = 300 kHz, RBW = 100 kHz, TRX 1/2, Test Port 1 to 4 400 MHz ≤ frequency ≤ 6000 MHz ±0.2 dB (20 to 30°C) 0 to –40 dB, Input level ≥ –55 dBm ±0.4 dB (20 to 30°C) 0 to –40 dB, Input level ≥ –65 dBm TRX 1/2, Test Port 5 to 12 400 MHz ≤ frequency ≤ 7300 MHz ±0.2 dB (20 to 30°C) 0 to –40 dB, Input level ≥ –55 dBm ±0.4 dB (20 to 30°C) 0 to –40 dB, Input level ≥ –65 dBm
Modulation analysis Maximum analysis bandwidth	TRX 1/2, Test Port 1 to 4 25 MHz 400 MHz ≤ setting frequency < 500 MHz 80 MHz 500 MHz ≤ setting frequency < 600 MHz 200 MHz 600 MHz ≤ setting frequency ≤ 6000 MHz However, the results measured in the following condition are not guaranteed: (setting frequency – analysis bandwidth / 2) < 400 MHz, 6000 MHz < (setting frequency + analysis bandwidth / 2) TRX 1/2, Test Port 5 to 12 25 MHz 400 MHz ≤ setting frequency < 500 MHz 80 MHz 500 MHz ≤ setting frequency < 600 MHz 200 MHz 600 MHz ≤ setting frequency ≤ 7300 MHz However, the results measured in the following condition are not guaranteed: (setting frequency – analysis bandwidth / 2) < 400 MHz

Appendix B Software Licenses

This product includes the package software listed in the following table. Package software in the table is not included our software licensing.

For details on each license, go to the following website:
https://my.anritsu.com/myproduct/home

Table B-1 Packages and Licenses

Package Name	License	License Reference No.
acl	LGPLv2+	(*03)
acpid	GPLv2+	(*01)
apr	ASL 2.0 and BSD with advertising and ISC and BSD	(*06)(*08)(*12)(*73)
apr-uti	ASL 2.0	(*06)
audit	GPLv2+	(*01)
audit-libs	GPLv2+	(*01)
authconfig	GPLv2+	(*01)
autoconf	GPLv2+ and GFDL	(*01)(*76)
autogen-libopts	LGPLv3+	(*04)
automake	GPLv2+ and GFDL and Public Domain and MIT	(*01)(*76)(*73)
avahi-libs	LGPLv2+	(*03)
basesystem	Public Domain	–
bash	GPLv3+	(*04)
bc	GPLv2+	(*01)
bind-libs-lite	ISC	(*12)
bind-license	ISC	(*12)
binutils	GPLv3+	(*02)
bison	GPLv3+	(*02)
boost-date-time	Boost and MIT and Python	(*07)(*73)(*19)
boost-system	Boost and MIT and Python	(*07)(*73)(*19)
boost-thread	Boost and MIT and Python	(*07)(*73)(*19)
btrfs-progs	GPLv2	(*01)
byacc	Public Domain	–
bzip2	BSD	(*72)
bzip2-libs	BSD	(*72)
ca-certificates	Public Domain	–
centos-logos	Copyright c 2014 The CentOS Project. All rights reserved.	(*77)
centos-release	GPLv2	(*01)
chkconfig	GPLv2	(*01)
chrony	GPLv2	(*01)
coreutils	GPLv3+	(*02)
cpio	GPLv3+	(*02)
cpp	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*02)(*30)(*72)
cracklib	LGPLv2+	(*03)
cracklib-dicts	LGPLv2+	(*03)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
cronie	MIT and BSD and ISC and GPLv2+	(*73)(*72)(*12)(*01)
cronie-anacron	MIT and BSD and ISC and GPLv2+	(*73)(*72)(*12)(*01)
crontabs	Public Domain and GPLv2	(*01)
cryptsetup-libs	GPLv2+ and LGPLv2+	(*01)(*03)
cscope	BSD and GPLv2+	(*72)(*01)
ctags	GPLv2+ and LGPLv2+ and Public Domain	(*01)(*03)
curl	MIT	(*73)
cyrus-sasl-lib	BSD with advertising	(*08)
dbus	GPLv2+ or AFL	(*01)(*05)
dbus-glib	AFL and GPLv2+	(*05)(*01)
dbus-libs	GPLv2+ or AFL	(*01)(*05)
dbus-python	MIT	(*73)
device-mapper	GPLv2	(*01)
device-mapper-libs	LGPLv2	(*03)
dhclient	ISC	(*12)
dhcp-common	ISC	(*12)
dhcp-libs	ISC	(*12)
diffstat	MIT	(*73)
diffutils	GPLv3+	(*02)
dmidecode	GPLv2+	(*01)
dosfstools	GPLv3+	(*02)
doxygen	GPL+	(*02)
dracut	GPLv2+ and LGPLv2+	(*01)(*03)
dracut-config-rescue	GPLv2+ and LGPLv2+	(*01)(*03)
dracut-network	GPLv2+ and LGPLv2+	(*01)(*03)
dwz	GPLv2+ and GPLv3+	(*01)(*02)
dyninst	LGPLv2+	(*03)
e2fsprogs	GPLv2	(*01)
e2fsprogs-libs	GPLv2 and LGPLv2	(*01)(*03)
ebtables	GPLv2+	(*01)
ed	GPLv3+, GFDL	(*02)(*76)
efivar-libs	LGPLv2.1	(*03)
elfutils	GPLv3+ and (GPLv2+ or LGPLv3+)	(*02)(*01)(*04)
elfutils-default-yama-scope	GPLv2+ or LGPLv3+	(*01)(*04)
elfutils-libelf	GPLv2+ or LGPLv3+	(*01)(*04)
elfutils-libs	GPLv2+ or LGPLv3+	(*01)(*04)
emacs-filesystem	GPLv3+	(*02)
ethtool	GPLv2	(*01)
expat	MIT	(*73)
file	BSD	(*72)
file-libs	BSD	(*72)
filesystem	Public Domain	–
findutils	GPLv3+	(*02)
fipscheck	BSD	(*72)
fipscheck-lib	BSD	(*72)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
firewalld	GPLv2+	(*01)
firewalld-filesystem	GPLv2+	(*01)
flex	BSD and LGPLv2+	(*03)(*72)
freetype	(FTL or GPLv2+) and BSD and MIT and Public Domain and zlib with acknowledgement	(*09)(*01)(*72)(*73)(*79)
fxload	GPLv2+	(*01)
ftp	BSD with advertising	(*08)
gawk	GPLv3+ and GPL and LGPLv3+ and LGPL and BSD	(*02)
gcc	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*01)(*84)(*02)
gcc-c++	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*01)(*84)(*02)
gcc-gfortran	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*01)(*84)(*02)
gdb	GPLv3+ and GPLv3+ with exceptions and GPLv2+ and GPLv2+ with exceptions and GPL+ and LGPLv2+ and BSD and Public Domain	(*01)(*02)
gdbm	GPLv3+	(*02)
GeoIP	LGPLv2+ and GPLv2+ and CC-BY-SA	(*01)(*03)(*83)
gettext	GPLv3+ and LGPLv2+	(*02)(*03)
gettext-common-devel	GPLv3+	(*02)
gettext-devel	LGPLv2+ and GPLv3+	(*02)(*03)
gettext-libs	LGPLv2+ and GPLv3+	(*02)(*03)
git	GPLv2	(*01)
glib2	LGPLv2+	(*03)
glibc	LGPLv2+ and LGPLv2+ with exceptions and GPLv2+	(*01)(*02)
glibc-common	LGPLv2+ and LGPLv2+ with exceptions and GPLv2+	(*01)(*02)
glibc-devel	LGPLv2+ and LGPLv2+ with exceptions and GPLv2+	(*01)(*02)
glibc-headers	LGPLv2+ and LGPLv2+ with exceptions and GPLv2+	(*01)(*02)
gmp	LGPLv3+ or GPLv2+	(*01)(*02)
gnupg2	GPLv3+	(*02)
gnutls	GPLv3+ and LGPLv2+	(*02)(*03)
gobject-introspection	GPLv2+, LGPLv2+, MIT	(*01)(*03)(*73)
gpgme	LGPLv2+	(*03)
gpg-pubkey	pubkey	–
grep	GPLv3+	(*02)
groff-base	GPLv3+ and GFDL and BSD and MIT	(*02)(*72)(*73)(*76)
grub2	GPLv3+	(*02)
grub2-common	GPLv3+	(*02)
grub2-pc	GPLv3+	(*02)
grub2-pc-modules	GPLv3+	(*02)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
grub2-tools	GPLv3+	(*02)
grub2-tools-extra	GPLv3+	(*02)
grub2-tools-minimal	GPLv3+	(*02)
grubby	GPLv2+	(*01)
gzip	GPLv3+ and GFDL	(*02)(*76)
hardlink	GPL+	(*02)
hostname	GPLv2+	(*01)
hwdata	GPLv2+	(*01)
indent	GPLv3+	(*03)
info	GPLv3+	(*03)
initscripts	GPLv2 and GPLv2+	(*01)
intltool	GPLv2 with exceptions	(*01)
iprutils	CPL	(*75)
ipset	GPLv2	(*01)
ipset-libs	GPLv2	(*01)
iproute	GPLv2+ and Public Domain	(*01)
iptables	GPLv2	(*01)
iptables-services	GPLv2	(*01)
iputils	BSD and GPLv2+	(*01)
irqbalance	GPLv2	(*01)
iwl100-firmware	Redistributable, no modification permitted	(*80)
iwl105-firmware	Redistributable, no modification permitted	(*80)
iwl135-firmware	Redistributable, no modification permitted	(*80)
iwl2000-firmware	Redistributable, no modification permitted	(*80)
iwl2030-firmware	Redistributable, no modification permitted	(*80)
iwl3160-firmware	Redistributable, no modification permitted	(*80)
iwl6000g2a-firmware	Redistributable, no modification permitted	(*80)
iwl6000g2b-firmware	Redistributable, no modification permitted	(*80)
iwl7260-firmware	Redistributable, no modification permitted	(*80)
iwl7265-firmware	Redistributable, no modification permitted	(*80)
kbd	GPLv2+	(*01)(*10)
kbd-legacy	GPLv2+	(*01)
kbd-misc	GPLv2+	(*01)
kernel	GPLv2	(*01)
kernel-devel	GPLv2	(*01)
kernel-headers	GPLv2	(*01)
kernel-tools	GPLv2	(*01)
kernel-tools-libs	GPLv2	(*01)
kexec-tools	GPLv2	(*01)
keyutils-libs	GPLv2+ and LGPLv2+	(*01)(*03)
kmod	LGPLv2+	(*03)
kmod-libs	LGPLv2+	(*03)
kpartx	GPL+	(*02)
krb5-libs	MIT	(*73)
libfastjson	MIT	(*73)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
libsemanage	LGPLv2+	(*03)
libpciaccess	MIT	(*73)
lua	MIT	(*73)
libgcrypt	LGPLv2+	(*03)
libedit	BSD	(*72)
libassuan	LGPLv2+ and GPLv3+	(*01)(*02)
libidn	LGPLv2+ and GPLv3+ and GFDL	(*76)(*01)(*02)
libstdc++	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*72)(*01)(*02)(*03)
libgomp	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*72)(*01)(*02)(*03)
libstdc++	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*72)(*01)(*02)(*03)
libmodman	LGPLv2+	(*03)
libcroco	LGPLv2	(*03)
lsscsi	GPLv2+	(*01)
linux-firmware	GPL+ and GPLv2+ and MIT and Redistributable, no modification permitted	(*73)(*03)
libsepol	LGPLv2+	(*03)
libselinux	Public Domain	–
libdb	BSD and LGPLv2 and Sleepycat	(*31)
libuuid	BSD	(*37)
libdrm	MIT	(*73)
libattr	LGPLv2+	(*03)
libacl	LGPLv2+	(*03)
libxml2	MIT	(*73)
libgpg-error	LGPLv2+	(*03)
libmpc	LGPLv3+ and GFDL	(*76)(*02)
libestr	LGPLv2+	(*03)
libtool	GPLv2+ and LGPLv2+ and GFDL	(*76)(*01)(*03)
less	GPLv3+	(*03)
lsof	zlib and Sendmail and LGPLv2+	(*03)(*79)
libunistring	LGPLv3+	(*04)
libnfnetlink	GPLv2+	(*01)
lzo	GPLv2+	(*01)
libss	MIT	(*73)
libdwarf	LGPLv2	(*03)
libselinux-utils	Public Domain	–
libproxy	LGPLv2+	(*03)
libverto	MIT	(*73)
libblkid	LGPLv2+	(*03)
libpwquality	BSD or GPLv2+	(*72)(*01)
logrotate	GPL+	(*02)
libcurl	MIT	(*73)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
libuser	LGPLv2+	(*03)
libgcc	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*30)
libstdc++-devel	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*02)(*04)(*72)
libseccomp	LGPLv2	(*03)
libpcap	BSD with advertising	(*72)
libpipeline	GPLv3+	(*03)
libutempter	LGPLv2+	(*03)
libcom_err	MIT	(*73)
libcap-ng	LGPLv2+	(*03)
libcap	LGPLv2+	(*03)
libffi	MIT and Public Domain	(*73)
libmnl	LGPLv2+	(*03)
libtasn1	GPLv3+ and LGPLv2+	(*02)(*03)
libsysfs	LGPLv2+	(*03)
libnetfilter_conntrack	GPLv2+	(*01)
libquadmath	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*02)(*04)(*72)
libgfortran	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*02)(*04)(*72)
libquadmath-devel	GPLv3+ and GPLv3+ with exceptions and GPLv2+ with exceptions and LGPLv2+ and BSD	(*02)(*04)(*72)
libdb-utils	BSD and LGPLv2 and Sleepycat	(*03)(*72)
libmount	LGPLv2+	(*03)
libgnome-keyring	GPLv2+ and LGPLv2+	(*01)(*03)
libselinux-python	Public Domain	–
libtirpc	SISSL and BSD	(*72)
libssh2	BSD	(*36)
man-pages	GPL+ and GPLv2+ and BSD and MIT and Copyright only and IEEE	(*39)
mariadb-libs	GPLv2 with exceptions and LGPLv2 and BSD	(*03)(*01)(*73)
mailx	BSD with advertising and MPLv1.1	(*38)
make	GPLv2+	(*01)
microcode_ctl	GPLv2+ and Redistributable, no modification permitted	(*01)
m4	GPLv3+	(*03)
mpfr	LGPLv3+ and GPLv3+ and GFDL	(*76)(*02)(*04)
man-db	GPLv2+ and GPLv3+	(*01)(*02)
mozjs17	GPLv2+ or LGPLv2+ or MPLv1.1	(*15)(*01)(*03)
man-pages-ja	Freely redistributable without restriction	–
ntpdate	(MIT and BSD and BSD with advertising) and GPLv2	(*73)(*01)
neon	LGPLv2+	(*03)
nsawb	GPLv2	(*01)
net-tools	GPLv2+	(*01)
nspr	MPLv2.0	(*16)
nss	MPLv2.0	(*16)
numactl-libs	LGPLv2 and GPLv2	(*01)(*03)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
ncurses-libs	MIT	(*73)
newt	LGPLv2	(*03)
nss-util	MPLv2.0	(*16)
nss-softokn-freebl	MPLv2.0	(*16)
nss-pem	MPLv1.1	(*15)
nss-sysinit	MPLv2.0	(*16)
ncurses-base	MIT	(*73)
ntp	(MIT and BSD and BSD with advertising) and GPLv2	(*01)(*73)
nss-softokn	MPLv2.0	(*16)
nettle	LGPLv2+	(*03)
nss-softokn-freebl	MPLv2.0	(*16)
ncurses	MIT	(*73)
newt-python	LGPLv2	(*03)
nss-tools	MPLv2.0	(*16)
openssh-server	BSD	(*72)
openssh	BSD	(*72)
os-prober	GPLv2+ and GPL+	(*01)
openssl	OpenSSL	(*18)
openldap	OpenLDAP	(*17)
openssh-clients	BSD	(*72)
openssl-libs	OpenSSL	(*18)
pygpgme	LGPLv2+	(*03)
python-urlgrabber	LGPLv2+	(*03)
python-configobj	BSD	(*72)
pam	BSD and GPLv2+	(*72)(*01)
python	Python	(*19)
psmisc	GPLv2+	(*01)
perl-Pod-Simple	GPL+ or Artistic	(*02)
perl-Carp	GPL+ or Artistic	(*02)
perl-Time-Local	GPL+ or Artistic	(*02)
parted	GPLv3+	(*03)
perl-Filter	GPL+ or Artistic	(*02)
perl-libs	GPL+ or Artistic	(*02)
perl-Pod-Escapes	GPL+ or Artistic	(*02)
postfix	IBM and GPLv2+	(*02)
perl-parent	GPL+ or Artistic	(*02)
plymouth-scripts	GPLv2+	(*02)
pcre	BSD	(*45)
pth	LGPLv2+	(*03)
python-pycurl	LGPLv2+ or MIT	(*03)(*73)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
python-perf	GPLv2	(*01)
python-firewall	GPLv2+	(*01)
python-slip	GPLv2+	(*01)
python-iniparse	MIT	(*73)
pyliblzma	LGPLv3+	(*04)
python-linux-procfs	GPLv2	(*01)
plymouth-core-libs	GPLv2+	(*01)
pkgconfig	GPLv2+	(*01)
python-decorator	BSD	(*72)
python-libs	Python	(*19)
pinentry	GPLv2+	(*01)
perl-TermReadKey	(Copyright only) and (Artistic or GPL+)	(*02)
perl-XML-Parser	GPL+ or Artistic	(*02)
perl-Data-Dumper	GPL+ or Artistic	(*02)
p11-kit-trust	BSD	(*42)
patch	GPLv3+	(*03)
patchutils	GPLv2+	(*01)
perl-Error	(GPL+ or Artistic) and MIT	(*02)(*73)
pciutils-libs	GPLv2+	(*01)
perl-Getopt-Long	GPLv2+ or Artistic	(*02)
perl-PathTools	(GPL+ or Artistic) and BSD	(*02)(*72)
perl-File-Temp	GPL+ or Artistic	(*02)
perl-Scalar-List-Utils	GPL+ or Artistic	(*02)
perl-constant	GPL+ or Artistic	(*02)
perl-Socket	GPL+ or Artistic	(*02)
perl-Exporter	GPL+ or Artistic	(*02)
perl-Storable	GPL+ or Artistic	(*02)
perl-macros	GPL+ or Artistic	(*02)
perl-Pod-Usage	GPL+ or Artistic	(*02)
perl-Text-ParseWords	GPL+ or Artistic	(*02)
perl-Pod-Perldoc	GPL+ or Artistic	(*02)
perl-HTTP-Tiny	GPL+ or Artistic	(*02)
pciutils	GPLv2+	(*01)
p11-kit	BSD	(*42)
perl-Git	GPLv2	(*01)
plymouth	GPLv2+	(*01)
polkit	LGPLv2+	(*03)
popt	MIT	(*73)
perl-srpm-macros	GPLv3+	(*03)
pakchois	LGPLv2+	(*03)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
python-schedutils	GPLv2	(*01)
python-slip-dbus	GPLv2+	(*01)
pyxattr	LGPLv2+	(*03)
python-gobject-base	LGPLv2+ and MIT	(*03)(*73)
python-pyudev	LGPLv2+	(*03)
procps-ng	GPL+ and GPLv2 and GPLv2+ and GPLv3+ and LGPLv2+	(*78)
perl-Test-Harness	GPL+ or Artistic	(*02)
perl-Thread-Queue	GPL+ or Artistic	(*02)
perl	(GPL+ or Artistic) and (GPLv2+ or Artistic) and Copyright Only and MIT and Public Domain and UCD	(*02)(*04)(*73)
passwd	BSD or GPL+	(*02)
perl-File-Path	GPL+ or Artistic	(*02)
perl-threads-shared	GPL+ or Artistic	(*02)
perl-Time-HiRes	GPL+ or Artistic	(*02)
perl-threads	GPL+ or Artistic	(*02)
perl-Encode	GPL+ or Artistic	(*02)
perl-podlators	GPL+ or Artistic	(*02)
polkit-pkla-compat	LGPLv2+	(*03)
policycoreutils	GPLv2	(*01)
qrencode-libs	LGPLv2+	(*03)
rcs	GPLv3+	(*03)
readline	GPLv3+	(*03)
rootfiles	Public Domain	–
redhat-rpm-config	GPL+	(*02)
rpm-python	GPLv2+	(*02)
rpcbind	BSD	(*72)
rpm-sign	GPLv2+	(*01)
rpm-libs	GPLv2+ and LGPLv2+ with exceptions	(*01)
rpm-build-libs	GPLv2+ and LGPLv2+ with exceptions	(*01)
rsync	GPLv3+	(*03)
rsyslog	(GPLv3+ and ASL 2.0)	(*04)(*06)
rpm-build	GPLv2+	(*01)
rpm	GPLv2+	(*01)
setserial	GPL+	(*02)
setup	Public Domain	(*02)
slang	GPLv2+	(*01)
sudo	ISC	(*12)
systemd-sysv	LGPLv2+	(*03)
sqlite	Public Domain	–
selinux-policy-targeted	GPLv2+	(*01)
systemtap-client	GPLv2+	(*01)

Table B-1 Packages and Licenses (Cont'd)

Package Name	License	License Reference No.
systemd	LGPLv2+ and MIT and GPLv2+	(*01)(*03)
sed	GPLv3+	(*03)
smartmontools	GPLv2+	(*01)
systemtap	GPLv2+	(*01)
shared-mime-info	GPLv2+	(*01)
shadow-utils	BSD and GPLv2+	(*72)(*01)
selinux-policy	GPLv2+	(*01)
systemtap-runtime	GPLv2+	(*01)
subversion-libs	ASL 2.0	(*06)
subversion	ASL 2.0	(*06)
sudo	ISC	(*12)
swig	GPLv3+ and BSD	(*48)
snappy	BSD	(*47)
systemd-libs	LGPLv2+ and MIT	(*03)(*73)
systemtap-devel	GPLv2+	(*01)
systemtap	GPLv2+	(*01)
sysvinit-tools	GPLv2+	(*01)
tcp_wrappers-libs	BSD	(*49)
telnet	BSD	(*72)
tzdata	Public Domain	–
trousers	BSD	(*60)
tar	GPLv3+	(*03)
tuned	GPLv2+	(*01)
tcpdump	BSD with advertising	(*50)
TinyXML 2.6.2		(*27)
unzip	BSD	(*61)
ustr	MIT or LGPLv2+ or BSD	(*73)(*03)(*72)
util-linux	GPLv2 and GPLv2+ and LGPLv2+ and BSD with advertising and Public Domain	(*01)(*03)(*72)
vsftpd	GPLv2 with exceptions	(*74)
virt-what	GPLv2+	(*01)
vim-minimal	Vim	–
VXI-11 1.12		(*28)
which	GPLv3	(*03)
xz-libs	LGPLv2+	(*01)(*02)(*03)
xz	LGPLv2+	(*01)(*02)(*03)
xfsprogs	GPL+ and LGPLv2+	(*01)(*02)
yum-metadata-parser	GPLv2	(*01)
yum-plugin-fastestmirror	GPLv2+	(*01)
yum	GPLv2+	(*01)
zip	zlib and Boost	(*71)
zlib	BSD	(*72)

Appendix C Sample format for performance test result sheet

Use the following test result sheet when performing the MU887002A performance tests. Duplicate this sheet as necessary for tests.

C.1 Test Condition..... C-2
C.2 VSWR..... C-3
C.3 SG Level Accuracy C-4
C.4 SG Spurious C-11
C.5 SA Level Accuracy..... C-13
C.6 SA Linearity C-17
C.7 Pathloss Measurement Accuracy (MX887092A)..... C-20

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 vary depending on the option and port.
When using the result sheet, fill in the TRX number and the port number to be measured.

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
Without MU887002A-007 TRX 1/2 Port 1 to 12 With MU887002A-007 TRX 1/2 Port 1 to 4	Table C.2-2
With MU887002A-007 TRX 1/2 Port 5 to 12	Table C.2-3

Table C.2-2 VSWR

Test port: TRX Port

Frequency (MHz) f	Worst VSWR Frequency (MHz)	Worst VSWR	Spec.
400 ≤ f < 450			< 1.4
450 ≤ f ≤ 2700			< 1.3
2700 < f ≤ 3800			< 1.4
3800 < f ≤ 6000			< 1.4

Table C.2-3 VSWR

Test port: TRX Port

Frequency (MHz) f	Worst VSWR Frequency (MHz)	Worst VSWR	Spec.
400 ≤ f < 450*1			< 1.4
450 ≤ f ≤ 2700*1			< 1.3
2700 < f ≤ 3800*1			< 1.4
3800 < f ≤ 6000*1			< 1.4
5800 ≤ f ≤ 7400*2			< 1.7

*1: setting frequency < 5900 MHz
*2: setting frequency ≥ 5900 MHz

C.3 SG Level Accuracy

Performance test result sheets of SG Level Accuracy vary depending on the option, port, and frequency.
When using the result sheet, fill in the TRX number and the port number to be measured.

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
Without MU887002A-007 TRX 1/2 Port 1 to 12
With MU887002A-007 TRX 1/2 Port 1 to 4

Frequency*1 (MHz)	MU887002A Setting Frequency*2 (MHz)	TRX 1/2 Port 1 to Port 12
400	400.001	Table C.3-3
1318	1318.001	Table C.3-3
2236	2236.001	Table C.3-3
3154	3154.001	Table C.3-3
3800	3800.001	Table C.3-4
4072	4072.001	Table C.3-4
4990	4990.001	Table C.3-4
5908	5908.001	Table C.3-4
6000	5999.999	Table C.3-4

Table C.3-2 Measurement condition and the table number of result sheet
With MU887002A-007 TRX 1/2 Port 5 to 12

Frequency*1 (MHz)	MU887002A Setting Frequency*2 (MHz)	TRX 1/2 Port 1 to Port 12
400	400.001	Table C.3-3
1318	1318.001	Table C.3-3
2236	2236.001	Table C.3-3
3154	3154.001	Table C.3-3
3800	3800.001	Table C.3-4
4072	4072.001	Table C.3-4
4990	4990.001	Table C.3-4
5908	5908.001	Table C.3-5
6818	6818.001	Table C.3-5
7300	7299.999	Table C.3-5

*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 (MU887002A Settings - Frequency) in the performance test result sheet

Table C.3-3 SG Level Accuracy

Test port: TRX Port Freq. ≤ 3800 MHz

Frequency (MHz)	MU887002A Settings		Power Meter	Signal Analyzer					MU887002A	
	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)
		–5 = P1		#1	F0		= B1		= (B1 - A + C) - P1	±1.0
		–10 = P2					= B2		= (B2 - A + C) - P2	±1.0
		–15 = P3					= B3		= (B3 - A + C) - P3	±1.0
		–20 = P4					= B4		= (B4 - A + C) - P4	±1.0
		–22 = P5	= C				= A = B5		= 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

Appendix

Table C.3-3 SG Level Accuracy (Cont'd)

Frequency (MHz)	MU887002A Settings		Power Meter	Signal Analyzer					MU887002A	
	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)
	= F0	–55 = P12				= D	= G12	= B12 = G12 + D	= (B12 - A + C) - P12	±1.0
		–60 = P13					= G13	= B13 = G13 + D	= (B13 - A + C) - P13	±1.0
		–65 = P14		#2	F0		= 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
		–110 = P23					= G23	= B23 = G23 + D	= (B23 - A + C) - P23	±1.0
		–115 = P24		#5			= G24	= B24 = G24 + D	= (B24 - A + C) - P24	±1.0
		–120 = P25				= D	= G25	= B25 = G25 + D	= (B25 - A + C) - P25	±1.0

Table C.3-4 SG Level Accuracy

Test port: TRX Port 3800 MHz < Freq.

Frequency (MHz)	MU887002A Settings		Power Meter	Signal Analyzer					MU887002A	
	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		#1	F0			B1	= (B1 - A + C) - P1	±1.3
		–10 = P2						= B2	= (B2 - A + C) - P2	±1.3
		–15 = P3						= B3	= (B3 - A + C) - P3	±1.3
		–20 = P4						= B4	= (B4 - A + C) - P4	±1.3
		–22 = P5	= C					= A = B5	= 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.3-4 SG Level Accuracy (Cont'd)

Frequency (MHz)	MU887002A Settings		Power Meter	Signal Analyzer					MU887002A	
	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

Table C.3-5 SG Level Accuracy

Test port: TRX Port 5900 MHz ≤ Freq.

Frequency (MHz)	MU887002A Settings		Power Meter	Signal Analyzer					MU887002A	
	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	±1.3
		–15 = P2					= B2		= (B2 - A + C) - P2	±1.3
		–20 = P3					= B3		= (B3 - A + C) - P3	±1.3
		–22 = P4	= C				= A = B4		= C - P4	±1.3
		–25 = P5					= B5		= (B5 - A + C) - P5	±1.3
		–30 = P6					= B6		= (B6 - A + C) - P6	±1.3
		–35 = P7					= B7		= (B7 - A + C) - P7	±1.3
		–40 = P8					= B8		= (B8 - A + C) - P8	±1.3
		–45 = P9					= B9		= (B9 - A + C) - P9	±1.3
	= F0	–50 = P10		#2		= D	= G10	= B10 = G10 + D	= (B10 - A + C) - P10	±1.3

Appendix C

Table C.3-5 SG Level Accuracy (Cont'd)

Frequency (MHz)	MU887002A Settings		Power Meter	Signal Analyzer					MU887002A	
	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 = P11		#2	F0		= G11	= B11 = G11 + D	= (B11 - A + C) - P11	±1.3
		–60 = P12					= G12	= B12 = G12 + D	= (B12 - A + C) - P12	±1.3
		–65 = P13					= G13	= B13 = G13 + D	= (B13 - A + C) - P13	±1.3
		–70 = P14					= G14	= B14 = G14 + D	= (B14 - A + C) - P14	±1.3
		–75 = P15					= G15	= B15 = G15 + D	= (B15 - A + C) - P15	±1.3
		–80 = P16		#3			= G16	= B16 = G16 + D	= (B16 - A + C) - P16	±1.3
		–85 = P17					= G17	= B17 = G17 + D	= (B17 - A + C) - P17	±1.3
		–90 = P18					= G18	= B18 = G18 + D	= (B18 - A + C) - P18	±1.3
		–95 = P19					= G19	= B19 = G19 + D	= (B19 - A + C) - P19	±1.3
		–100 = P20					= G20	= B20 = G20 + D	= (B20 - A + C) - P20	±1.3

C.4 SG Spurious

Performance test result sheets of SG Spurious vary depending on the option and port.
When using the result sheet, fill in the TRX number and the port number to be measured.

The value to write in column of "Output Level" is –5 dBm.

Table C.4-1 SG Spurious
Without MU887002A-007 TRX 1/2 Port 1 to 12
With MU887002A-007 TRX 1/2 Port 1 to 4

Test port: TRX Port

MU887002A 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	
	400						< –25
	600						< –25
	850						< –25
	900						< –25
	1000						< –25
	1500						< –25
	1800						< –25
	1950						< –25
	2200						< –25
	2450						< –25
	2650						< –25
	3000						< –25

Table C.4-2 SG Spurious
With MU887002A-007 TRX 1/2 Port 5 to 12

Test port: TRX Port

MU887002A 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	
	400						< –25
	600						< –25
	850						< –25
	900						< –25
	1000						< –25
	1500						< –25
	1800						< –25
	1950						< –25
	2200						< –25
	2450						< –25
	2650						< –25
	3000						< –25
	3300						< –25
	3650						< –25

C.5 SA Level Accuracy

Performance test result sheets of SA Level Accuracy vary depending on the option, port, and frequency.
When using the result sheet, fill in the TRX number and the port number to be measured.

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
Without MU887002A-007 TRX 1/2 Port 1 to 12
With MU887002A-007 TRX 1/2 Port 1 to 4

MU887002A Setting Frequency (MHz)	Signal Generator Setting Frequency (MHz)	Table number
1318	1317.9	Table C.5-3
2236	2235.9	Table C.5-3
3154	3153.9	Table C.5-3
3800	3799.9	Table C.5-3
4072	4071.9	Table C.5-4
4990	4989.9	Table C.5-4
6000	5999.9	Table C.5-4

Table C.5-2 Measurement condition and the table number of result sheet
With MU887002A-007 TRX 1/2 Port 5 to 12

MU887002A Setting Frequency (MHz)	Signal Generator Setting Frequency (MHz)	Table number
1318	1317.9	Table C.5-3
2236	2235.9	Table C.5-3
3154	3153.9	Table C.5-3
3800	3799.9	Table C.5-3
4072	4071.9	Table C.5-4
4990	4989.9	Table C.5-4
5900	5900.1	Table C.5-5
6818	6817.9	Table C.5-5
7300	7299.9	Table C.5-5

Table C.5-3 SA Level Accuracy (400 ≤ f ≤ 3800 MHz)

Test port: TRX Port

MU887002A Settings		Signal Generator Settings		Power Meter	Signal Analyzer	MU887002A		
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. 0 dBm		= 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. -30 dBm		= A14	= A14 - C30	±0.5
			= P25	= C25 = approx. -25 dBm	= S			
	–35		P25 – 10		= S1	= A15	= A15 - (C25 + S1 - S)	±0.7
	–40					= A16	= A16 - (C25 + S1 - S)	±0.7
	–45					= A17	= A17 - (C25 + S1 - S)	±0.7
	–50					= A18	= A18 - (C25 + S1 - S)	±0.7
	–55					= A19	= A19 - (C25 + S1 - S)	±0.7
	–60					= A20	= A20 - (C25 + S1 - S)	±0.9
	–65					= A21	= A21 - (C25 + S1 - S)	±0.9

Table C.5-4 SA Level Accuracy (3800 < f ≤ 6000 MHz)

Test port: TRX Port

MU887002A Settings		Signal Generator Settings		Power Meter	Signal Analyzer	MU887002A		
Frequency (MHz)	Input Level (dBm)	Frequency (MHz)	Level (dBm)	Level (dBm)	Level (dBm)	Level (dBm)	Accuracy (dB)	Spec. (dB)
	+35					= A1	= A1 - C0	±0.7
	+30					= A2	= A2 - C0	±0.7
	+25					= A3	= A3 - C0	±0.7
	+20					= A4	= A4 - C0	±0.7
	+15					= A5	= A5 - C0	±0.7
	+10					= A6	= A6 - C0	±0.7
	+5					= A7	= A7 - C0	±0.7
	0		= P0	= C0 = approx. 0 dBm		= A8	= A8 - C0	±0.7
	–5					= A9	= A9 - C30	±0.7
	–10					= A10	= A10 - C30	±0.7
	–15					= A11	= A11 - C30	±0.7
	–20					= A12	= A12 - C30	±0.7
	–25					= A13	= A13 - C30	±0.7
	–30		= P30	= C30 = approx. -30 dBm		= A14	= A14 - C30	±0.7
			= P25	= C25 = approx. -25 dBm	= S			
	–35		P25 – 10			= A15	= A15 - (C25 + S1 - S)	±0.9
	–40					= A16	= A16 - (C25 + S1 - S)	±0.9
	–45					= A17	= A17 - (C25 + S1 - S)	±0.9
	–50					= A18	= A18 - (C25 + S1 - S)	±0.9
	–55					= A19	= A19 - (C25 + S1 - S)	±0.9
	–60					= A20	= A20 - (C25 + S1 - S)	±1.1
	–65				= S1	= A21	= A21 - (C25 + S1 - S)	±1.1

Table C.5-5 SA Level Accuracy (5900 < f ≤ 7300 MHz)

Test port: TRX Port

MU887002A Settings		Signal Generator Settings		Power Meter	Signal Analyzer	MU887002A		
Frequency (MHz)	Input Level (dBm)	Frequency (MHz)	Level (dBm)	Level (dBm)	Level (dBm)	Level (dBm)	Accuracy (dB)	Spec. (dB)
	+30					= A1	= A1 - C0	±1.0
	+25					= A2	= A2 - C0	±1.0
	+20					= A3	= A3 - C0	±1.0
	+15					= A4	= A4 - C0	±1.0
	+10					= A5	= A5 - C0	±1.0
	+5					= A6	= A6 - C0	±1.0
	0		= P0	= C0 = approx. 0 dBm		= A7	= A7 - C0	±1.0
	–5					= A8	= A8 - C0	±1.0
	–10					= A9	= A9 - C30	±1.0
	–15					= A10	= A10 - C30	±1.0
	–20					= A11	= A11 - C30	±1.0
	–25					= A12	= A12 - C30	±1.0
	–30		= P30	= C30 = approx. -30 dBm		= A13	= A13 - C30	±1.0
			= P25	= C25 = approx. -25 dBm	= S			
	–35					= A14	= A14 - (C25 + S1 - S)	±1.2
	–40					= A15	= A15 - (C25 + S1 - S)	±1.2
	–45					= A16	= A16 - (C25 + S1 - S)	±1.2
	–50		P25 – 10			= A17	= A17 - (C25 + S1 - S)	±1.2
	–55					= A18	= A18 - (C25 + S1 - S)	±1.2
	–60					= A19	= A19 - (C25 + S1 - S)	±1.8
	–65				= S1	= A20	= A20 - (C25 + S1 - S)	±1.8

C.6 SA Linearity

Performance test result sheets of SA Linearity vary depending on the option, port, and frequency.
When using the result sheet, fill in the TRX number and the port number to be measured.

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
Without MU887002A-007 TRX 1/2 Port 1 to 12
With MU887002A-007 TRX 1/2 Port 1 to 4

MU887002A Setting Frequency (MHz)	Signal Generator Setting Frequency (MHz)	Input Level (dBm)	Table number
1318	1317.9	0	Table C.6-3
2236	2235.9	0	Table C.6-3
3154	3153.9	0	Table C.6-3
4072	4071.9	0	Table C.6-3
4990	4989.9	0	Table C.6-3
6000	5999.9	0	Table C.6-3
1318	1317.9	–25	Table C.6-4
2236	2235.9	–25	Table C.6-4
3154	3153.9	–25	Table C.6-4
4072	4071.9	–25	Table C.6-4
4990	4989.9	–25	Table C.6-4
6000	5999.9	–25	Table C.6-4

Table C.6-2 Measurement condition and the table number of result sheet With MU887002A-007 TRX 1/2 Port 5 to 12

MU887002A Setting Frequency (MHz)	Signal Generator Setting Frequency (MHz)	Input Level (dBm)	Table number
1318	1317.9	0	Table C.6-3
2236	2235.9	0	Table C.6-3
3154	3153.9	0	Table C.6-3
4072	4071.9	0	Table C.6-3
4990	4989.9	0	Table C.6-3
5900	5900.1	0	Table C.6-3
6818	6817.9	0	Table C.6-3
7300	7299.9	0	Table C.6-3
1318	1317.9	–25	Table C.6-4
2236	2235.9	–25	Table C.6-4
3154	3153.9	–25	Table C.6-4
4072	4071.9	–25	Table C.6-4
4990	4989.9	–25	Table C.6-4
5900	5900.1	–25	Table C.6-4
6818	6817.9	–25	Table C.6-4
7300	7299.9	–25	Table C.6-4

Table C.6-3 SA Linearity (0 dBm)

Test port: TRX Port

MU887002A Settings		Signal Generator Settings		Power Meter	Signal Analyzer	MU887002A		
Frequency (MHz)	Input Level (dBm)	Frequency (MHz)	Level (dBm)	Level (dBm)	Level (dBm)	Level (dBm)	Accuracy (dB)	Spec. (dB)
	0		= P0	= C0 = approx. 0 dBm	= S	= A	–	–
			P0 – 10		= S1	= B1	= (B1 - A) - (S1 - S)	±0.2
			P0 – 20		= S2	= B2	= (B2 - A) - (S2 - S)	±0.2
			P0 – 30		= S3	= B3	= (B3 - A) - (S3 - S)	±0.2
			P0 – 40		= S4	= B4	= (B4 - A) - (S4 - S)	±0.2

Table C.6-4 SA Linearity (–25 dBm)

Test port: TRX Port

MU887002A Settings		Signal Generator Settings		Power Meter	Signal Analyzer	MU887002A		
Frequency (MHz)	Input Level (dBm)	Frequency (MHz)	Level (dBm)	Level (dBm)	Level (dBm)	Level (dBm)	Accuracy (dB)	Spec. (dB)
	–25		= P25	= C25 = approx. –25 dBm	= S	= A	–	–
			P25 – 10		= S1	= B1	= (B1 - A) - (S1 - S)	±0.2
			P25 – 20		= S2	= B2	= (B2 - A) - (S2 - S)	±0.2
			P25 – 30		= S3	= B3	= (B3 - A) - (S3 - S)	±0.2
			P25 – 40		= S4	= B4	= (B4 - A) - (S4 - S)	±0.4

C.7 Pathloss Measurement Accuracy (MX887092A)

Table C.7-1 Pathloss Measurement Accuracy (MX887092A)

Test port: TRX Port 11

Frequency (MHz)	Accuracy (dB)	Spec. (dB)
400		±0.5
1852.5		±0.5
2700		±0.5
3300		±0.9
4920		±0.9
6000		±0.9

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:	In the process of starting up
	Orange blink:	Unexpected error occurred.*1
	Green:	Normal*2
	Green blink:	In the process of starting up, powering off, calibration
Remote	Green:	Remote controlled
	Green blink:	Error occurred by the remote controlling.*3
	Off:	Not remote controlled.
1	Orange blink:	Updating software*1
	Off:	No error
2 (for VSG)	Red:	Error occurred when loading Waveform file.
	Green:	RF signal output is on.
	Orange:	An external loss correction error occurred. (RF output signals are lower than the specified level.)
	Orange blink:	Updating software*1, Loading the waveform file.
	Off:	RF signal output is off.
3 (for VSA)	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
	Orange blink:	Updating software*1
	Off:	No error

*1: The module cannot be controlled via remote interface.

*2: Even if the lamp lights green after the power is turned on, the module may not be controlled remotely for about one minute. When remote control is enabled, the error lamp (Error) on MT8870A stops flashing in red and lights off. If the beep setting is on, a beeping sound will also be made at the 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.

Notes:

- When the power is supplied, all the lamps are lit orange. However, sometimes the lamp 3 of TRX 2 lights green, this is not an error.
- When the power is turned off, all the lamps are lit orange. However, after all the lamps lit off, some lamps may light red, this is not an 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 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 lamp 3 is lit in red, 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, MX887018A, MX887019A, MX887065A, MX887067A, MX887068A	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?

Appendix E MX887092A

This appendix describes the MX887092A Pathloss Measurement Function.

E.1 Outline E-2
E.1.1 Overview E-2
E.1.2 Configuration E-3
E.1.3 Specifications E-3
E.2 License Registration/Viewing E-5
E.3 Pathloss Measurement E-6
E.3.1 Measurement E-7
E.3.2 Setting Pathloss value E-9
E.3.3 Example of Use E-10
E.4 SCPI Command E-12
E.4.1 List of Commands E-13
E.4.2 Details of Commands E-15
E.5 Native Command E-29
E.5.1 List of Commands E-30

E.1 Outline

E.1.1 Overview

Function

The MX887092A Pathloss Measurement Function provides the function to measure the path loss value from the test port end of the MU887002A to the connection end of the DUT by connecting the Open and Short devices alternately to the DUT and measuring the reflection signals from them.

The measured pathloss value can be used as loss correction data in 3.5.3 ”Correcting cable losses".

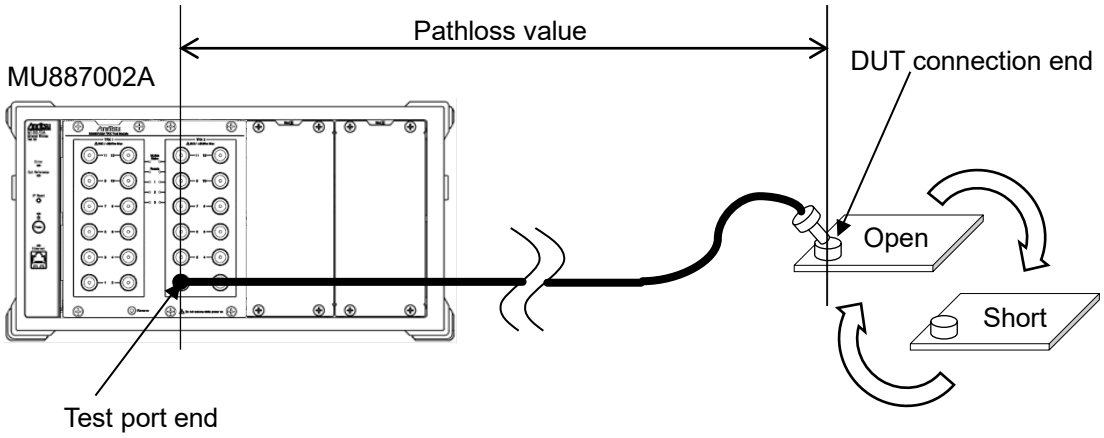


Figure E.1.1-1 Connection example

Note:

You need to arrange the Open and Short devices by yourself.

E.1.2 Configuration

The following table lists the MX887092A configuration.

Table E.1.2-1 Configuration

Model	Name	Remarks
MX887092A	Pathloss Measurement Function for MU887002A	MU887000A/01A is not supported.

E.1.3 Specifications

The following specification values are those under the conditions after 30-min warm-up at stable ambient temperature.

(Typ.): Typical indicates the reference data at 20 to 30°C and is nota guaranteed value.

(Nom): Nominal indicates the design value and is not guaranteed.

Table E.1.3-1 Specifications

Item	Specifications
Frequency range	400 MHz ≤ setting frequency ≤ 6000 MHz
Functions	Pathloss measurement using the MU887002A is possible by installing it on the MT8870A Universal Wireless Test Set or MT8872A Universal Wireless Test Set.
Application	MU887002A TRX 1/2 Port 1 to 12
Measurement Range	0 dB ≤ Pathloss ≤ 15 dB 1 m ≤ route length ≤ 3 m (When using a coaxial cable with a relative permittivity of 2.03) Refer to Figure E.1.3-1, Figure E.1.3-2, Figure E.1.3-3.
Measurement Accuracy	400 MHz ≤ setting frequency ≤ 2700 MHz, 11 moving average intervals, route length 1.5 m, a coaxial cable with a relative permittivity of 2.03 ±0.3 dB (Typ.) ±0.5 dB 3300 MHz ≤ setting frequency ≤ 6000 MHz, 11 moving average intervals, route length 1.5 m, a coaxial cable with a relative permittivity of 2.03 ±0.6 dB (Typ.) ±0.9 dB
Measurement reproducibility	At Pathloss ≤ 6 dB, 11 moving average intervals ±0.1 dB (Nom.) At Pathloss ≤ 15 dB, 11 moving average intervals ±0.15 dB (Nom.)

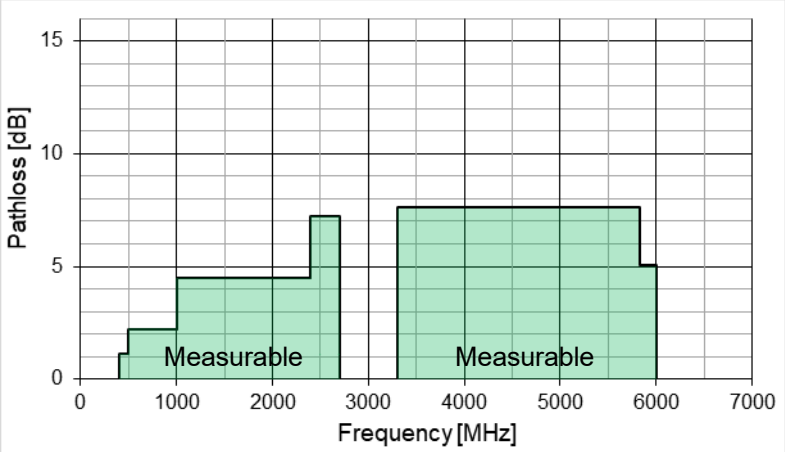


Figure E.1.3-1 Measurable Range (Route length 1 m)

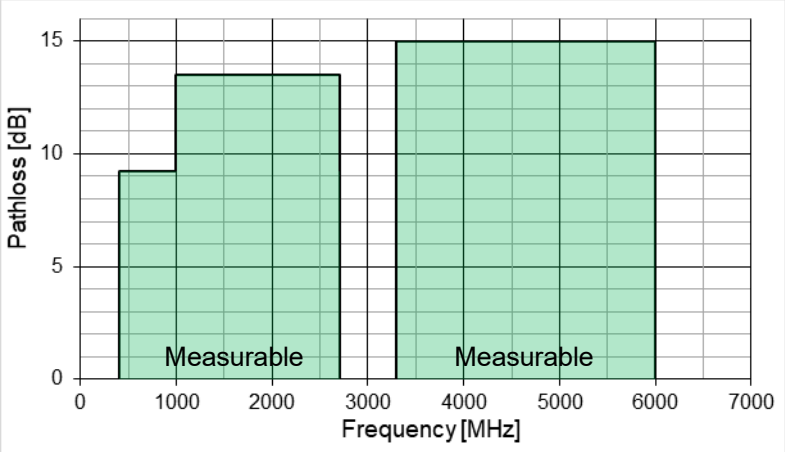


Figure E.1.3-2 Measurable Range (Route length 1.5 m)

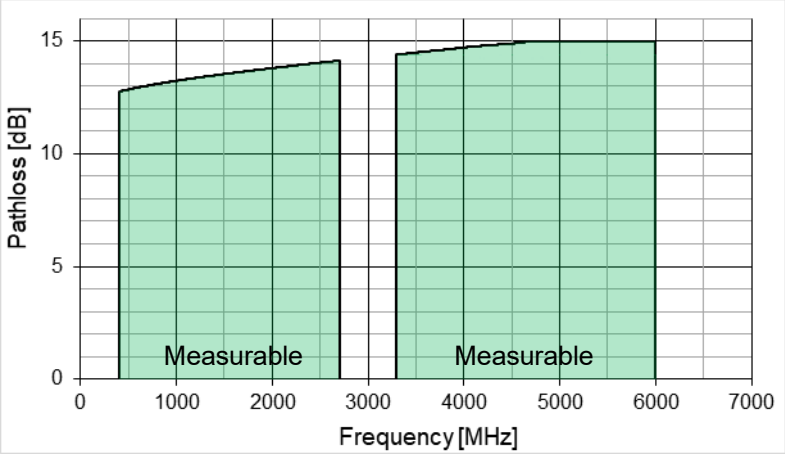


Figure E.1.3-3 Measurable Range (Route length 3 m)

E.2 License Registration/Viewing

License Registration

Before the MX887092A Pathloss Measurement Function can be used, the software license must be registered in the MT8870A.
Refer to Chapter 8 “Utility Tool” in the MU887002A TRX Test Module Operation Manual for the license registration procedure.

Note:

MX887092A is available only for MU887002A.
The supported frequency range is 400 MHz to 6 GHz even for MU887002A with the MU887002A-007 7 GHz Extension Function.

Viewing license

Refer to the commands on Section 5, 6

- MX887092A software license

```
MFINFO?
:SYSTem:INFormation:MAINframe?
```

UtilityTool — Refer to 8.3.4 “Viewing information on devices”

- MX887092A expiry date of the software license

```
KEYINFO?
:SYSTem:BASE:LICense:INFormation?
```

E.3 Pathloss Measurement

The MX887092A Pathloss Measurement Function provides the function to measure the path loss value from the test port end of the MU887002A to the connection end of the DUT by connecting the Open and Short devices alternately to the DUT and measuring the reflection signals from them.

⚠ CAUTION

Warm-up the MU887002A 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.

E.3.1 Measurement

Perform measurement by sending a remote command from the control PC.
Up to eight path indexes can be configured for each test port.

Checking the correction version and inspection version

- Utility > Pathloss Information

  Refer to 8.3.4 “Viewing information on devices”

  Check the correction version and inspection version.
- Correction version or supported frequency range of the calibration value

  `SYST:INF:CAL:PLOS:CORR:VERS?`

  `:SYSTem:INFormation:CALibration:PLOSs:CORRection:VERSion?`
- Inspection version of the calibration value

  `SYST:INF:CAL:PLOS:CORR:INSP:VERS?`

  `:SYSTem:INFormation:CALibration:PLOSs:CORRection:INSPection:VERSion?`

Settings before measurement

- Clearing pathloss values and setting

  `INIT:CAL:MEAS:PLOS:CLE`

  `:INITiate:CALibration:MEASurement:PLOSs:CLEar`
- Pathloss measurement frequency

  `CONF:CAL:MEAS:PLOS:MPAT:FREQ`

  `:CONFigure:CALibration:MEASurement:PLOSs:MPATh:FREQuency`
- Pathloss measurement frequency of Path index 1

  `CONF:CAL:MEAS:PLOS:FREQ`

  `:CONFigure:CALibration:MEASurement:PLOSs:FREQuency`
- Number of intervals for moving average in pathloss calculation.

  `CONF:CAL:MEAS:PLOS:AFT`

  `:CONFigure:CALibration:MEASurement:PLOSs:AFTaps`

Measurement

- Pathloss measurement with the Open device

  `INIT:CAL:MEAS:PLOS:OPEN`

  `:INITiate:CALibration:MEASurement:PLOSs:OPEN`
- Pathloss measurement with the Short device

  `INIT:CAL:MEAS:PLOS:SHOR`

  `:INITiate:CALibration:MEASurement:PLOSs:SHORt`
- Pathloss measurement status with the Open device

  `FETC:CAL:MEAS:PLOS:OPEN?`

  `:FETCh:CALibration:MEASurement:PLOSs:OPEN?`
- Pathloss measurement status with the Short device

  `FETC:CAL:MEAS:PLOS:SHOR?`

  `:FETCh:CALibration:MEASurement:PLOSs:SHORt?`

Calculation

- Calculating the pathloss value
  `INIT:CAL:MEAS:PLOS:EVAL`
  `:INITiate:CALibration:MEASurement:PLOSs:EVALuate`
- Pathloss calculation status
  `FETC:CAL:MEAS:PLOS:EVAL?`
  `:FETCh:CALibration:MEASurement:PLOSs:EVALuate?`
- Frequency of pathloss calculation result
  `FETC:CAL:MEAS:PLOS:EVAL:FREQ?`
  `:FETCh:CALibration:MEASurement:PLOSs:EVALuate:FREQuency?`
- Pathloss calculation result (Gain)
  `FETC:CAL:MEAS:PLOS:EVAL:GAIN?`
  `:FETCh:CALibration:MEASurement:PLOSs:EVALuate:GAIN?`

E.3.2 Setting Pathloss value

The measured pathloss values can be set as loss correction data.
Refer to 3.5.3 “Correcting cable losses” for details

Notes:

- Since the pathloss value is a gain, its sign is reversed to use it as a correction value.
- The same loss correction value is used for DL (Rx) and UL (Tx).

E.3.3 Example of Use

The table below shows the procedure for measuring path loss and calculating the pathloss value.
Port 11 and pathloss index 1 are used as examples.

Table E.3.5-1 Pathloss measurement procedure

No.	Procedure	Command
1	Set up the instruments.	
2	Clear (initialize) pathloss values and settings.	:INIT:CAL:MEAS:PLOS:CLE
3	Set the number of intervals for moving average used in the pathloss calculation. (e.g. 15)	:CONF:CAL:MEAS:PLOS:AFT 15
4	Set the pathloss measurement frequency. (e.g. 3 points of 1000 MHz, 2000 MHz, 3000 MHz)	:CONF:CAL:MEAS:PLOS:MPAT:FREQ PORT11,1,3,1000MHZ,2000MHZ,3000MHZ
5	Attach the Open device to the DUT connection end where to be connected to Port 11.	
6	Perform a pathloss measurement with Open device.	:INIT:CAL:MEAS:PLOS:OPEN PORT11,1
7	Wait for the measurement to finish.	*OPC?
8	Check that the pathloss measurement with Open device has completed successfully.	:FETC:CAL:MEAS:PLOS:OPEN? PORT11,1
9	Repeat steps 4 to 8 for Ports 1 to 10 and 12.	
10	Attach the Short device to the DUT connection end where to be connected to Port 11.	
11	Perform a pathloss measurement with Short device.	:INIT:CAL:MEAS:PLOS:SHOR PORT11,1
12	Wait for the measurement to finish.	*OPC?
13	Check that the pathloss measurement with Short device has completed successfully.	:FETC:CAL:MEAS:PLOS:SHOR? PORT11,1
14	Repeat steps 10 to 13 for Ports 1 to 10 and 12.	
15	Calculate the pathloss value.	:INIT:CAL:MEAS:PLOS:EVAL PORT11,1
16	Check that the pathloss calculation has finished successfully.	:FETC:CAL:MEAS:PLOS:EVAL? PORT11,1
17	Query the frequency of the pathloss calculation result.	:FETC:CAL:MEAS:PLOS:EVAL:FREQ? PORT11,1
18	Query the pathloss calculation result (Gain).	:FETC:CAL:MEAS:PLOS:EVAL:GAIN? PORT11,1
19	Repeat steps 15 to 18 for Ports 1 to 10 and 12.	

Table E.3.5-1 Setting Pathloss measurement result

No.	Procedure	Command
1	Set the language mode (SCPI).	:SYST:LANG SCPI
2	Set the Loss correction table index. e.g. Correction table index 1	:CALC:EXTL:TABL:SETT 1
3	Delete correction value set in the loss correction table.	:CALC:EXTL:TABL:DEL
4	Set the frequency and pathloss calculation result (Gain) into the loss correction table. Make sure to reverse the sign of the pathloss calculation result (Gain) to save it into the loss correction table. The same loss correction value is used for DL (Rx) and UL (Tx). An arbitrary correction value can be set for the ports not set. e.g. Port: 1 Frequency: 701000000 Hz Gain: −0.21	:CALC:EXTL:TABL:VAL:ALL 701000000,0.21,0.21,0,0,0,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0
5	Perform step 4 at all frequencies in the path loss calculation results.	
6	Enable the loss correction table.	:ROUT:EXTL:TABL:SWIT ON

E.4 SCPI Command

This section describes the details of SCPI commands for the pathloss measurement.
To switch to the SCPI command mode, send the command SYST:LANG SCPI.

The following table shows the rules for describing messages.

[ ]	Messages or parameters in square brackets can be omitted.
|	Choose one of several choices. A | B | C | D indicates a choice of A, B, C, and D.
{ }	Curly brackets are used to group options. For A | B({C | D}), choose A, B(C), or B(D).

E.4.1 List of Commands

Pathloss measurement

Function	Command	Query	Response
Clear pathloss data	:INITiate:CALibration:MEASurement:PLOSs:CLEar [<port>[,<pathindex>]]	-----	-----
Pathloss measurement frequency	:CONFigure:CALibration:MEASurement:PLOSs:MPATh:FREQuency <port>,<pathindex>,<numEntries>,<frequency_1>,…,<frequency_numentries>	:CONFigure:CALibration:MEASurement:PLOSs:MPATh:FREQuency? <port>,<pathindex>	<numentries>,<frequency_1>,<frequency_2>,…,<frequency_numentries>
Pathloss measurement frequency for path index 1	:CONFigure:CALibration:MEASurement:PLOSs:FREQuency <port>,<numentries>,<frequency_1>,…,<frequency_numentries>	:CONFigure:CALibration:MEASurement:PLOSs:FREQuency? <port>	<numentries>,<frequency_1>,<frequency_2>,…,<frequency_numentries>
Number of intervals for moving average	:CONFigure:CALibration:MEASurement:PLOSs:AFTaps <avgfiltertaps>	:CONFigure:CALibration:MEASurement:PLOSs:AFTaps?	<avgfilterTaps>
Pathloss measurement with Open	:INITiate:CALibration:MEASurement:PLOSs:OPEN <port>[, <pathindex>]	-----	-----
Pathloss measurement with Short	:INITiate:CALibration:MEASurement:PLOSs:SHORt <port>[, <pathindex>]	-----	-----

Appendix

Pathloss measurement

Function	Command	Query	Response
Pathloss measurement status with Open	-----	:FETCh:CALibration:MEASurement:PLOSs:OPEN? <port>[,<pathindex>]	<status>
Pathloss measurement status with Short	-----	:FETCh:CALibration:MEASurement:PLOSs:SHORt? <port>[,<pathindex>]	<status>
Calculate pathloss value	:INITiate:CALibration:MEASurement:PLOSs:EVALuate <Port>[, <pathindex>]	-----	-----
Pathloss calculation status	-----	:FETCh:CALibration:MEASurement:PLOSs:EVALuate? <port>[,<pathindex>]	<status>
Frequency of pathloss calculation result	-----	:FETCh:CALibration:MEASurement:PLOSs:EVALuate:FREQuency? <port>[,<pathindex>]	<numEntries>,<frequency_1>,<frequency_2>,…,<frequency_numentries>
Pathloss calculation result (Gain)	-----	:FETCh:CALibration:MEASurement:PLOSs:EVALuate:GAIN? <port>[,<pathindex>]	<numEntries>,<gain_1>,<gain_2>,…,<gain_numentries>
Correction version of calibration value	-----	:SYSTem:INFormation:CALibration:PLOSs:CORRection:VERSion? [<index>]	<ver> or <type>
Inspection version of calibration value	-----	:SYSTem:INFormation:CALibration:PLOSs:CORRection:INSPection:VERSion?	<ver>

E.4.2 Details of Commands

Refer to 5.2 “Details of Commands” for Conventions used in the command table and the suffix code list.

This section explains the SCPI commands in alphabetical order.

:CONFigure:CALibration:MEASurement:PLOSs:AFTaps

Number of intervals for moving average

Function

Sets or queries the number of intervals for moving average used for pathloss calculation.

Command

```
:CONFigure:CALibration:MEASurement:PLOSs:AFTaps <avgfiltertaps>
```

Query

```
:CONFigure:CALibration:MEASurement:PLOSs:AFTaps?
```

Response

```
<avgfiltertaps>
```

Parameter

<avgfiltertaps>	Number of intervals for moving average
Range	1 to 41
Default	11

Example of Use

To set the number of intervals to 1:

```
:CONF:CAL:MEAS:PLOS:AFT 1
:CONF:CAL:MEAS:PLOS:AFT?
>1
```

Remarks

If the MX887092A license is not found, a license error will occur.

:CONFigure:CALibration:MEASurement:PLOSs:FREQuency

Pathloss measurement frequency for path index 1

Function

Sets or queries the pathloss measurement frequencies for the path index 1.

Command

```
:CONFigure:CALibration:MEASurement:PLOSs:FREQuency
<port>,<numentries>,<frequency_1>,…,<frequency_numentries>
```

Query

```
:CONFigure:CALibration:MEASurement:PLOSs:FREQuency? <port>
```

Response

```
<numentries>,<frequency_1>,<frequency_2>,…,<frequency_numentries>
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<numentries>	Number of measurement frequencies
Range	1 to 1000
Default	0 (Not configurable)
<frequency_1>,<frequency_2>,…,<frequency_numentries>	Measurement frequency
Range	400 to 6000 MHz
Resolution	1 Hz
Response unit	Hz
Suffix code	HZ, KHZ, KZ, MHZ, MZ, GHZ, GZ, uses Hz when omitted.

Example of Use

To set seven pathloss measurement frequencies for Path index 1 of Test Port 1:

```
:CONF:CAL:MEAS:PLOS:FREQ
PORT1,7,411111111HZ,500000KHZ,600000KZ,700MHZ,800MZ,1GHZ,0.4GZ
:CONF:CAL:MEAS:PLOS:FREQ? PORT1
>7,411111111,500000000,600000000,700000000,800000000,1000000000,400000000
```

Related Command

```
:INITiate:CALibration:MEASurement:PLOSs:CLEar
:CONFigure:CALibration:MEASurement:PLOSs:MPATh:FREQuency
```

Remarks

If the MX887092A license is not found, a license error will occur.

:CONFigure:CALibration:MEASurement:PLOSs:MPATh:FREQuency

Pathloss measurement frequency

Function

Sets or queries the pathloss measurement frequency.

Command

```
:CONFigure:CALibration:MEASurement:PLOSs:MPATh:FREQuency
<port>,<pathindex>,<numentries>,<frequency_1>,…,<frequency_numentries>
```

Query

```
:CONFigure:CALibration:MEASurement:PLOSs:MPATh:FREQuency?
<port>,<pathindex>
```

Response

```
<numentries>,<frequency_1>,<frequency_2>,…,<frequency_numentries>
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
<numentries>	Number of measurement frequencies
Range	1 to 1000
Default	0 (Not configurable)
<frequency_1>,<frequency_2>,…,<frequency_numentries>	Measurement frequency
Range	400 to 6000 MHz
Resolution	1 Hz
Response unit	Hz
Suffix code	HZ, KHZ, KZ, MHZ, MZ, GHZ, GZ, uses Hz when omitted.

Example of Use

To set seven pathloss measurement frequencies for Path index 1 of Test Port 1:

```
:CONF:CAL:MEAS:PLOS:MPAT:FREQ
PORT1,1,7,411111111HZ,500000KHZ,600000KZ,700MHZ,800MZ,1GHZ,0.4GZ
:CONF:CAL:MEAS:PLOS:MPAT:FREQ? PORT1,1
>7,411111111,500000000,600000000,700000000,800000000,1000000000,400000000
```

Related Command

```
:INITiate:CALibration:MEASurement:PLOSs:CLEar
:CONFigure:CALibration:MEASurement:PLOSs:FREQuency
```

Remarks

If the MX887092A license is not found, a license error will occur.

:FETCh:CALibration:MEASurement:PLOSs:EVALuate?

Pathloss calculation status

Function

Queries the status of pathloss calculation.

Query

```
:FETCh:CALibration:MEASurement:PLOSs:EVALuate? <port>[,<pathindex>]
```

Response

```
<status>
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
Omitted	1
<status>	Status of pathloss calculation
0	In preparation
1	Normal termination
2	Inappropriate calculation result

Example of Use

To query the status of pathloss calculation for Path index 1 of Test Port 1:

```
:FETC:CAL:MEAS:PLOS:EVAL? PORT1,1
>1
```

Related Command

```
:INITiate:CALibration:MEASurement:PLOSs:EVALuate
```

Remarks

If the MX887092A license is not found, a license error will occur.

:FETCh:CALibration:MEASurement:PLOSs:EVALuate:FREQuency?

Frequency of pathloss calculation result

Function

Queries the frequency of pathloss calculation result.

Query

```
:FETCh:CALibration:MEASurement:PLOSs:EVALuate:FREQuency?
<port>[,<pathindex>]
```

Response

```
<numentries>,<frequency_1>,<frequency_2>,…,<frequency_numentries>
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
Omitted	1
<numentries>	Number of frequencies
Range	0 to 1000
Default	0
<frequency_1>,<frequency_2>,…,<frequency_numentries>	Measurement frequency
Range	400 000 000 to 6000 000 000
Response unit	Hz

Example of Use

To query the frequencies of the pathloss calculation result for Path index 1 of Test Port 1:

```
:FETC:CAL:MEAS:PLOS:EVAL:FREQ? PORT1,1
>7,411111111,500000000,600000000,700000000,800000000,1000000000,400000000
```

Related Command

```
:FETCh:CALibration:MEASurement:PLOSs:EVALuate?
:FETCh:CALibration:MEASurement:PLOSs:EVALuate:GAIN?
```

Remarks

If the MX887092A license is not found, a license error will occur.

:FETCh:CALibration:MEASurement:PLOSs:EVALuate:GAIN?

Pathloss calculation result (Gain)

Function

Queries the pathloss calculation result (Gain).

Query

```
:FETCh:CALibration:MEASurement:PLOSs:EVALuate:GAIN? <port>[,<pathindex>]
```

Response

```
<numentries>,<gain_1>,<gain_2>,…,<gain_numentries>
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
Omitted	1
<numentries>	
Range	0 to 1000
Default	0
<gain_1>,<gain_2>,…,<gain_numentries>	
	Pathloss value (Gain)
Range	–100.00 to 100.00
Response unit	dB

Details

The value queried by this command is the gain value. To use the queried value as cable loss correction data, its sign must be reversed.

Example of Use

To query the pathloss values for Path index 1 of Test Port 1:

```
:FETC:CAL:MEAS:PLOS:EVAL:GAIN? PORT1,1
>7,-0.26,-0.30,-0.30,-0.27,-0.26,-0.26,-0.29
```

Related Command

```
:FETCh:CALibration:MEASurement:PLOSs:EVALuate?
:FETCh:CALibration:MEASurement:PLOSs:EVALuate:FREQuency?
```

Remarks

If the MX887092A license is not found, a license error will occur.

:FETCh:CALibration:MEASurement:PLOSs:OPEN?

Pathloss measurement status with Open

Function

Queries the status of the pathloss measurement performed with the Open device.

Query

```
:FETCh:CALibration:MEASurement:PLOSs:OPEN? <port>[,<pathindex>]
```

Response

```
<status>
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
Omitted	1
<status>	Measurement status
0	In preparation
1	Normal termination

Example of Use

To query the status of the pathloss measurement, performed with the Open device, for Path index 1 of Test Port 1:

```
:FETC:CAL:MEAS:PLOS:OPEN? PORT1,1
>1
```

Related Command

```
:INITiate:CALibration:MEASurement:PLOSs:OPEN
```

Remarks

If the MX887092A license is not found, a license error will occur.

:FETCh:CALibration:MEASurement:PLOSs:SHORt?

Pathloss measurement status with Short

Function

Queries the status of the pathloss measurement performed with the Short device.

Query

```
:FETCh:CALibration:MEASurement:PLOSs:SHORt? <port>[,<pathindex>]
```

Response

```
<status>
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
Omitted	1
<status>	Measurement status
0	In preparation
1	Normal termination

Example of Use

To query the status of the pathloss measurement, performed with the Short device, for Path index 1 of Test Port 1:

```
:FETC:CAL:MEAS:PLOS:SHOR? PORT1,1
>1
```

Related Command

```
:INITiate:CALibration:MEASurement:PLOSs:SHORt
```

Remarks

If the MX887092A license is not found, a license error will occur.

:INITiate:CALibration:MEASurement:PLOSs:CLEar

Clear Pathloss data

Function

Clears (initializes) the pathloss values and settings

Command

```
:INITiate:CALibration:MEASurement:PLOSs:CLEar [<port>[,<pathindex>]]
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
Omitted	All Ports
<pathindex>	Path index
Range	1 to 8
Omitted	All path indexes

Details

Clears (initializes) the following data.

- Measurement frequency
- Pathloss measurement value/status with the Open device
- Pathloss measurement value/status with the Short device
- Pathloss calculation value/status

When all parameters are omitted, the following data is also initialized.

- The number of intervals for moving average

Example of Use

To clear the pathloss values and settings for Path index 1 of Test Port 1:

```
:INIT:CAL:MEAS:PLOS:CLE PORT1,1
```

Remarks

- The data described in Details cannot be cleared (initialized) by the *RST command.
- If the MX887092A license is not found, a license error will occur.

:INITiate:CALibration:MEASurement:PLOSs:EVALuate

Calculate pathloss value

Function

Calculates the pathloss value.

Command

```
:INITiate:CALibration:MEASurement:PLOSs:EVALuate <port>[,<pathindex>]
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
Omitted	1

Details

Before using this command, perform the following:

- Perform the pathloss measurement using the Open and Short devices for the path index of the port to be specified in this command.
- Set the measurement frequencies and the number of intervals for moving average.

Example of Use

To calculate the pathloss value for Path index 1 of Test Port 1:

```
:INIT:CAL:MEAS:PLOS:EVAL PORT1,1
```

Related Command

```
:FETCh:CALibration:MEASurement:PLOSs:EVALuate?
```

Remarks

- If the MX887092A license is not found, a license error will occur.
- When necessary data is not complete, an execution error occurs.

:INITiate:CALibration:MEASurement:PLOSs:OPEN

Pathloss measurement with Open

Function

Performs the pathloss measurement with the Open device.

Command

```
:INITiate:CALibration:MEASurement:PLOSs:OPEN <port>[,<pathindex>]
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
Omitted	1

Details

Before using this command, attach the Open device to the DUT connection end connected to the test port to be specified by the parameter.

Example of Use

To perform the pathloss measurement for Path index 1 of Test Port 1 with the Open device:

```
:INIT:CAL:MEAS:PLOS:OPEN PORT1,1
```

Related Command

```
:FETCh:CALibration:MEASurement:PLOSs:OPEN?
```

Remarks

- If the MX887092A license is not found, a license error will occur.
- If there is not enough free space on the waveform memory, an execution error will occur. In this case, delete unnecessary waveforms from the waveform memory. For how to check available waveform memory space and delete waveform files, refer to the following commands:

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

:INITiate:CALibration:MEASurement:PLOSs:SHORt

Pathloss measurement with Short

Function

Performs the pathloss measurement with the Short device.

Command

```
:INITiate:CALibration:MEASurement:PLOSs:SHORt <port>[,<pathindex>]
```

Parameter

<port>	Test port number
Range	Test Port 1 to 12
PORT1	Test Port 1
…	…
PORT12	Test Port 12
<pathindex>	Path index
Range	1 to 8
Omitted	1

Details

Before using this command, attach the Short device to the DUT connection end connected to the test port to be specified by the parameter.

Example of Use

To perform the pathloss measurement for Path index 1 of Test Port 1 with the Short device:

```
:INIT:CAL:MEAS:PLOS:SHOR PORT1,1
```

Related Command

```
:FETCh:CALibration:MEASurement:PLOSs:SHORt?
```

Remarks

- If the MX887092A license is not found, a license error will occur.
- If there is not enough free space on the waveform memory, an execution error will occur. In this case, delete unnecessary waveforms from the waveform memory. For how to check available waveform memory space and delete waveform files, refer to the following commands:

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

:SYSTem:INFormation:CALibration:PLOSs:CORRection:INSPection:VERSion?

Inspection version of calibration value

Function

Queries the inspection version of the calibration value.

Query

```
:SYSTem:INFormation:CALibration:PLOSs:CORRection:INSPection:VERSion?
```

Response

<ver> Inspection version

Parameter

None

Details

Queries the inspection version of the calibration value.

Example of Use

To query the inspection version of the calibration value:

```
:SYST:INF:CAL:PLOS:CORR:INSP:VERS?
> 0
```

Related Command

```
:SYSTem:INFormation:CALibration:PLOSs:CORRection:VERSion?
```

:SYSTem:INFormation:CALibration:PLOSs:CORRection:VERSion?

Correction version of calibration value

Function

Queries the correction version or the supported frequency range of the calibration value.

Query

```
:SYSTem:INFormation:CALibration:PLOSs:CORRection:VERSion? [<index>]
```

Response

When the parameter is 0 or omitted

<ver>	Correction version
Range	0 to 2147483647

When the parameter is 1

<type>	Supported frequency range
6GHZ	400 MHz to 6 GHz
6GHZ&7GHZ	400 MHz to 7.3 GHz (To be designed)

Parameter

<index>	Response type
0	Correction version
1	Supported frequency range
Omitted	0

Details

Queries the correction version of the calibration value.

Queries the supported frequency range of the calibration value.

Example of Use

To query the correction version of the calibration value:

```
:SYST:INF:CAL:PLOS:CORR:VERS?
> 0
```

To query the supported frequency range of the calibration value:

```
SYST:INF:CAL:PLOS:CORR:VERS? 1
> 6GHZ
```

Related Command

```
:SYSTem:INFormation:CALibration:PLOSs:CORRection:INSPection:VERSion?
```

E.5 Native Command

This section describes the details of Native commands for the pathloss measurement.
To switch to the Native command mode, send the command SYST:LANG NAT.

In the Native command mode, use the SCPI command in short format.

Example:
:INITiate:CALibration:MEASurement:PLOSs:CLEar
→ INIT:CAL:MEAS:PLOS:CLE
:CONFigure:CALibration:MEASurement:PLOSs:MPATh:FREQuency
→ CONF:CAL:MEAS:PLOS:MPAT:FREQ

The following table shows the rules for describing messages.

[ ]	Messages or parameters in square brackets can be omitted.
|	Choose one of several choices. A | B | C | D indicates a choice of A, B, C, and D.
{ }	Curly brackets are used to group options. For A | B({C | D}), choose A, B(C), or B(D).

E.5.1 List of Commands

Pathloss measurement

Function	Command	Query	Response
Clear pathloss data	INIT:CAL:MEAS:PLOS:CLE [port,[pathIndex]]	-----	-----
Pathloss measurement frequency	CONF:CAL:MEAS:PLOS:MPAT:FREQ port,pathIndex,numentries, frequency_1,…,frequency_numentries	CONF:CAL:MEAS:PLOS:MPAT:FREQ? port,pathindex	numentries,frequency_1,frequency_2,…,frequency_numentries
Pathloss measurement frequency for path index 1	CONF:CAL:MEAS:PLOS:FREQ port,numentries,frequency_1,…,frequency_numentries	CONF:CAL:MEAS:PLOS:FREQ? port	numentries,frequency_1,frequency_2,…,frequency_numentries
Number of intervals for moving average	CONF:CAL:MEAS:PLOS:AFT avgfiltertaps	CONF:CAL:MEAS:PLOS:AFT?	avgfilterTaps
Pathloss measurement with Open	INIT:CAL:MEAS:PLOS:OPEN port[,pathindex]	-----	-----
Pathloss measurement with Short	INIT:CAL:MEAS:PLOS:SHOR port[,pathindex]	-----	-----

Pathloss measurement

Function	Command	Query	Response
Pathloss measurement status with Open	-----	FETC:CAL:MEAS:PLOS:OPEN? port[,pathindex]	status
Pathloss measurement status with Short	-----	FETC:CAL:MEAS:PLOS:SHOR? port[,pathindex]	status
Calculate pathloss value	INIT:CAL:MEAS:PLOS:EVAL port[,pathindex]	-----	-----
Pathloss calculation status	-----	FETC:CAL:MEAS:PLOS:EVAL? port[,pathindex]	status
Frequency of pathloss calculation result	-----	FETC:CAL:MEAS:PLOS:EVAL:FREQ? port[,pathindex]	numEntries,frequency_1,frequency_2,…,frequency_numentries
Pathloss calculation result (Gain)	-----	FETC:CAL:MEAS:PLOS:EVAL:GAIN? port[,pathindex]	numEntries,gain_1,gain_2,…,gain_numentries
Correction version of calibration value	-----	SYST:INF:CAL:PLOS:CORR:VERS? [index]	ver or type
Inspection version of calibration value	-----	SYST:INF:CAL:PLOS:CORR:INSP:VERS?	ver



Index

References are page numbers.

A

Application Software 1-6

B

Batch Upgrade 8-33
bit definitions of the standard event status register 3-27
bit definitions of the status byte register ..3-25

C

Common parameters 4-19
configurations of registers 3-21
Configure Network 8-23
Confirming a Connection 2-7

D

Decimal Values Corresponding to Bits of Registers 3-23

E

electrostatic 2-9
Ethernet Default Settings 3-4

F

Front Panel 2-2
FTP 3-8

I

Initialization 3-3
Installing Module 2-5
IP Address Default Settings 3-4

L

Level accuracy 7-12, 7-25
Level calibration 3-15
Linearity 7-35
Log Operation 8-27
Loss Correction 3-17

M

Measurement Accuracy 7-43
Measuring Instruments for Performance Test 7-3

N

Native
Pathloss measurement E-30, E-31
Native command mode 3-2
Network settings 8-23
Normal mode 4-4

O

Operation Mode 4-3
operation status register 3-29
Optional Accessories 1-8

P

Pattern Gap 4-7
product information 3-11

Q

questionable status register 3-33

R

Register License 8-35
Removing Module 2-6

S

SCPI
Pathloss measurement E-13, E-14
SCPI command mode 3-2
Sequence Mode 4-17
Sequence Table Configuration 4-18
service request enable register 3-24
SG sequencer data 4-24
SG sequencer table 4-22
SG spurious 7-22
Software Licenses B-1
Specifications A-2, A-6
status byte register 3-24
Status indication of lamps D-1
Status Registers 3-21
System Settings 3-9

T

Transferring files 8-31

U

Upgrade.....8-28
Utility Tool..... 8-1
Compatible Versions 8-8
NI-VISA Version..... 8-8
Operating Environment..... 8-8

V

VSG 4-2
VSWR 7-8

W

Warranty Service..... 1-8
Waveform file 4-6
Waveform file configuration 4-7
Waveforms..... 1-7
Waveforms list table.....4-21