MeasureReady™ M81 Synchronous Source Measure System User’s Manual

Lake Shore Cryotronics, Inc.
575 McCorkle Blvd.
Westerville, Ohio
43082-8888 USA
Fax: (614) 891-1392
Telephone: (614) 891-2243

sales@lakeshore.com
support@lakeshore.com
www.lakeshore.com

Firmware version 2.0

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 Overview

The Lake Shore M81-SSM Synchronous Source Measure System is a modular system for applying voltage and current stimulus to experimental systems, as well as measuring voltages and currents produced by experimental systems.

This manual applies to 3 model numbers that support varying numbers of connected modules:

M81-SSM-2	M81-SSM Synchronous Source Measure System with 1 source channel and 1 measure channel
M81-SSM-4	M81-SSM Synchronous Source Measure System with 2 source channels and 2 measure channels
M81-SSM-6	M81-SSM Synchronous Source Measure System with 3 source channels and 3 measure channels

The MeasureReady™ M81-SSM provides a modular approach to multi-channel voltage/current sourcing and measuring of signals from DC to 100 kHz AC while also ensuring very tight timing synchronization between up to 6 total channels due to a unique synchronous sampling architecture called MeasureSync™ technology. This proprietary Lake Shore measurement architecture is a unique approach to ensuring that all source and measure signals remain inherently aligned in time with no user intervention required beyond selecting the desired signal types, amplitudes, frequencies and timing references for each source and measure channel in a given setup.

The compact instrument coordinates all source and measure channel updates to ensure precise timing alignment with the single main sampling system which forms the basis of the MeasureSync™ technology and architecture. In addition to the multi-channel timing synchronization benefits, MeasureSync™ technology also ensures consistent and near real time sampling performance across the M81-SSM's full DC to 100 kHz signal bandwidth and across all connected source and measure amplifier modules. And it does so regardless of the distance between instrument and local/benchtop, rack or remotely mounted signal amplifier modules, which can be located up to 2 m away (10 m optional) from the instrument mainframe, further adding flexibility to each test setup while maintaining full specified timing synchronization and signal sourcing and measurement performance.

The ability to locate source and measure signal amplifiers as close as possible to apparatus, samples or devices under test minimizes signal level cabling length and the associated cabling leakage, noise and increased settling times, any of which can be limiting factors for ultra-low signal levels transport and voltage/current characterization applications.

The M81-SSM architecture and unique technologies provide the configuration flexibility of a modular system without the usual mechanical or performance limitations inherent in card cage or other mainframe centric form product yielding significant cost, time and performance advantages in low level electrical measurement applications.

1.2 Features

Unique real-time sampling architecture for synchronous sourcing and measuring

- MeasureSync™ technology for simultaneous source module update and measure module sampling timing across all channels
- DC/AC amplitude and phase detection is user-selectable on all measure channels
- Common DAC/ADC sampling clock ensures highly precise and consistent source/measure timing coordination between 3 sources and 3 measures

Designed for scientific-grade low-level measurement applications

- Linear module power supply architecture for lowest possible source/measure noise
- Fully analog signal paths between data converters, modules, and the device under test (DUT)
- Remote modules for the shortest possible signal path to the DUT, which separates sensitive analog circuits from digital circuits and unwanted sources of interference typical of traditional single-enclosure instrument designs

The absolute precision of DC plus the detection sensitivity performance of AC instrumentation

- All source and measure channels are capable of DC and AC to 100 kHz signals
- Optimized for fundamental, harmonic, and phase AC plus DC biased measurements
- Modularity allows for flexible, user-configured modules to suit a specific application

Unique, flexible instrument/distributed module architecture

- Remote-mountable amplifier modules are interchangeable between instruments
- Modules are dynamically recognized when the system is reconfigured
- Uses a clean, simple UI and a common programming API for fast setup and a shorter learning curve

1.3 Functions

The M81-SSM provides:

Up to 3 source channels capable of:

- DC excitation
- AC excitation up to 100 kHz

Up to 3 measure channels capable of:

- DC measurement
- AC measurement (RMS, Peak)
- Lock-in amplifier measurement (X, Y, R, θ)

And additionally:

- A reference input for acquiring external frequencies
- A reference output to provide a phase reference to external equipment
- A configurable analog output
- 2 general purpose digital inputs
- 2 general purpose digital outputs

1.4 Specifications

Full specifications about the M81-SSM are provided on lakeshore.com.

Power requirement	100, 120, 220, 240 VAC, ±10%, 50 or 60 Hz, 140 VA
Size	216 mm wide × 87 mm high × 369 mm deep (8.5 in × 3.4 in × 14.5 in)
Weight	5.7 kg (12.6 lb)
Approval	CE mark
Temperature	15 °C to 35 °C at rated accuracy; 5 °C to 40 °C at reduced accuracy

1.5 Safety Summary and Symbols

Observe these general safety precautions during all phases of instrument operation, service, and repair. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended instrument use. Lake Shore Cryotronics, Inc. assumes no liability for user failure to comply with these requirements.

The M81-SSM protects the operator and surrounding area from electric shock or burn, mechanical hazards, excessive temperature, and spread of fire from the instrument. Environmental conditions outside of the conditions below may pose a hazard to the operator and surrounding area.

- Indoor use
- Altitude to 2000 m
- -20 °C to 70 °C, <90% relative humidity non-condensing
- Overvoltage category II
- Pollution degree 2
- Mains fluctuations up to 10%

Warning: Always plug the power cord into an easily accessible, properly grounded receptacle to ensure safe instrument operation.

Warning: Position the instrument in such a way to enable easy access to the disconnecting device. Failure to comply could result in death or injury to personnel.

Follow the guidelines below to ensure safety

Ground the Instrument

To minimize shock hazard, the instrument is equipped with a 3-conductor AC power cable. Plug the power cable into an approved 3-contact electrical outlet or use a 3-contact adapter with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable meet Underwriters Laboratories (UL) and International Electrotechnical Commission (IEC) safety standards.

Ventilation

The instrument has ventilation holes in its side covers. Do not block these holes when the instrument is operating.

Do Not Operate in an Explosive Atmosphere

Do not operate the instrument in the presence of flammable gases or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

Keep Away from Live Circuits

Operating personnel must not remove instrument covers. Refer component replacement and internal adjustments to qualified maintenance personnel. Do not replace components with power cable connected. To avoid injuries, always disconnect power and discharge circuits before touching them. Do not position the instrument so that it is difficult to disconnect the power cord.

Child Safety

This equipment is not suitable for use in locations where children are likely to be present.

Do Not Substitute Parts or Modify Instrument

Do not install substitute parts or perform any unauthorized modification to the instrument. Return the instrument to an authorized Lake Shore Cryotronics, Inc. representative for service and repair to ensure that safety features are maintained. If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

Cleaning

Do not submerge instrument. Clean only with a damp cloth and mild detergent. Exterior only.

Desktop Installation

When installing the instrument in a desktop environment, ensure it is mounted on a flat, level surface.

Improper Use

If the instrument is used in a manner that is not specified by Lake Shore, the safety protections provided by the instrument are no longer guaranteed, and may be impaired.

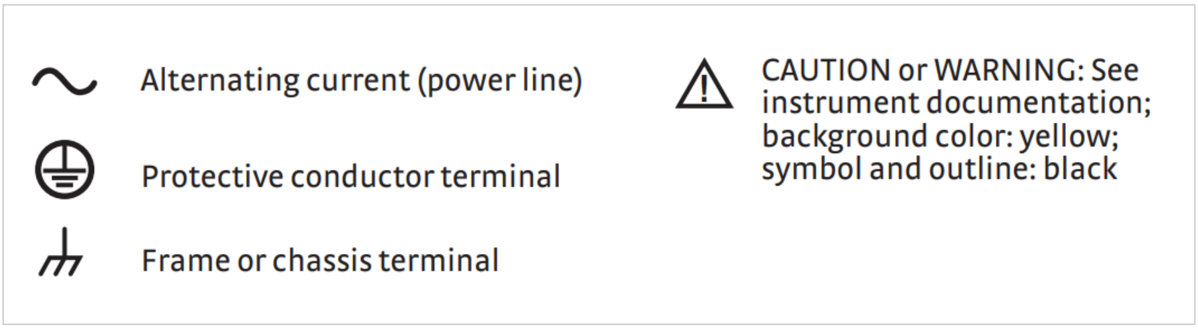


Figure 1.1: Safety symbols

1.6 Legal Disclaimers

1.6.1 PROPRIETARY RIGHTS

Methods and apparatus disclosed and described herein have been developed solely on company funds of Lake Shore Cryotronics, Inc. No government or other contractual support or relationship whatsoever has existed which in any way affects or mitigates proprietary rights of Lake Shore Cryotronics, Inc. in these developments. Methods and apparatus disclosed herein may be subject to U.S. Patents existing or applied for.

Lake Shore Cryotronics, Inc. reserves the right to add, improve, modify, or withdraw functions, design modifications, or products at any time without notice. Lake Shore shall not be liable for errors contained herein or for incidental or consequential damages in connection with furnishing, performance, or use of this material.

1.6.2 LIMITED WARRANTY STATEMENT

WARRANTY PERIOD: THREE (3) YEARS

1. Lake Shore warrants that products manufactured by Lake Shore (the "Product") will be free from defects in materials and workmanship for three years from the date the Product leaves Lake Shore (the "Warranty Period"). If Lake Shore receives notice of any such defects during the Warranty Period and the defective Product is shipped freight prepaid back to Lake Shore, Lake Shore will, at its option, either repair or replace the Product (if it is so defective) without charge for parts, service labor or associated customary return shipping cost to the Purchaser. Replacement for the Product may be by either new or equivalent in performance to new. Replacement or repaired parts, or a replaced Product, will be warranted for only the unexpired portion of the original warranty or 90 days (whichever is greater).
2. Lake Shore warrants the Product only if the Product has been sold by an authorized Lake Shore employee, sales representative, dealer or an authorized Lake Shore original equipment manufacturer (OEM).
3. The Product may contain remanufactured parts equivalent to new in performance or may have been subject to incidental use when it is originally sold to the Purchaser.
4. The Warranty Period begins on the date the Product ships from Lake Shore's plant.
5. This limited warranty does not apply to defects in the Product resulting from (a) improper or inadequate installation (unless OT&V services are performed by Lake Shore), maintenance, repair or calibration, (b) fuses, software, power surges, lightning and non-rechargeable batteries, (c) software, interfacing, parts or other supplies not furnished by Lake Shore, (d) unauthorized modification or misuse, (e)

operation outside of the published specifications, (f) improper site preparation or site maintenance (g) natural disasters such as flood, fire, wind, or earthquake, or (h) damage during shipment other than original shipment to you if shipped through a Lake Shore carrier.

6. This limited warranty does not cover: (a) regularly scheduled or ordinary and expected recalibrations of the Product; (b) accessories to the Product (such as probe tips and cables, holders, wire, grease, varnish, feed throughs, etc.); (c) consumables used in conjunction with the Product (such as probe tips and cables, probe holders, sample tails, rods and holders, ceramic putty for mounting samples, Hall sample cards, Hall sample enclosures, etc.); or, (d) non-Lake Shore branded Products that are integrated with the Product.
7. To the extent allowed by applicable law,, this limited warranty is the only warranty applicable to the Product and replaces all other warranties or conditions, express or implied, including, but not limited to, the implied warranties or conditions of merchantability and fitness for a particular purpose. Specifically, except as provided herein, Lake Shore undertakes no responsibility that the products will be fit for any particular purpose for which you may be buying the Products. Any implied warranty is limited in duration to the warranty period. No oral or written information, or advice given by the Company, its Agents or Employees, shall create a warranty or in any way increase the scope of this limited warranty. Some countries, states or provinces do not allow limitations on an implied warranty, so the above limitation or exclusion might not apply to you. This warranty gives you specific legal rights and you might also have other rights that vary from country to country, state to state or province to province.
8. Further, with regard to the United Nations Convention for International Sale of Goods (CISC), if CISG is found to apply in relation to this agreement, which is specifically disclaimed by Lake Shore, then this limited warranty excludes warranties that: (a) the Product is fit for the purpose for which goods of the same description would ordinarily be used, (b) the Product is fit for any particular purpose expressly or impliedly made known to Lake Shore at the time of the conclusion of the contract. (c) the Product is contained or packaged in a manner usual for such goods or in a manner adequate to preserve and protect such goods where it is shipped by someone other than a carrier hired by Lake Shore.
9. Lake Shore disclaims any warranties of technological value or of non-infringement with respect to the Product and Lake Shore shall have no duty to defend, indemnify, or hold harmless you from and against any or all damages or costs incurred by you arising from the infringement of patents or trademarks or violation or copyrights by the Product.
10. THIS WARRANTY IS NOT TRANSFERRABLE. This warranty is not transferrable.
11. Except to the extent prohibited by applicable law, neither Lake Shore nor any of its

subsidiaries, affiliates or suppliers will be held liable for direct, special, incidental, consequential or other damages (including lost profit, lost data, or downtime costs) arising out of the use, inability to use or result of use of the product, whether based in warranty, contract, tort or other legal theory, regardless whether or not Lake Shore has been advised of the possibility of such damages. Purchaser's use of the Product is entirely at Purchaser's risk. Some countries, states and provinces do not allow the exclusion of liability for incidental or consequential damages, so the above limitation may not apply to you.

12. This limited warranty gives you specific legal rights, and you may also have other rights that vary within or between jurisdictions where the product is purchased and/or used. Some jurisdictions do not allow limitation in certain warranties, and so the above limitations or exclusions of some warranties stated above may not apply to you.
13. Except to the extent allowed by applicable law, the terms of this limited warranty statement do not exclude, restrict or modify the mandatory statutory rights applicable to the sale of the product to you.

1.6.3 CERTIFICATION

Lake Shore certifies that this product has been inspected and tested in accordance with its published specifications and that this product met its published specifications at the time of shipment. The accuracy and calibration of this product at the time of shipment are traceable to the United States National Institute of Standards and Technology (NIST); formerly known as the National Bureau of Standards (NBS).

1.6.4 FIRMWARE LIMITATIONS

Lake Shore has worked to ensure that the firmware is as free of errors as possible, and that the results you obtain from the instrument are accurate and reliable. However, as with any computer-based software, the possibility of errors exists.

In any important research, as when using any laboratory equipment, results should be carefully examined and rechecked before final conclusions are drawn. Neither Lake Shore nor anyone else involved in the creation or production of this firmware can pay for loss of time, inconvenience, loss of use of the product, or property damage caused by this product or its failure to work, or any other incidental or consequential damages. Use of our product implies that you understand the Lake Shore license agreement and statement of limited warranty.

1.6.5 FIRMWARE LICENSE AGREEMENT

The firmware in this instrument is protected by United States copyright law and international treaty provisions. To maintain the warranty, the code contained in the firmware must not be modified. Any changes made to the code is at the user's risk. Lake Shore will assume no responsibility for damage or errors incurred as result of any changes made to the firmware.

Under the terms of this agreement you may only use the firmware as physically installed in the instrument. Archival copies are strictly forbidden. You may not decompile, disassemble, or reverse engineer the firmware. If you suspect there are problems with the firmware, return the instrument to Lake Shore for repair under the terms of the Limited Warranty specified above. Any unauthorized duplication or use of the firmware in whole or in part, in print, or in any other storage and retrieval system is forbidden.

1.6.6 TRADEMARK ACKNOWLEDGMENT

Many manufacturers and sellers claim designations used to distinguish their products as trademarks. Where those designations appear in this manual and Lake Shore was aware of a trademark claim, they appear with initial capital letters and the ™ or ® symbol.

- MeasureReady™, MeasureLINK™, CalCurve™, Cernox®, SoftCal™, Rox™, 2Dex™, TiltView™, and TruZero™ are trademarks of Lake Shore Cryotronics, Inc.
- Java® is a registered trademark of Oracle.
- LabVIEW® is a registered trademark of National Instruments.
- Mac® is a registered trademark of Apple, Inc., registered in the U.S and other countries.
- Microsoft Windows® and Excel® are registered trademarks of Microsoft Corporation in the United States and other countries.
- Stycast® is a trademark of Henkel Corporation.
- USB Type-C™ and USB-C are trademarks of USB Implementers Forum.

1.6.7 EU DECLARATION OF CONFORMITY

EU DECLARATION OF CONFORMITY

CE

This declaration of conformity is issued under the sole responsibility of the manufacturer.

Manufacturer:
Lake Shore Cryotronics, Inc.
575 McCorkle Boulevard
Westerville, OH 43082, USA

Hereby declares that the product:

Model(s):	M81-SSM-2, M81-SSM-4, M81-SSM-6 M81-BCS-10, M81-VM-10, M81-VS-10, M81-CM-10
Description:	MeasureReady® M81 Synchronous Measure Source System

is in conformity with the relevant Union harmonization legislation:

Low Voltage Directive	2014/35/EU
EMC Directive	2014/30/EU
RoHS Directive	2011/65/EU including amendments Directive (EU) 2015/863

and that the equipment is in conformity with the following harmonized and other appropriate standards:

EN 61010-1:2010
Overvoltage Category II
Pollution Degree 2

EN 61326-1:2013

EN 61000-3-2: 2014

EN 61000-3-3: 2013
Class A

EN 63000:2018

Signed for and on behalf of:
Place, Date:
Westerville, OH USA
19-JUL-2021

Scott Ayer
Director of Quality & Compliance

1.6.8 UL AND CSA CERTIFICATE

CERTIFICATE OF COMPLIANCE

Certificate Number: **SGSNA/21/SUW/00089**

Contract Number: 801167
Certificate Project Number: SUW-CERT210600076

Certified Product: MeasureReady M81 Synchronous Source Measure System (SSMS)
Trademarks: Lake Shore Cryotronics
Model(s): M81-SSM-6, M81-SSM-4, M81-SSM-2
Technical Data: 100, 120, 220, 240 Vac, 50/60 Hz, 140 VA

Certificate Holder: Lake Shore Cryotronics, Inc.
575 McCorkle Boulevard, 43082, Westerville, Ohio, United States of America

This certificate supercedes previous certificates issued with the same certificate number. Certification is valid when products are indicated on the SGS directory of certified products at www.sgs.com or using the QR code below. The product is certified according to ISO/IEC Guide 17067, Conformity assessment - Fundamentals of product certification, System 3, and in accordance with:

UL 61010-1, 3rd Ed., Rev. July 19, 2019
CAN/CSA C22.2 No. 61010-1, 3rd Ed., A1: 2018

Authorized by:

Effective date: 25 August 2021

Paul Krauss
Certifier

Page 1 of 1

This certificate is issued by the company under its General Conditions for Certification Services accessible at https://www.sgs.com/en/terms-and-conditions. Attention is drawn to the limitations of liability defined therein and in the Test Report here above mentioned which findings are reflected in this Certificate. Any unauthorized alteration, forgery or falsification of the content or appearance of this document is unlawful and offenders may be prosecuted to the fullest extent of the law.

Certification Body

Consumer and Retail Services, a division of SGS North America Inc.
620 Old Peachtree Road, Ste. 100, Suwanee, GA 30024, USA
t +1 770 570 1800 f +1 770 277 1240 www.sgs.com

1.6.9 ELECTROMAGNETIC COMPATIBILITY (EMC)

Electromagnetic compatibility (EMC) of electronic equipment is a growing concern worldwide. Emissions of and immunity to electromagnetic interference is now part of the design and manufacture of most electronics. To qualify for the CE Mark, the M81-SSM meets or exceeds the requirements of the European Radio Equipment Directive 2014/53/EU.

The instrument was tested under normal operating conditions with interface cables attached. If the installation and operating instructions in this user's manual are followed, there should be no degradation in EMC performance.

Exposure to RF interference greater than that found in a typical laboratory environment may disturb the sensitive measurement circuitry of the instrument.

Pay special attention to instrument cabling. Improperly installed cabling may defeat even the best EMC protection. For the best performance from any precision instrument, follow the grounding and shielding instructions in the user's manual. In addition, the installer of the instrument should consider the following:

- Shield measurement and computer interface cables.
- Leave no unused or unterminated cables attached to the instrument.
- Make cable runs as short and direct as possible. Higher radiated emissions are possible with long cables.
- Do not tightly bundle cables that carry different types of signals.

CHAPTER

TWO

INSTALLATION

This chapter provides general installation instructions for the MeasureReady™ M81-SSM Synchronous Source Measure System. Please read this entire chapter before installing the instrument and powering it on to ensure the best possible performance and to maintain safety.

2.1 Inspection and Unpacking

Inspect shipping containers for external damage before opening them. Photograph any container that has significant damage before opening it. Inspect all items for both visible and hidden damage that occurred during shipment. If there is visible damage to the contents of the container, contact the shipping company and Lake Shore immediately, preferably within five days of receipt of goods, for instructions on how to file a proper insurance claim. Lake Shore products are insured against damage during shipment, but a timely claim must be filed before Lake Shore will take further action. Procedures vary slightly with shipping companies. Keep all damaged shipping materials and contents until instructed to either return or discard them.

Open the shipping container and keep the container and shipping materials until all contents have been accounted for. Check off each item on the packing list as it is unpacked. Instruments themselves may be shipped as several parts. The items included with the MeasureReady™ M81-SSM are listed below. Contact Lake Shore immediately if there is a shortage of parts or accessories. Lake Shore is not responsible for any missing items if not notified within 60 days of shipment.

If you need to return the instrument for recalibration, replacement or repair, a return authorization (RMA) number must be obtained from a Lake Shore representative.

Items Included with M81-SSM:

- MeasureReady™ M81-SSM
- Line power cord
- USB A to USB-Type C™ adapter

- Quick Start Guide

2.2 Front Panel

The M81-SSM has a 5 in capacitive touch, color TFT display with LED backlight, which is used to control the instrument and display information.

Figure 2.1: M81-SSM front panel

1. **Power button** Tap to put the M81-SSM into sleep mode. Long press to power off the M81-SSM.
2. **TiltView™ touchscreen** The screen adjusts from a 0° to a 47° viewing angle, whether mounted in a rack or on a benchtop.

2.2.1 TiltView™ Touchscreen

The TiltView™ feature makes seeing the screen and operating the touch interface comfortable from any angle, even when mounted in a rack. The screen adjusts from a 0° to a 47° viewing angle, whether mounted in a rack or on a benchtop.

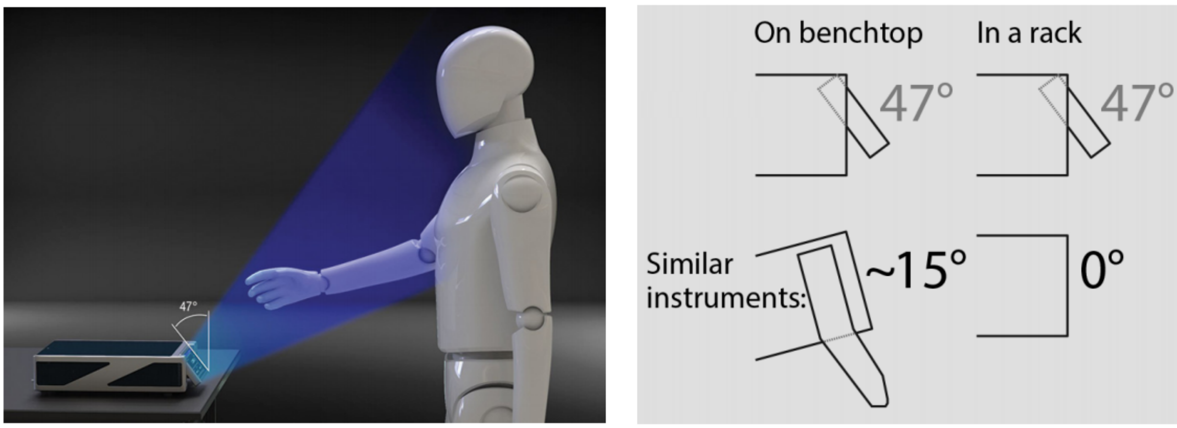


Figure 2.2: TiltView™ angle comparison

2.3 Rear Panel

This section provides a description of the M81-SSM rear panel connections.

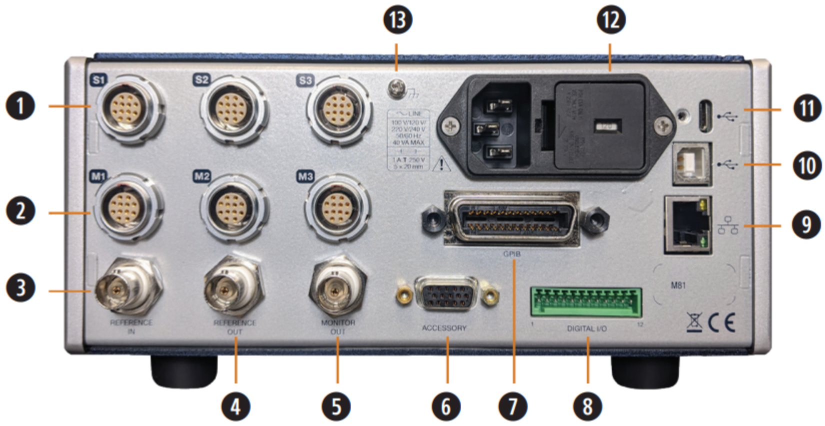


Figure 2.3: M81-SSM rear panel

1. **Source module connectors** Connect source modules to these connectors.
2. **Measure module connectors** Connect measure modules to these connectors.
3. **Reference in BNC connector** Allows connection of an external phase reference. See *Reference In* (page 54).
4. **Reference out BNC connector** Provides phase reference to external equipment. See *Reference Out* (page 55).
5. **Monitor out BNC connector** Configurable analog output. See *Monitor Out* (page 56).
6. **DB15 accessory connector** For future use.
7. **IEEE-488 connector** Connects the instrument to a IEEE-488 bus.
8. **Digital I/O** This terminal block provides configurable digital inputs and outputs. See *Digital I/O* (page 53).
9. **RJ-45 Ethernet interface** The Ethernet interface is provided to allow connection to a computer network.
10. **USB serial communications interface** The USB interface is provided to connect with most modern computers.
11. **USB Type-C™ interface** Connect to USB devices such as flash drives. Compatible with screw locking connectors.
12. **Line input assembly** Power is provided through the line input assembly.
13. **Chassis ground connection** Screw for chassis ground connection (M3×6).

See *connector definitions* (page 312) for more information on these connectors.

2.3.1 Line Input Assembly

This section describes how to properly connect the M81-SSM to line power. Please follow these instructions carefully to ensure proper operation of the instrument and the safety of operators.

For further instructions, see *Fuse and Line Voltage Selection* (page 310).

Line Voltage

The M81-SSM has four different AC line voltage configurations so that it can be operated from line power anywhere in the world. The nominal voltage and voltage range of each configuration is shown below. (The recommended setting for 230 V operation is 240 V.)

Nominal	Minimum	Maximum
100 V	90 V	110 V
120 V	108 V	132 V
220 V	198 V	242 V
240 V	216 V	264 V

Caution: AC line voltage is set at Lake Shore, but it is good to verify that the AC line voltage indicator in the fuse drawer window is appropriate before turning the instrument on. The instrument may be damaged if turned on with the wrong voltage selected.

2.3.2 Line Fuse and Fuse Holder

The line fuse is an important safety feature of the M81-SSM. If a fuse ever fails, it is important to replace it with the value and type indicated on the rear panel for the line voltage setting. The letter T on the fuse rating indicates that the instrument requires a time-delay or slow-blow fuse.

Fuse	1.6 A **T** 250 V

2.3.3 Power Cord

The M81-SSM includes a 3-conductor power cord that mates with the IEC 320-C14 line cord receptacle. Line voltage is present on the two outside conductors and the center conductor is a safety ground. The safety ground attaches to the instrument chassis and protects the user in case of a component failure. A CE approved power cord is included with instruments shipped to Europe; a domestic power cord is included with all other instruments (unless otherwise specified when ordered).

Warning: Always plug the power cord into a properly grounded receptacle to ensure safe instrument operation.

Note: If the power supply cord is damaged or lost, it must be replaced. Contact Lake Shore for a replacement to ensure proper voltage, current and type of cord. The power supply cord must not exceed 3 m (10 ft) in length.

The delicate nature of the measurements being taken with this instrument may necessitate additional grounding including ground strapping of the instrument chassis. In these cases the operator's safety should remain the highest priority and low impedance from the instrument chassis to safety ground should always be maintained.

2.4 Grounding Considerations

2.4.1 Chassis vs Measure Common

- **Chassis ground**: The exterior of the M81-SSM and modules are connected to the chassis ground of the power supply to the M81-SSM.
- **Measure common**: This internal isolated signal serves as the zero reference for all single ended signals in the system.

Earth ground connections may be made to the M81-SSM via a screw on the rear panel. An earth ground screw is also available on each module. The shell of each BNC connector is measure common.

For flexibility, measure common and chassis ground are isolated in the M81-SSM, allowing them to be joined externally at the point that is best suited for a particular application.

For best performance, chassis ground and measure common should be joined at a single location. If measure common is not connected to earth ground, the system may be more sensitive to electrostatic pickup. If measure common is connected to earth ground at multiple locations, the system may be more sensitive to interference coupled into the system via ground loop. For samples with high impedance, the connection between earth ground and measure common should be located as close as possible to the sample.

CHAPTER

THREE

OPERATION

This chapter provides instructions for the operating features of the MeasureReady™ M81-SSM Synchronous Source Measure System.

3.1 Module Management

3.1.1 Module Loading

After connecting a module, it must be loaded before it can be used. This is done by tapping the module's `Load` button on the front panel. Modules can be safely connceted when the instrument is powered off and are loaded automatically when the instrument is powered on.

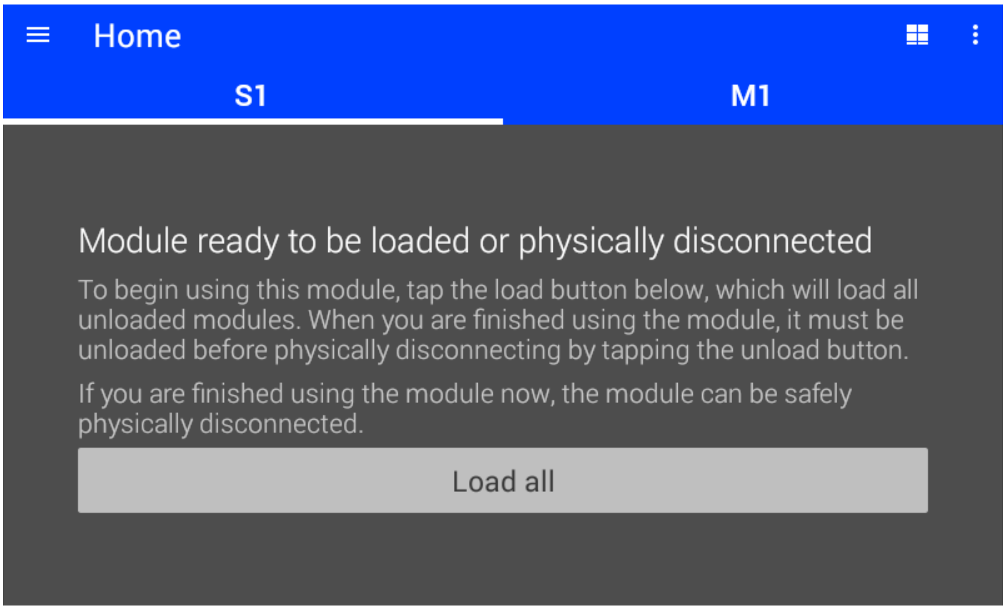


Figure 3.1: Load module button

Before disconnecting a module, it must be unloaded. This is done by scrolling to the bottom of the front panel and tapping the module's `Unload` button. Once the module is unloaded it can be safely physically disconnected. Modules can be safely disconnceted when the instrument is powered off.

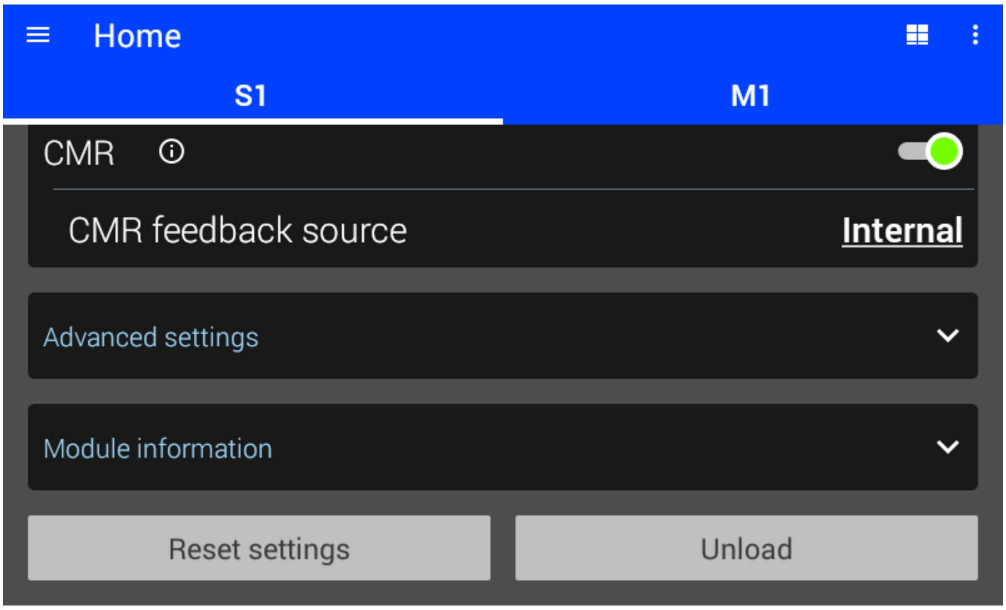


Figure 3.2: Unload and reset module buttons

SCPI Interface: *SOURce#:LOAD* (page 246) *SOURce#:UNLoad* (page 264) *SENSe#:LOAD* (page 209) *SENSe#:UNLoad* (page 216)

3.1.2 Module Names

You can provide your module with a name of your choice. This can be useful to track where specific modules are connected, for example "Hall Voltage." To name your module, navigate to the module's page and open the module information section. The name will be displayed on the front panel and can also be used to specify modules when connecting in scripting environments.

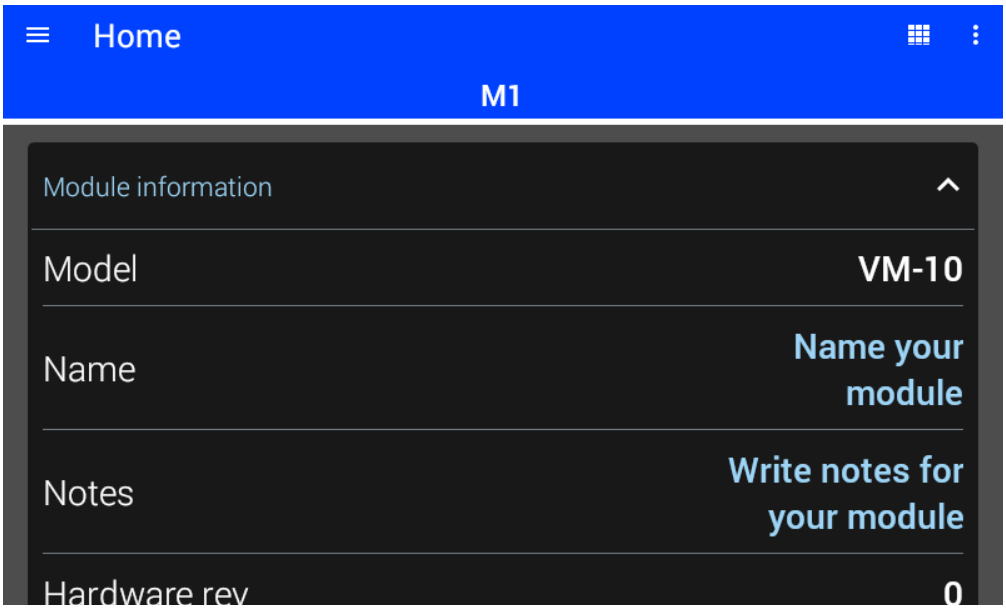


Figure 3.3: Module detail view

SCPI Interface: *SENSe#:NAME* (page 210) *SOURce#:NAME* (page 247)

For example, with the python driver:

```
from lakeshore import SSMSystem

head = SSMSystem()
my_module = dut.get_measure_module_by_name('My Module')
```

3.1.3 Module Notes

The modules notes section alows you to add notes relevant to individual modules. These notes are stored on the module, and are not lost when the module is unloaded or physically disconnected. In this way, it is possible to store details related to specific measurements or experiments, contact information, or ownership details on the module. This information can be modified at any time by navigating to the module's page and open the module information section. An abreviated version can be observed in the module information section, but up to 500 characters can be set and viewed by tapping on the setting.

SCPI Interface: *SOURce#:NOTes* (page 248) *SOURce#:NOTes?* (page 249) *SENSe#:NOTes* (page 211) *SENSe#:NOTes?* (page 211)

3.2 Basic Source Operation

To configure a source module on the front panel, navigate to the tab for the module.

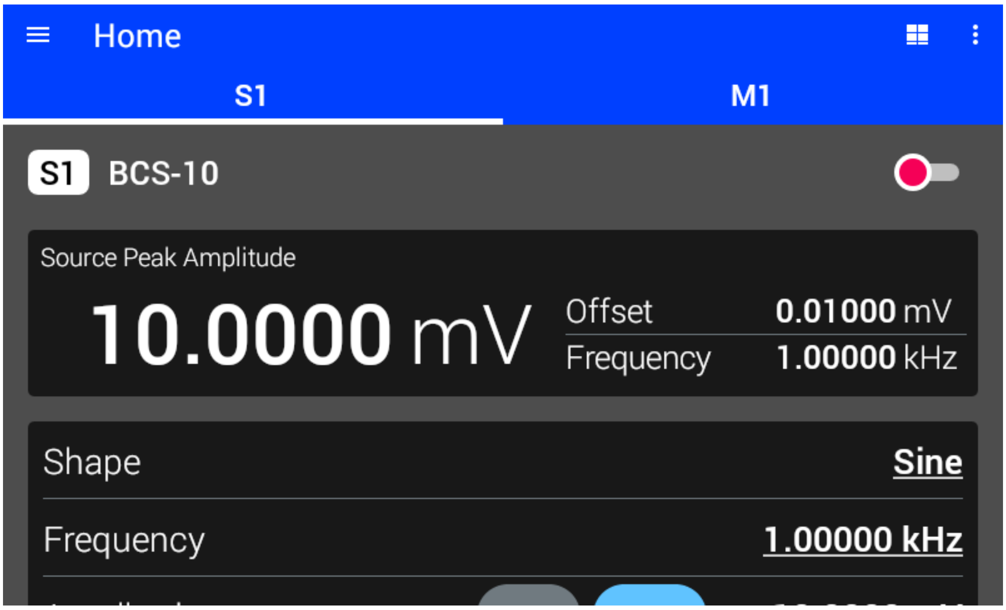


Figure 3.4: Basic source settings

Note that additional settings as applicable to a specific module will also be displayed here. For information on those settings, see *Modules* (page 77).

3.2.1 Enable/Disable

A source module's output can be enabled or disabled with the slider switch. When the switch is red, the output is disconnected and no excitation is applied. When the switch is green, the output is connected and excitation is applied. Always make sure that all other settings are configured appropriately before enabling the output.

Interface Command: *SOURce#:STATe* (page 255)

3.2.2 Shape

The shape setting determines whether the output is a DC signal or an AC signal.

Interface Command: *SOURce#:FUNCtion[:SHAPe]* (page 244)

3.2.3 Frequency

The source frequency can be set by tapping the frequency and entering a new frequency value.

Interface Command: *SOURce#:FREQuency[:FIXed]* (page 242)

3.2.4 Amplitude

When shape is DC, the amplitude directly sets the total excitation level. When shape is not DC, the amplitude sets the peak level of the excitation waveform. When shape is SINE, the amplitude can also be set with the RMS command. The instrument will calculate the peak amplitude based on the value provided.

Interface Commands:
SOURce#:VOLTage[:LEVel][:AMPLitude][:PEAK] (page 278)
SOURce#:VOLTage[:LEVel][:AMPLitude]:RMS (page 277)
SOURce#:CURRent[:LEVel][:AMPLitude][:PEAK] (page 234)
SOURce#:CURRent[:LEVel][:AMPLitude]:RMS (page 233)

3.2.5 Offset

The offset is added to the excitation waveform allowing for simultaneous AC and DC excitation.

Interface Commands:
SOURce#:VOLTage[:LEVel]:OFFSet (page 276)
SOURce#:CURRent[:LEVel]:OFFSet (page 232)

Note: Offset is not used if shape is DC and will be automatically configured to zero.

3.2.6 Duty

The duty cycle is the ratio of the excitation waveform that is active and is expressed as a % value between 0 and 1.0. It only applies to square and triangle shaped output. How the output is affected by the duty cycle is discussed in the following sections.

Square Wave Duty

Square wave source shape is a periodic waveform consisting of near instantaneous transitions between DC source values. The reciprocal of the source frequency determines the full period (high and low values) of a cycle while the source amplitude and offset determine the values of the high and low levels of the periodic source signal. At zero offset, the high and low values are the set value of the amplitude and the corresponding negative amplitude.

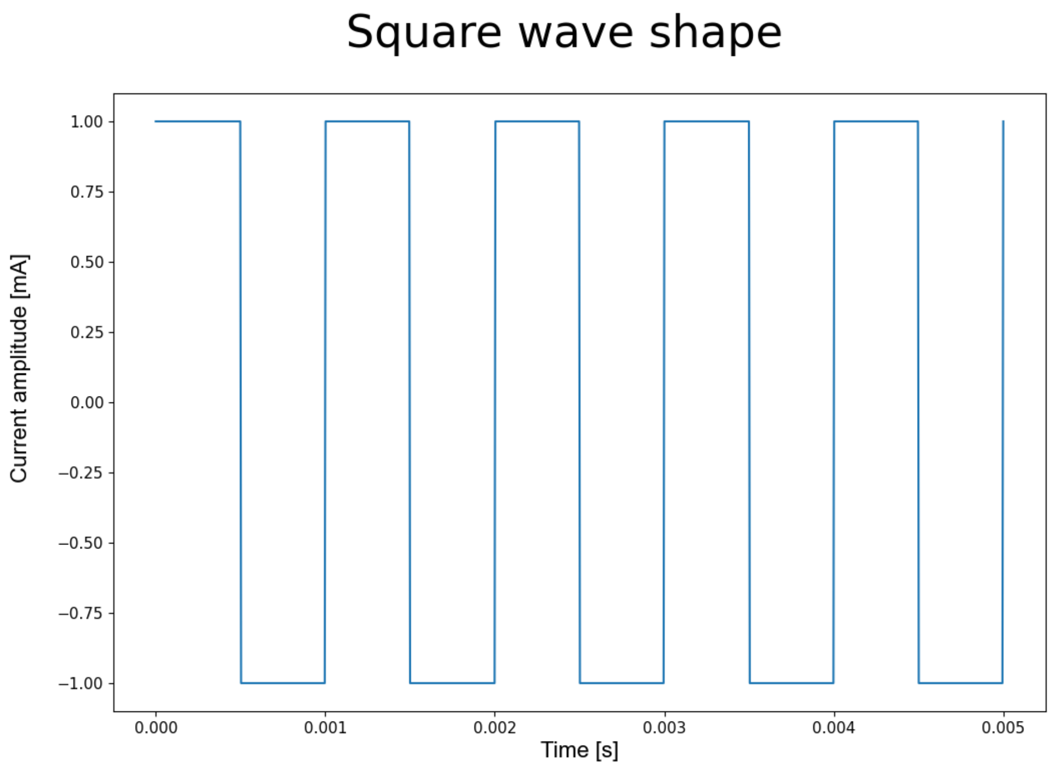


Figure 3.5: Square waveform shape

At 50% duty (duty: 0.500), the source value will spend equal time at the high and low setting of the waveform. The duty represents the fraction of the full cycle that the waveform spends at the high value; for values greater than 0.5, the periodic signal will spend more time at the high value and for values less than 0.5 the signal will be at the low value of the waveform for a comparatively longer time. For a square waveform on the

M81-SSM, the duty can be set from values of 0.001 to 0.999 — that is 0.1% to 99.9% duty cycle. Duty values of 0 and 1 are valid inputs but will result in a constant or DC waveform.

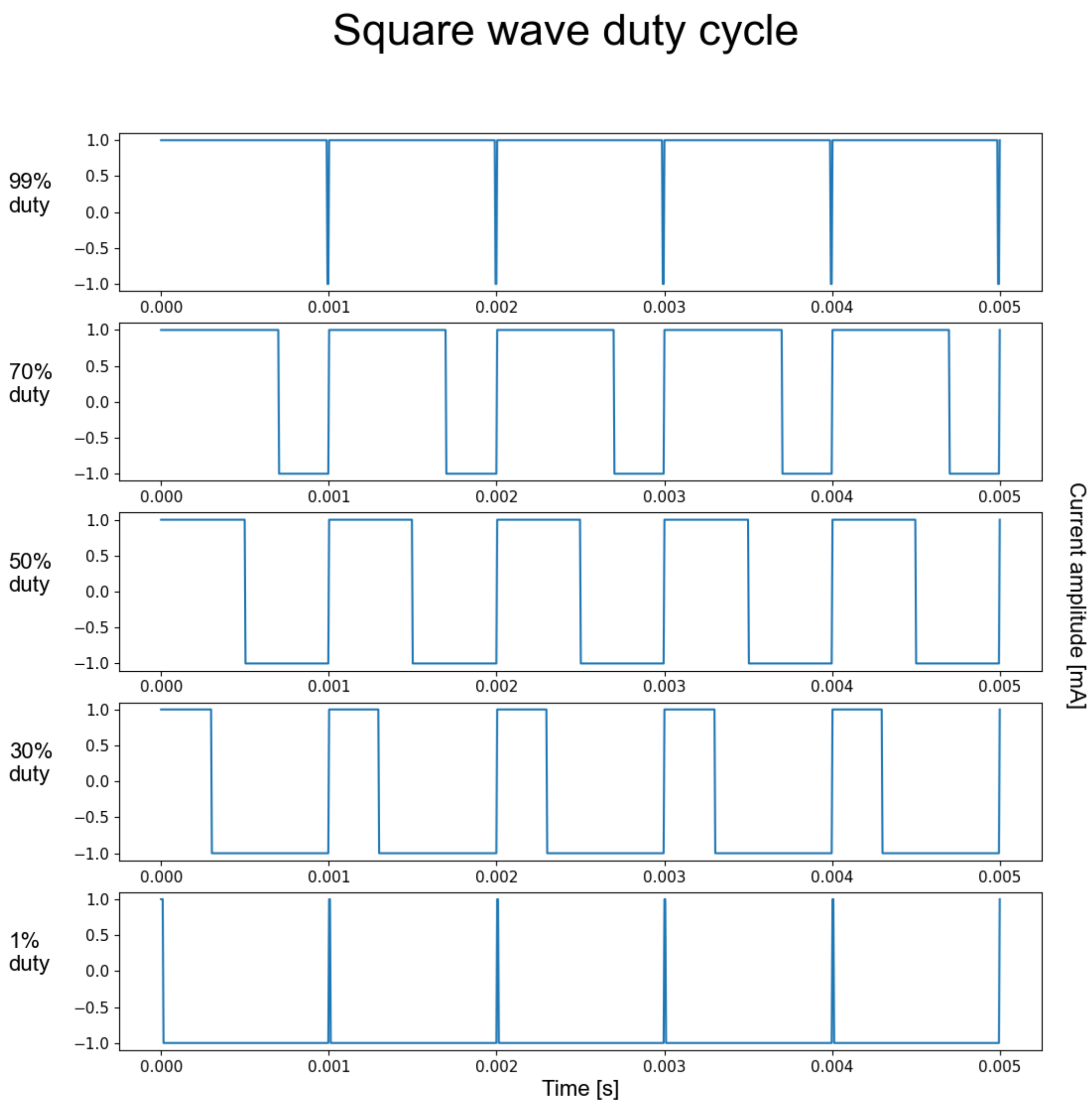


Figure 3.6: Duty setting changes the fraction of cycle time at the maximum DC value. Each trace has the same source frequency.

Triangle Wave Duty

The M81-SSM can also source triangle waveforms. With zero offset, the triangle wave is a periodic function consisting of a segment with source values that increase, linearly with time, from the negative amplitude value to the positive amplitude value and a second segment with source value that decreases, linearly with time, from the positive amplitude to the negative amplitude. The cycle period, consisting of both positive and negative slope segments, is the reciprocal of the waveform frequency.

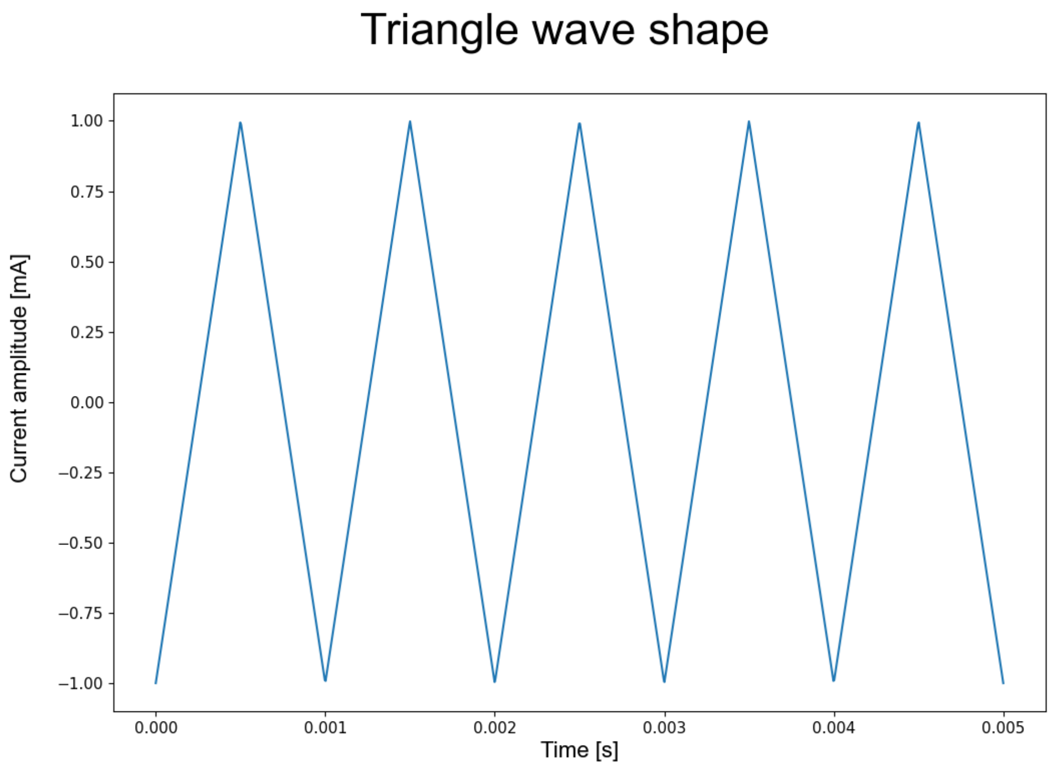


Figure 3.7: Triangle waveform shape.

With a triangle shape, the duty is defined by the fraction of the cycle period with positive slope. Just as with the square waveform, the duty can be set from 0 to 1 with 0.1% increments. Duty values greater than 0.5 generate waveforms with longer times with positive slope and duty values below 0.5 generate longer times with negative slope. With a duty setting of 0 or 1, the M81-SSM sources a so-called sawtooth waveform.

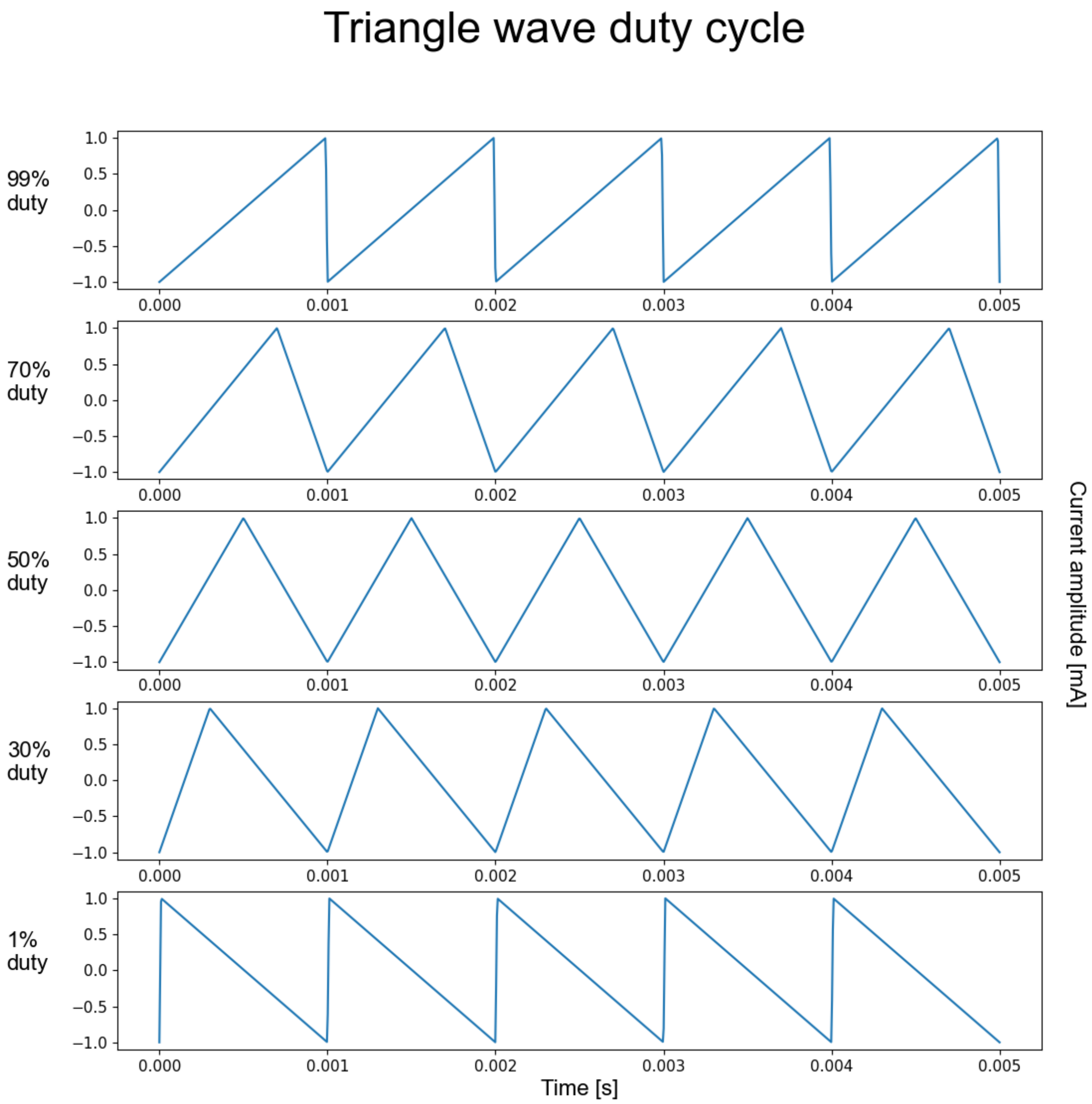


Figure 3.8: Duty sets the fraction of the cycle period with positive slope. Each trace has the same source frequency.

3.3 Basic Measurement Operation

To configure a measure module on the front panel, navigate to the tab for the module. Note that additional settings as applicable to a specific module will also be displayed here. For information on those settings, see *Modules* (page 77).

This first setting is the mode setting. The mode setting will determine what type of processing (DC, AC, or Lock-in) is performed by the M81-SSM on the signal received from the module. The other settings displayed will change based on what is relevant in the selected mode.

To change the mode on the front panel, tap the mode and select a new mode.

Interface Command: *SENSe#:MODE* (page 209)

3.3.1 DC Mode

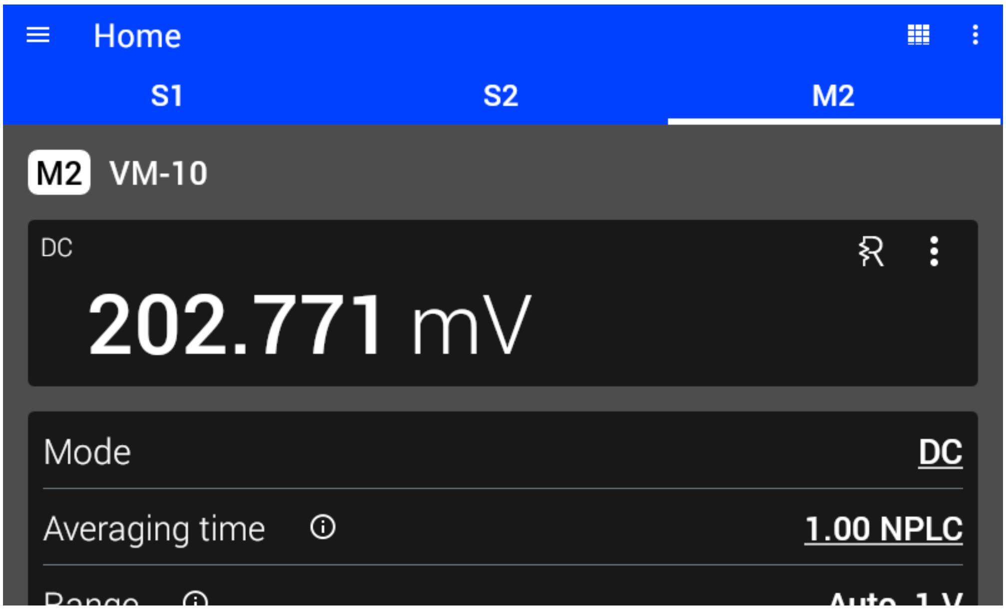


Figure 3.9: DC mode

Display

In DC mode, the display can be configured to show either the DC measurement, Relative DC measurement, or calculated resistance. Tap the three dots or the resistance symbol in the upper right hand corner of the display area to change which value is displayed. When displaying the Relative DC measurement, the ZERO/CLEAR button can be used to modify the baseline for the relative measurement. For more information on the calculated resistance, see *Calculated Resistance* (page 44). For more information on the Relative DC measurement, see *Relative Measurement* (page 46).

Averaging Time

The averaging time is specified in number of power line cycles (NPLC). Note that the MeasureReady™ M81-SSM detects the power line frequency in your country. For example, if your power line frequency is 60 Hz, setting an NPLC averaging time of 30 NPLCs would mean an averaging time of 0.5 s. For best rejection of line-related interference, an integer number of NPLC should be selected.

Interface Command: *SENSe#:NPLCycles* (page 212)

3.3.2 AC Mode

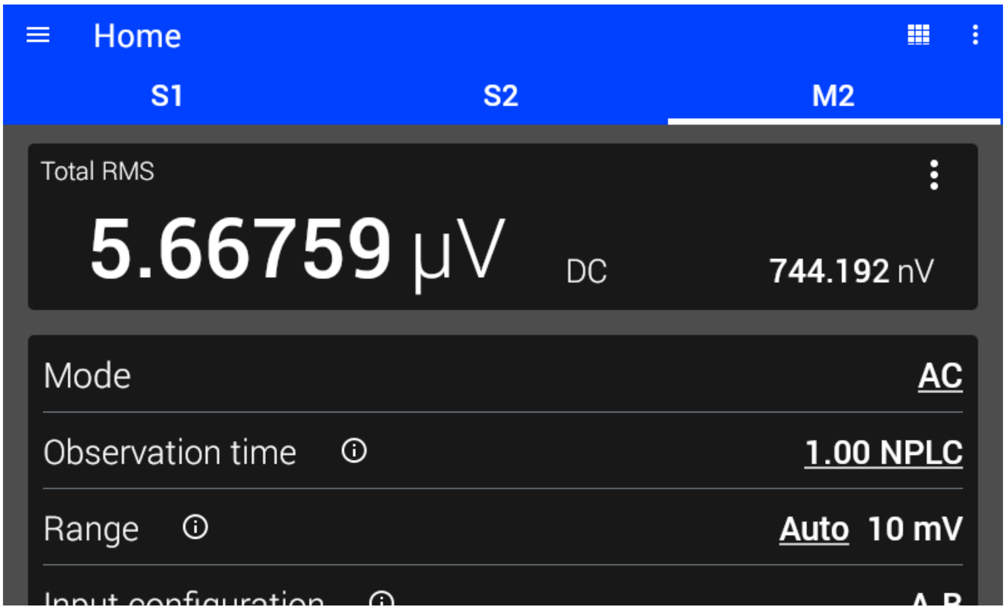


Figure 3.10: AC mode

Display

In AC mode, the display can be configured to show either total RMS and DC measurements, relative RMS measurement, or +Peak, -Peak, and Peak to Peak measurements. Tap the three dots in the upper right hand corner of the display area to change what is displayed. When displaying the Relative RMS measurement, the ZERO/CLEAR button can be used to modify the baseline for the relative measurement. For more information on the Relative RMS measurement, see *Relative Measurement* (page 46).

- **Total RMS** The total power of the input signal, including AC and DC components within the observation window
- **DC** The DC level of the input signal
- **+Peak** The highest input level detected during the observation window
- **-Peak** The lowest input level detected during the observation window
- **Peak to Peak** The span between the highest and lowest input levels detected during the observation window

Observation Time

The observation time is specified in number of power line cycles (NPLC). Note that the M81-SSM detects the power line frequency in your country. For example, if your power line frequency is 60 Hz, setting an averaging time of 30 NPLC would mean an averaging time of 0.5 s. For best rejection of line-related interference, an integer number of NPLC should be selected.

Interface Command: *SENSe#:NPLCycles* (page 212)

3.3.3 Lock-In Mode

In lock-in mode, the M81-SSM will detect signals that are coherent with the configured reference signal. For more information about the underlying concepts of lock-in operation, see section *Lock-In Amplifier Background* (page 48).

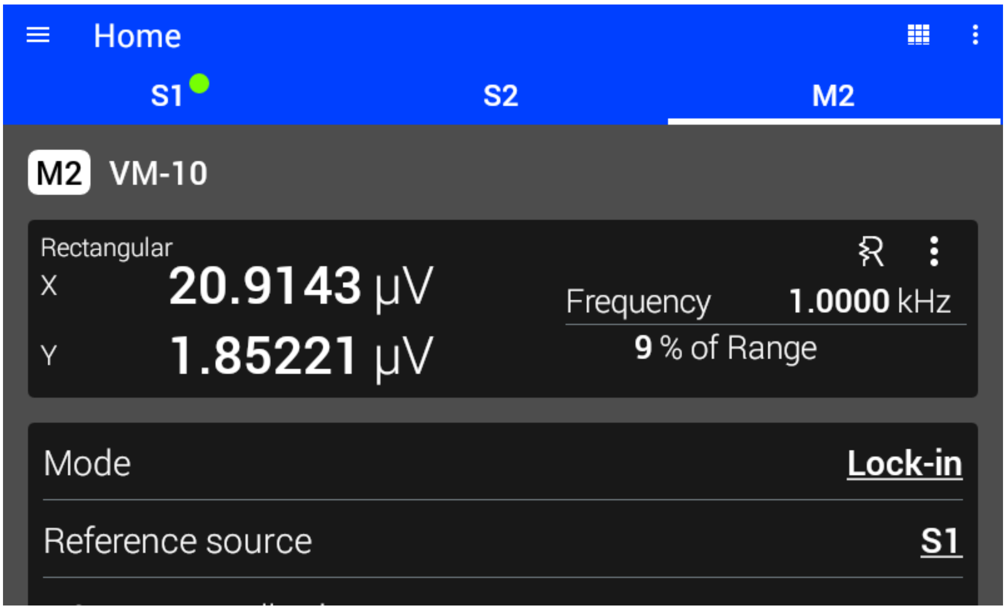


Figure 3.11: Lock-in mode display

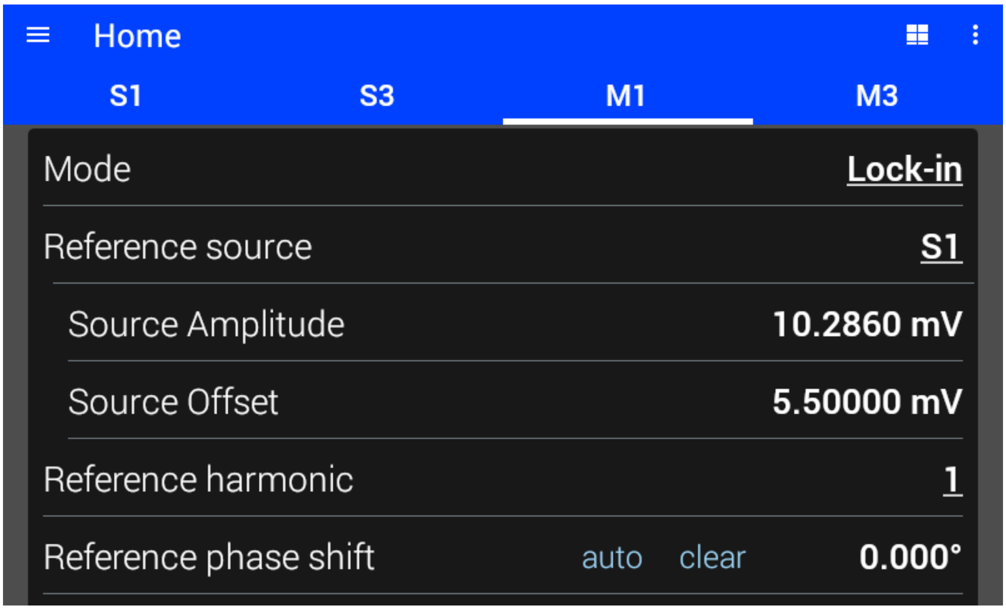


Figure 3.12: Lock-in mode settings

Display

In lock-in mode, the display can be configured to show either rectangular coordinates (X, Y), polar coordinates (R,), or the calculated resistance. In addition, the percent of range and frequency are always displayed. Tap the three dots or the resistance symbol in the upper right hand corner of the display area to change what is displayed. For more information on the calculated resistance, see *Calculated Resistance* (page 44).

- **X** The real (in-phase) component of the detected signal
- **Y** The imaginary (quadrature) component of the detected signal
- **R** The magnitude of the detected signal
- **θ** The phase of the detected signal relative to the reference signal
- **% of Range** The amount of the present range being used by the total input signal, including any noise
- **Frequency** The frequency of the reference signal

Reference Source

The measure channel will detect frequencies which are coherent with the specified reference source. For example, if a sample is being excited with a current by S1, the reference source can be set to S1. The reference source can also be set to Ref In. In this case the source will track the frequency detected on the reference input.

When reference source is set to a source channel, two additional read-only settings will appear. These settings display the amplitude and offset for the specified source module. However, the below states maybe shown.

- **Disconnected** The selected source channel does not have a source module connected.
- **Unloaded** The selected source channel has a source module connected, but the source module is not loaded.
- **Module error** The selected source channel has an error on the module, which can be viewed by selecting the relevant source channel tab.

Interface Command: *SENSe#:LIA:RSOurce* (page 206)

Reference Harmonic

The reference harmonic can be set to detect signals at a frequency that is a harmonic of the reference. For example, if the reference frequency is 1 kHz, setting reference harmonic to 2 would result in detecting signals at 2 kHz, which are coherent with the reference.

Interface Command: *SENSe#:LIA:DHARmonic* (page 201)

Reference Phase Shift

The reference phase shift is applied to the reference source before using the reference for demodulation. Tap **auto** to set the phase shift to result in zero degrees of indicated angle. Note that the measurement should be settled before tapping **auto**. Tap **clear** to set the phase shift to zero.

Interface Command: *SENSe#:LIA:DPHase* (page 202)

Time Constant

The time constant setting determines the bandwidth of the PSD output filter. Longer time constants will result in lower equivalent noise bandwidth (ENBW) at the cost of longer settle times. See table below.

Interface Command: *SENSe#:LIA:TIMEconstant* (page 208)

Rolloff

The rolloff setting determines the slope of the PSD output filter in the stop band. Steeper rolloff will result in lower ENBW at the cost of longer settle times. Steeper rolloff will also provide better rejection of interfering signals that are near the frequency of interest.

Interface Command: *SENSe#:LIA:ROLLoff* (page 206)

For time constant τ, the ENBW and time to settle can be seen in the *choosing lock-in filter settings table* (page 52).

3.4 Dashboard View

The M81-SSM can display the measurements and source settings of multiple modules using the dashboard. The dashboard is customizable, and some settings may be edited from this view. The dashboard is saved as part of a *Settings profile* (page 60).

To navigate to the dashboard, tap the icon on the top right that resembles a grid of boxes. This is the dashboard icon.

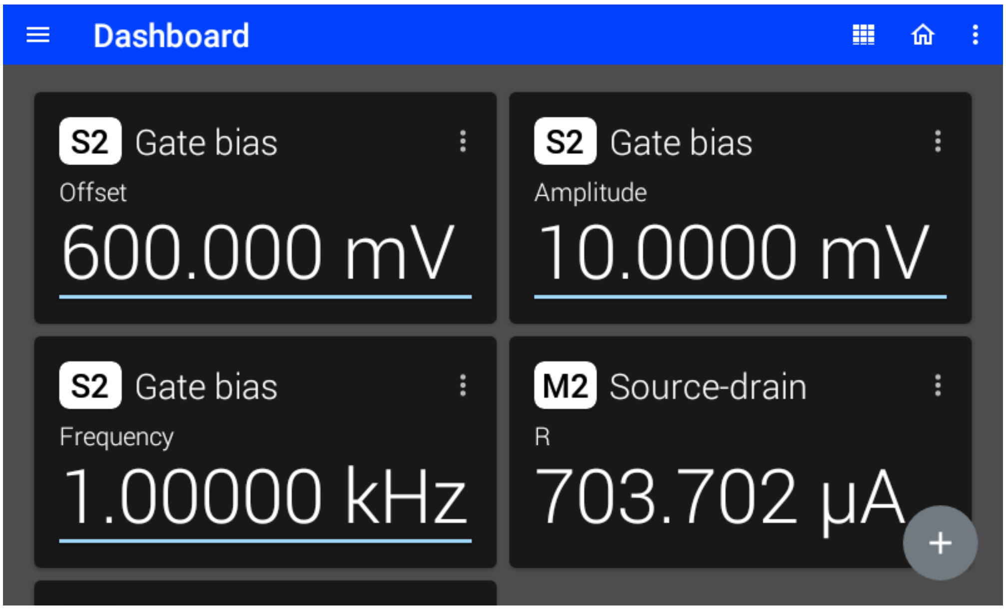


Figure 3.13: example dashboard

3.4.1 Configuring the dashboard

To add a panel to the dashboard, tap the "+" icon. You will be prompted to select a module and a parameter. A panel will then appear on the dashboard containing the selected parameter.

To change the contents of a panel or to delete a panel, tap the three dots in the top right corner of the panel. Remove all of the panels tap the three dots in the top right corner of the screen.

To rearrange the panels tap and hold the panel you would like to move. You may then drag it to the desired position.

If you would like to fit more panels on the screen, tap the dashboard icon again.

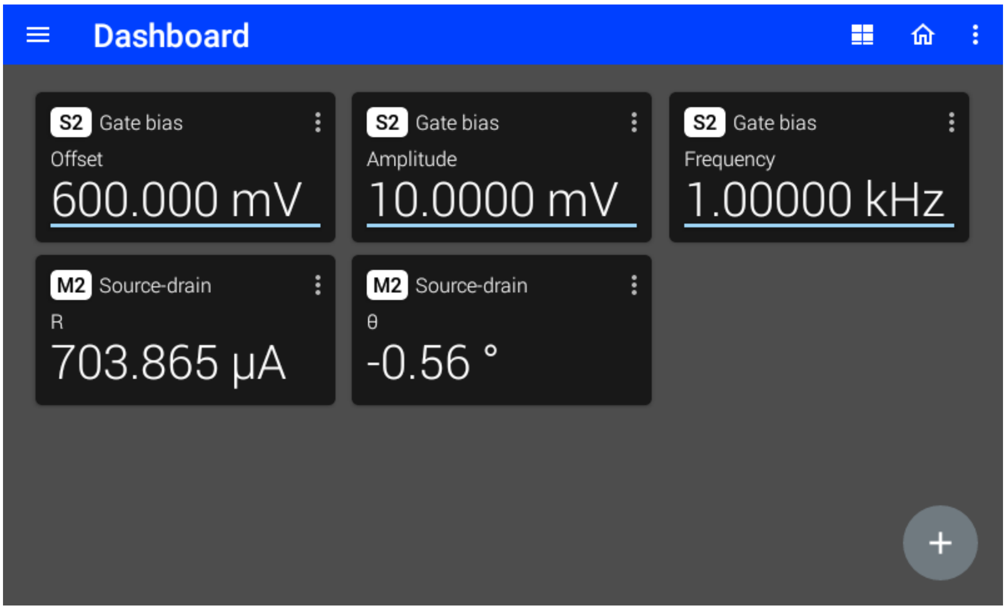


Figure 3.14: example dashboard with small panels

To return to the home screen, tap the icon in the top right that resembles a house.

3.5 Advanced Source Operation

To configure advanced settings, scroll down to the advanced section, and tap it to expand.

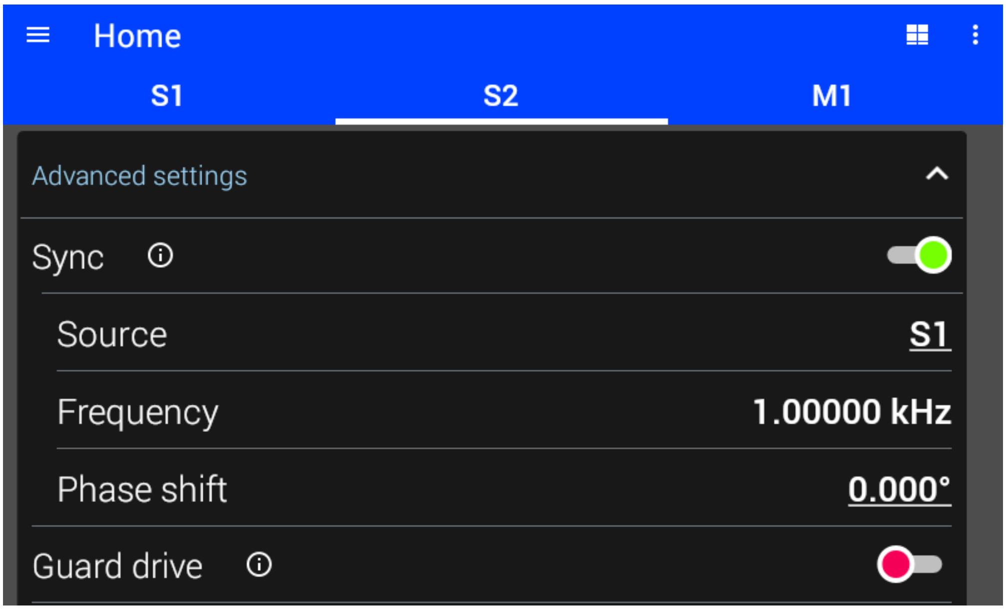


Figure 3.15: Advanced source settings

3.5.1 Sync

The sync feature allows the frequency of the source channel to track another frequency in the system.

Sync source

By default, sync is disabled and each channel generates its own frequency. When sync is enabled, the sync source can be set to the desired system frequency. Then this source channel will track that frequency, rather than generating its own.

For example, to generate the same frequency on S1 and S2, configure S1 normally, then enable Sync on S2 and select S1 as the Sync source for S2. The frequency configured on S1 will now dictate the frequency of S2.

The sync source can also be set to Ref In. In this case the source will track the frequency detected on the reference input.

Sync Phase Shift

When the sync feature is activated, a phase shift can be specified. The specified phase shift will be applied between the positive-going zero crossing of the signal specified by sync source and the positive-going zero crossing of this source channel.

Note: If sync is disabled, when two source channels are set to exactly the same frequency, the phase relationship is not controlled. To control the phase relationship of two signals, the sync feature must be used.

Interface Commands:

SOURce#:SYNChronize[:STATe] (page 263)

SOURce#:SYNChronize:SOURce (page 263)

SOURce#:SYNChronize:PHASe (page 262)

3.5.2 Dark Mode

If dark mode is enabled, the indicator LEDs on the front of the module will be turned off.

Interface Command: *SOURce#:DMODe* (page 236)

3.6 Advanced Measurement Operation

To configure advanced settings, scroll down to the advanced section, and tap it to expand.

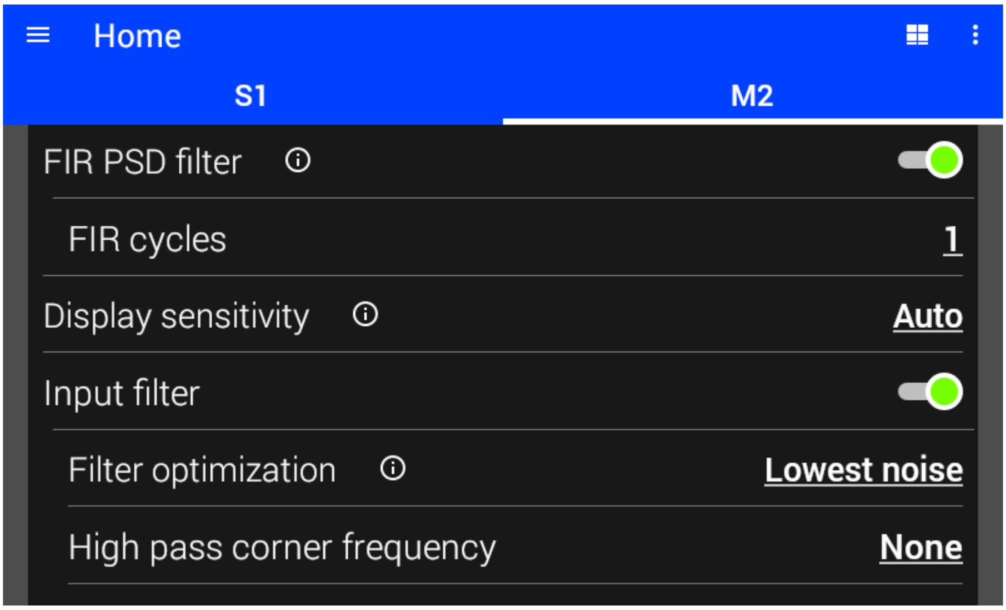


Figure 3.16: Advanced lock-in settings

3.6.1 FIR PSD Filter

When this filter is enabled, a separate FIR filter is applied to the PSD output in addition to the standard IIR output filter specified by the time constant and rolloff settings. This FIR filter is a moving average over a configurable number of cycles (periods) of the carrier frequency. This additional filter helps reject the carrier and harmonics of the carrier when making lock-in measurements.

Interface Command: *SENSe#:LIA:FIR[:STATe]* (page 204)

3.6.2 IIR PSD Filter

The IIR filter is extremely good at attenuating/rejecting signals outside of its pass band. Unlike the frequency response of the FIR filter, the IIR does not "bounce" and allow certain frequencies through. This means that if there is a large interfering signal the IIR will do a better job of removing it than the FIR.

However because of the exponential settling behavior of the IIR filter, it can take a long time for a measurement to be acceptably settled. Any input changes will live on in the IIR filter forever, getting smaller as time goes on but never going away completely. This means that it can take a very long time for large input disturbances to no longer be a significant error in the lock-in measurement.

Interface Command: *SENSe#:LIA:IIR[:STATe]* (page 205)

3.6.3 Digital Filter

When this filter is enabled, a high pass filter is applied to the lock-in readings before demodulation. The corner frequency of the filter adjusts automatically such that the response time of the lock-in measurement is not affected. The gain and phase of the filter is also compensated to avoid affecting the measurement. The filter is effective at reducing noise when measuring lock-in signals with large DC components, especially at low frequencies.

Interface Command: *SENSe#:DIGital:FILTer:HPASs[:STATe]* (page 191)

3.6.4 Dark Mode

If dark mode is enabled, the indicator LEDs on the front of the module will be turned off.

Interface Command: *SENSe#:DMODe* (page 193)

3.6.5 Calculated Resistance

By tapping the resistance "R" in the upper right hand corner of the display area, it is possible to view the calulated resistance when the measure module is in either dc or lock-in mode. To exit resistance view, either the three dots can be tapped to select the desired view, or the reading symbol can be tapped to return to the default view. When in dc mode, the display will show the selected source module's excitation that is being used in addition to the resistance value. When in lock-in mode, the display will show the selected source module's excitation and frequency in addition to the resistance value.

Presently, only two resistance calculations are supported. The first calculation requires a source module generating a DC shape with a measure module in DC mode. The second calculation requires a source module generating a sine shape with a measure module in lock-in mode.

In some cases, the resistance value may appear as an error code. These error messages can be tapped for more information about what is causing the issue.

- **S MISS** The selected source module is not connected.
- **S UNL** The selected source module is not loaded.

- **INVALID** The source module is not compatible with the measure module for calculating resistance. This occurs when the two modules use the same units, such as a VM-10 and VS-10.
- **E TYPE** The excitation type is not valid. This occurs when the measure mode and source shape do not create a supported resistance calculation.
- **DIV BY 0** The sourced or measured current is 0.
- **M ERR** There is an active error on the measure module.
- **S ERR** There is an active error on the selected source module.

Note: The displayed value may also show inaccurate noise values when the selected source is not enabled.

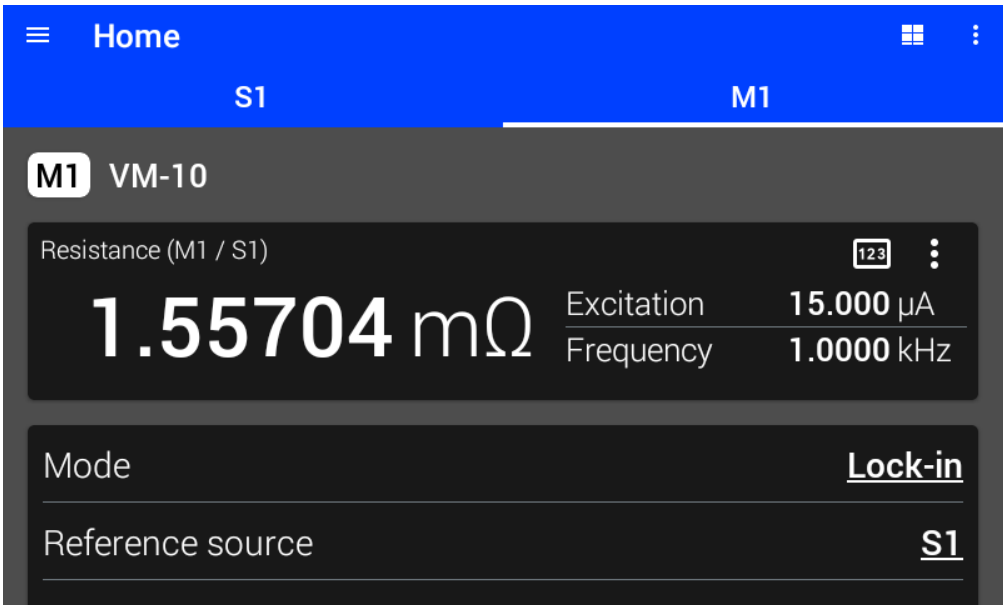


Figure 3.17: Lock-in resistance calculation

The source module being used to calculate resistance can be changed by scrolling down to the advanced section, and tapping to expand. The resistance is calcuated using one voltage and one current module. Therefore, for a module to be compatible, the source module must be voltage when the measure module is current, or vice versa.

Interface Command: *CALCulate:SENSe#:RESistance:SOURce* (page 153)

3.6.6 Relative Measurement

Relative measurements allow the removal of offsets from measured readings that could be introduced by sources external to the experiment. By using the zero button, existing offset will be set into the baseline value which is subtracted from the primary reading being displayed. When a baseline value is set, the zero button will be replaced with a clear button, which will set the baseline value back to zero.

By tapping the 3 dots in the upper right hand corner of the display area, it is possible to select Relative DC or Relative RMS, when in DC or AC modes respectively. The zero and clear button will allow for quick modification of the baseline value, shown in the bottom left of the display.

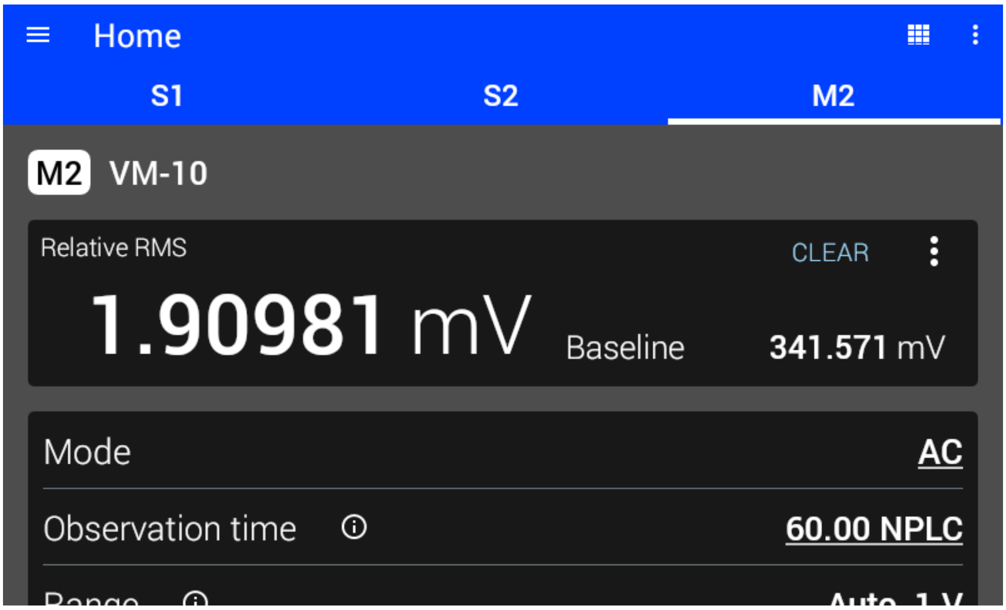


Figure 3.18: Relative RMS measurement

The baseline value can also be controlled by scrolling down to the advanced section, and tapping to expand. Tapping the advanced setting also allows for setting a custom baseline value.

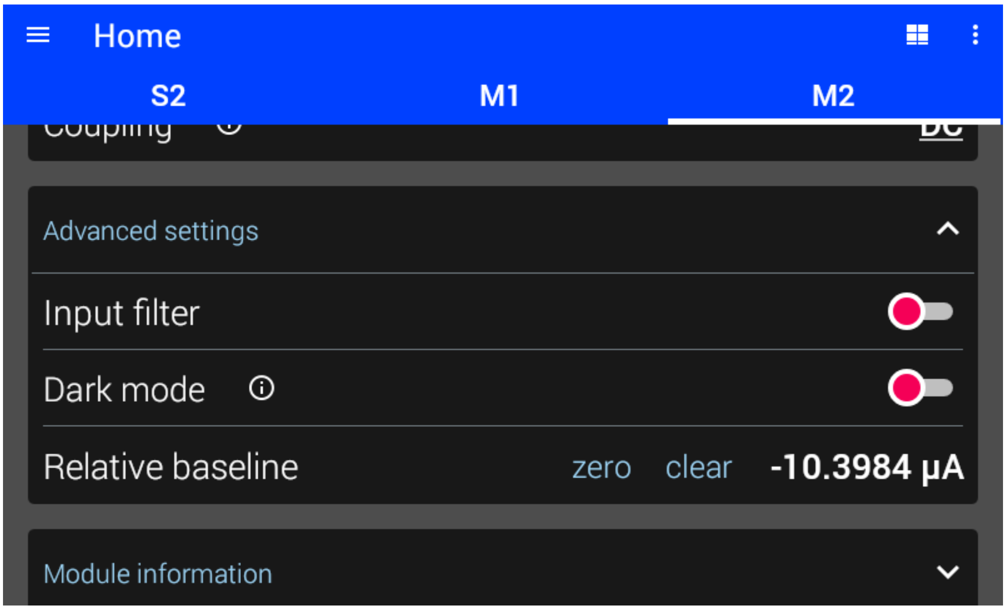


Figure 3.19: Relative measurement setting and baseline value

Interface commands:

FETCh:SENSe#:DC:RELative? (page 158)

FETCh:SENSe#:DC:RELative:IMMediate? (page 158)

FETCh:SENSe#:RMS:RELative? (page 165)

FETCh:SENSe#:RMS:RELative:IMMediate? (page 164)

READ:SENSe#:DC:RELative? (page 178)

READ:SENSe#:RMS:RELative? (page 182)

SENSe#:RELative:ZERO (page 214)

SENSe#:RELative:BASEline (page 213)

SENSe#:RELative:BASEline? (page 213)

3.7 Lock-In Amplifier Background

3.7.1 What is a Lock-In Amplifier?

A lock-in amplifier can detect signals at a known reference frequency. This reference frequency is either generated internally, or detected from an external source using a phase locked loop (PLL). The reference signal is then supplied to a phase sensitive detector (PSD). This is the core of the lock-in's signal processing, consisting of a multiplier and a low pass filter:

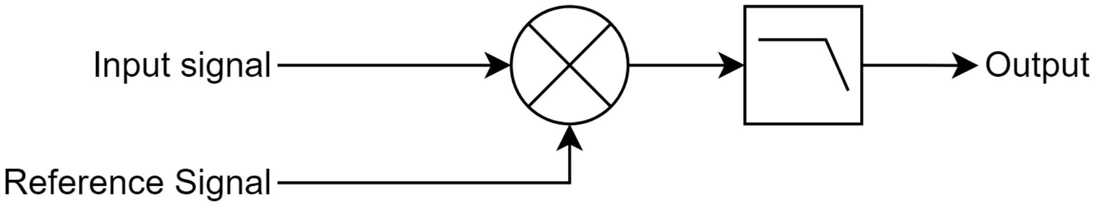


Figure 3.20: Phase sensitive detector

Additional filtering is often employed, but consider a filter with an output $y(t)$, which is an average over a single period T of the reference signal. For an given input $x(t)$ and a sinusoidal reference, this gives:

$$y(t) = \frac{1}{T}\int_{t-T}^{t} x(t)\,\sin(\frac{2\pi}{T}t)dt$$

For an input signal that is also a sine wave at the same frequency as the reference, the output of this filter has a non-zero mean corresponding to half the amplitude of the input signal. For example if the input is a 0.5 V sine wave at 1 kHz in phase with a 1 kHz reference:

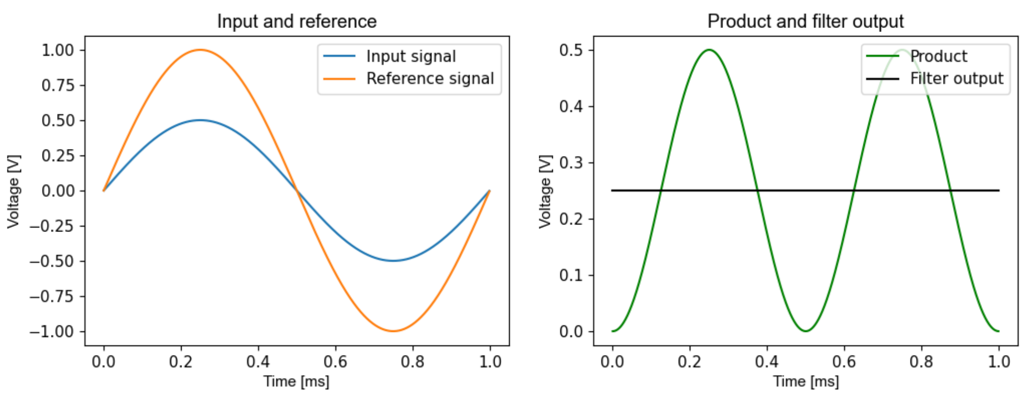


Figure 3.21: Sine wave input

However for other frequencies, the output has zero mean. For example if the input is a 0.5 V sine wave at 2.6 kHz:

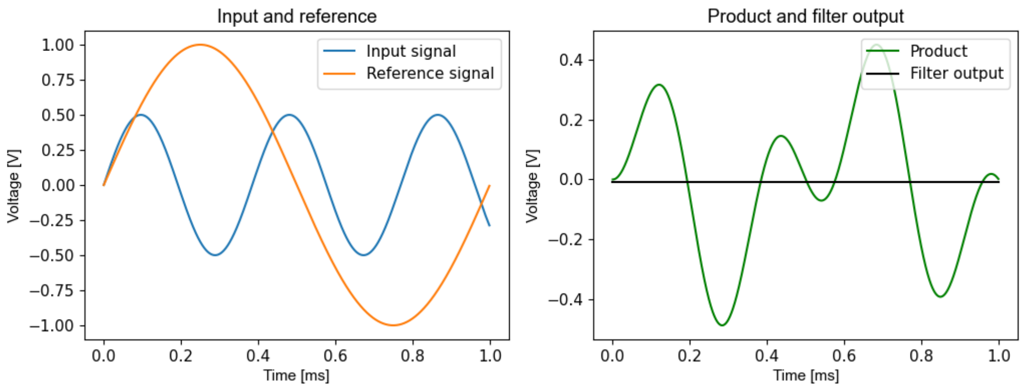


Figure 3.22: 2.6 kHz input

The product has no mean, and because the difference in frequencies is 1.6 kHz, the slowest frequency component of the product is 1.6 kHz, so this is easily rejected by the low pass filter. Frequencies close to to the reference frequency will not be rejected. Lower cut-off frequencies for the low pass filter make the PSD more selective.

3.7.2 Why use a Lock-In Amplifier?

The selective nature of the phase sensitive detector is very useful when a signal of interest must be extracted from noise. The components of the noise at other frequencies are rejected by the filter.

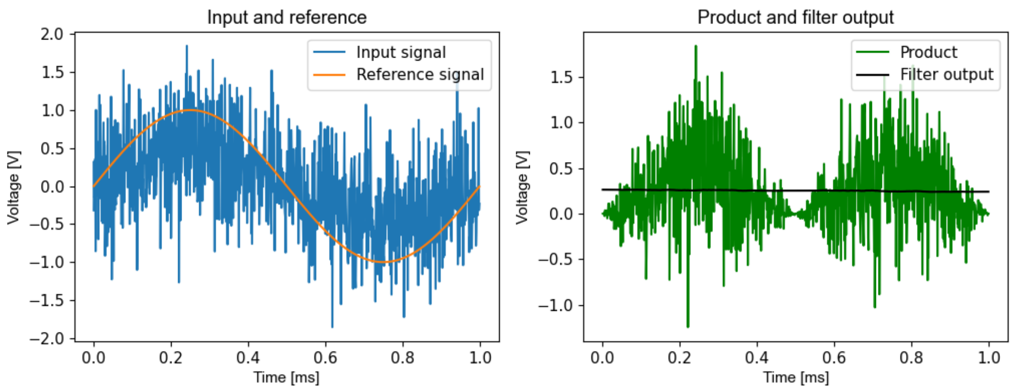


Figure 3.23: Noisy input

3.7.3 Other Lock-In Functions

The PSD described above responds to signals that are in phase with the reference. A signal that is shifted 90° from the reference produces zero mean.

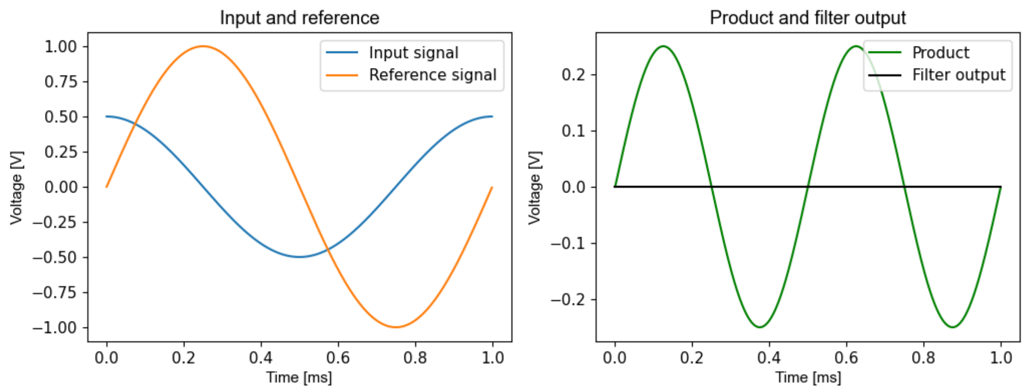


Figure 3.24: Phase shifted input

To provide a full picture of the signal that is at the reference frequency, each channel of the M81-SSM has two PSDs. The reference fed to the second PSD is shifted 90°. This makes that detector sensitive to the signals that are out of phase with the reference signal.

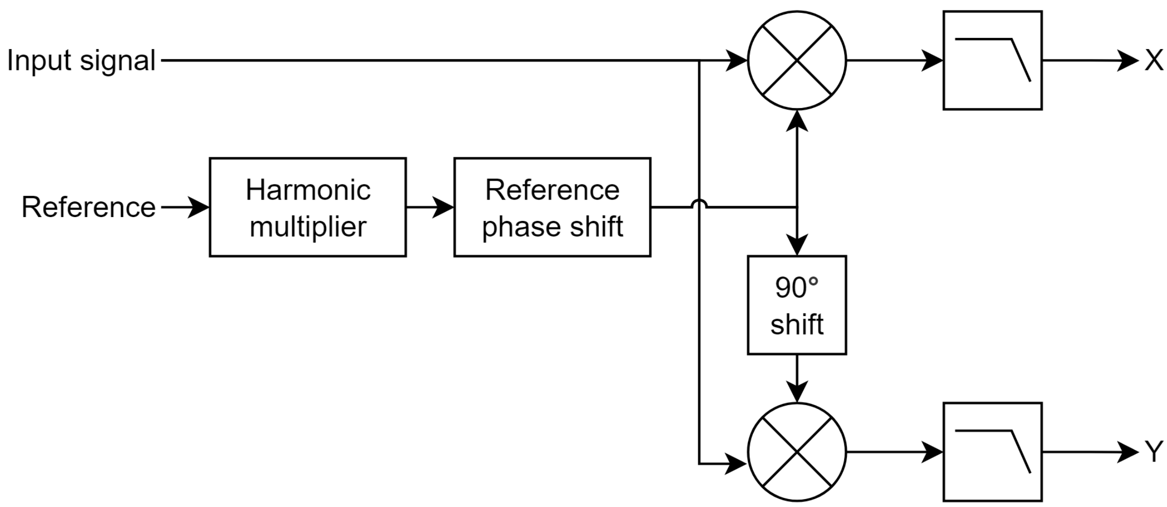


Figure 3.25: Full detector

Together, these two detectors can detect the magnitude and phase of the input signal. The M81-SSM can display this information in either rectangular coordinates (X, Y) or polar coordinates (R, θ). The polar coorinates are converted from the detector outputs as follows:

$$R = \sqrt{X^2 + Y^2}$$

$$\theta = atan2(Y, X)$$

As shown above, the M81-SSM also has the capability of multiplying the reference frequency. This allows the detection of input signals that are harmonics of the reference frequency.

Additionally a phase shift can be applied to the reference. This modifies the phase of input signal, which results in a measurement of zero degrees.

3.7.4 Choosing Lock-In Filter Settings

There are software filters that affect the behavior of the lock-in measurement. This section explains how to determine the software filter settings for the best possible lock-in measurement. The hardware filters in measure modules are not connected to these software filters.

Time constant and roll off

The primary lock-in filter is an infinite impulse response (IIR) filter. The time constant and roll off settings control this filter.

The filter provides the choice between settling time (measurement speed) and noise in the measurement band (equivalent noise bandwidth or ENBW). Longer settling times provide more repeatable measurements at the expense of the measurement rate.

A step change in the measured signal will cause the output to change forever. It is important to be aware of this behavior when choosing the time constant and roll off settings of the lock-in IIR filter. Capturing dynamically changing signals requires low settling times.

The table below shows the ENBW and amount of time needed to settle within a certain percentage of the final value. These are shown in terms of number of time constants (τ). For example, if the lock-in time constant is 1 s and the rolloff is 18 dB it will take 11.23 s for the lock-in indication to be within 0.1% of the final value after an input step change. If the time constant is 0.1 s it will take only 1.123 s for the lock-in indication to be within 0.1% of the final value after a step change.

Example calculations

Rolloff setting [dB]	ENBW [Hz]	1% settle time [s]	0.1% settle time [s]	0.01% settle time [s]	10 ppm settle time [s]
6	$\frac{1}{4\tau}$	4.61τ	6.91τ	9.21τ	11.51τ
12	$\frac{1}{8\tau}$	6.64τ	9.23τ	11.76τ	14.24τ
18	$\frac{3}{32\tau}$	8.41τ	11.23τ	13.93τ	16.55τ
24	$\frac{5}{64\tau}$	9.98τ	13.03τ	15.91τ	18.65τ

FIR cycles

In addition to the primary IIR filter there is an *FIR PSD Filter* (page 43) that averages over one period (cycle) of the reference frequency to reduce the effect of the reference signal on the lock-in measurement. The number of cycles is configurable. More cycles result in improved rejection of the reference signal interference at the cost of longer settling times. For example if the FIR filter is configured to average over 10 cycles and the reference frequency is 7 Hz the filter averaging time will be $\frac{10}{7\ Hz}$ = 1.429 s.

3.8 Digital I/O

The M81-SSM provides 2 digital outputs and 2 digital inputs.

To configure the digital I/O, open the settings menu (top left corner of the screen). Then tap GPIO configuration.

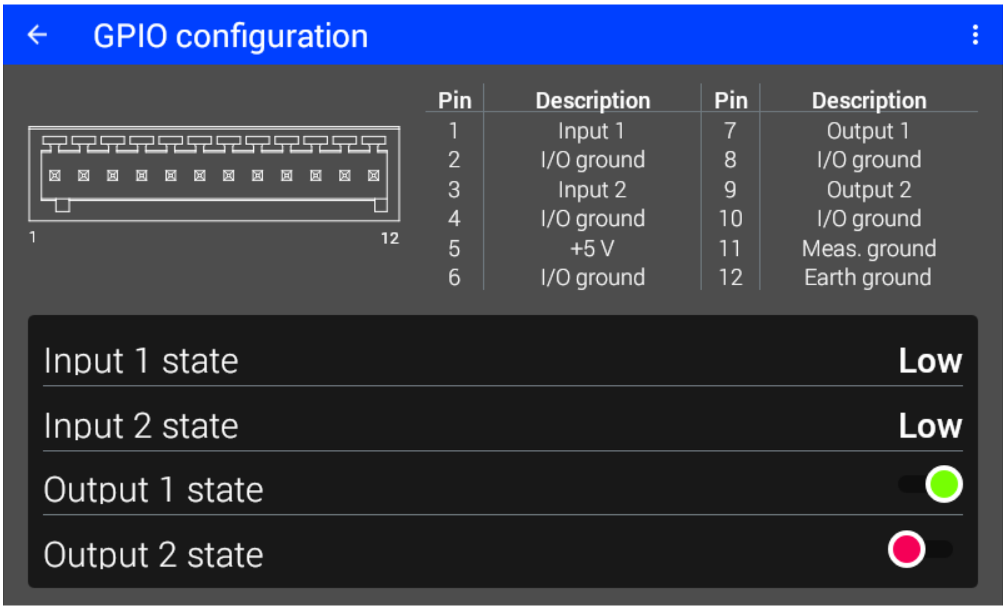


Figure 3.26: GPIO configuration

3.8.1 Input States

The present states read by the digital inputs will be displayed as either Low or High.

3.8.2 Output States

The state driven on the digital outputs can be configured with the slider switches. When the switch is red, a logic Low is driven on the output. When the switch is green, a logic High is driven on the output.

3.9 Reference In

The M81-SSM provides a reference input for acquiring external frequencies for use as a reference using a phase-locked loop (PLL). This reference can be used for lock-in measurements on any connected measure module. The maximum voltage limits of the Reference Input are +/- 10 V peak.

To configure the reference input, open the settings menu (top left corner of the screen). Then tap Reference configuration.

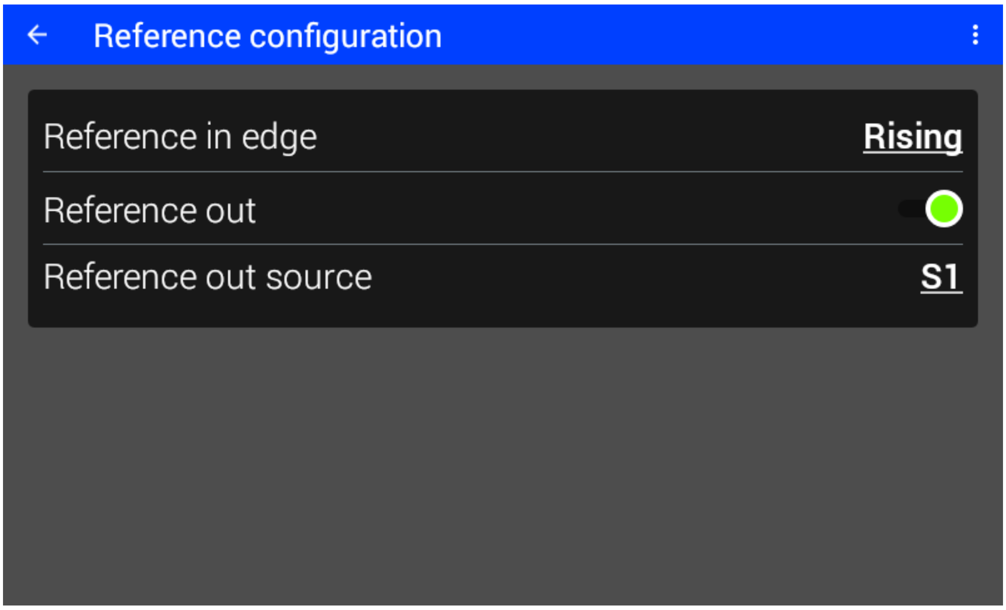


Figure 3.27: Reference configuration

For use by a lock-in measurement, see *Reference Source* (page 37).

3.9.1 Edge

The edge setting determines if the PLL is sensitive to the rising or falling edge of the reference input signal. If the edge is set to Rising, the positive going zero crossing of the reference signal will be regarded as zero degrees. If the edge is set to Falling, the negative going zero crossing of the reference signal will be regarded as zero degrees.

Interface Command: *INPut:REFerence:EDGe* (page 167)

3.10 Reference Out

The M81-SSM is capable of providing a square wave reference output that matches the frequency and phase of a selected source channel.

To configure the reference output, open the settings menu (top left corner of the screen). Then tap Reference configuration.

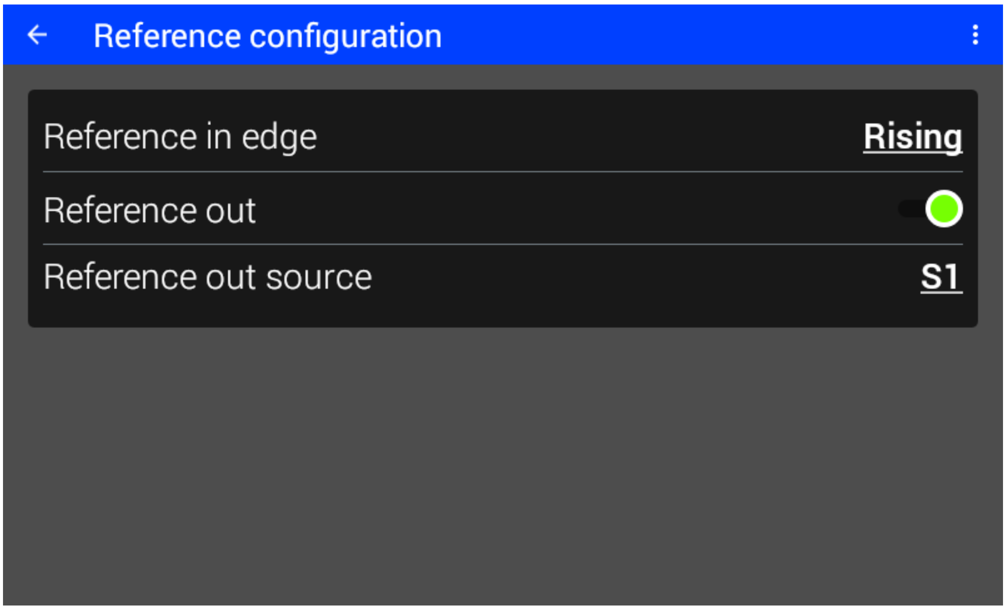


Figure 3.28: Reference configuration

3.10.1 Enable/Disable

The reference output can be enabled or disabled with the slider switch. When the switch is red, the reference output is disabled. When the switch is green, the reference output is enabled.

Interface Command: *OUTPut:REFerence[:STATe]* (page 172)

3.10.2 Source

The reference output can be configured to track any of the source channels. This is configured with the source setting (S1, S2, or S3).

Interface Command: *OUTPut:REFerence:SOURce* (page 171)

3.11 Monitor Out

The M81-SSM provides a monitor output that can be used to observe the analog signal that is processed by a measure module or a manually specified DC signal. The monitor out signal will not reflect the accuracy, bandwidth, or other specifications of the measure module. Calibration corrections are not applied and other performance limitations are introduced, including offset and bandwidth limitations of the monitor output signal.

To configure the monitor output, open the settings menu (top left corner of the screen). Then tap Monitor out configuration.

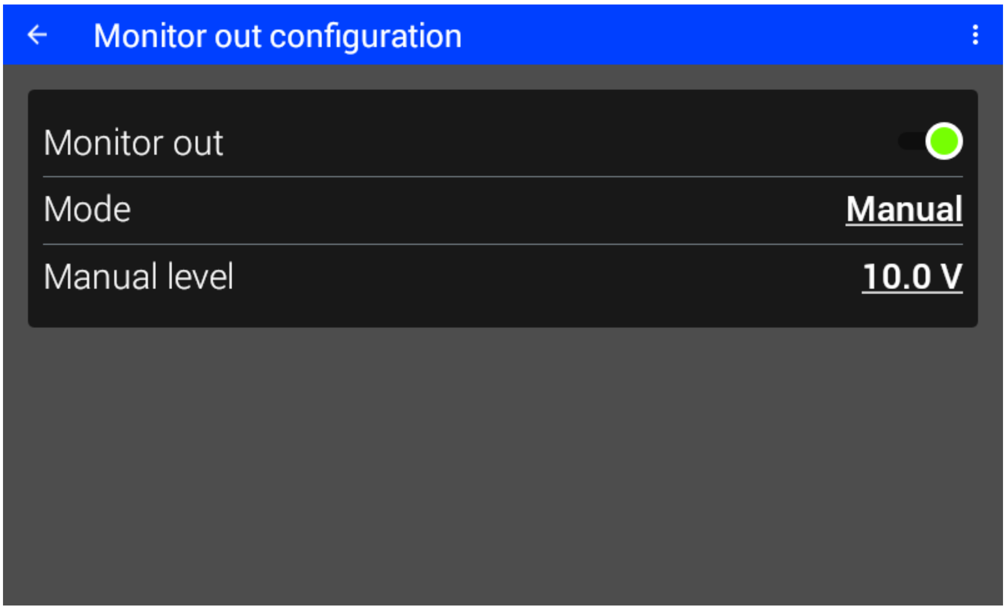


Figure 3.29: Monitor out configuration

3.11.1 Enable/Disable

The monitor output can be enabled or disabled with the slider switch. When the switch is red, the monitor output is disabled. When the switch is green, the monitor output is enabled.

Interface Command: *OUTPut:MONitor[:STATe]* (page 170)

3.11.2 Mode

The monitor output has multiple modes of operation. This is configured with the Mode setting.

Interface Command: *OUTPut:MONitor:MODe* (page 169)

Measure Channel

If the source is configured to a measure channel (M1, M2, or M3), the monitor out will provide the analog signal that is output by the connected measure module.

Note that the scaling of this voltage is dependant upon the configuration of the connected measure module such as range. The scaling for the present configuration will be displayed below the source selection.

Manual

If the source is configured to manual, a DC voltage can be configured to be output on the monitor output. To set the level tap Manual level to select the output voltage.

Interface Command: *OUTPut:MONitor:MLEVel* (page 168)

3.12 Self Calibration

3.12.1 Head Self Calibration

For maximum accuracy, the M81-SSM has the ability to calibrate all of the circuitry to an internal precision standard. The internal standard must be factory calibrated annually.

For full specified accuracy, a self calibration must be performed within 24 h of use.

To initiate a self-calibration sequence, navigate to the calibration page. Here, you can also view information about the last self and factory calibrations.

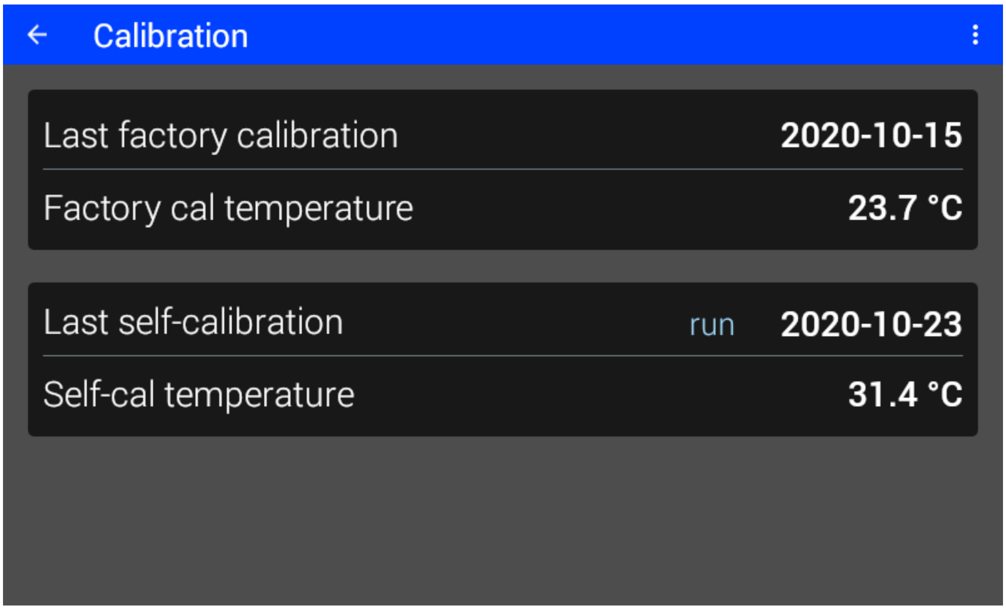


Figure 3.30: Calibration view

A self cal can also be initiated from the SCPI interface using *CALibration:SCALibration:RUN* (page 155). The status of the self calibration is avaiable with the query *CALibration:SCALibration:STATus?* (page 156).

3.12.2 Module Self Calibration

Some modules support a self-calibration sequence for maximum accuracy. For specified accuracy, a self calibration must be performed within 24 h of use.

To initiate a self-calibration sequence, navigate to the module's page and open the module information section. Here, you can also view the date of the most recent self-calibration of this module.

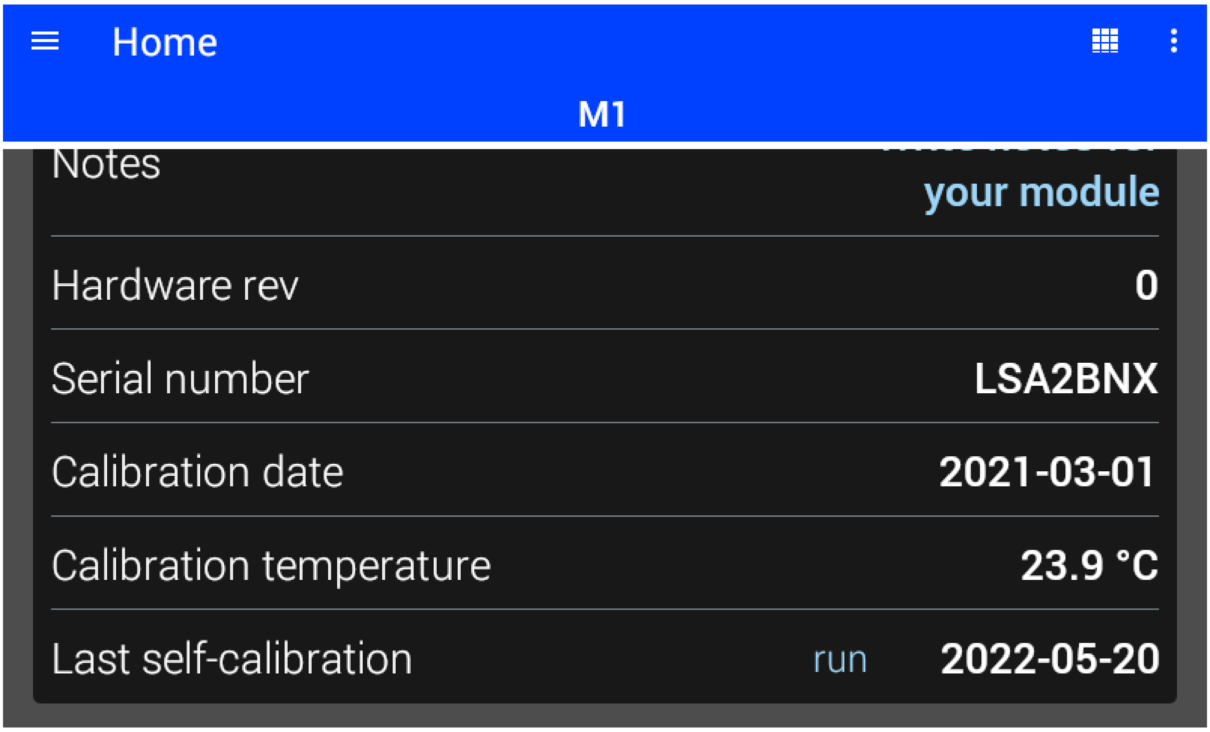


Figure 3.31: Module detail view

A self cal can also be initiated from the SCPI interface using *SENSe#:SCALibration:RUN* (page 214), or *SOURce#:SCALibration:RUN* (page 254).

3.13 Settings Profiles

The M81-SSM can help manage settings for an experiment using settings profiles. A settings profile contains all of the settings for the M81-SSM for all connected modules. A profile with all of the settings properly set for an experiment can be captured, then later recalled to establish a convenient, repeatable settings state.

To manage settings profiles, open the settings menu (top left corner of the screen) to go to navigation. Then tap Settings profiles.

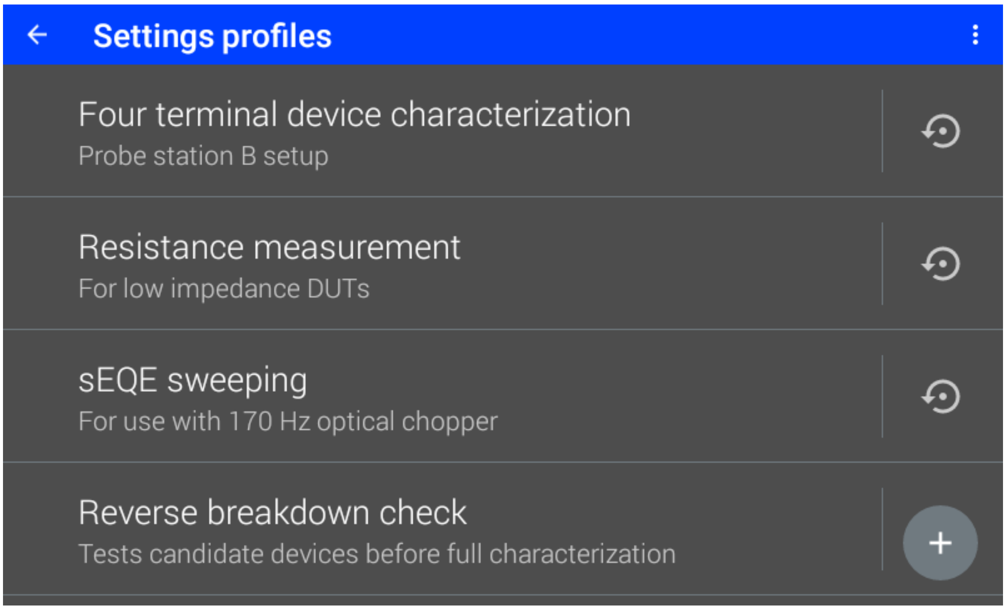


Figure 3.32: example Settings profiles page

3.13.1 Creating a Profile

To create a new profile, first make sure that the desired modules are connected and that all settings are configured as desired for the M81-SSM and all modules.

Then, navigate to settings profiles and click the "+" icon. By default, a numbered name will be created. You can also choose a name for the profile. Then click save.

You can then tap on the profile in the list to view details and optionally add a description.

3.13.2 Recalling a Profile

To recall a profile, first make sure that the same modules are connected as when the profile was created. You can see what modules a given profile expects in the details page for that profile.

Then navigate to the details page for that profile and tap the restore button. You can also tap the restore icon next to the profile name in the profiles list.

Note: When the profile is recalled, all sources will be disabled.

If the *Dashboard view* (page 39) was configured when the profile was saved, the dashboard will also be restored.

When recalling a profile from an attached USB storage device, it will not be imported to the instrument.

3.13.3 Updating a Profile

To update a profile, first recall the profile, then make any desired changes to the profile. Then, navigate to the profile details page and tap the update icon in the upper right corner of the screen. This action cannot be performed on a profile that is on an external USB storage device.

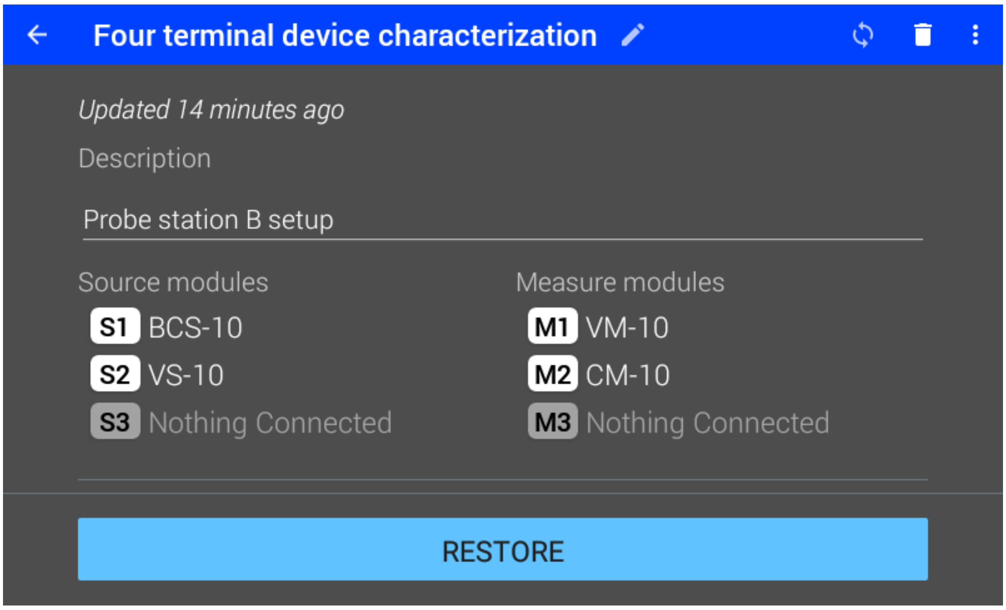


Figure 3.33: Editing a profile

3.13.4 Importing a Profile

To import a profile, first insert an external USB storage device into the back of the instrument. Then tap the import button located next to the restore button on the profiles overview page after the profiles on the drive are loaded. It is also possible to import a profile while on the details page by tapping the import button at the bottom of the screen.

By selecting the overflow menu (three dots in the top right corner of the screen) on the profiles overview page it is possible to import all profiles from an external storage device.

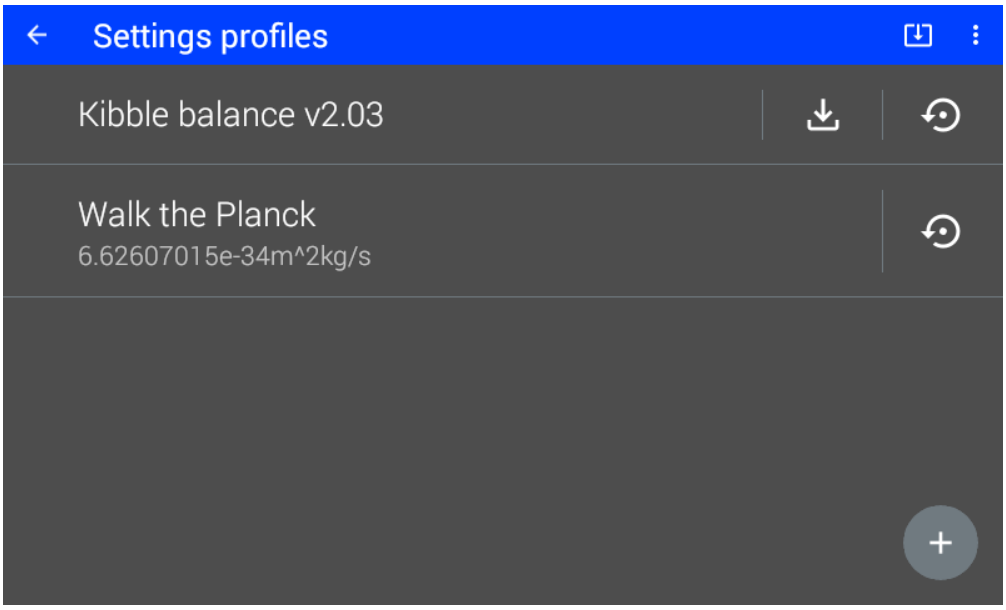


Figure 3.34: Importing a profile

3.13.5 Exporting a Profile

To export a profile, first insert an external USB storage device into the back of the instrument. Then, from a profile's details page, tap the export button next to the overflow menu (three dots in the top right corner of the screen).

It is also possible to export all profiles by selecting the overflow menu and selecting the export all option.

Caution: While it is possible to view and modify a settings profile when it is on an external storage device, it is strongly discouraged. Attempting to restore a profile that has been manually modified may have unintended effects.

3.13.6 Remote Interface

Settings profiles can also be managed via the remote interface:

- *PROFile:CREAte* (page 173)
- *PROFile:LIST?* (page 175)
- *PROFile:DESCription* (page 174)
- *PROFile:DESCription?* (page 174)
- *PROFile:JSON?* (page 175)
- *PROFile:REName* (page 176)
- *PROFile:UPDate* (page 177)
- *PROFile:RESTore* (page 176)
- *PROFile:RESTore[:VALid]?* (page 177)
- *PROFile:DELete* (page 173)
- *PROFile:DELete:ALL* (page 174)

3.14 Default Settings

3.14.1 Settings Reset

To reset to default values, open the settings menu (top left corner of the screen). Then tap System settings, then Reset.

The M81-SSM has two types of settings: instrumentation and system. Instrumentation settings includes settings that are unique to the instrument, such as shape, frequency, amplitude, and range. System settings are settings common to most instruments, such as volume, brightness, and TCP/IP settings.

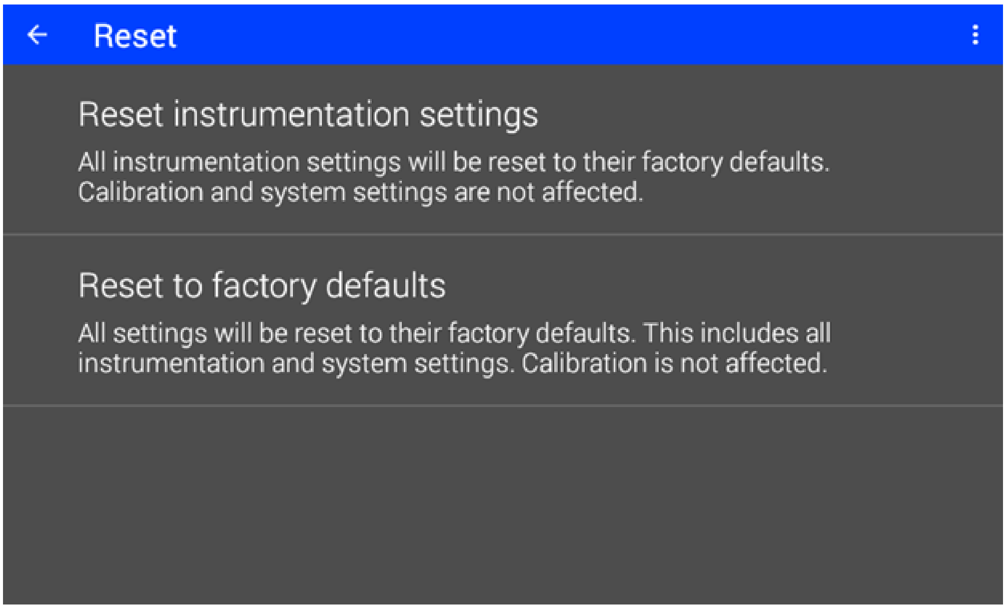


Figure 3.35: Reset page

The tables below list the settings of the M81-SSM upon initial power-on or after settings reset.

For the default settings of each module see the *Modules* (page 77) section.

To save and restore settings for the M81-SSM and connected modules, see the *Settings profiles* (page 60) section.

3.14.2 Instrumentation Settings

Reference Configuration

Setting	Default state
Reference in edge	Rising
Reference out	Disabled
Reference out source	S1

GPIO Configuration

Setting	Default state
Output 1 state	Disabled
Output 2 state	Disabled

Monitor Out Configuration

Setting	Default state
Monitor out	Disabled
Mode	M1
Manual level (for manual mode)	0 V

3.14.3 System Settings

Connectivity — Ethernet

Setting	Default state
SCPI over TCP	Disabled
TCP port	7777
IP configuration	DHCP
Static IP	192.168.0.12
Static gateway	192.168.0.1
Static subnet mask	255.255.255.0
DNS 1	8.8.8.8
DNS 2	8.8.4.4

Connectivity — Virtual Serial Port

Setting	Default state
Baud rate	921600
Data bits	8
Stop bits	1
Parity	None
Flow control	RTS/CTS

Connectivity — GPIB

Setting	Default state
Address	12

Display and Sound

Setting	Default state
Brightness	Suggested (50%)
Volume	50%
Front panel lock	Disabled
Dashboard	Cleared, normal panel size
Settings profiles	Cleared

Date and Time

Setting	Default state
Automatic date a time	Disabled

3.15 Parameter Sweeping

Note: Parameter sweeping is a beta feature and is subject to change.

The M81-SSM provides the ability to configure a high speed parameter sweep on either DC source amplitude or AC frequency. The source must be set to the appropriate shape for the configured parameter sweep. All source modules configured to sweep will do so synchronously. To set up a source module to sweep, the source mode must be changed from fixed to sweep. Any attached source module can be changed between fixed and sweep mode via the *SOURce#:VOLTage:MODE* (page 266), *SOURce#:CURRent:MODE* (page 224), or *SOURce#:FREQuency:MODE* (page 238) commands.

3.15.1 Sweep Set Up

To set up a parameter sweep the span, duration, and number of points must be defined.

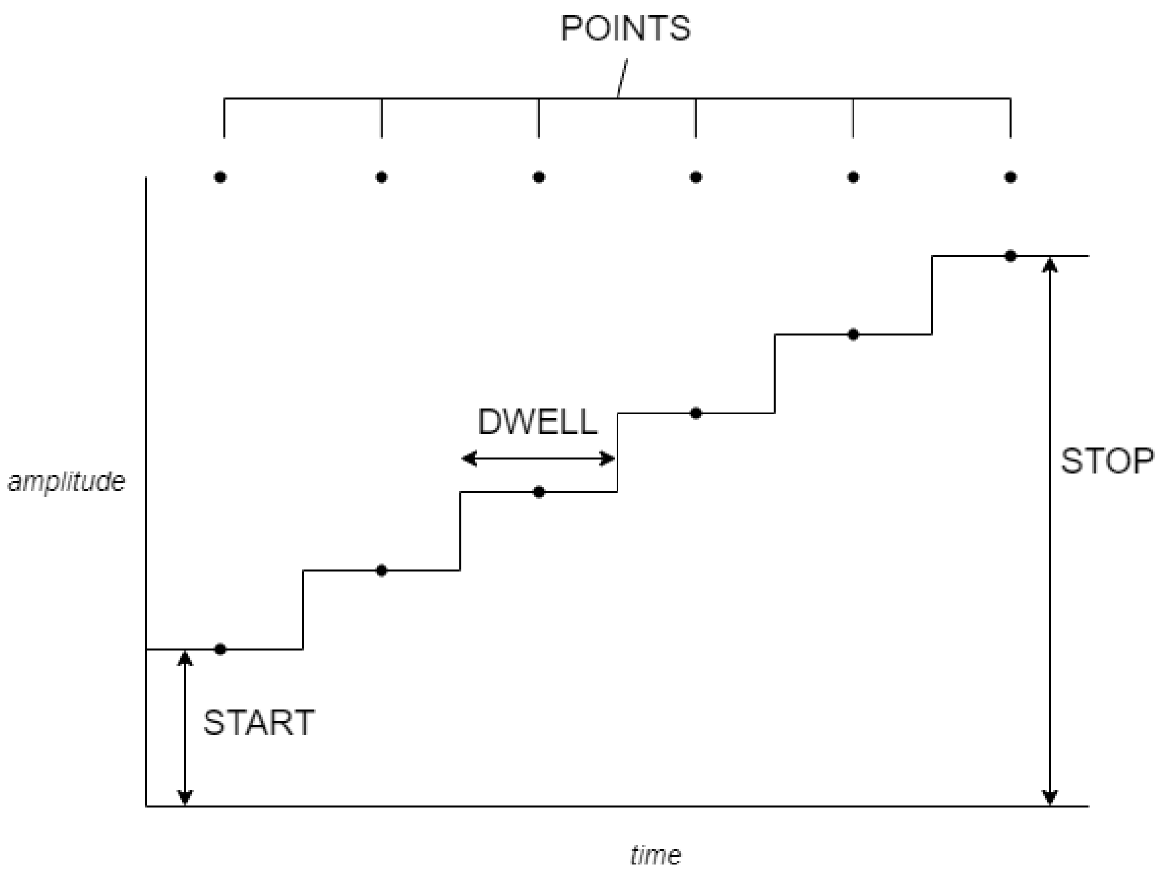


Figure 3.36: Sweeping settings

Span

The span is used when referring to the difference between the start and stop values. Span is configured by setting both the start and stop voltage or the start and stop current of the sweep. When setting the span, it is possible to set the stop value lower than the start value. This will result in a sweep that decreases in either voltage or current, depending on whether the module is sourcing voltage or current. The term span may be used elsewhere for the sake of explaining calculations.

Interface Commands:

SOURce#:VOLTage:STARt (page 274)

SOURce#:VOLTage:STOP (page 276)

SOURce#:CURRent:STARt (page 230)

SOURce#:CURRent:STOP (page 231)

SOURce#:FREQuency:STARt (page 239)

SOURce#:FREQuency:STOP (page 240)

Duration

Duration is used when referring to the length of the parameter sweep. The duration of the parameter sweep can be set in two seperate ways. Setting the dwell time will directly affect the overall run time of the sweep by setting the duration of each step in the sweep. Presently, dwell times are restricted to multiples of 200 µs per step, and must be a minimum of 200 µs. When a dwell time is set to a value that is invalid the instrument will adjust to the nearest valid value. For example, 0.0006 seconds is a valid dwell time, however, 0.0005 seconds is invalid. In this case, the instrument will apply the dwell time of 0.0006 seconds, which can be seen with the relevant query.

It is also possible to set the overall duration of the parameter sweep, allowing the instrument to calculate the closest dwell time to achieve the desired duration. The overall time is also bound by the same rules as dwell time.

Dwell time is calculated using the following formula: $DWELl = TIME/POINts$

Interface Commands:

SOURce#:SWEep:DWELl (page 258)

SOURce#:SWEep:TIME (page 261)

Points

The number of points defined in the stepped sweep. A higher value will create higher granularity in the span of the sweep, but will increase the total duration of the parameter sweep. Setting the number of points also directly affects the step size.

Alternatively, setting the step size of the sweep will set the number of points according to the following formula: $STEP = SPAN/(POINts - 1)$

It is important to remember that the number of points specified should include the ending value. For example, if the desired sweep is 0 Volts to 10 Volts with a step size of 1 Volt, the number of points should be 11.

Note: Step cannot be set when using logarithmic spacing.

Interface Commands:

SOURce#:SWEep:POINts (page 259)
SOURce#:CURRent:STEP (page 230)
SOURce#:VOLTage:STEP (page 275)
SOURce#:FREQuency:STEP (page 240)

Direction

The direction of the sweep controls whether the instrument sweeps up or down. While sweeping up, the sweep will begin at the start value and end at the stop value. Alternatively, when sweeping down, the sweep will begin at the stop value and end at the start value.

When round trip sweeping is enabled the instrument will begin and end at the same value. The instrument will repeat the middle value and double the number of points. For example a sweep with a start value of 1, a stop value of 5, and 5 points would have output values like so:

Direction: UP	Direction: DOWN
1	5
2	4
3	3
4	2
5	1
5	1
4	2
3	3
2	4
1	5

Interface Commands:

SOURce#:SWEep:DIRection (page 256)

SOURce#:SWEep:DIRection:RTRip[:STATe] (page 256)

Linear and logarithmic sweeps

The sweep steps may be linearly or logarithmically spaced. For example a sweep with a start value of 1, a stop value of 10, and 10 points would have output values like so:

Linear spacing	Logarithmic spacing
1	1
2	1.291549665
3	1.668100537
4	2.15443469
5	2.782559402
6	3.593813664
7	4.641588834
8	5.994842503
9	7.742636827
10	10

Logarithmic sweeps are useful when sweeping values over multiple orders of magnitude. It is important to note that logarithmic sweeps cannot include zero.

Interface Commands:

SOURce#:SWEep:SPACing (page 260)

3.15.2 Initiating the Sweep

By default, data sweeping is closely tied to data stream to allow for easy collection of data during parameter sweeps. To disable this behavior the *SWEep:INITiate:TRACe[:STATe]* (page 288) command can be used. After all of the sweep parameters have been configured, a data stream should be configured to capture relevant data. Once the data stream has been configured, calling *TRACe:STARt* (page 296) will start data streaming collection and simultaneously initiate any configured parameter sweeps.

To initiate a sweep without a data stream, use the *SWEep:INITiate* (page 288) command. Any configured sweeps will be initiated immediately. This command initiates the sweep regardless of the state set by *SWEep:INITiate:TRACe[:STATe]* (page 288). Initiating a sweep without a data stream is useful when ramping the output from one value to another to minimize discontinuities.

3.15.3 Parameter Sweep Status

To determine whether a parameter sweep is in progress use the following query: *SOURce#:SWEep[:STATus]?* (page 261)

3.15.4 Aborting a Sweep

To abort all sweeps in progress use the command *SWEep:ABORt* (page 287). Aborting leaves the output enabled and at the last sweep point applied. The following actions will also cause a sweep in progress to abort:

- Changing any setting on a source module while it is sweeping.
- Resetting a data stream in progress

3.15.5 Example Parameter Sweep and Data Stream

The following example illustrates how to configure a current parameter sweep on the first source channel and read the resulting measured voltage on the first measure channel.

Set the source module to a linear sweep.

```
SOURce1:SWEep:SPACing LINear
SOURce1:CURRent:MODE SWEep
```

Set the span of the sweep to start at 25 mA and end at 100 mA.

```
SOURce1:CURRent:STARt .025
SOURce1:CURRent:STOP .100
```

Set the dwell time to 5 ms per step.

```
SOURce1:SWEep:DWELl 0.005
```

Set the number of points to 100, which is equivalent to step size of 75 μA.

```
SOURce1:SWEep:POINts 100
```

Configure the data stream and start it to begin the sweep and collect the data. Set the data stream to provide information in a comma separated format. Set the data stream to provide the source channel 1 amplitude, measure channel 1 DC measurement, and measure channel 1 settling status. Set the data stream rate to 200 points per second to match the 5 ms dwell time.

```
TRACe:RESet
TRACe:FORMat:ENCOding CSV
TRACe:FORMat:ELEMents SAMP,1,MDC,1,MSET,1
TRACe:RATE 200
TRACe:STARt 100
```

3.16 Example Measurement Setups

The M81-SSM can be configured for a wide variety of measurements. The following examples show the recommended wiring for a few common applications.

3.16.1 Resistance Measurement

Current source with voltage measurement

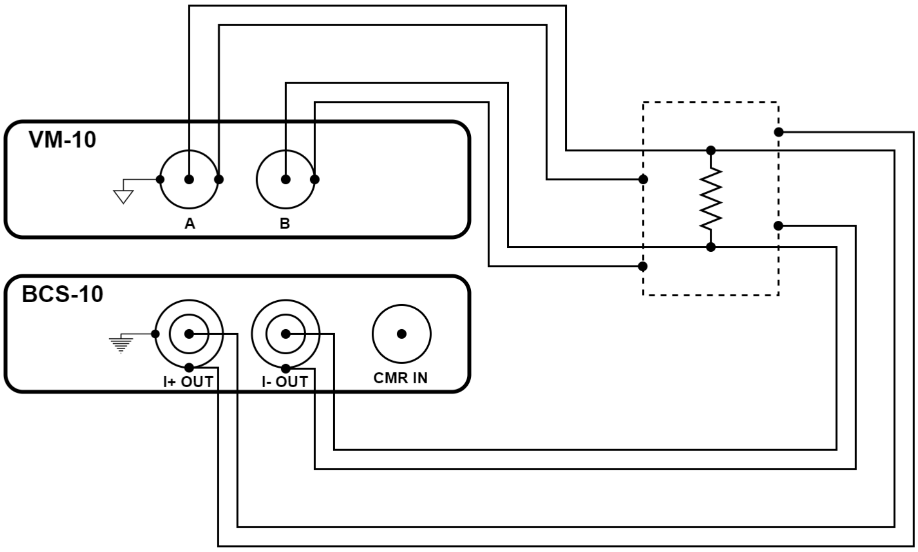


Figure 3.37: Resistance measurement using a BCS-10 module and a VM-10 module

The configuration illustrated above is a four-wire resistance measurement. It accurately determines resistance from milliohms to over a hundred gigaohms. Guarded tri-axial current source connections improve accuracy and settle time when measuring large resistors. This configuration resolves micro-ohm changes when set up to do lock-in measurements.

The configuration uses the *balanced current source (BCS-10) module* (page 81) to provide current and measures the resulting voltage with the *voltage measure (VM-10) module* (page 87).

Earth and measure ground are connected together at the enclosure. The outer shell of the BCS-10 triaxial connectors provides earth ground while the outer shell of the VM-10 BNC connectors provides measure ground.

Voltage source with current measurement

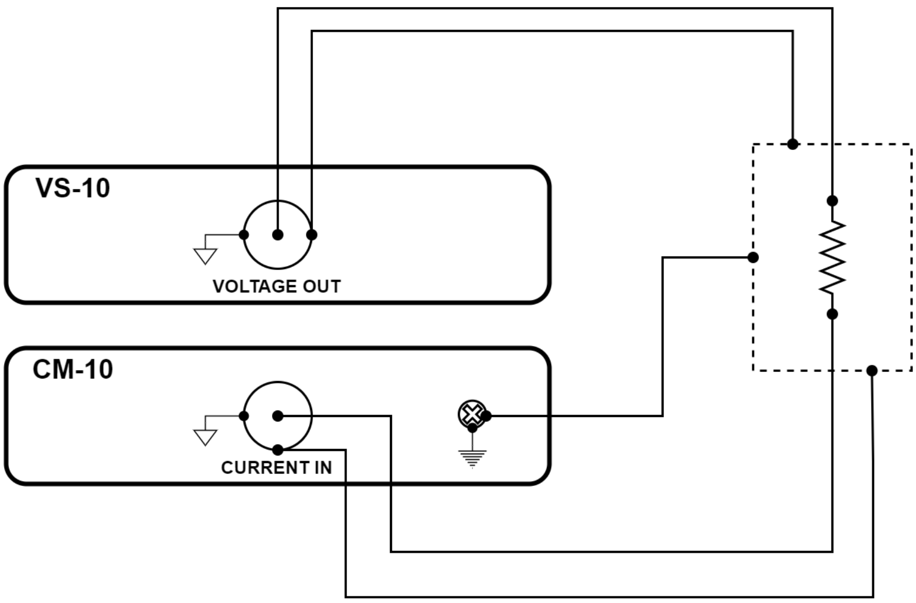


Figure 3.38: Resistance measurement using a VS-10 module and a CM-10 module

The configuration illustrated above is a two-wire resistance measurement. It is capable of accurately determining resistance from a few ohms to over a hundred gigaohms. This configuration can resolve micro-ohm changes when set up to do lock-in measurements.

The configuration uses the *voltage source (VS-10) module* (page 77) to provide voltage and measures the resulting current with the *current measure (CM-10) module* (page 92).

Earth and measure ground are connected together at the enclosure. Earth ground is connected using a screw on the rear of the CM-10 module. The BNC shells of the modules provide measure ground.

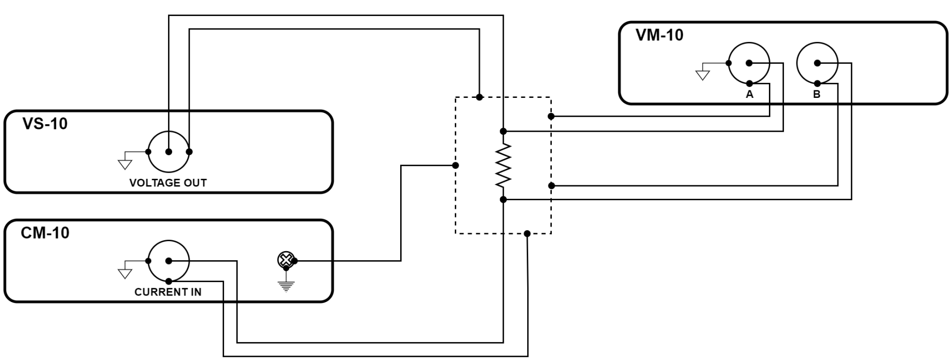


Figure 3.39: Resistance measurement using VS-10, CM-10, and VM-10 modules

Adding a CM-10 module to this configuration provides a four-wire measurement. This improves the accuracy of measuring small resistances.

3.16.2 Diode Characterization

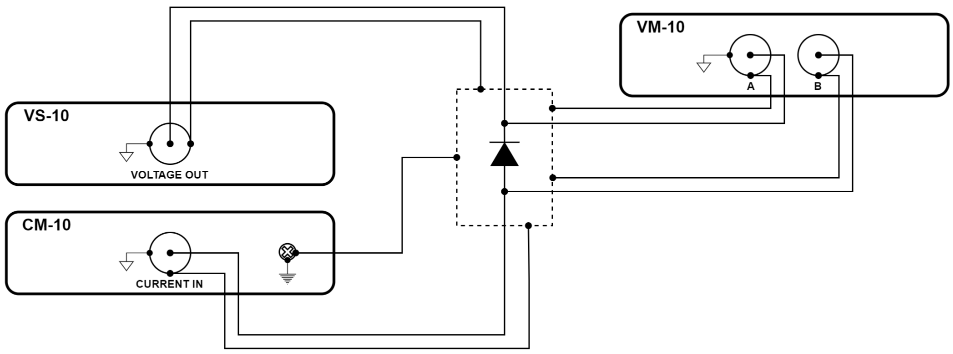


Figure 3.40: Diode measurement using VS-10, CM-10, and VM-10 modules

The configuration illustrated above is a four-wire diode measurement. It is capable of quickly and accurately determining the I-V characteristics of a diode.

The configuration uses the *voltage source (VS-10) module* (page 77) to provide voltage and measures the resulting current with the *current measure (CM-10) module* (page 92). A *voltage measure (VM-10) module* (page 87) creates a four-wire measurement to improve accuracy when the diode is conducting large amounts of current.

The *parameter sweeping* (page 67) feature and the *data streaming* (page 116) feature allow for rapid I-V characterization. Lock-in capabilities allow for highly sensitive measurements.

Earth and measure ground are connected together at the enclosure. Earth ground is connected using a screw on the rear of the CM-10 module. The BNC shells of the modules provide measure ground.

3.16.3 Transistor Characterization

Field Effect Transistor

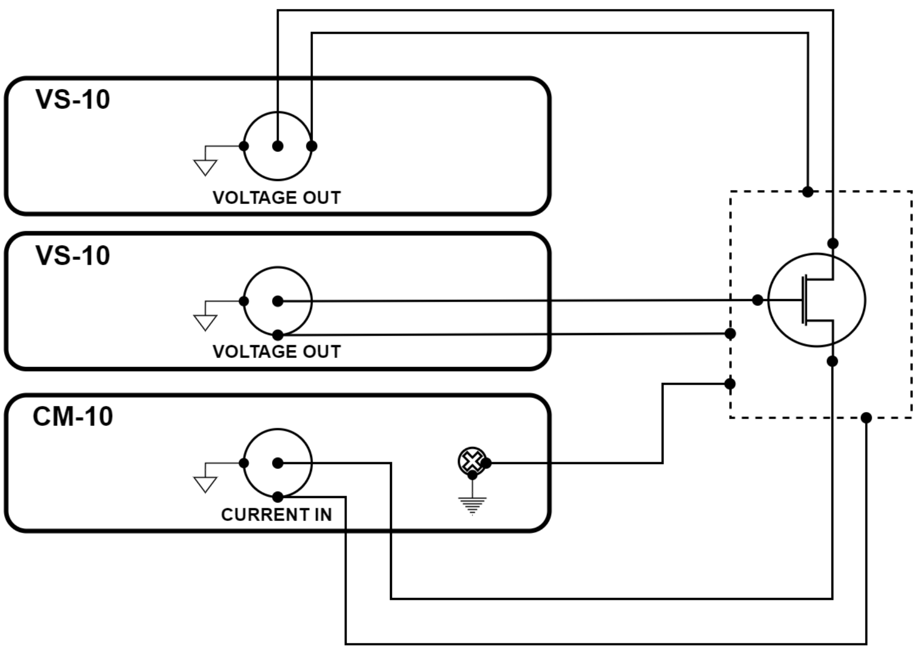


Figure 3.41: FET transistor measurement using VS-10, CM-10, and VM-10 modules

The configuration illustrated above is a FET transistor measurement. This configuration can quickly and accurately determine the I-V characteristics of a FET transistor.

The configuration uses two *voltage source (VS-10) modules* (page 77) to provide voltage to the gate and drain terminals. The *current measure (CM-10) module* (page 92) holds the source terminal at zero volts.

The *parameter sweeping* (page 67) feature and the *data streaming* (page 116) feature allow for rapid I-V characterization. Lock-in capabilities allow for highly sensitive measurements.

Earth and measure ground are connected together at the enclosure. Earth ground is connected using a screw on the rear of the CM-10 module. The BNC shells of the modules provide measure ground.

CHAPTER

FOUR

MODULES

This chapter provides operational instructions for the specific modules in the Lake Shore M81-SSM Synchronous Source Measure System.

4.1 VS-10

The VS-10 is a DC and AC voltage source that provides a high precision voltage output. The VS-10 provides voltage up to 10 V and 100 mA.

The VS-10 provides:

- Configurable ranges from 10 V to 10 mV
- Separate AC and DC ranging
- Configurable current limit

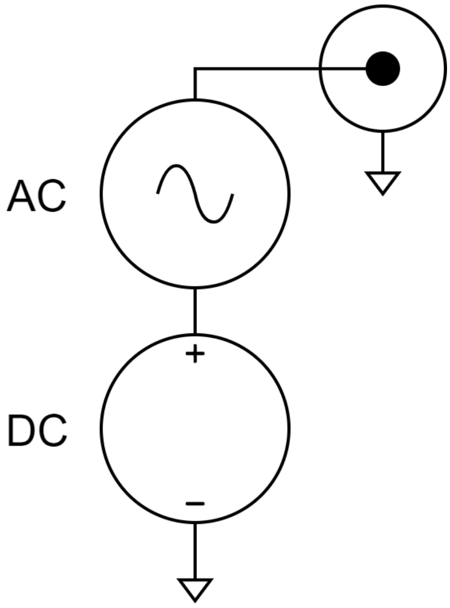


Figure 4.1: Module equivalent diagram

The settings for the VS-10 can be configured on the page for the module:

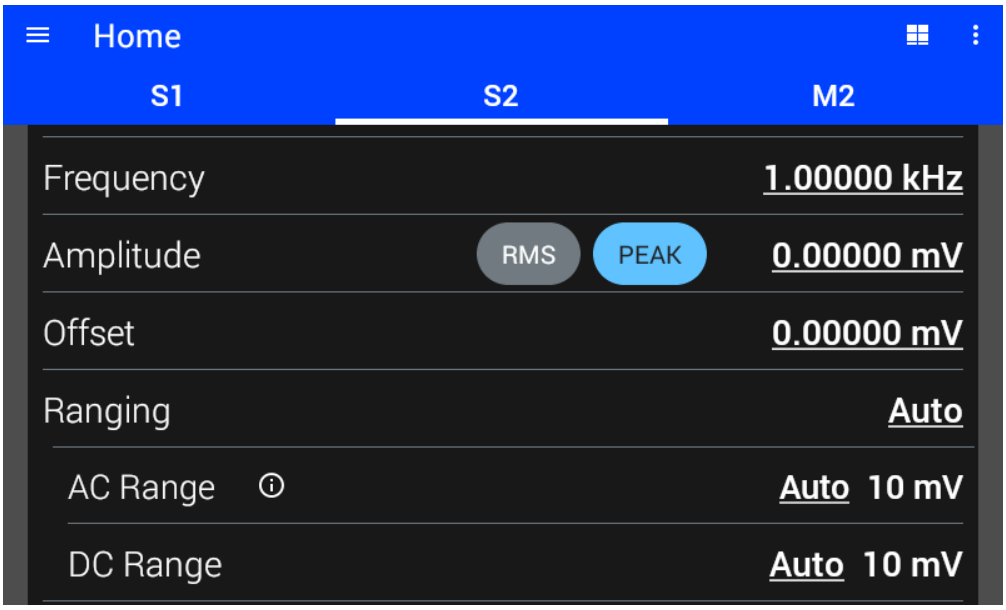


Figure 4.2: VS-10 settings

4.1.1 Range

The range determines the largest voltage that can be sourced by the module. In general, the lowest range that can be used will provide the best performance. See instrument specifications for the performance characteristics of each range.

The VS-10 has the capability to use different ranges for the AC and DC signals. This can be used to combine low level AC excitation with a larger DC excitation.

If auto is selected, the M81-SSM will select the lowest voltage range that will support the amplitude and offset each time they are configured.

The available ranges are 10 V, 1 V, 100 mV, and 10 mV.

Interface Commands:

SOURce#:VOLTage:RANGe:AUTO (page 271)

SOURce#:VOLTage:RANGe:DC (page 272)

SOURce#:VOLTage:RANGe:AC (page 270)

4.1.2 Current Protection Level

A hardware current protection level can be configured to limit the amount of current that the VS-10 will provide. This is a DC current limit and cannot limit high frequency (>100 Hz) outputs.

By definition, a voltage source will adjust the current it provides in order to maintain a constant voltage across a load. When the current protection level is reached, the source acts as a constant current source. The source will remain in this state until the output voltage is reduced, or the load resistance is increased.

Interface Command: *SOURce#:CURRent:PROTection* (page 225)

4.1.3 High and low voltage output limits

The voltage output limits are software limits preventing the user from entering an output that could potentially damage the module's load.

When the shape is not DC, the limit is applied to the sum of the offset and amplitude. Low voltage output limit is the minimum voltage that a user will be allowed to enter, while the high voltage output limit is the maximum voltage the user will be allowed to enter. Both voltage limits are bound between -10 V and 10 V with the expectation that the lower voltage output limit must be a lower value than the high voltage output limit.

Interface Command: *SOURce#:VOLTage:LIMit:HIGH* (page 264) Interface Command: *SOURce#:VOLTage:LIMit:LOW* (page 265)

4.1.4 Default Settings

The table below lists the settings of the VS-10 upon initial power on or after settings are reset. For more information about default settings see the *Default settings* (page 64) section.

Setting	Default state
Shape	Sine
Frequency	1 kHz
Amplitude	0 mV
Offset	0 mV
Ranging	Auto
AC range	Auto, 10 mV
DC range	Auto, 10 mV
Current protection level	100 mA
Sync	Disabled
Sync source	S1 (or S2 if module is S1)
Sync phase shift	0 degrees
Dark mode	Disabled
High voltage output limit	10 V
Low voltage output limit	-10 V

4.2 BCS-10

The BCS-10 is a unique type of current source called a balanced current source. It provides up to 100 mA of DC and AC output current.

The BCS-10 provides:

- A balanced current output
- Configurable ranges from 100 mA to 10 nA
- Common mode control
- Configurable AC coupling

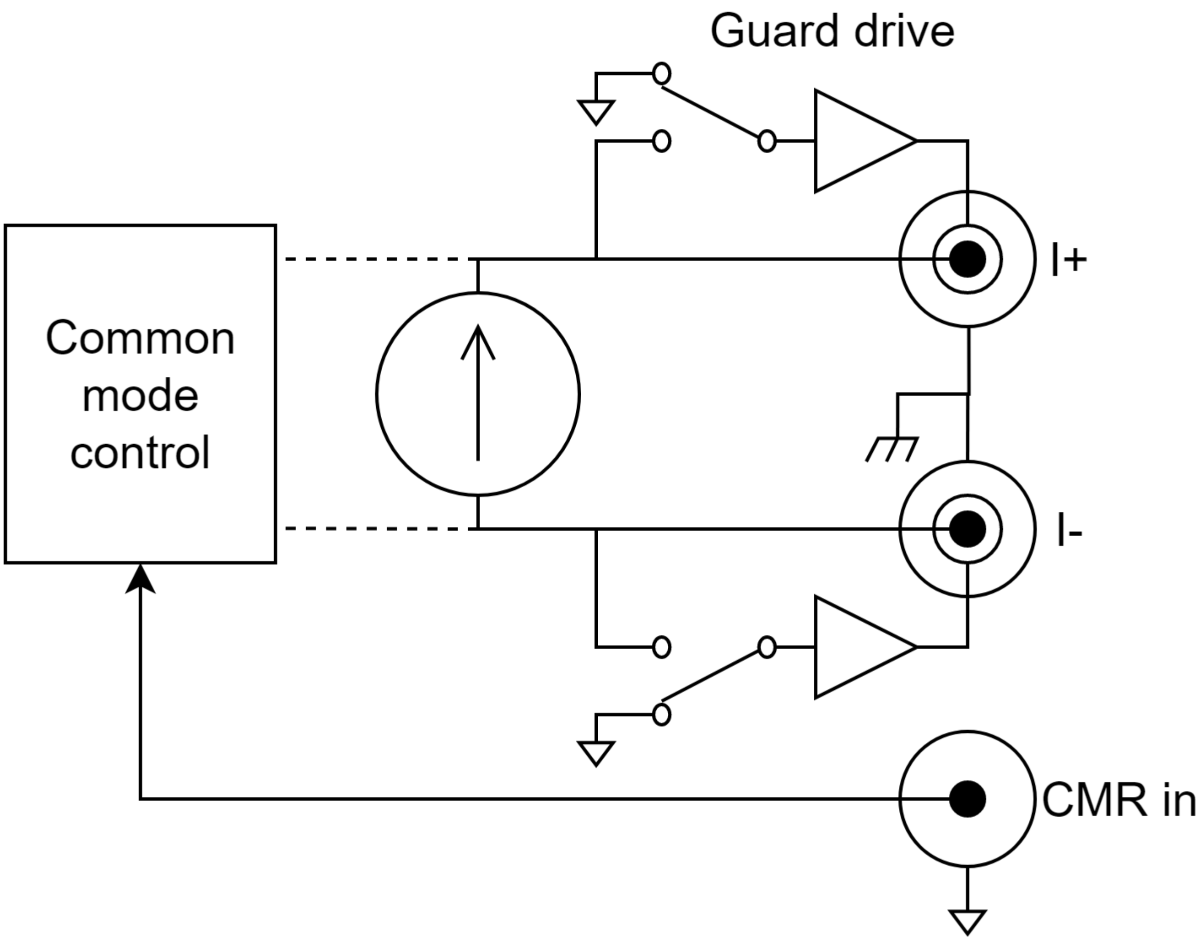


Figure 4.3: Module equivalent diagram

The settings for the BCS-10 can be configured on the page for the module:

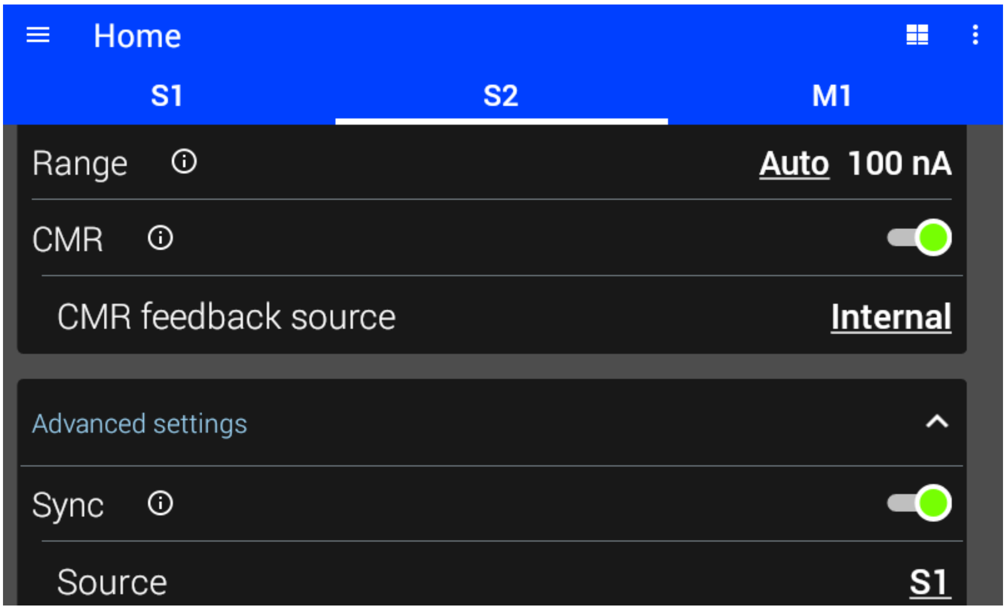


Figure 4.4: BCS-10 settings

Caution: If the BCS-10 is disconnected from a load, it will output the full compliance voltage of 20V. For sensitive devices, ensure that the source is disabled when connecting the module to the device. For devices that are sensitive to, or may be damaged by such voltages, the user must take measures to ensure that the source is disabled prior to disconnection from and/or re-connection to the device under test or add external voltage limiting devices if such operation cannot be ensured.

4.2.1 Range

The range determines the largest current that can be sourced by the module. In general, the lowest range that can be used will provide the best performance. See instrument specifications for the performance characteristics of each range.

If auto is selected, the M81-SSM will select the lowest current range that will support the amplitude and offset each time they are configured.

The available ranges are 100 mA, 10 mA, 1 mA, 100 µA, 10 µA, 1 µA, 100 nA, and 10 nA.

Interface Commands:

SOURce#:CURRent:RANGe:AUTO (page 227)
SOURce#:CURRent:RANGe (page 226)

4.2.2 Coupling

If coupling is set to AC, signals below 1.6 Hz are blocked from the output. This minimizes offsets on the output if DC and low frequency signals are not needed. When coupling is set to DC, the BCS-10 has a low output impedance at low frequencies. DC coupling is needed when sourcing low frequencies or output shapes that have DC components such as square waves.

Interface Command: *SOURce#:COUPling* (page 220)

4.2.3 CMR

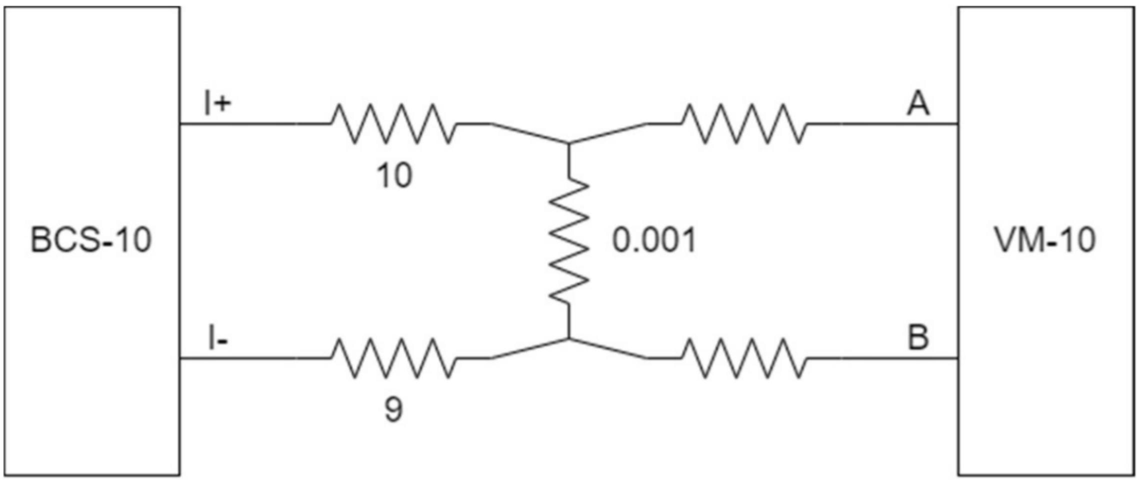
The common mode reduction (CMR) servo circuit controls the common mode voltage on the differential output of the current source.

Should CMR be on or off?

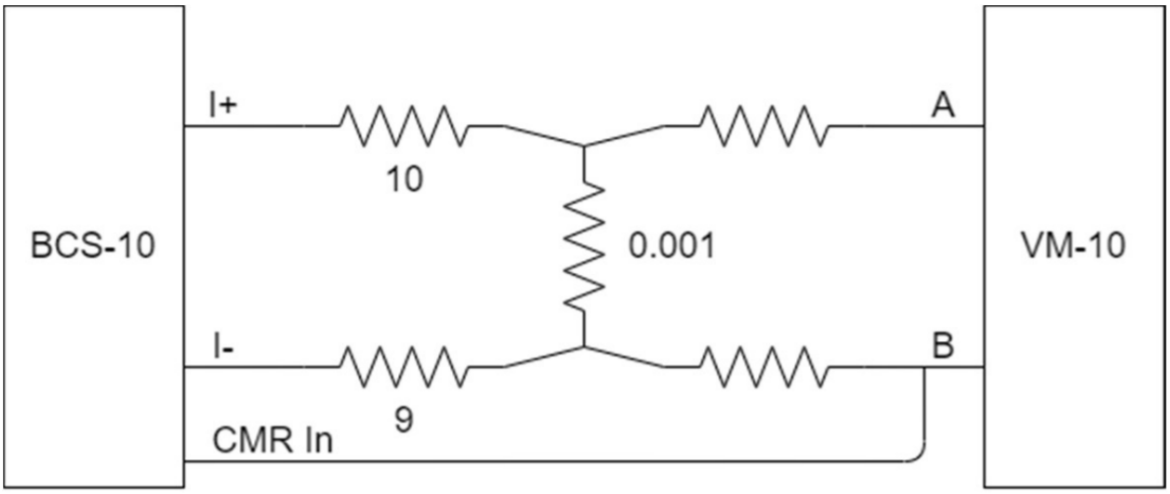
The BCS-10 is unique in that both the I+ and I- outputs are high impedance. This means that both the I+ and I- leads can be at any voltage between +10 V and -10 V. Something needs to determine what this voltage will be. In many cases, there is no external low impedance path to measure common (for example in a 4-wire resistor measurement with the VM-10). In such cases, CMR should be enabled. This means that the BCS will determine the operating voltage. In other cases, the load of the BCS may be grounded externally. In such cases, CMR should be disabled.

If CMR is on, should I use internal or external CMR?

In most applications, internal CMR is suitable. However, external CMR can be used to reduce errors arising from common mode voltage. This is especially relevant for low resistance measurements. Consider a 4 wire resistor measurement like the following:



Suppose the BCS-10 is supplying a 100 mA excitation current. This current through a 1 mΩ device under test will produce a differential voltage of 100 µV. However, with internal CMR mode, the BCS will produce an average voltage of 0 on its output terminals. In doing so, it will produce a common mode voltage of 100 mV because of the 1 Ω difference in the current leads. The VM-10 now has to measure a 100 µV signal in the presence of 100 mV of common mode. The VM-10 has the ability to reject at least 80 dB of common mode. Even with this rejection, the VM will indicate 110 µV, resulting in a 10% error. This error can be resolved by using external CMR, connected as follows:



In external CMR mode, the BCS-10 will adjust the I+ and I- outputs to produce 0 V at the CMR sense point, which has been connected to the VM-10 B input. Now the VM-10 is presented with only 100 µV of common mode voltage, significantly reducing the error.

Interface Commands:

SOURce#:CMR[:STATe] (page 220)

SOURce#:CMR:SOURce (page 219)

4.2.4 Guard Drive

If Guards are enabled, the inner sheath is driven with the voltage present on the center conductor. The guard minimizes effective capacitance and leakage current. In either setting, the guard should not be connected to ground.

Interface Command: *SOURce#:GUARd* (page 245)

4.2.5 Disable On Overload

If Disable on Overload is enabled, the BCS-10 firmware will automatically disable the output when an overload is detected. Note that this feature is implemented in firmware, and will not respond as instantaneous as a hardware implemented compliance would.

Interface Command: *SOURce#:DOCompliance* (page 237)

4.2.6 High and Low Current Output Limits

The current output limits are software limits preventing the user from entering an output that could potentially damage the module's load.

When the shape is not DC, the limit is applied to the sum of the offset and amplitude. Low current output limit is the minimum current that a user will be allowed to enter, while the high current output limit is the maximum current the user will be allowed to enter. Both current limits are bound between -100 mA and 100 mA with the expectation that the lower current output limit must be a lower value than the high current output limit.

Interface Command: *SOURce#:CURRent:LIMit:HIGH* (page 222) Interface Command: *SOURce#:CURRent:LIMit:LOW* (page 223)

4.2.7 Default Settings

The table below lists the settings of the BCS-10 upon initial power on or after settings are reset. For more information about default settings see the *Default settings* (page 64) section.

Setting	Default state
Shape	Sine
Frequency	1 kHz
Amplitude	0 pA
Offset	0 pA
Ranging	Auto, 100 nA
Common mode reduction (CMR)	Enabled
CMR feedback source	Internal
Sync	Disabled
Sync source	S1 (or S2 if module is S1)
Sync phase shift	0 degrees
Guard drive	Disabled
Coupling	Auto, DC
Dark mode	Disabled
Disable on Overload	Disabled
High current output limit	100 mA
Low current output limit	-100 mA

4.3 VM-10

The VM-10 is a single-ended or differential voltage measurement amplifier that can operate up to 10 V.

The VM-10 provides:

- Single-ended (A) or differential (A-B) voltage measurement
- Configurable ranges from 10 V to 10 mV
- Seamless range changes
- Configurable AC input coupling
- A configurable hardware filter

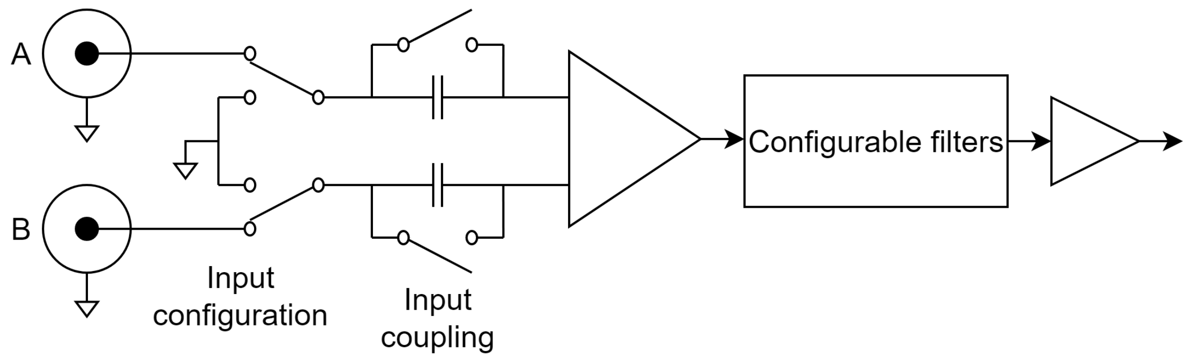


Figure 4.5: Module equivalent diagram

The settings for the VM-10 can be configured on the page for the module:

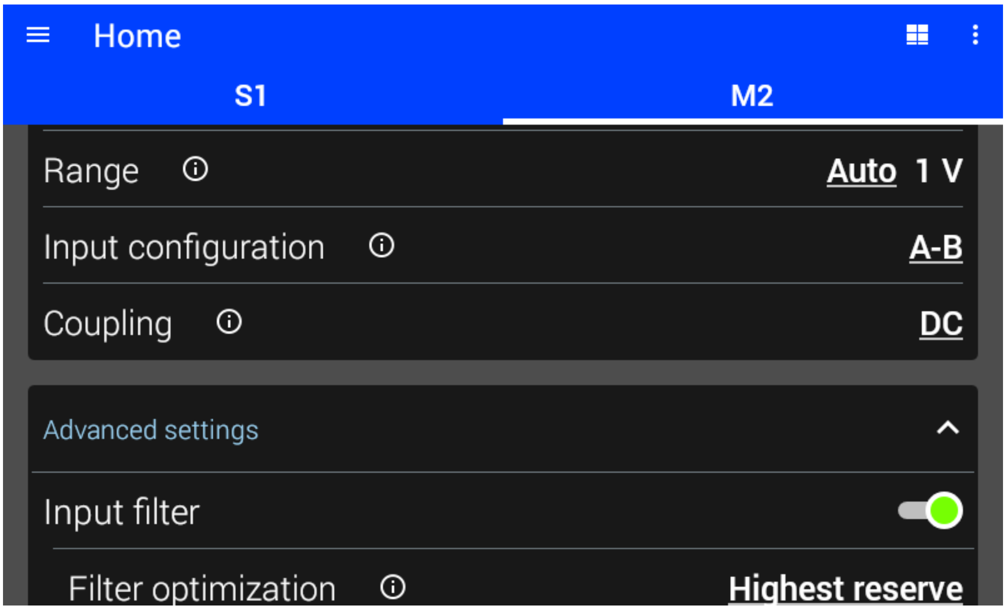


Figure 4.6: VM-10 settings

4.3.1 Range

The range determines the largest voltage that can be measured by the module. In general, the lowest range that can be used will provide the best performance. See instrument specifications for the performance characteristics of each range. The list of range bandwidths can be located on the instrument by tapping the info circle on the range setting.

If auto is selected, the VM-10 will automatically change ranges based on the input signal. The auto ranging algorithm uses the maximum and minimum peak measurements to make decisions. The algorithm will observe the signal over the specified averaging time or time constant before making a decision. Therefore it will take longer to change ranges when the averaging time is longer, for example when there is a large number of NPLCs.

Available ranges are 10 V, 1 V, 100 mV, and 10 mV.

Note: The 10 V and 1 V ranges are not available when the input filter is enabled and filter optimization is set to highest reserve.

Interface Commands:

SENSe#:VOLTage:RANGe:AUTO (page 216)

SENSe#:VOLTage:RANGe (page 216)

4.3.2 Input Configuration

The input configuration determines what voltage is measured by the VM-10.

- **A-B** The VM-10 measures the difference between the voltage on input A and input B.
- **A** The VM-10 measures the voltage between input A and measure ground.
- **Ground** The input is internally connected to measure ground.

The indicator LEDs on the VM-10 will change to indicate what inputs are presently being used by the measurement.

Interface Command: *SENSe#:CONFiguration* (page 188)

4.3.3 Input Coupling

If Coupling is set to AC, a high pass filter with a cut-off frequency of 0.16 Hz is engaged. This blocks DC signals from being amplified but may reduce measurement accuracy.

Interface Command: *SENSe#:COUPling* (page 189)

Note: The VM-10's input bias current will cause a small DC offset voltage when coupling is set to AC. AC coupling should not be used when DC measurements are needed.

4.3.4 Input Filter

The VM-10 has the capability to filter the input signal. Both a high pass and low pass filter are available.

This filter is a hardware filter located in the module. This is in addition to any digital filtering configured in the M81-SSM. This is useful for rejecting large interfering signals and preventing them from being amplified.

Filter Optimization

In *lowest noise* mode, some gain is placed before the filters. This configuration provides the lowest noise, but causes overloads if an interfering signal is too large.

In *highest reserve* mode, all gain will be placed after the filters. This configuration can tolerate the largest interference, but the noise is higher.

Note: The 10 V and 1 V measurement ranges are not available when the input filter is enabled and filter optimization is set to highest reserve.

Optimization

The VM-10 can apply gain before and/or after the filter. The gain allocation is controlled with the Filter optimization setting.

In Lowest noise mode, some gain is placed before the filters. This configuration provides the lowest noise, but causes overloads if an interfering signal is too large.

In Highest reserve mode, all gain will be placed after the filters. This configuration can tolerate the largest interference, but the noise is higher.

Interface Commands:

SENSe#:FILTer[:STATe] (page 198)

SENSe#:FILTer:OPTimization (page 197)

SENSe#:FILTer:HPASs:FREQuency (page 194)

SENSe#:FILTer:HPASs:ATTenuation (page 194)

SENSe#:FILTer:LPASs:FREQuency (page 196)

SENSe#:FILTer:LPASs:ATTenuation (page 195)

Corner Frequency

The corner frequency can be selected from the listed options for each filter. If the corner frequency is set to None, the filter is disabled.

Rolloff

Each filter can be configured for single pole (6 dB/oct) or two pole (12 dB/oct) operation.

4.3.5 Default Settings

The table below lists the settings of the VM-10 upon initial power on or after settings are reset. For more information about default settings see the *Default settings* (page 64) section.

Setting	Default state
Mode	DC
Averaging time	1 NPLC
Range	Auto, 10 mV
Input configuration	A-B
Coupling	DC
Input filter	Disabled
Filter optimization	Lowest noise
Low pass corner frequency	None
Low pass rolloff	6 dB
Dark mode	Disabled

4.4 CM-10

The CM-10 is a transimpedance amplifier that can measure currents up to 100 mA.

The CM-10 provides:

- Configurable ranges from 100 mA to 1 nA
- Settable bias voltage
- A configurable hardware filter

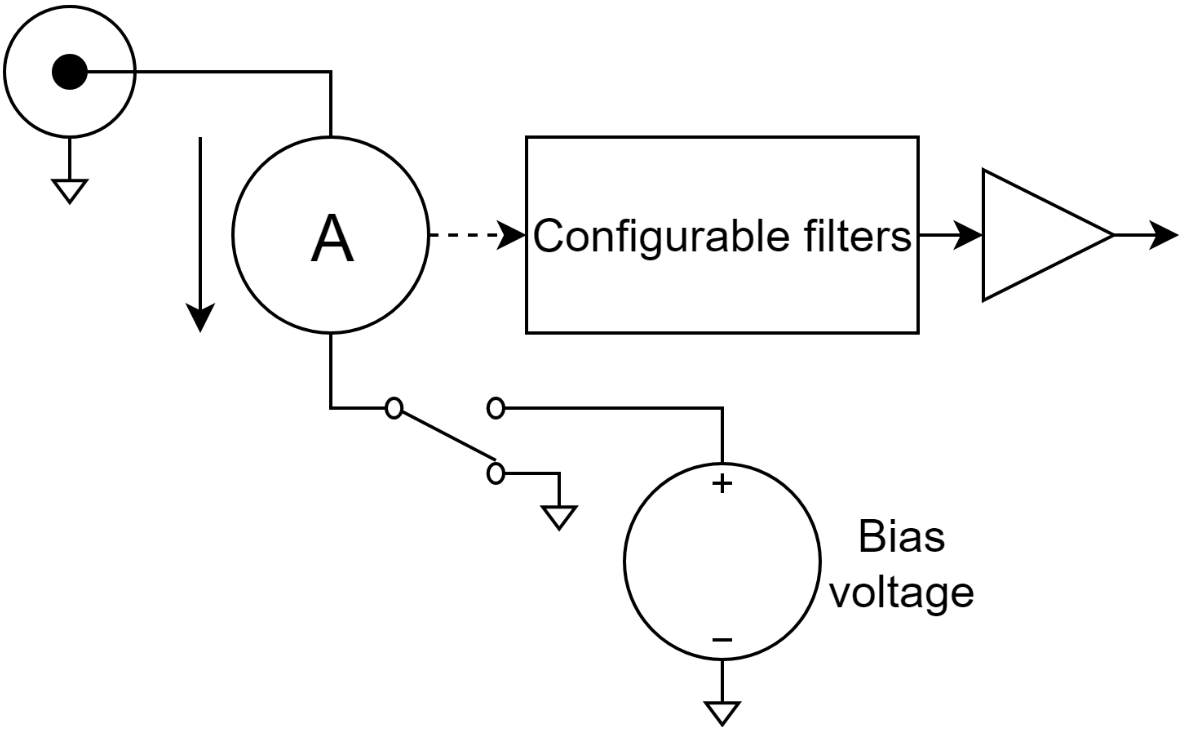


Figure 4.7: Module equivalent diagram

Note: The BNC shell is measure common.

The settings for the CM-10 can be configured on the page for the module:

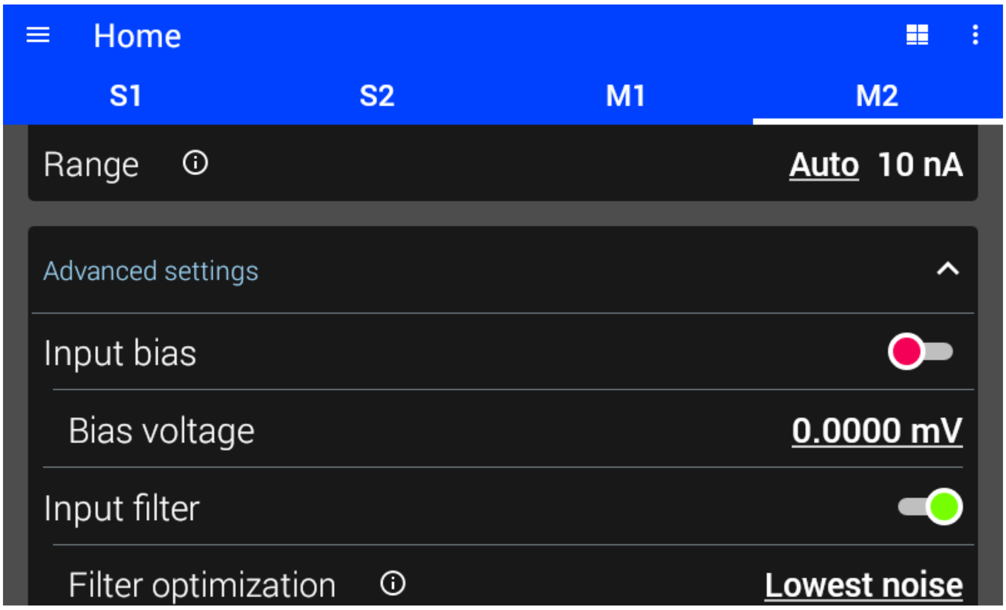


Figure 4.8: CM-10 settings

4.4.1 Range

The range determines the largest current that can be measured by the module. In general, the lowest range that can be used will provide the best performance. See instrument specifications for the performance characteristics of each range. The list of range bandwidths can be located on the instrument by tapping the info circle on the range setting.

If auto is selected, the CM-10 will automatically change ranges based on the input signal. The auto ranging algorithm uses the maximum and minimum peak measurements to make decisions. The algorithm will observe the signal over the specified averaging time or time constant before making a decision. Therefor it will take longer to change ranges when the averaging time is longer, for example when there is a large number of NPLCs.

Available ranges are 100 mA, 10 mA, 1 mA, 100 μA, 10 μA, 1 μA, 100 nA, 10 nA, and 1 nA.

Interface Commands:

SENSe#:CURRent:RANGe:AUTO (page 190)

SENSe#:CURRent:RANGe (page 189)

4.4.2 Input Bias Voltage

When the Bias Voltage is disabled, the input node of the CM-10 is a virtual ground.

The Bias Voltage can be enabled, which drives the set Bias Voltage on the input node. This can be used to bias a connected device.

Note: If Input bias is enabled in the 100 mA range the span of measurable current is reduced proportional to the Bias voltage.

Interface Commands:

SENSe#:BIAS[:STATe] (page 187)
SENSe#:BIAS:VOLTage[:DC] (page 186)

4.4.3 Input Filter

The CM-10 has the capability to filter the input signal. Both a high-pass and low-pass filter are available.

The input filter is a hardware filter located in the module, which is in addition to any digital filtering configured in the M81-SSM. This is useful for rejecting large interfering signals and preventing them from being amplified.

Filter Optimization

In *Lowest noise* mode, some gain is placed before the filters. This configuration provides the lowest noise, but causes overloads if an interfering signal is too large.

In *Highest reserve* mode, all gain will be placed after the filters. This configuration can tolerate the largest interference, but the noise is higher.

Note: The 100 mA range is not available when the input filter is enabled and filter optimization is set to highest reserve.

Optimization

The CM-10 can apply gain before and/or after the filter. The gain allocation is controlled with the Filter optimization setting.

In Lowest noise mode, more gain is placed before the filters. This configuration provides the lowest noise, but causes overloads if an interfering signal is too large.

In Highest reserve mode, more gain will be placed after the filters. This configuration can tolerate the largest interference, but the noise is higher.

Interface Commands:

SENSe#:FILTer[:STATe] (page 198)
SENSe#:FILTer:OPTimization (page 197)
SENSe#:FILTer:HPASs:FREQuency (page 194)
SENSe#:FILTer:HPASs:ATTenuation (page 194)
SENSe#:FILTer:LPASs:FREQuency (page 196)
SENSe#:FILTer:LPASs:ATTenuation (page 195)

Corner Frequency

The corner frequency can be selected from the listed options for each filter.

If the corner frequency is set to None, the filter is disabled.

Rolloff

Each filter can be configured for single-pole (6 dB/oct) or two-pole (12 dB/oct) operation.

4.4.4 Frequency Range Threshold

The frequency range threshold defines the max allowable input signal frequency for a given current range. When the input current range is set to Auto, a current range will be automatically chosen depending on the signal's frequency and the range's specified -3 dB bandwidth. All ranges of the CM-10 have frequency limitations. These limitations can be viewed by tapping the info circle on the module's range setting.

The threshold value is normalized to the -3 dB bandwidth of each range and can be set from 0.0 to 3.0 (default is 0.1, which means 10 % of the -3 dB point). For example, since a 10 μA range has a -3 dB bandwidth of 8 kHz, a threshold setting of 0.1 will allow a frequency up to 800 Hz before the current range is increased to a range that has a higher -3 dB bandwidth.

4.4.5 Default Settings

The table below lists the settings of the CM-10 upon initial power on or after settings are reset. For more information about default settings see the *Default settings* (page 64) section.

Setting	Default state
Mode	DC
Averaging time	1 NPLC
Range	Auto, 1 nA
Input bias	Disabled
Bias voltage	0.0 mV
Input filter	Disabled
Filter optimization	Lowest noise
Low pass corner frequency	None
Low pass roll off	6 dB
Dark mode	Disabled

4.5 BNC Adapter

The BNC Adapter cable connects to either a source or a measure port. The BNC Adapter is useful in cases where the capabilities of a module are not needed, such as connecting to a specialized amplifier. For example, the BNC Adapter could connect to the analog input of a magnet power supply, allowing the M81 to control its output.

When connected to a source port, it acts as a single-ended voltage source with a fixed output range of ±10V.

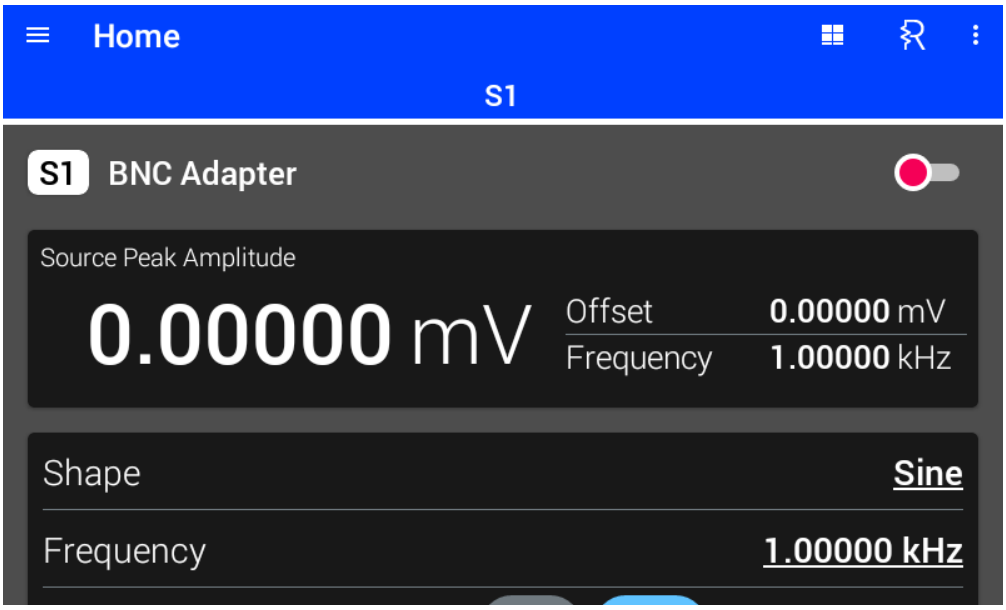


Figure 4.9: BNC Adapter connected to a source port

When connected to a measure port, it acts as a single-ended voltage measurement with a fixed input range of ±10V.

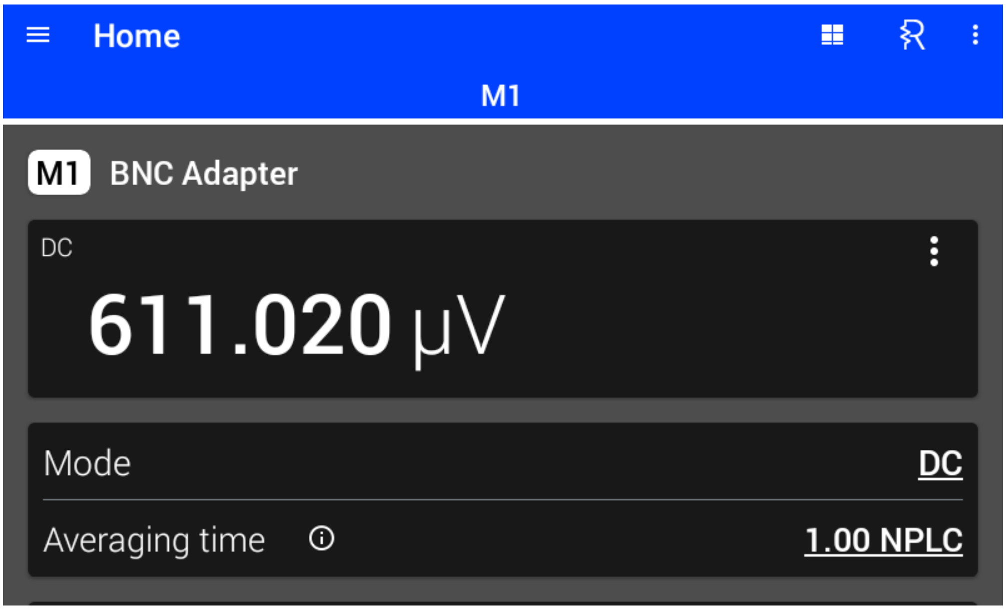


Figure 4.10: BNC Adapter connected to a measure port

4.5.1 Default Settings

When connected to a source port

The table below lists the settings of the BNC Adapter upon initial power on or after settings are reset when connected to a source port.

Setting	Default state
Shape	Sine
Frequency	1 kHz
Amplitude	0 mV
Offset	0 mV

When connected to a measure port

The table below lists the settings of the BNC Adapter upon initial power on or after settings are reset when connected to a measure port.

Setting	Default state
Mode	DC
Averaging time	1 NPLC

For more information about default settings see the *Default settings* (page 64) section.

CHAPTER FIVE

FIRMWARE UPDATE

This chapter provides instructions for updating the MeasureReady™ M81-SSM Synchronous Source Measure System firmware. There are two options for updating the instruments firmware:

- Connect to the internet (recommended)
- Provide the update via USB drive

5.1 Connected Updates

First, the MeasureReady™ M81-SSM must be connected to the internet via the *ethernet interface* (page 108).

Lake Shore recommends enabling the "Notify when updates are available" setting to ensure your M81-SSM is up-to-date with the latest features and updates. If used with a valid network connection, the M81-SSM will display a notification on the notification bar as soon as an update is available, but will not apply the update until instructed.

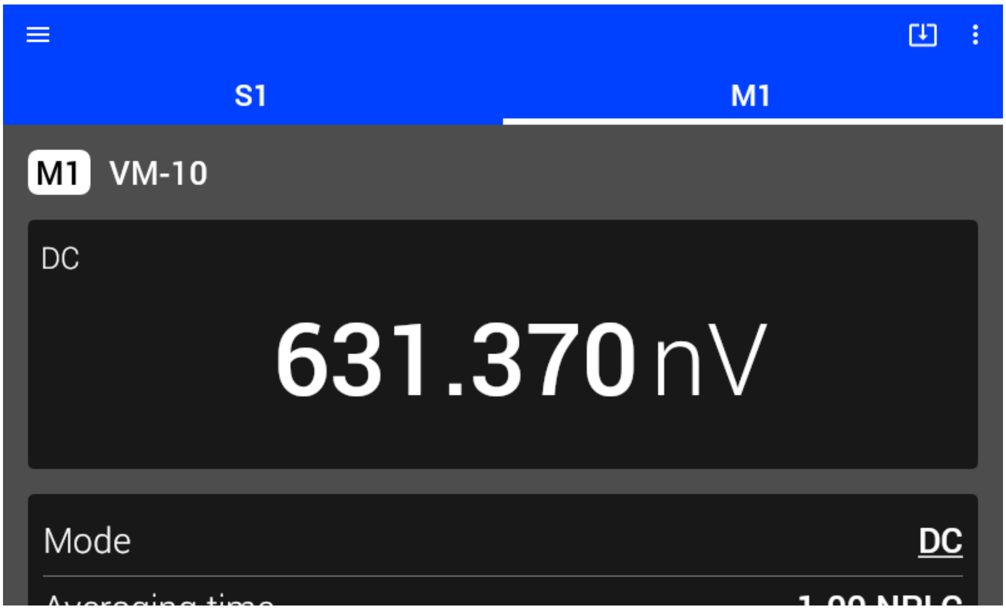


Figure 5.1: Firmware update notification

If the M81-SSM notifies you that a firmware update is available, tap "Update available". A pop-up notification will appear, prompting you to install. Click Install and follow any on-screen instructions.

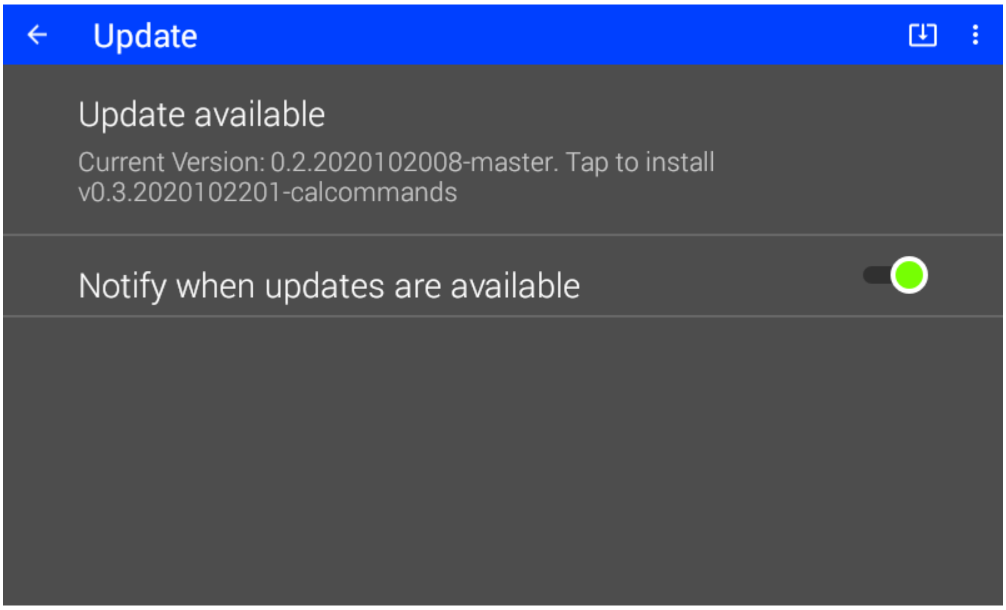


Figure 5.2: Install a firmware update

5.2 Manual Updates

Updating an offline instrument is accomplished with the following steps:

1. Go to lakeshore.com and navigate to the M81-SSM downloads page. Click on M81-SSM Firmware Update under Software.
2. On the Firmware Update page, click on the desired version number, which links to the .lsfw firmware update file.
3. Copy or save the .lsfw file onto the root directory of a USB flash drive. Make sure it is the only .lsfw file in that location.
4. Plug the flash drive into the USB-C port on the back of the instrument; if you don't have a newer flash drive with a USB-C connection, use the adapter supplied with the instrument.
5. On the M81-SSM front panel, tap the Settings menu (top left corner of the screen). Then tap System settings. Touch Update.
6. Click "Updates are available. Click here to install," and follow the on-screen instructions to update the firmware.

Note: Firmware comprises many files aggregated to form a Lake Shore firmware (.lsfw) package. The instrument stores versions of its .lsfw packages in its internal memory. This local memory location is called the "local repository." This allows you to downgrade the firmware to older versions that remain on the instrument in the local repository.

CHAPTER

SIX

REMOTE OPERATION

This chapter provides operational instructions for the remote interface for the MeasureReady™ M81-SSM Synchronous Source Measure System. The MeasureReady™ M81-SSM supports the following remote interfaces for direct user control:

- USB
- Ethernet
- GPIB

Once connected via any of these interfaces, the M81-SSM can be remotely operated in any of the following ways:

- **Python**: Using the Lake Shore Python driver
- **LabVIEW**: LabVIEW driver
- **Other**: IVI driver
- **Other**: Directly sending *SCPI commands* (page 134) from your application

The M81-SSM adheres to the Standard Commands for Programmable Instruments (SCPI) standard, regardless of the remote interface selected for user control.

6.1 USB Interface

The USB interface provides a convenient way to connect to most modern computers. The USB interface is implemented as a virtual serial com port connection. This implementation provides a simple migration path for modifying existing RS-232 based remote interface software. It also provides a simpler means of communicating than a standard USB implementation.

6.1.1 Physical Connection

The M81-SSM has a B-type USB connector on the rear panel. This is the standard connector used on USB peripheral devices, and it allows the common USB A-type to B-type cable to be used to connect the M81-SSM to a host PC. The pin assignments for A-type and B-type connectors are shown in section 6.7. The maximum length of a USB cable, as defined by the USB 2.0 standard, is 5 m (16.4 ft). This length can be extended using USB hubs every 5 m (16.4 ft) up to five times, for a maximum total length of 30 m (98.4 ft).

Note: The M81-SSM also has a USB-C connector for firmware updates. Only the USB-B connection can be used for SCPI commands.

6.1.2 Hardware Support

The USB interface emulates an RS-232 serial port at a fixed 921,600 baud rate, but with the physical connections of a USB. This programming interface requires a certain configuration to communicate properly with the M81-SSM. The proper configuration parameters are listed in the table below.

Baud rate	921600
Data bits	8
Parity	None
Stop bits	1
Flow control	RTS/CTS
Command terminator	LF (CR ignored)
Query response terminator	CR LF

The USB hardware connection uses the full speed (12,000,000 bit/s) profile of the USB 2.0 standard; however, since the interface uses a virtual serial com port at a fixed data rate, the data throughput is still limited to a baud rate of 921,600 bit/s.

6.1.3 Driver Installation

Installing the Driver from Windows® Update

The USB driver is available through Windows® Update. This is the recommended method for installing the driver, as it will ensure that you always have the latest version of the driver installed.

1. Connect the USB cable from the M81-SSM to the computer.
2. Turn on the M81-SSM.
3. When the Found New Hardware wizard appears, select **Locate and install driver software (recommended)**.
4. The Found New Hardware wizard should automatically connect to Windows® Update and install the drivers.
5. When the Found New Hardware wizard finishes installing the driver, a confirmation message stating “the software for this device has been successfully installed” will appear. Click **Close to complete the installation**.

Installing the Driver from the Web

If you are unable to install the driver from Windows® Update, refer to the instructions below to install the driver from the web. These procedures assume that you are logged into a user account that has administrator privileges.

The USB driver is available on the Lake Shore website. To install the driver, it must be downloaded from the website and extracted. Use the procedures below to download, extract, and install the driver using Windows®.

Download the driver

1. Locate the USB driver from https://www.lakeshore.com/resources/software.
2. Right-click on the USB driver download link, and select **Save as**.
3. Save the driver to a convenient place, and take note where the driver was downloaded.

Extract the driver

The downloaded driver is in a ZIP compressed archive. The driver must be extracted from this file. Windows® provides built-in support for ZIP archives. If this support is disabled, a third-party application, such as WinZip™ or 7-Zip, must be used.

1. Right-click on the file and click **extract all**.

2. An Extract Compressed (Zipped) Folders dialog box will appear. It is recommended the default folder is not changed. Take note of this folder location.
3. Click to clear the **Show extracted files when complete** checkbox, and click Extract.

Manually install the driver

The following section describes how to manually install the driver using Windows®. To install the driver, you must be logged in to a user account that has administrator privileges.

1. Connect the USB cable from the M81-SSM to the computer.
2. Turn on the M81-SSM.
3. If the Found New Hardware wizard appears, click **Ask me again later**.
4. Open Device Manager. Use this procedure to open Device Manager.
    a. Click the Windows® **Start** button and type Device Manager in the **Start Search** box.
    b. Click on the Device Manager link in the Search Results Under Programs dialog box.
    c. If User Account Control is enabled click **Continue** on the User Account Control prompt.
5. Click **View** and ensure the **Devices by Type** check box is selected.
6. In the main window of Device Manager, locate **Other Devices** in the list of device types. In many instances this will be between Network adapters and Ports (COM & LPT). If the Other Devices item is not already expanded, click the + icon. M81-SSM should appear indented underneath Other Devices. If it is not displayed as M81-SSM, it might be displayed as USB Device. If neither are displayed, click Action and then Scan for hardware changes, which may open the Found New Hardware wizard automatically. If the Found New Hardware wizard opens, click Cancel.
7. Right-click M81-SSM and click **Update Driver Software**.
8. Click **Browse my computer for driver software**.
9. Click **Browse** and select the location of the extracted driver.
10. Ensure the **Include subfolders** check box is selected and click **Next**.
11. When the driver finishes installing a confirmation message stating "Windows has successfully updated your driver software" should appear. Click **Close** to complete the installation.

6.2 GPIB Interface

The M81-SSM provides a GPIB interface for controlling the instrument from a computer via a GPIB controller.

6.2.1 Setting the Address

All GPIB devices must be assigned a fixed address that is unique within the bus. The default address of the M81-SSM is 12.

To view or edit the GPIB address of the M81-SSM, tap the Settings menu (top left corner of the screen). Then tap System settings. GPIB settings can be found in the Connectivity box. Touch the item to configure the address.

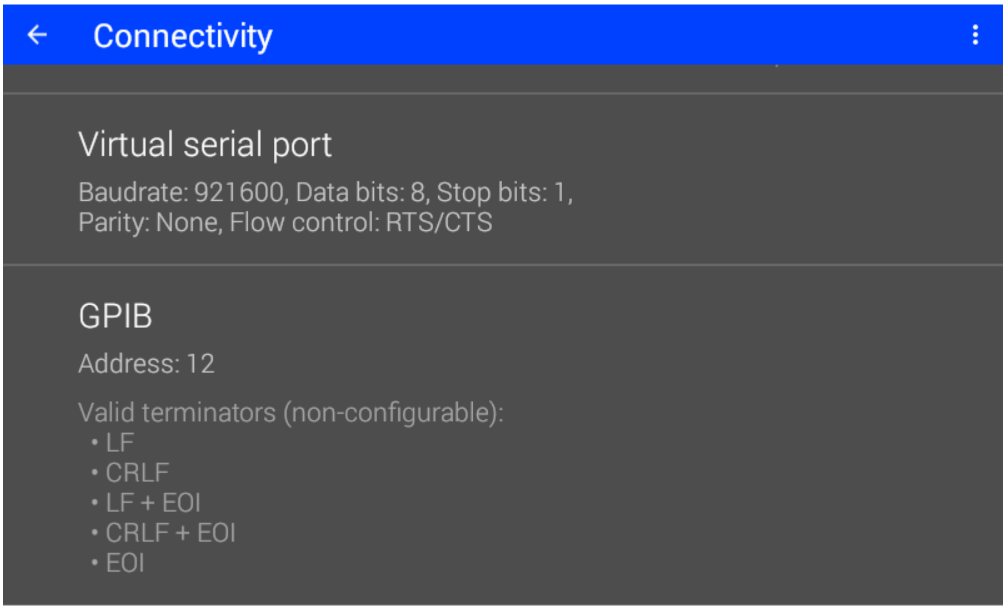


Figure 6.1: GPIB settings

6.2.2 GPIB Modes

Local

In Local mode the front panel is not locked out, indicating that there is currently no communication occuring on the GPIB bus.

Remote

In Remote mode the GPIB bus is being used to communicate with the instrument, and therefore the front panel is locked out. In this mode, touching the front panel screen presents an option to go to Local mode, which will set the mode to Local and unlock the front panel. Note that any subsequent GPIB communication will set the mode back to Remote. Therefore during continuous active communication over GPIB it is not possible to interact with the front panel.

Remote with Local Lockout

Remote with Local Lockout mode behaves the same as Remote mode except that there will be no option presented on the front panel to go to Local mode. This mode can only be set from the GPIB interface, and can only be changed from the GPIB interface, except that cycling power on the instrument will always return it to Local mode.

6.3 Ethernet Interface

The Ethernet interface provides a means of connecting the M81-SSM to an Ethernet-based computer network. These networks provide the ability to communicate across large distances, often using existing equipment (the internet, pre-existing local networks). The Ethernet interface of the M81-SSM provides the following capabilities:

- Send SCPI commands via TCP socket
- Firmware updates

6.3.1 Network Configuration

Configuration and status for these network configurations can be found from the front panel by tapping the Settings menu (top left corner of the screen). Then tap System settings. Ethernet settings can be found in the Connectivity box. Touch the item to see further settings.

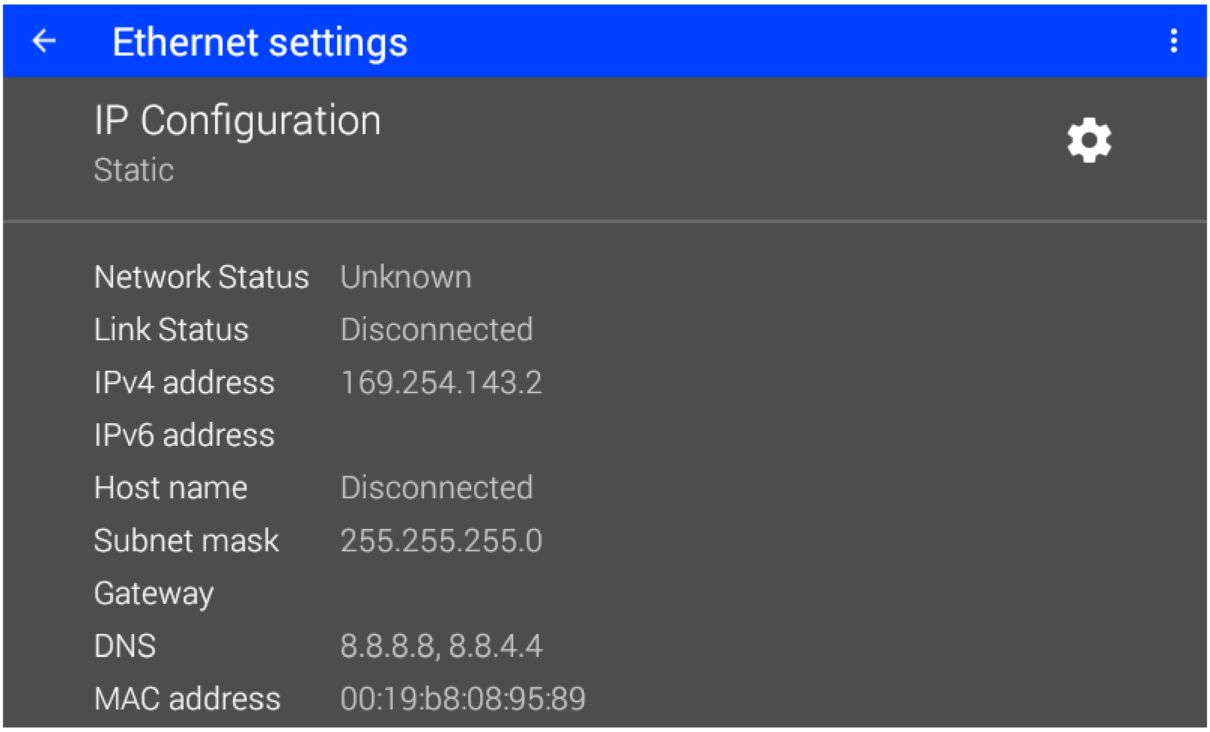


Figure 6.2: Ethernet settings

6.3.2 Network Address Parameters

Network address parameters include the IP address, the subnet mask, gateway address, and DNS address. The network address parameters of the M81-SSM can be configured using one of two methods: DHCP or static-IP.

- *IP address:* an IP address is required for a device to communicate using TCP/IP, which is the protocol generally used for Ethernet devices and the M81-SSM. The M81-SSM supports both IPv4 and IPv6. However, for simplicity, references to the IP protocol from this point forward will be referring to IPv4. An IP address is a 32-bit logical address used to differentiate devices on a network. It is most often given in dotted decimal notation, such as nnn.nnn.nnn.nnn where nnn is a decimal number from 0 to 255.
- *Subnet mask:* a sub network, or subnet, is a group of devices within a network that have a common, designated IP address routing prefix. A subnet mask is a 32-bit "bit mask" that signifies which part of the IP address represents the subnet routing prefix, and which part represents the device's address on the subnet. A subnet mask is most often given in dotted decimal notation, such as nnn.nnn.nnn.nnn where nnn is a decimal number from 0 to 255. When converted to a binary notation, the 32-bit subnet mask should consist of a contiguous group of ones, followed by a contiguous group of zeros. The ones represent which bits in the IP address refer to the subnet, and the zeros represent which bits refer to the device address. For example, the default Static-IP Address of the M81-SSM is 192.168.0.128, and the default static subnet mask is 255.255.255.0. Converting this subnet mask to binary shows that the first 24 bits are ones, and the last 8 bits are zeros. This means that the first 24 bits of the Static-IP Address (192.168.0) represent the subnet, and the last 8 bits (12) represent the device.
- *Gateway address:* a gateway is a network traffic routing device that is used to route communication between networks. If a gateway is not used, then devices on a network can only communicate with other devices on that same network. A gateway address is the IP address of the gateway on a network. Contact the network administrator for the gateway address for your network.
- *DNS address:* a domain name system (DNS) is a service that translates names into IP addresses. This service allows for using human readable names for devices on a network. As an example, when a web browser attempts to retrieve the web page at www.lakeshore.com, the browser first performs a forward-lookup on the assigned DNS server to attempt to retrieve the IP address that is represented by the name www.lakeshore.com. If successful, the web browser then uses the retrieved IP address to communicate with the web server that hosts the website at www.lakeshore.com.

6.3.3 Network Address Configuration Methods

The network address parameters of the M81-SSM can be configured using one of two methods: DHCP or Static-IP. DHCP is an automatic configuration method while static-IP requires manual configuration. If supported by the server, DHCP can also be used to automatically configure DNS server addresses, as well as IP address parameters.

Dynamic host configuration protocol (DHCP): DHCP is a method of automatically configuring the IP address, subnet mask, and gateway of Ethernet devices on a network. This method provides simple automatic configuration for users connecting to a network that provides a DHCP server. The network DHCP server will provide an IP address, subnet mask, and gateway address. Depending on the DHCP server configuration, it may also provide primary DNS and secondary DNS addresses as well. DHCP is the simplest method of IP configuration. DHCP does have the disadvantage of not necessarily preserving the IP address through a device reconfiguration, as well as the possibility of being automatically reconfigured when the DHCP "lease" expires. Contact your network administrator to find out the DHCP lease policy on your network.

Static-IP: static-IP is a method of manually configuring the IP address, subnet mask, and gateway of network enabled devices. When using the static-IP method, the IP address, subnet mask, and gateway must be configured appropriately for the connected network, or for the connected PC, in order to establish connection to the network. A major advantage to the Static-IP method is that the IP address will not change during device reconfiguration (power cycle). Disadvantages of using the static-IP method include the requirement of knowing how your network is configured in order to choose the correct configuration parameters.

Domain name: A domain is a collection of network devices that are managed according to some common characteristic of its members. Domains can contain subdomains, which are subsets within the domain. The hierarchy can contain several dot-separated levels, which flow from right to left. For example, lakeshore.com contains the top-level-domain "com" and the subdomain "lakeshore". When using the domain name system (DNS) to connect to a specific host device on a network, the device's hostname is tacked onto the left of the domain name. For example, the "www"" in www.lakeshore.com refers to the Lake Shore web server, located within the internet domain "lakeshore.com."

6.3.4 TCP Socket Communication

A TCP socket connection interface is provided as the communication medium for the network interface of the M81-SSM. A TCP socket connection, or simply "socket connection," is a common connection protocol used by Ethernet devices. The transmission control protocol (TCP) is commonly used for creating a communication channel between one program on one computer and one program on another computer, for example a web browser on a PC and a web server on the Internet. In the case of the M81-SSM, the protocol is used to create a communication channel between one program on one computer and the command line interface of the M81-SSM. TCP uses error correction and collision avoidance schemes that make it a very reliable form of network communication, but has drawbacks of having nondeterministic timing and can encounter relatively large delays depending on network conditions. These delays can be on the order of seconds. Sockets use port numbers to identify sending and receiving endpoints on network devices. This allows for multiple separate communication links to exist on each device.

Note: The port number used for TCP socket connections on the M81-SSM is 7777 by default. The port number can be configured from the Connectivity screen in System settings (top left corner of the screen).

6.3.5 SCPI Over TCP

There may be scenarios where the user desires to have the M81-SSM connected to the internet but does not want anyone to be able to connect to the M81-SSM and start controlling it. The SCPI over TCP setting allows for this. If the setting is disabled, network connections are still allowed, enabling firmware updates or autodate time to still function. For example, remote control of the M81-SSM will not be allowed even if another user knows the IP address and socket of the M81-SSM.

In order to issue SCPI commands over TCP and remotely control the instrument, this setting needs to be enabled. This can be done by touching the switch, as shown below.

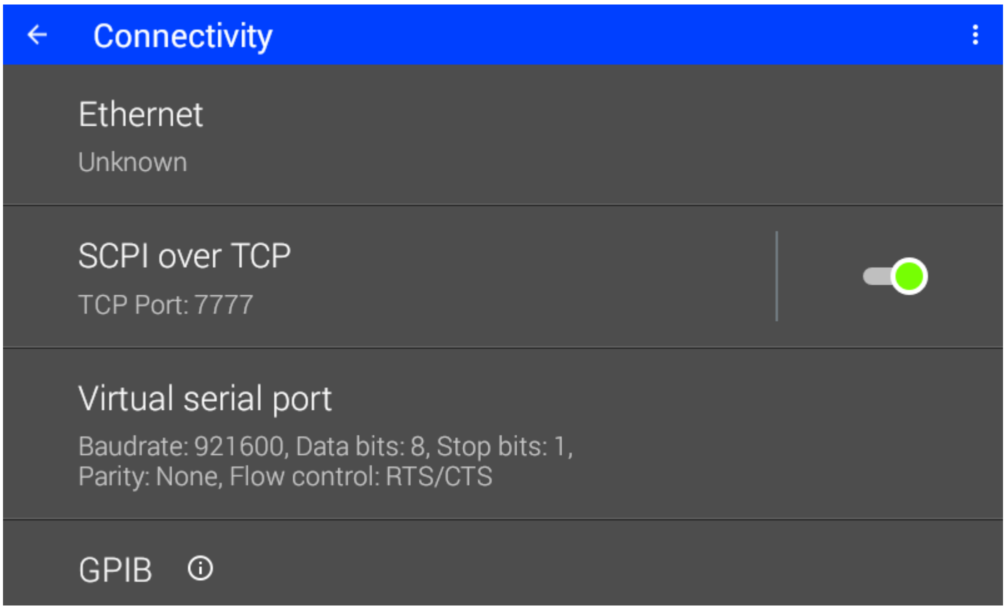


Figure 6.3: SCPI over TCP

6.4 Read and fetch queries

The M81-SSM provides read and fetch queries for easy collection of measurement data. Upon receiving the read query the M81-SSM will wait the averaging time (number of NPLCs) before returning a measurement. Upon receiving the fetch query the M81-SSM will return with the most recent measurement. For general measurements, the read command is typically preferred over the fetch command.

For example, to read the RMS value from M1 waiting the averaging time:

```
READ:SENSe1:RMS?
```

To collect data at regular intervals over time, use *data streaming* (page 116) instead.

Note: The basic READ? and FETCh? queries will not return while settling between range changes. Settle time depends on the averaging time. A measure module may go through multiple ranges before finding the optional range. Therefore it is important to configure your remote interface to avoid timeouts when using long averaging times. This is best accomplished by polling the *settling status bit* (page 122) before querying a new measurement.

6.4.1 Read Multiple

To collect a snapshot of multiple values once all readings have settled and after waiting for the highest configured NPLC, *READ[:MULTiple]?* (page 185) can be used. This command uses the data specifiers described in *data streaming* (page 116).

For example, to collect DC measurements from M1 and RMS measurements from M2:

```
READ:MULTiple? MDC,1,MRMS,2
```

By using read multiple instead of separate read commands, the data is synchronized in time.

6.4.2 Long running read measurements

When autorange is enabled, the read query accommodates range changes by resetting its averaging time on range change. This means that in some cases the time for a read command to complete cannot be determined a priori and communications can time out during a read command. For these cases, the read measurements can be started and the result queried after it is finished.

Use the *READ:SENSe#:STARt* (page 183) command to start a read measurement. One read measurement can be started for each measurement channel available. Use the *READ[:MULTiple]:STARt* (page 185) command to start a read multiple measurement. Paramenter pairs must be specified for the read multiple measurement. One read multiple measurement can be started. These measurements are all independent of each other.

To query if the measurement is running, use the *READ:SENSe#:RUNNing?* (page 183) query for a single read measurement, or *READ[:MULTiple]:RUNNing?* (page 185)

To abort a running measurement, use the *READ:SENSe#:ABORt* (page 178) command for a single read measurement, or the *READ[:MULTiple]:ABORt* (page 184) command for a read multiple measurement.

Once the measurement is completed, the results can be queried with the *READ:SENSe#:RESult:DC?* (page 180) and similar queries for a single read measurement. All applicable measurement data can be queried without running a new measurement. For a multiple reading, use the *READ[:MULTiple]:RESult?* (page 184) query. This query will return the data specified in the start command's parameter pairs.

For example, to collect the RMS value from M1:

```
READ:SENSe1:STARt
READ:SENSe1:RUNNing?
READ:SENSe1:RESult:DC?
```

For example, to collect DC measurements from M1 and RMS measurements from M2:

```
READ:MULTiple:STARt MDC,1,MRMS,2
READ:MULTiple:RUNNing?
READ:MULTiple:RESult?
```

6.4.3 Fetch Multiple

To collect a snapshot of multiple values at once, *FETCh[:MULTiple]?* (page 167) can be used. This command uses the data specifiers described in *data streaming* (page 116).

For example, to collect X and Y measurements from M1:

```
FETCh:MULTiple? MX,1,MY,1
```

By using fetch multiple instead of separate fetch commands, the data is synchronized in time.

6.4.4 Overload and Unlock

Readings that are presently not available because an overload will return 9.90E37 (SCPI value indicating Inf).

Readings that are presently not available because the PLL is unlocked will return 9.91E37 (SCPI value indicating NaN).

6.4.5 Settling

When a measurement is settling, the read command will wait until the reading has finished settling, and will subsequently wait the averaging time (NPLC). The fetch command will simply wait until the reading has finished settling to return a value.

6.4.6 Fetch Immediate

The *Fetch Immediate Queries* (page 136) will always return a value regardless of settling. This can be used in cases where the query must always return a value as soon as possible. If the reading is settling the value returned will be 9.91E37 (SCPI value indicating NaN).

6.5 Data Streaming

The M81-SSM allows for fast and synchronized data transfer via the data stream. The instrument has a configurable data logger that places data into a real-time capture buffer. The data is then transferred at the maximum throughput to a SCPI buffer, from which it can be extracted via SCPI commands.

MeasureSync™ tightly synchronizes the data captured by the data logger. It can then be extracted from the data buffer without any real-time considerations as with the fetch or read commands.

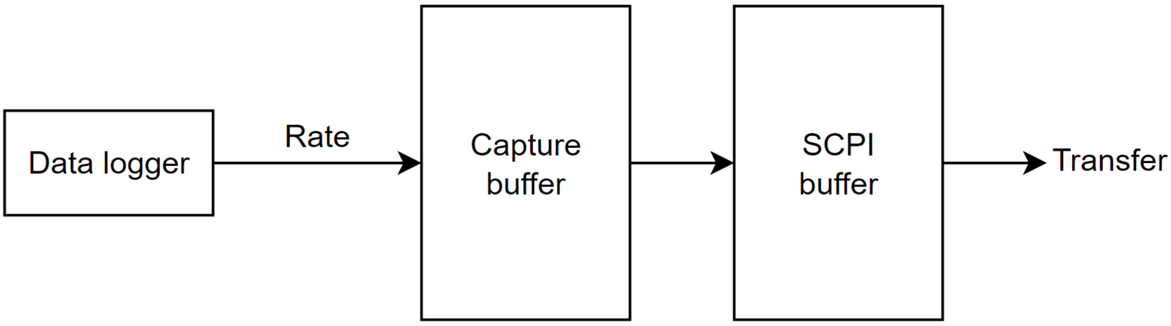


Figure 6.4: Data stream architecture

Note: This section details how to stream data via direct SCPI commands. If you are using a driver, please refer to that driver's documentation for a higher level data streaming interface.

6.5.1 Configuring the Stream

What is streamed

What is streamed can be configured using the *TRACe:FORMat:ELEMents* (page 292) command. Up to 10 can be specified. Max update rate is specified in kSamples/second. If elements are specified that update at a lower max update rate than the requested stream rate, duplicate sample values are provided. For example, streaming MRMS (5 kSa max) and MPPeak (1 kSa max) at 5 kSa would return five unique MRMS values but only one unique MPPeak value per five rows.

Table 6.1: Data sources

Mnemonic	Description	Bytes	Type	Max update rate
RTIMe	Relative Time: Time elapsed since first row in this stream in seconds.	8	double (d)	5 kSa
SAMPlitude	Source Amplitude: Present value of amplitude setting for this source module.	8	double (d)	5 kSa
SOFFset	Source Offset: Present value of offset setting for this source module.	8	double (d)	5 kSa
SFRequency	Source Frequency: Present value of the frequency setting for this source module.	8	double (d)	5 kSa
SRANge	Source Range: Largest value that can be sourced on the present range of this source module.	4	float (f)	5 kSa
SVLimit	Source Voltage Limit: State of V limit for this source module. True if limit is engaged.	1	bool (?)	5 kSa
SILimit	Source Current Limit: State of I limit for this source module. True if limit is engaged.	1	bool (?)	5 kSa
SRSettling	Source Readback Settling: State of the source readback for this module. True if readback is settling.	1	bool (b)	5 kSa
SSWeeping	Source Sweeping: Sate of the source sweeping for this module. True is the source is presently sweeping a parameter.	1	bool (?)	5 kSa
MDC	Measure DC: DC measurement for this measure module. The module must be in DC or AC mode.	8	double (d)	5 kSa
MRMS	Measure RMS: RMS measurement for this measure module. The module must be in DC or AC mode.	8	double (d)	5 kSa
MPPeak	Measure Positive Peak: Positive peak measurement for this measure module.	8	double (d)	1 kSa

Continued on next page

Table 6.1 – continued from previous page

Mnemonic	Description	Bytes	Type	Max update rate
MNPeak	Measure Negative Peak: Negative peak measurement for this measure module.	8	double (d)	1 kSa
MPTPeak	Measure Peak to Peak: Peak to peak measurement for this measure module.	8	double (d)	1 kSa
MX	Measure X: X measurement from the lock-in for this measure module. The module must be in lock-in mode.	8	double (d)	5 kSa
MY	Measure Y: Y measurement from the lock-in for this measure module. The module must be in lock-in mode.	8	double (d)	5 kSa
MR	Measure R: Magnitude measurement from the lock-in for this measure module. The module must be in lock-in mode.	8	double (d)	1 kSa
MTHeta	Measure Theta: Angle measurement from the lock-in for this measure module. The module must be in lock-in mode.	8	double (d)	1 kSa
MRANge	Measure Range: Largest value that can be measured on present range of this measure module	4	float (f)	5 kSa
MOVerload	Measure Overload: Overload status of this measure module. True if overloaded.	1	bool (?)	5 kSa
MSETtling	Measure Settling: Settling status of this measure module. True if settling.	1	bool (?)	5 kSa
MUNLock	Measure Unlock: Lock status of the PLL for this measure module. The module must be in lock-in mode.	1	bool (?)	5 kSa
MRFRequency	Measure Reference Frequency: Frequency measurement from the PLL for this measure module. True if unlocked.	8	double (d)	1 Sa

Continued on next page

Table 6.1 – continued from previous page

Mnemonic	Description	Bytes	Type	Max update rate
GPIStates	General Purpose Input States: Bit-stuffed value of general purpose input states. module index ignored.	1	unsigned char (B)	5 kSa
GPOStates	General Purpose Output States: Bit-stuffed value of general purpose output states. module index ignored.	1	unsigned char (B)	5 kSa

Rate

The rate is specified using the *TRACe:RATE* (page 295) command. The rate must be the maximum rate divided by an integer value. The command will select the closest available rate to the requested rate. The query *TRACe:RATE?* (page 295) will return the actual, rounded streaming rate.

There are two constraints on the rate at which data can be streamed. The first is the capture buffer size. Data may be streamed at any rate (up to the maximum for that source) for streams that are short enough to fit within the capture buffer size of 16 kB. For streams larger than the capture buffer, the rate must be slower than the maximum throughput of 50 kB/s.

Encoding

Two encoding types are available:

- **CSV**: Data is returned as comma separated values. This is often easier to process, but will not achieve the maximum throughput.
- **B64**: Data is returned in a binary format that has been B64 encoded (no padding, little endian).

The encoding type must be selected before starting the data stream. This is done via the *TRACe:FORMat:ENCOding* (page 293) command.

6.5.2 Data Transfer

For any encoding, data can be extracted from the SCPI buffer one row at a time or multiple rows at a time. *TRACe:DATA[:SINGle]?* (page 292) provides a single row. *TRACe:DATA:ALL?* (page 290) provides all available rows. Once data is read by either query, it is no longer in the buffer.

If no data is available, the response of either query will be empty. Number of unread rows in the SCPI buffer can be queried with *TRACe:DATA:COUNt?* (page 291).

In either encoding, values appear in the order they are specified by *TRACe:FORMat:ELEMents* (page 292).

For the following examples, assume the following was sent for *TRACe:FORMat:ELEMents* (page 292):

```
TRACe:FORMat:ELEMents SAMPLITUDE,1,MX,2,MOVERLOAD,2
```

CSV Encoding

In the CSV encoding, a single row looks like this with values in a single row separated by commas:

```
"3.14159,2.71828,False"
```

TRACe:DATA:ALL? (page 290) separates rows with a semicolon at the end of each row like this:

```
"3.14159,2.71828,False;1.41421,1.61803,True;"
```

B64 Encoding

In the B64 encoding, a single row consists of each value in that row. The above example would consist of the following raw bytes:

```
EA 2E 44 54 FB 21 09 40 03 57 14 8B 0A BF 05 40 00
```

The representation is little endian. No padding bytes are inserted. The format of the present configuration is available via the *TRACe:FORMat:ENCOding:B64:BFORmat?* (page 293) query.

This would then be base 64 encoded for transfer over SCPI as:

```
"6i5EVPshCUADVxSLCr8FQAA="
```

If multiple rows are available *TRACe:DATA:ALL?* (page 290) will concatenate the results with no separator. The length of a single row is available via the *TRACe:FORMat:ENCOding:B64:BCOunt?* (page 293) query.

Overflow

If data is not read from the SCPI buffer quickly enough and the SCPI buffer is full, overflow will occur. *TRACe:DATA:OVERflow?* (page 291) can be used to check if any overflow has occurred during this data stream.

6.6 Status and Error Reporting

6.6.1 Status System Overview

The MeasureReady™ M81-SSM implements a status system compliant to the SCPI-99 standard. The SCPI status system is derived from the status system called out in Chapter 11 of the IEEE 488.2 standard. The status system provides a method of recording and reporting instrument information. At the center of the status system is the status byte register. This register contains summary bits from other status registers, providing the user one register to periodically query to determine if further interrogation of the instrument is required.

In addition to the status byte and its complementary service request enable register, the status system is made up of standard, questionable, and operation status register sets. Each one of those register sets comprises three types of registers: condition, event, and enable. In addition, the status system contains an output buffer and error queue. A diagram of the status system is shown below.

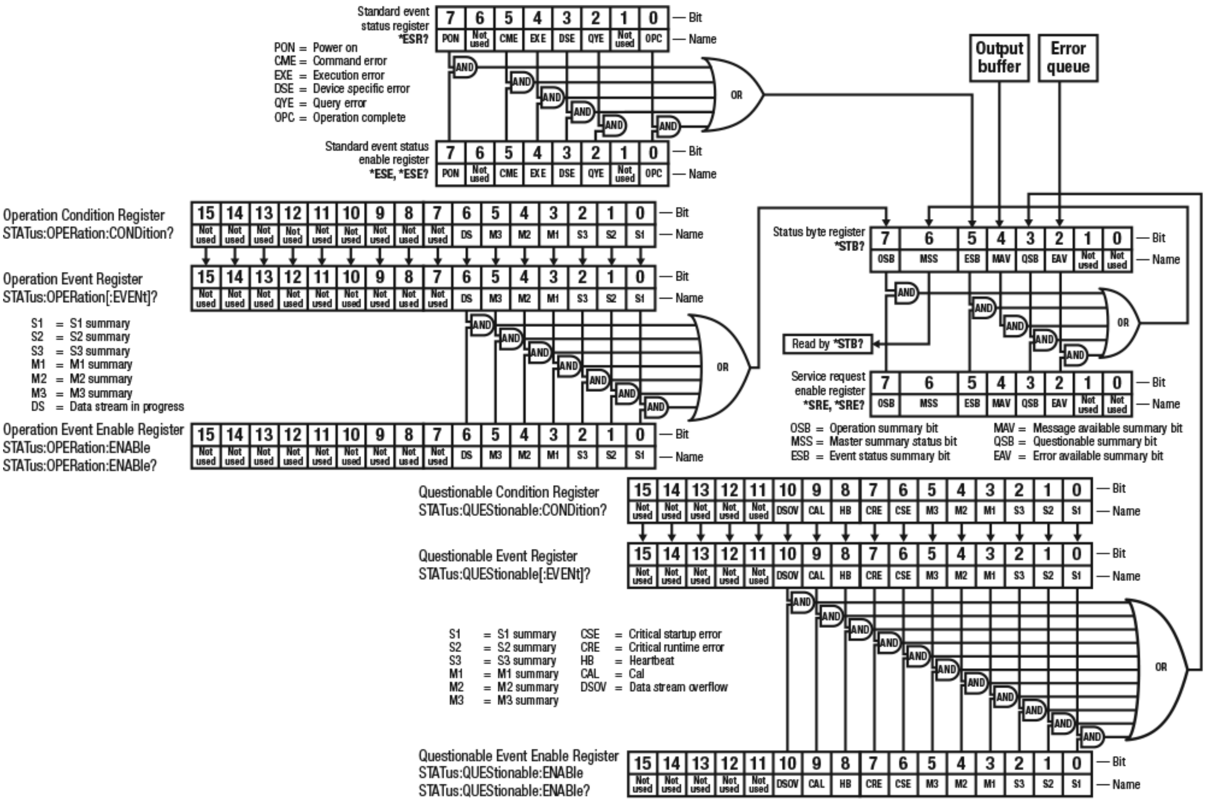


Figure 6.5: M81-SSM status system

Status Byte Register

The status byte register, typically referred to as the status byte, is a non-latching, read-only register that contains all of the summary bits from the register sets. The status of the summary bits are controlled from the register sets as explained in section 4.3.2.1 to section 4.3.2.5. The status byte also contains the Master Summary Status (MSS) bit. This bit is used to report if any of the summary bits are set via the **STB?* command. The status of the MSS bit is controlled by the summary bits and the service request enable register.

Service Request Enable Register

The service request enable register determines which summary bits in the status byte will set the MSS bit of the status byte. The user may write to or read from the service request enable register. Each status byte summary bit is logically ANDed to the corresponding enable bit of the service request enable register. When a service request enable register bit is set by the user, and the corresponding summary bit is set in the status byte, the MSS bit of the status byte will be set.

Conditional Registers

Each register set (except the standard event register set) includes a condition register. The condition register constantly monitors the instrument status. The data bits are real-time and are not latched or buffered. The register is read-only.

Event Registers

Each register set includes an event register. Bits in the event register correspond to various system events and latch when the event occurs. Once an event bit is set, subsequent events corresponding to that bit are ignored. Set bits remain latched until the register is cleared by a query command (such as **ESR?*) or a **CLS* command. The register is read-only.

Enable Registers

Each register set includes an enable register. An enable register determines which bits in the corresponding event register will set the summary bit for the register set in the status byte. The user may write to or read from an enable register. Each event register bit is logically ANDed to the corresponding enable bit of the enable register. When an enable register bit is set by the user, and the corresponding bit is set in the event register, the output (summary) of the register will be set, which in turn sets the master summary status bit of the status byte register.

Reading Registers

Any register in the status system may be read using the appropriate query command. The response to a query will be a decimal value that corresponds to the binary-weighted sum of all bits in the register.

Bit	Decimal	Weighting
0	1	2^0
1	2	2^1
2	4	2^2
3	8	2^3
4	16	2^4
5	32	2^5
6	64	2^6
7	128	2^7

Programming Registers

The only registers that may be programmed by the user are the enable registers. All other registers in the status system are read-only registers. To program an enable register, send a decimal value that corresponds to the desired binary-weighted sum of all bits in the register

Status Byte Register

The summary messages from the event registers and the output buffer set or clear the summary bits of the status byte register. These summary bits are not latched. Clearing an event register will clear the corresponding summary bit in the status byte register. Reading all messages in the output buffer, including any pending queries, will clear the message available bit. Reading all errors out of the queue will clear the error available bit. The bits of the status byte register are described as follows:

- **Operation Summary (OSB), Bit 7**: This bit is set when an enabled operation event has occurred
- **Master Summary Status (MSS), Bit 6**: This bit is set when a summary bit and the summary bit's corresponding enable bit in the service request enable register are set. A **STB?* will read the status of the MSS bit (along with all of the summary bits), but also will not clear it. To clear the MSS bit, either clear the event register that set the summary bit or disable the summary bit in the service request Enable register.
- **Event Summary (ESB), Bit 5**: This bit is set when an enabled standard event has occurred
- **Message Available (MAV), Bit 4**: This bit is set when a message is available in the output buffer
- **Questionable Summary (QSB), Bit 3**: This bit is set when an enabled questionable event has occurred

- **Error Available (EAV), Bit 2**: This bit is set when an error is available in the error queue

Service Request Enable Register

The service request enable register is programmed by the user and determines which summary bits of the status byte may set bit 6 (MSS). Enable bits are logically ANDed with the corresponding summary bits. Whenever a summary bit is set by an event register and its corresponding enable bit is set by the user, bit 6 will be set. The Service Request Enable command (**SRE*) programs the Service Request Enable Register and the query command (**SRE?*) reads it.

Standard Event Status Register Set

The standard event status register reports the following interface-elated instrument events: power on detected, command syntax errors, command execution errors, query errors, and operation complete. Any or all of these events may be reported in the standard event summary bit through the enable register. The standard event status enable command (**ESE*) programs the enable register and the query command (**ESE?*) reads it. **ESR?* reads and clears the standard event status register.

- **Power On (PON), Bit 7**: This bit is set to indicate an instrument off-on transition
- **Command Error (CME), Bit 5**: This bit is set if a command error has been detected since the last reading. This means that the instrument could not interpret the command due to a syntax error, an unrecognized header, unrecognized terminators, or an unsupported command
- **Execution Error (EXE), Bit 4**: This bit is set if an execution error has been detected. This occurs when the instrument is instructed to do something not within its capabilities. A typical example of this are command parameters that are outside the instrument’s acceptable bounds.
- **Device Specific Error (DSE), Bit 3**: This bit is set if an error occurs that does not fall into another category defined as a standard event. Examples include if the M81-SSM is unable to set its date/time or time zone.
- **Query Error (QYE), Bit 2**: This bit indicates a query error. It occurs rarely and involves loss of data because the output queue is full.
- **Operation Complete (OPC), Bit 0**: When **OPC* is sent, this bit will be set when the instrument has completed all pending operations. The operation of this bit is not related to the **OPC?* command, which is a separate interface feature.

Operation Register Set

The operation event register reports the instrument events that are considered part of normal operation. Any or all of these events may be reported in the operation event summary bit through the enable register. The operation event enable command (*STATus:OPERation:ENABle*) programs the enable register and the query command (*STATus:OPERation:ENABle?*) reads it. *STATus:OPERation[:EVENt]?* reads and clears the operation event register. *STATus:OPERation:CONDition?* reads the operation condition register.

- **S1-3 Summary, Bit 0-2**: This bit is a summary of the operation register for a particular source module
- **M1-3 Summary, Bit 3-5**: This bit is a summary of the operation register for a particular mesure module
- **Data Stream in Progress (DS), Bit 6**: This bit is set when a datastream is in progress.

The source modules have their own operational register sets and are accessed from the *STATus:OPERation:SOURce#* subsystem with the following bit definition:

- **V Limit (VLIM), Bit 0**: This bit is set when the module voltage limit is activated.
- **I Limit (ILIM), Bit 1**: This bit is set when the module current limit is activated.
- **Sweeping (SWP), Bit 2**: This bit is set when the module is sweeping a parameter.

The measure modules have their own operational register sets and are accessed from the *STATus:OPERation:SENSe#* subsystem with the following bit definition:

- **Overload (OVLD), Bit 0**: This bit is set when the measurement is higher than the maximum for the range.
- **Settling (SETL), Bit 1**: This bit is set when the module hardware is settling due to a configuration change.
- **Unlocked (UNLK), Bit 2**: This bit is set if the PLL has not attained lock on the input frequency.

Questionable Register Set

The questionable status register reports various states of the instrument that could indicate the quality of the output signal may be compromised. Any or all of these events may be reported in the questionable event summary bit through the enable register. The questionable event enable command (*STATus:QUEStionable:ENABle*) programs the enable register and the query command (*STATus:QUEStionable:ENABle?*) reads it.

STATus:QUEStionable[:EVENt]? reads and clears the questionable event register. *STATus:QUEStionable:CONDition?* reads the questionable condition register.

- **S1-3 Summary, Bit 0-2**: This bit is a summary of the questionable register for a particular measure module
- **M1-3 Summary, Bit 3-5**: This bit is a summary of the questionable register for a particular source module
- **Critical Startup Error (CSE), Bit 6**: This bit is set if a hardware issue was detected during startup
- **Critical Runtime Error (CRE), Bit 7**: This bit is set if a hardware issue was detected during runtime
- **Internal Communication Failure (HB), Bit 8**: This bit is set if internal communications are interrupted
- **Calibration Error (CAL), Bit 9**: This bit is set if corrupted or default calibration data is detected
- **Data Stream Overflow (DSOV), Bit 10**: This bit is set if data was lost during a stream because of buffer overflow

The source and measure modules have their own questionable register sets. These register sets are accessed through the *STATus:QUEStionable:SOURce#* and *STATus:QUEStionable:SENSe#* subsystems respectfully. They share the following bit definitions:

- **Read Error (RE), Bit 0**: This bit is set when module communication fails.
- **Unrecognized Module Error (UME), Bit 1**: This bit is set when the connected module was successfully identified, but is not supported by the instrument firmware. The firmware must be updated to support this module.
- **Port Direction Error (PDE), Bit 2**: This bit indicates that the module is plugged into the wrong port type. Source modules must be connected to source ports. Measure modules must be connected to measure ports.
- **Factory Calibration Failure (FCF), Bit 3**: This bit is set if the factory calibration process failed for this module.
- **Self Calibration Failure (SCF), Bit 4**: This bit is set if the last self calibration attempt failed.

6.6.2 Error Messages

As called out in the SCPI-99 specification, the M81-SSM implements an error queue that contains coded error and status messages thrown during operation. SCPI 99 defines error messages with a negative (-) prefix as standard errors, common to all SCPI compliant instruments. Error messages with a positive prefix (+) are allocated to instrument manufacturers for instrument specific messages. Presently, the M81-SSM does not implement any instrument specific messages. All M81-SSM specific errors and status are captured in the status register system.

Coded error and status messages can be retrieved and cleared over the remote interface using the following commands:

```
SYSTem:ERRor:ALL?
SYSTem:ERRor:CLEar
SYSTem:ERRor:COUNt?
SYSTem:ERRor[:NEXT]?
```

The queue is implemented with a "First In, First Out" (FIFO) approach. This means, if the M81-SSM adds multiple messages to the queue in a given period of time, issuing the SYSTem:ERRor[:NEXT]? query will return the message that was added to the queue first, and, subsequently remove it from the queue.

SCPI 99 categorizes its standard errors into logical groups that match bit definitions in the standard event status register. When the M81-SSM adds an error message to the queue, a bit in the standard event status register will be set as well. The error code range table shown below lists the error code ranges, title, and the corresponding bit in the standard event status register that gets set when the error or status is added to the queue.

Error code range	Description	Standard event register bit
-100 to -184	Command errors	5
-200 to -294	Execution errors	4
-300 to -365	Device specific errors	3
-400 to -440	Query errors	2

A comprehensive list of standard SCPI error codes descriptions can be found in chapters 21.8.9 through 21.8.16 of the official SCPI 99 standard, located on the IVI website.

Note: The M81-SSM does not implement every error code listed in the standard.

6.7 SCPI Usage Guide

The M81-SSM adheres to the Standard Commands for Programmable Instruments (SCPI) command language. SCPI is a standardized command language, presently maintained by the Interchangeable Virtual Instruments (IVI) Foundation, used for controlling test and measurement instruments. SCPI provides instrument control with a standardized command syntax and style.

The following sections provide a brief overview of the SCPI language, syntax, and behavior, that will allow a new instrument user to get up and running quickly. The complete SCPI specification can be found on the IVI website.

6.7.1 Command Parameter Data Types

Boolean	The Boolean datatype in SCPI is used to indicate whether a setting should be enabled or disabled, where 0 or OFF represents disable and 1 or ON represents enable.
NR1	Integer number with no decimal point. Example: 123.
NRf	A decimal number that can be represented as an integer (77), a real number (77.325), or an exponential number (7732.5E02).
String	Character string enclosed in double quotes. Example: "abc".
Character	Short mnemonic alphanumeric strings used to represent specific commands or functions, enabling concise and convenient identification of operations or queries. Ex. SWE or SWEep.
Decimal Numeric	Superset of NRf where values such as MIN and MAX are also valid.

6.7.2 Command Types

The M81-SSM responds to two different types of SCPI commands: common and subsystem.

Common

SCPI builds upon the IEEE 488.2 standard. Because of this, several of the commands found in IEEE 488.2 are required by the M81-SSM to be considered SCPI compliant. These commands include, but are not limited to, functionality to reset (*RST), clear the status registers (*CLS), or identify the instrument (*IDN?).

These commands are identifiable by the fact that they:

1. Always consist of three letters and
2. Are always preceded by an asterisk

Subsystem

In addition to the common commands, SCPI defines subsystem commands. If the M81-SSM is considered a "system," then the logical grouping of its various functions can be considered different "subsystems," forming a hierarchical "tree." Each subsystem is composed of program headers. Together, these make up the individual SCPI command. The following is an example of several SCPI commands and how they fit into the hierarchical tree:

```
READ
    :SENSe#
        :DC?

SOURce#
    :COUPling?
    :COUPling <coupling>

SENSe#
    :MODE?
    :MODE <mode>
    :VOLTage
        :RANGe?
        :RANGe <range>
            :AUTO?
            :AUTO <state>
```

6.7.3 Queries

Many times, a command will have a corresponding query. Queries require a question mark at the end of the command header. For example:

```
READ:SENSe1:DC?
```

If a query requires a parameter, place the question mark at the end of the header and before the query parameters.

Infinity and NaN Values

As defined in the IEEE 488.2 standard, positive infinity is represented by 9.9E+37, negative infinity is represented by -9.9E+37, and NaN (not a number) is represented by 9.91E+37.

6.7.4 Optional Commands

If a command header is enclosed in brackets ([ ]), then it is optional. Do not include the brackets if you send the optional command to the instrument. For example, the following command messages are both valid:

```
SENSe1:FILTer:STATe?
SENSe1:FILTer?
```

6.7.5 Short and Long Form

SCPI specifies a short and long form for its command headers. The command header is completely spelled out in the long form while it is abbreviated to the first three or four letters for short form. The following is an example of the same command in both short and long form:

```
SENSe1:CONFiguration GROund
SENS1:CONF GRO
```

6.7.6 Number Suffix

When a command header is used to represent multiple instances of a given function of the instrument, the selection of which to use is designated by a numeric suffix attached to the command header. This suffix is applicable for both short and long forms. Examples:

```
READ:SENSe1:DC?
READ:SENS1:DC?
```

6.7.7 Multiple Commands in a Single Message

Many times, it is desirable to chain multiple commands and queries together into one message transmission. To do this, separate each command/query with a semicolon and be sure to include a colon before the next command, unless it is a common command that uses the "*" prefix.

If multiple commands are provided in a given message, the M81-SSM will process the commands in the order in which they were received. Subsequently, if multiple queries are included in chained message, the M81-SSM will respond to the queries in the order in which they were received. For example:

```
SENSe1:CONFiguration GROund;:READ:SENSe1:DC?;*OPC?
```

This will set measure module 1 input configuration to ground, request a DC measurement from measure module 1, then wait for an operation complete query response. What is returned might look like the following:

```
2.438927059283;1
```

6.7.8 Commands Within the Same Subsystem

Referring to the subsystem tree shown in section 4.2.1.2, if multiple commands within the same level of the SCPI tree are sent in the same chained message, the colon does not need to be included in order to reset the SCPI header path. An example of this is the RANGe and MODE headers within the SENSe subsystem. For example:

```
READ:SENSe1:LIA:X?;Y?
```

This will return both an X and Y measurement.

6.7.9 Terminators

All data in a given SCPI message is encoded in the American Standard Code for Information Interchange (ASCII) format. A special ASCII character, the line feed (LF, hex 0A, decimal 10), is required by the instrument to know where the SCPI message ends. The instrument also allows an optional carriage return (CR, hex 0D, decimal 13) to precede the line feed. The instrument query response terminator will include both carriage return and line feed.

6.7.10 Message Flow Control

It is important to remember that the user program is in charge of SCPI communication at all times, independent of the communications bus used to control the instrument. The instrument cannot initiate communication, determine which device should be transmitting at a given time, or guarantee timing between messages. All of this is the responsibility of the user program.

Most commands sent to the M81-SSM will execute within 100 ms of the last character being transmitted. However, the SCPI command protocol does not specify that an instrument return an acknowledge message when a given command is finished executing. Because of this, a user program could potentially flood and overload the instrument's SCPI buffer.

Therefore, it is highly recommended to append the Operation Complete Query (*OPC?) to the end of any given command string. The operation complete query is a SCPI mandated IEEE 488.2 query that will cause the instrument to send a "1" back to the user program when all commands have been processed. This scheme closes the timing loop and provides the proper balance for a user program to maximize command throughput yet not risk overrunning the instrument SCPI buffer. *OPC? can be appended onto a single command or a chained SCPI message. For example:

```
SENSe1:CONFiguration AB;:SENSe1:COUPling DC;*OPC?
```

This will set measure module 1 configuration to differential (A-B), coupling to DC and then wait for an operation complete query response.

6.8 M81-SSM SCPI Command Summary

6.8.1 Module Loading

These commands and queries provide functionality for loading connected modules before use and unloading modules before physically disconnecting them. See *Module Loading* (page 24) for details.

All Modules

- *SYSTem:LOAD* (page 290)

Source Modules

- *SOURce#:UNLoad* (page 264)
- *SOURce#:LOAD?* (page 247)

Measure Modules

- *SENSe#:UNLoad* (page 216)
- *SENSe#:LOAD?* (page 209)

6.8.2 Data Queries

These queries provide readings one value at a time. For faster, synchronized data transfer use *data streaming* (page 116) instead.

Read Queries

These queries provide a new reading waiting the averaging time (NPLC) after the reading has settled.

- *READ[:MULTiple]?* (page 185)
- *READ[:MULTiple]:STARt* (page 185)
- *READ[:MULTiple]:RUNNing?* (page 185)
- *READ[:MULTiple]:ABORt* (page 184)
- *READ[:MULTiple]:RESult?* (page 184)

Fetch Queries

These queries provide the latest available reading after the reading has settled.

- *FETCh:SENSe#:PPEak?* (page 163)
- *FETCh:SENSe#:PTPeak?* (page 164)
- *FETCh:SENSe#:RMS?* (page 165)
- *FETCh:SENSe#:RMS:RELative?* (page 165)
- *FETCh:TEMPerature?* (page 167)
- *FETCh:SENSe#:TEMPerature?* (page 165)
- *FETCh:SOURce#:TEMPerature?* (page 166)

Fetch Immediate Queries

The :IMMediate subsystem can be used to force a reading to be returned immediately without waiting for settling.

- *FETCh:SENSe#:DC:IMMediate?* (page 157)
- *FETCh:SENSe#:DC:RELative:IMMediate?* (page 158)
- *FETCh:SENSe#:LIA:R:IMMediate?* (page 159)
- *FETCh:SENSe#:LIA:THETa:IMMediate?* (page 160)
- *FETCh:SENSe#:LIA:X:IMMediate?* (page 161)
- *FETCh:SENSe#:LIA:Y:IMMediate?* (page 161)
- *FETCh:SENSe#:NPEak:IMMediate?* (page 162)
- *FETCh:SENSe#:PPEak:IMMediate?* (page 163)
- *FETCh:SENSe#:PTPeak:IMMediate?* (page 163)
- *FETCh:SENSe#:RMS:IMMediate?* (page 164)
- *FETCh:SENSe#:RMS:RELative:IMMediate?* (page 164)

6.8.3 Data Streaming

These commands allow configuration of *data streaming* (page 116) as well as transfer of data.

Configuration

- *TRACe:FORMat:ELEMents* (page 292)
- *TRACe:FORMat:ELEMents?* (page 292)
- *TRACe:FORMat:ENCOding* (page 293)
- *TRACe:FORMat:ENCOding?* (page 294)
- *TRACe:FORMat:ENCOding:B64:BCOunt?* (page 293)
- *TRACe:FORMat:ENCOding:B64:BFORmat?* (page 293)
- *TRACe:FORMat:HEADer?* (page 294)
- *TRACe:RATE* (page 295)
- *TRACe:RATE?* (page 295)

Transfer

- *TRACe:STARt* (page 296)
- *TRACe:RESet* (page 295)
- *TRACe:DATA[:SINGle]?* (page 292)
- *TRACe:DATA:ALL?* (page 290)
- *TRACe:DATA:COUNt?* (page 291)
- *TRACe:DATA:OVERflow?* (page 291)
- *TRACe:ACTive?* (page 290)

6.8.4 Source Setup

These commands setup a source channel, and are available regardless of what source module is used.

- *SOURce#:STATe* (page 255)
- *SOURce#:STATe?* (page 256)
- *SOURce#:FUNCtion[:SHAPe]* (page 244)
- *SOURce#:FUNCtion[:SHAPe]?* (page 244)
- *SOURce#:FREQuency[:FIXed]* (page 242)
- *SOURce#:FREQuency[:FIXed]?* (page 243)
- *SOURce#:FREQuency[:CW]* (page 241)

- *SOURce#:FREQuency[:CW]?* (page 242)
- *SOURce#:DCYCle* (page 235)
- *SOURce#:DCYCle?* (page 235)
- *SOURce#:SYNChronize[:STATe]* (page 263)
- *SOURce#:SYNChronize[:STATe]?* (page 264)
- *SOURce#:SYNChronize:PHASe* (page 262)
- *SOURce#:SYNChronize:PHASe?* (page 262)
- *SOURce#:SYNChronize:SOURce* (page 263)
- *SOURce#:SYNChronize:SOURce?* (page 263)
- *SOURce#:PRESet* (page 252)

6.8.5 Measurement Setup

The measurement mode can be configured with the mode command. Further configuration is available based on the mode. The commands in this section are available regardless of what measure module is used.

- *SENSe#:MODE* (page 209)
- *SENSe#:MODE?* (page 210)
- *SENSe#:PRESet* (page 213)

DC Mode

- *SENSe#:NPLCycles* (page 212)
- *SENSe#:NPLCycles?* (page 212)

AC Mode

- *SENSe#:MODE* (page 209)
- *SENSe#:MODE?* (page 210)
- *SENSe#:NPLCycles* (page 212)
- *SENSe#:NPLCycles?* (page 212)

Lock-In Mode

Relative Measurement

6.8.6 Module Configuration

Each module has additional commands to configure functionality provided by that module.

BCS-10

VS-10

VM-10

- *SENSe#:VOLTage:RANGe* (page 216)
- *SENSe#:VOLTage:RANGe?* (page 217)
- *SENSe#:VOLTage:RANGe:AUTO* (page 216)
- *SENSe#:VOLTage:RANGe:AUTO?* (page 217)
- *SENSe#:CONFiguration* (page 188)
- *SENSe#:CONFiguration?* (page 188)
- *SENSe#:COUPling* (page 189)
- *SENSe#:COUPling?* (page 189)
- *SENSe#:FILTer[:STATe]* (page 198)
- *SENSe#:FILTer[:STATe]?* (page 198)
- *SENSe#:FILTer:HPASs:ATTenuation* (page 194)
- *SENSe#:FILTer:HPASs:ATTenuation?* (page 194)
- *SENSe#:FILTer:HPASs:FREQuency* (page 194)
- *SENSe#:FILTer:HPASs:FREQuency?* (page 195)
- *SENSe#:FILTer:LPASs:ATTenuation* (page 195)
- *SENSe#:FILTer:LPASs:ATTenuation?* (page 196)
- *SENSe#:FILTer:LPASs:FREQuency* (page 196)
- *SENSe#:FILTer:LPASs:FREQuency?* (page 197)
- *SENSe#:FILTer:OPTimization* (page 197)
- *SENSe#:FILTer:OPTimization?* (page 197)

CM-10

- *SENSe#:CURRent:RANGe* (page 189)
- *SENSe#:CURRent:RANGe:AUTO* (page 190)
- *SENSe#:CURRent:RANGe:AUTO?* (page 190)
- *SENSe#:CURRent:RANGe?* (page 191)
- *SENSe#:FRTHreshold* (page 199)
- *SENSe#:FRTHreshold?* (page 199)
- *SENSe#:BIAS[:STATe]* (page 187)

6.8.7 Auxiliary Inputs and Outputs

6.8.8 Calculate

These commands are used to configure and query calculated measurements.

- *CALCulate:SENSe#:RESistance?* (page 154)
- *CALCulate:SENSe#:RESistance:SOURce* (page 153)
- *CALCulate:SENSe#:RESistance:SOURce?* (page 154)

6.8.9 Statistics

These commands are used to obtain statistics from the instrument about the related measurement readings.

- *STATistic:MMAXimum[:MULTiple]?* (page 286)
- *STATistic:MINimum:RESet* (page 283)
- *STATistic:MAXimum:RESet* (page 279)
- *STATistic:SENSe#:RESet* (page 287)

Minimum

- *STATistic:MINimum:SENSe#:DC?* (page 283)
- *STATistic:MINimum:SENSe#:RMS?* (page 286)
- *STATistic:MINimum:SENSe#:PTPeak?* (page 285)
- *STATistic:MINimum:SENSe#:PPEak?* (page 285)
- *STATistic:MINimum:SENSe#:NPEak?* (page 285)
- *STATistic:MINimum:SENSe#:LIA:X?* (page 284)
- *STATistic:MINimum:SENSe#:LIA:Y?* (page 284)
- *STATistic:MINimum:SENSe#:LIA:R?* (page 283)
- *STATistic:MINimum:SENSe#:LIA:THETa?* (page 284)

Maximum

- *STATistic:MAXimum:SENSe#:DC?* (page 279)
- *STATistic:MAXimum:SENSe#:RMS?* (page 282)
- *STATistic:MAXimum:SENSe#:PTPeak?* (page 282)
- *STATistic:MAXimum:SENSe#:PPEak?* (page 281)
- *STATistic:MAXimum:SENSe#:NPEak?* (page 281)
- *STATistic:MAXimum:SENSe#:LIA:X?* (page 280)
- *STATistic:MAXimum:SENSe#:LIA:Y?* (page 281)
- *STATistic:MAXimum:SENSe#:LIA:R?* (page 280)
- *STATistic:MAXimum:SENSe#:LIA:THETa?* (page 280)

6.8.10 Miscellaneous

- *CALibration:DATE?* (page 154)
- *CALibration:SNUMber?* (page 156)
- *CALibration:TEMPerature?* (page 157)
- *CALibration:SCALibration:DATE?* (page 155)
- *CALibration:SCALibration:RESet* (page 155)
- *CALibration:SCALibration:RUN* (page 155)
- *CALibration:SCALibration:STATus?* (page 156)
- *CALibration:SCALibration:TEMPerature?* (page 156)
- *SENSe#:NAME* (page 210)
- *SENSe#:NAME?* (page 211)
- *SENSe#:NOTes* (page 211)
- *SENSe#:NOTes?* (page 211)
- *SENSe#:MODel?* (page 210)
- *SENSe#:SERial?* (page 215)
- *SENSe#:HWVersion?* (page 200)
- *SENSe#:SCALibration:DATE?* (page 214)
- *SENSe#:SCALibration:RESet* (page 214)

6.8.11 Digital I/O

6.8.12 Settings Profiles

6.8.13 Parameter Sweeping

6.8.14 System

- *SYSTem:LFRequency:DETected?* (page 289)
- *SYSTem:LFRequency:ERRor?* (page 289)
- *SYSTem:LOAD* (page 290)
- *SYSTem:PRESet* (page 300)
- *SYSTem:TIME* (page 300)
- *SYSTem:TIME?* (page 300)
- *SYSTem:TZONe* (page 300)
- *SYSTem:TZONe?* (page 300)
- *SYSTem:VERSion?* (page 301)

CALCulate:SENSe#:RESistance:ETYPe

Sets the present resistance excitation type of the specified module. For DC, the measure mode is DC and source shape is DC. For AC, the measure mode is Lock-in and source shape is Sine.

```
CALCulate:SENSe#:RESistance:ETYPe <excitation>
```

Parameters

Name	Type	Description
excitation	*Character*	The desired excitation type; either DC or AC.

Returns

None

CALCulate:SENSe#:RESistance:ETYPe?

Gets the present resistance excitation type of the specified module. This represents the combination of the specified measure module mode and the selected source module shape.

```
CALCulate:SENSe#:RESistance:ETYPe?
```

Returns

DC when the measure mode is DC and source shape is DC; AC when the measure mode is Lock-in and source shape is Sine; otherwise Invalid.

Return type is *Character*

CALCulate:SENSe#:RESistance:MODE

Sets the present resistance mode of the specified module.

```
CALCulate:SENSe#:RESistance:MODE <optimization>
```

Parameters

Name	Type	Description
optimization	*Character*	The desired mode; NOISe or POWer

Returns

None

CALCulate:SENSe#:RESistance:MODE?

Gets the preset resistance mode of the specified module.

```
CALCulate:SENSe#:RESistance:MODE?
```

Returns

The present resistance mode; NOISe or POWer.

Return type is *Character*

CALCulate:SENSe#:RESistance:OPTimize

Gets the preset state of resistance optimization on the specified module.

```
CALCulate:SENSe#:RESistance:OPTimize <state>
```

Parameters

Name	Type	Description
state	*Boolean*	The state of resistance optimization.

Returns

None

CALCulate:SENSe#:RESistance:OPTimize?

Gets the present state of optimization on the specified module. When optimization is enabled, the instrument will set other settings based on the selected resistance range and mode settings.

```
CALCulate:SENSe#:RESistance:OPTimize?
```

Returns

The state of the optimization. True if optimizing for resistance; else false.

Return type is *Boolean*

CALCulate:SENSe#:RESistance:RANGe

Sets the present resistance range of the specified module.

```
CALCulate:SENSe#:RESistance:RANGe <range>
```

Parameters

Name	Type	Description
range	*NRf*	The desired range in Ohms.

Returns

None

CALCulate:SENSe#:RESistance:RANGe?

Gets the present resistance range of the specified module.

```
CALCulate:SENSe#:RESistance:RANGe?
```

Returns

The resistance range in Ohms.

Return type is *NRf*

CALCulate:SENSe#:RESistance:SOURce

Sets the source module that should be used to calculate resistance.

```
CALCulate:SENSe#:RESistance:SOURce <reference>
```

Parameters

Name	Type	Description
reference	*Character*	The source module to reference. S1, S2 or S3 for M81-6. S1 or S2 for M81-4. S1 only for M81-2.

Returns

None

CALCulate:SENSe#:RESistance:SOURce?

Returns the source module that is currently being used to calculate the resistance.

```
CALCulate:SENSe#:RESistance:SOURce?
```

Returns

The source module being referenced. S1, S2 or S3 for M81-6. S1 or S2 for M81-4. S1 only for M81-2.

Return type is *Character*

CALCulate:SENSe#:RESistance?

Immediately returns the resistance in Ohms. May return a NaN if attempting to divide by zero, if the source is incompatible, or if either measure module or its designated source have an error.

```
CALCulate:SENSe#:RESistance?
```

Returns

Resistance in Ohms.

Return type is *NRf*

CALibration:DATE?

Queries the date of the calibration.

```
CALibration:DATE?
```

Returns

Date in the form of year, month, day, hour, minute, second

Return type is *NR1*

Examples

```
CALibration:DATE?
```

CALibration:SCALibration:DATE?

Queries the self-calibration date.

`CALibration:SCALibration:DATE?`

Returns

Date in the form of year, month, day, hour, minute, second

Return type is *NR1*

CALibration:SCALibration:RESet

Restore the factory self-calibration.

`CALibration:SCALibration:RESet`

Returns

None

CALibration:SCALibration:RUN

Runs a self-calibration sequence. This is overlapping (not a blocking operation).

`CALibration:SCALibration:RUN`

Returns

None

CALibration:SCALibration:STATus?

Returns the status of the head self cal.

`CALibration:SCALibration:STATus?`

Returns

Self-calibration status. NONE, RUNNing, PASSed, or FAILed.

Return type is *Character*

CALibration:SCALibration:TEMPerature?

Queries the self-calibration temperature.

`CALibration:SCALibration:TEMPerature?`

Returns

Temperature in °C

Return type is *NRf*

CALibration:SNUMber?

Queries the unit's serial number.

`CALibration:SNUMber?`

Returns

Serial number as a string

Return type is *String*

Examples

```
CALibration:SNUMber?
```

CALibration:TEMPerature?

Queries the calibration temperature.

`CALibration:TEMPerature?`

Returns

Temperature as a double

Return type is *NRf*

Examples

```
CALibration:TEMPerature?
```

FETCh:SENSe#:DC:IMMediate?

Immediately returns the present DC measurement for the specified module without waiting for the value to become valid. The module must be in DC or AC mode.

`FETCh:SENSe#:DC:IMMediate?`

Returns

DC measurement in module units.

Return type is *NRf*

FETCh:SENSe#:DC:RELative:IMMediate?

Immediately returns the present relative DC measurement for the specified module without waiting for the value to become valid. The module must be in DC mode.

```
FETCh:SENSe#:DC:RELative:IMMediate?
```

Returns

Relative DC measurement in module units.

Return type is *NRf*

FETCh:SENSe#:DC:RELative?

Returns the present relative DC measurement for the specified module. The module must be in DC mode.

```
FETCh:SENSe#:DC:RELative?
```

Returns

Relative DC measurement in module units.

Return type is *NRf*

FETCh:SENSe#:DC?

Returns the present DC measurement for the specified module. The module must be in DC or AC mode.

```
FETCh:SENSe#:DC?
```

Returns

DC measurement in module units.

Return type is *NRf*

FETCh:SENSe#:LIA:FREQuency?

Returns the present lock-in frequency. The module must be in *lock-in mode* (page 35).

`FETCh:SENSe#:LIA:FREQuency?`

Returns

Frequency in Hz.

Return type is *NRf*

FETCh:SENSe#:LIA:LOCK?

Returns the status of the lock-in reference frequency. The module must be in *lock-in mode* (page 35).

`FETCh:SENSe#:LIA:LOCK?`

Returns

1 if locked, 0 if unlocked.

Return type is *Boolean*

FETCh:SENSe#:LIA:R:IMMediate?

Immediately returns the present magnitude measurement from the lock-in for the specified module without waiting for the value to become valid. See *Lock-in mode* (page 35). The module must be in lock-in mode.

`FETCh:SENSe#:LIA:R:IMMediate?`

Returns

Magnitude measurement in module units.

Return type is *NRf*

FETCh:SENSe#:LIA:R?

Returns the present magnitude measurement from the lock-in for the specified module. See *Lock-in mode* (page 35). The module must be in lock-in mode.

```
FETCh:SENSe#:LIA:R?
```

Returns

Magnitude measurement in module units.

Return type is *NRf*

FETCh:SENSe#:LIA:THETa:IMMediate?

Immediately returns the present angle measurement from the lock-in for the specified module without waiting for the value to become valid. See *Lock-in mode* (page 35). The module must be in lock-in mode.

```
FETCh:SENSe#:LIA:THETa:IMMediate?
```

Returns

Angle measurement in degrees.

Return type is *NRf*

FETCh:SENSe#:LIA:THETa?

Returns the present angle measurement from the lock-in for the specified module. See *Lock-in mode* (page 35). The module must be in lock-in mode.

```
FETCh:SENSe#:LIA:THETa?
```

Returns

Angle measurement in degrees.

Return type is *NRf*

FETCh:SENSe#:LIA:X:IMMediate?

Immediately returns the present X measurement from the lock-in for the specified module without waiting for the value to become valid. The module must be in lock-in mode.

```
FETCh:SENSe#:LIA:X:IMMediate?
```

Returns

X measurement in module units.

Return type is *NRf*

FETCh:SENSe#:LIA:X?

Returns the present X measurement from the lock-in for the specified module. The module must be in lock-in mode.

```
FETCh:SENSe#:LIA:X?
```

Returns

X measurement in module units.

Return type is *NRf*

FETCh:SENSe#:LIA:Y:IMMediate?

Immediately returns the present Y measurement from the lock-in for the specified module without waiting for the value to become valid. The module must be in lock-in mode.

```
FETCh:SENSe#:LIA:Y:IMMediate?
```

Returns

Y measurement in module units.

Return type is *NRf*

FETCh:SENSe#:LIA:Y?

Returns the present Y measurement from the lock-in for the specified module. The module must be in lock-in mode.

```
FETCh:SENSe#:LIA:Y?
```

Returns

Y measurement in module units.

Return type is *NRf*

FETCh:SENSe#:NPEak:IMMediate?

Immediately returns the present *negative peak* (page 34) measurement for the specified module without waiting for the value to become valid.

```
FETCh:SENSe#:NPEak:IMMediate?
```

Returns

Negative peak measurement in module units.

Return type is *NRf*

FETCh:SENSe#:NPEak?

Returns the present *negative peak* (page 34) measurement for the specified module.

```
FETCh:SENSe#:NPEak?
```

Returns

Negative peak measurement in module units.

Return type is *NRf*

FETCh:SENSe#:PPEak:IMMediate?

Immediately returns the present *positive peak* (page 34) measurement for the specified module without waiting for the value to become valid.

`FETCh:SENSe#:PPEak:IMMediate?`

Returns

Positive peak measurement in module units.

Return type is *NRf*

FETCh:SENSe#:PPEak?

Returns the present *positive peak* (page 34) measurement for the specified module.

`FETCh:SENSe#:PPEak?`

Returns

Positive peak measurement in module units.

Return type is *NRf*

FETCh:SENSe#:PTPeak:IMMediate?

Immediately returns the present *peak to peak* (page 34) measurement for the specified module without waiting for the value to become valid.

`FETCh:SENSe#:PTPeak:IMMediate?`

Returns

Peak to peak measurement in module units.

Return type is *NRf*

FETCh:SENSe#:PTPeak?

Returns the present *peak to peak* (page 34) measurement for the specified module.

```
FETCh:SENSe#:PTPeak?
```

Returns

Peak to peak measurement in module units.

Return type is *NRf*

FETCh:SENSe#:RMS:IMMediate?

Immediately returns the present RMS measurement for the specified module without waiting for the value to become valid. The module must be in DC or AC mode.

```
FETCh:SENSe#:RMS:IMMediate?
```

Returns

RMS measurement in module units.

Return type is *NRf*

FETCh:SENSe#:RMS:RELative:IMMediate?

Immediately returns the present relative RMS measurement for the specified module without waiting for the value to become valid. The module must be in AC mode.

```
FETCh:SENSe#:RMS:RELative:IMMediate?
```

Returns

Relative RMS measurement in module units.

Return type is *NRf*

FETCh:SENSe#:RMS:RELative?

Returns the present relative RMS measurement for the specified module. The module must be in AC mode.

```
FETCh:SENSe#:RMS:RELative?
```

Returns

Relative RMS measurement in module units.

Return type is *NRf*

FETCh:SENSe#:RMS?

Returns the present RMS measurement for the specified module. The module must be in DC or AC mode.

```
FETCh:SENSe#:RMS?
```

Returns

RMS measurement in module units.

Return type is *NRf*

FETCh:SENSe#:TEMPerature?

Returns the temperature of the measure module.

```
FETCh:SENSe#:TEMPerature?
```

Returns

The temperature in °C.

Return type is *Character*

FETCh:SOURce#:READback:DC[:IMMediate]?

Returns the DC readback measurement for the specified module.

```
FETCh:SOURce#:READback:DC[:IMMediate]?
```

Returns

DC readback measurement in module units.

Return type is *NRf*

FETCh:SOURce#:READback:RMS[:IMMediate]?

Returns the DC readback measurement for the specified module.

```
FETCh:SOURce#:READback:RMS[:IMMediate]?
```

Returns

DC readback measurement in module units.

Return type is *NRf*

FETCh:SOURce#:TEMPerature?

Returns the temperature of the source module.

```
FETCh:SOURce#:TEMPerature?
```

Returns

The temperature in °C.

Return type is *Character*

FETCh:TEMPerature?

Returns the temperature of the instrument.

`FETCh:TEMPerature?`

Returns

The temperature in °C.

Return type is *Character*

FETCh[:MULTiple]?

Query multiple parameters sampled at the same moment in time.

`FETCh[:MULTiple]?`

Returns

Comma separated list of values corresponding to the input parameters and channels.

Return type is *Character*

Examples

```
To fetch DC from measure module 1, RMS from measure module 2, and Peak-to-
↪Peak from measure module 3:
  FETCh? MDC,1,MRMS,2,MPTPeak,3
Example result:
  1.23, 2.34, 3.45
```

INPut:REFerence:EDGe

Set the reference input edge.

`INPut:REFerence:EDGe <edge>`

Parameters

Name	Type	Description
edge	*Character*	Reference input edge. Rising or Falling.

Returns

None

INPut:REFerence:EDGe?

Query the reference input edge.

```
INPut:REFerence:EDGe?
```

Returns

Reference input edge. Rising or Falling.

Return type is *Character*

OUTPut:MONitor:MLEVel

Set the monitor out manual level in volts. Only relevant in manual mode.

```
OUTPut:MONitor:MLEVel <value>
```

Parameters

Name	Type	Description
value	*NRf*	Manual level in volts between -10.0V and +10.0V.

Returns

None

OUTPut:MONitor:MLEVel?

Query the monitor out manual level. Only relevant in manual mode.

```
OUTPut:MONitor:MLEVel?
```

Returns

Manual level in volts.

Return type is *NRf*

OUTPut:MONitor:MODe

Set the monitor output mode.

```
OUTPut:MONitor:MODe <source>
```

Parameters

Name	Type	Description
source	*Character*	Monitor output mode. M1, M2, M3, or MANual.

Returns

None

OUTPut:MONitor:MODe?

Query the monitor output mode.

```
OUTPut:MONitor:MODe?
```

Returns

Monitor output mode. M1, M2, M3, or MANual. Use MANual with :MLEVel.

Return type is *Character*

OUTPut:MONitor:SCALe?

Query the monitor out scaling value. If in manual mode, a value of 1 will be returned.

```
OUTPut:MONitor:SCALe?
```

Returns

Scaling value

Return type is *NRf*

OUTPut:MONitor[:STATe]

Set the monitor output status.

```
OUTPut:MONitor[:STATe] <state>
```

Parameters

Name	Type	Description
state	*Boolean*	1 to enable monitor output, 0 to disable monitor output.

Returns

None

OUTPut:MONitor[:STATe]?

Query the monitor output status.

`OUTPut:MONitor[:STATe]?`

Returns

1 if monitor output is enabled, 0 if monitor output is disabled.

Return type is *Boolean*

OUTPut:REFerence:SOURce

Set the reference output mode.

`OUTPut:REFerence:SOURce <source>`

Parameters

Name	Type	Description
source	*Character*	Reference output mode. S1, S2, or S3 for M81-6. S1 or S2 for M81-4. S1 only for M81-2.

Returns

None

OUTPut:REFerence:SOURce?

Query the reference output mode.

`OUTPut:REFerence:SOURce?`

Returns

Reference output mode. S1, S2, or S3 for M81-6. S1 or S2 for M81-4. S1 only for M81-2.

Return type is *Character*

OUTPut:REFerence[:STATe]

Set the reference output status.

`OUTPut:REFerence[:STATe] <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable reference output, 0 to disable reference output.

Returns

None

OUTPut:REFerence[:STATe]?

Query the reference output status.

`OUTPut:REFerence[:STATe]?`

Returns

1 if reference output is enabled, 0 if reference output is disabled.

Return type is *Boolean*

PROFile:CREAte

Create a new profile using the present instrument configuration.

```
PROFile:CREAte <name>,<description>
```

Parameters

Name	Type	Description
name	*String*	Unique name to give the profile.
description	*String*	Optional description of the profile.

Returns

None

PROFile:DELete

Delete a profile.

```
PROFile:DELete <name>
```

Parameters

Name	Type	Description
name	*String*	The name of the profile to delete.

Returns

None

PROFile:DELete:ALL

Delete all profiles.

```
PROFile:DELete:ALL
```

Returns

None

PROFile:DESCription

Set a profile's description. Any existing description will be overwritten.

```
PROFile:DESCription <name>,<description>
```

Parameters

Name	Type	Description
name	*String*	Name of the profile to set the description.
description	*String*	The new description of the profile.

Returns

None

PROFile:DESCription?

Query a profile's description.

```
PROFile:DESCription? <name>
```

Parameters

Name	Type	Description
name	*String*	Name of the profile to query the description for.

Returns

The profile's description.

Return type is *String*

PROFile:JSON?

Query a JSON representation of a profile.

`PROFile:JSON? <name>,<pretty>`

Parameters

Name	Type	Description
name	*String*	The name of the profile to query.
pretty	*Boolean*	1 to format the JSON string with indentation and newline, 0 for a single line.

Returns

JSON representation of the profile.

Return type is *String*

PROFile:LIST?

Query a list of the saved profile names.

`PROFile:LIST?`

Returns

Saved profile names as comma separated strings.

Return type is *String*

PROFile:REName

Rename a profile. New name must be unique.

```
PROFile:REName <name>,<newName>
```

Parameters

Name	Type	Description
name	*String*	The name of the profile to rename.
newName	*String*	The new name of the profile.

Returns

None

PROFile:RESTore

Restore a profile.

```
PROFile:RESTore <name>
```

Parameters

Name	Type	Description
name	*String*	The name of the profile to restore.

Returns

None

PROFile:RESTore[:VALid]?

Query if a profile is valid to restore.

`PROFile:RESTore[:VALid]? <name>`

Parameters

Name	Type	Description
name	*String*	The name of the profile to validate.

Returns

1 if the profile is valid to restore, 0 otherwise.

Return type is *Boolean*

PROFile:UPDate

Update a profile with the present instrument configuration.

`PROFile:UPDate <name>`

Parameters

Name	Type	Description
name	*String*	The name of the profile to update.

Returns

None

PROFile[:SUMMary]?

Query the description and module models of the profile setting.

```
PROFile[:SUMMary]? <name>
```

Parameters

Name	Type	Description
name	*String*	The name of the profile to query.

Returns

The description and module models of the profile setting as comma separated strings.

Return type is *String*

READ:SENSe#:ABORt

Abort the read initiated by the READ:SENSE#:START command.

```
READ:SENSe#:ABORt
```

Returns

None

READ:SENSe#:DC:RELative?

Acquires and returns the relative DC measurement for the specified module. The value is returned after waiting for the configured NPLC to complete. The module must be in DC mode.

```
READ:SENSe#:DC:RELative?
```

Returns

Relative DC measurement in module units.

Return type is *Character*

READ:SENSe#:DC?

Acquires and returns the DC measurement for the specified module. The value is returned after waiting for the configured NPLC to complete. The module must be in DC or AC mode.

```
READ:SENSe#:DC?
```

Returns

DC measurement in module units.

Return type is *Character*

READ:SENSe#:NPEak?

Acquires and returns the *negative peak* (page 34) measurement for the specified module. The value is returned after waiting for the configured NPLC to complete.

```
READ:SENSe#:NPEak?
```

Returns

Negative peak measurement in module units.

Return type is *Character*

READ:SENSe#:PPEak?

Acquires and returns the *positive peak* (page 34) measurement for the specified module. The value is returned after waiting for the configured NPLC to complete.

```
READ:SENSe#:PPEak?
```

Returns

Positive peak measurement in module units.

Return type is *Character*

READ:SENSe#:PTPeak?

Acquires and returns the *peak to peak* (page 34) measurement for the specified module. The value is returned after waiting for the configured NPLC to complete.

`READ:SENSe#:PTPeak?`

Returns

Peak to peak measurement in module units.

Return type is *Character*

READ:SENSe#:RESult:DC:RELative?

Queries the relative DC measurement from the last read initiated by the READ:SENSE#:START command. The module must be in DC mode.

`READ:SENSe#:RESult:DC:RELative?`

Returns

Relative DC measurement in module units.

Return type is *Character*

READ:SENSe#:RESult:DC?

Queries the DC measurement from the last read initiated by the READ:SENSE#:START command. The module must be in DC or AC mode.

`READ:SENSe#:RESult:DC?`

Returns

DC measurement in module units.

Return type is *Character*

READ:SENSe#:RESult:NPEak?

Queries the *negative peak* (page 34) measurement from the last read initiated by the READ:SENSE#:START command.

`READ:SENSe#:RESult:NPEak?`

Returns

Negative peak measurement in module units.

Return type is *Character*

READ:SENSe#:RESult:PPEak?

Queries the *positive peak* (page 34) measurement from the last read initiated by the READ:SENSE#:START command.

`READ:SENSe#:RESult:PPEak?`

Returns

Positive peak measurement in module units.

Return type is *Character*

READ:SENSe#:RESult:PTPeak?

Queries the *peak to peak* (page 34) measurement from the last read initiated by the READ:SENSE#:START command.

`READ:SENSe#:RESult:PTPeak?`

Returns

Peak to peak measurement in module units.

Return type is *Character*

READ:SENSe#:RESult:RMS:RELative?

Queries the relative RMS measurement from the last read initiated by the READ:SENSE#:START command. The module must be in AC mode.

`READ:SENSe#:RESult:RMS:RELative?`

Returns

Relative RMS measurement in module units.

Return type is *Character*

READ:SENSe#:RESult:RMS?

Queries the RMS measurement from the last read initiated by the READ:SENSE#:START command. The module must be in DC or AC mode.

`READ:SENSe#:RESult:RMS?`

Returns

RMS measurement in module units.

Return type is *Character*

READ:SENSe#:RMS:RELative?

Acquires and returns the relative *RMS* (page 34) measurement for the specified module. The value is returned after waiting for the configured NPLC to complete. The module must be in AC mode.

`READ:SENSe#:RMS:RELative?`

Returns

RMS measurement in module units.

Return type is *Character*

READ:SENSe#:RMS?

Acquires and returns the RMS measurement for the specified module. The value is returned after waiting for the configured NPLC to complete. The module must be in DC or AC mode.

`READ:SENSe#:RMS?`

Returns

RMS measurement in module units.

Return type is *Character*

READ:SENSe#:RUNNing?

Query if the read initiated by the READ:SENSE#:START command is running.

`READ:SENSe#:RUNNing?`

Returns

The measurement is running.

Return type is *Boolean*

READ:SENSe#:STARt

Starts acquiring a single reading for the specified module. This command returns immediately after starting the read operation. The read operation will wait for the configured NPLC to complete. The state can be queried with READ:SENSE#:RUNNING?. The results can be queried with READ:SENSE#:RESULT queries when operation is complete.

`READ:SENSe#:STARt`

Returns

None

READ[:MULTiple]:ABORt

Abort the multiple reading initiated by the READ:MULTIPLE:START command.

`READ[:MULTiple]:ABORt`

Returns

None

READ[:MULTiple]:RESult?

Query the results of the last multiple reading initiated by the READ:MULTIPLE:START command.

`READ[:MULTiple]:RESult?`

Returns

Comma separated list of values corresponding to the input parameters and channels.

Return type is *Character*

Examples

```
If the command was setup to read DC from measure module 1, RMS from␣
↪measure module 2,
and Peak-to-Peak from measure module 3 with the following command:
  READ:MULTiple:START MDC,1,MRMS,2,MPTPeak,3
Then the results can be queried with:
  READ:MULTiple:RESULT?
Which return results similar to:
  1.25, 2.41, 3.38
```

READ[:MULTiple]:RUNNing?

Query if the multiple reading initiated by the READ:MULTIPLE:START command is running.

```
READ[:MULTiple]:RUNNing?
```

Returns

The measurement is running.

Return type is *Boolean*

READ[:MULTiple]:STARt

Starts acquiring readings for multiple parameters to be measured at the same time. This command returns immediately after starting the read operation. The read operation will wait for the highest configured NPLC to complete. The state can be queried with READ:MULTIPLE:RUNNING?. The results can be queried with READ:MULTIPLE:RESULT? when operation is complete.

```
READ[:MULTiple]:STARt
```

Returns

None

Examples

```
To start reading DC from measure module 1, RMS from measure module 2, and␣
↪Peak-to-Peak from measure module 3:
  READ:START MDC,1,MRMS,2,MPTPeak,3
```

READ[:MULTiple]?

Query multiple parameters to be measured at the same time. The value is returned after waiting for the highest configured NPLC to complete.

```
READ[:MULTiple]?
```

Returns

Comma separated list of values corresponding to the input parameters and channels.

Return type is *Character*

Examples

```
To read DC from measure module 1, RMS from measure module 2, and Peak-to-
↪Peak from measure module 3:
  READ? MDC,1,MRMS,2,MPTPeak,3
Example result:
  1.25, 2.41, 3.38
```

SENSe:NCHannels?

Query the number of measure channels available on this instrument.

`SENSe:NCHannels?`

Returns

The number of measure channels available on this instrument.

Return type is *Character*

SENSe#:BIAS:VOLTage[:DC]

Set the bias voltage of the specified module.

`SENSe#:BIAS:VOLTage[:DC] <biasVoltage>`

Parameters

Name	Type	Description
biasVoltage	*NRf*	Bias voltage in V.

Returns

None

SENSe#:BIAS:VOLTage[:DC]?

Query the bias voltage of the specified module.

```
SENSe#:BIAS:VOLTage[:DC]?
```

Returns

Bias voltage in V.

Return type is *NRf*

SENSe#:BIAS[:STATe]

Set the input bias state for the specified module.

```
SENSe#:BIAS[:STATe] <state>
```

Parameters

Name	Type	Description
state	*Boolean*	1 to enable input bias, 0 to disable input bias.

Returns

None

SENSe#:BIAS[:STATe]?

Query the input bias state for the specified module.

```
SENSe#:BIAS[:STATe]?
```

Returns

1 if input bias is enabled, 0 if input bias is disabled.

Return type is *Boolean*

SENSe#:CONFiguration

Set the input configuration of the specified module.

`SENSe#:CONFiguration <configuration>`

Parameters

Name	Type	Description
configuration	*Character*	Input configuration. AB, A, or GROund.

Returns

None

SENSe#:CONFiguration?

Query the input configuration of the specified module.

`SENSe#:CONFiguration?`

Returns

Input configuration. AB, A, or GROund.

Return type is *Character*

SENSe#:COUPling

Set the input coupling of the specified module.

`SENSe#:COUPling <coupling>`

Parameters

Name	Type	Description
coupling	*Character*	Input coupling. AC or DC.

Returns

None

SENSe#:COUPling?

Query the input coupling of the specified module.

`SENSe#:COUPling?`

Returns

Input coupling. AC or DC.

Return type is *Character*

SENSe#:CURRent:RANGe

Set the current range of the specified module.

`SENSe#:CURRent:RANGe <range>`

Parameters

Name	Type	Description
range	*NRf*	Current range in A.

Returns

None

Remarks

If autorange is enabled, then setting the range will disable it.

SENSe#:CURRent:RANGe:AUTO

Set the autorange state for the specified module.

`SENSe#:CURRent:RANGe:AUTO <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable autorange, 0 to disable autorange.

Returns

None

SENSe#:CURRent:RANGe:AUTO?

Query the autorange state for the specified module.

`SENSe#:CURRent:RANGe:AUTO?`

Returns

1 if autorange is enabled, 0 if autorange is disabled.

Return type is *Boolean*

SENSe#:CURRent:RANGe?

Query the present current range of the specified module.

`SENSe#:CURRent:RANGe?`

Returns

Current range in A.

Return type is *NRf*

SENSe#:DIGital:FILTer:HPASs[:STATe]

Set the digital LIA high-pass filter state

`SENSe#:DIGital:FILTer:HPASs[:STATe] <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable, 0 to disable

Returns

None

SENSe#:DIGital:FILTer:HPASs[:STATe]?

Query the digital LIA high-pass filter state

```
SENSe#:DIGital:FILTer:HPASs[:STATe]?
```

Returns

1 if enabled, 0 if disabled

Return type is *Boolean*

SENSe#:DISPlay[:MODe]

Set the readings display mode of the specified module.

```
SENSe#:DISPlay[:MODe] <displayMode>
```

Parameters

Name	Type	Description
displayMode	*Char*	Display mode. DEFault, DC, RDC (Relative DC), PTPeak (Peak to Peak), TRMS (Total RMS), RRMS (Relative RMS), POLar, or RECTangular. When setting DEFault, the mode will be set to the most appropriate mode for the module when the module is displayed on the front panel home page.

Returns

None

SENSe#:DISPlay[:MODe]?

Query the readings display mode of the specified module on the home page.

```
SENSe#:DISPlay[:MODe]?
```

Returns

DEFault, DC, RDC (Relative DC), PTPeak (Peak to Peak), TRMS (Total RMS), RRMS (Relative RMS), POLar, or RECTangular.

Return type is *Character*

SENSe#:DMODe

Sets the dark mode for the specified module. If dark mode is enabled, the indicator LEDs on the front of the module will be turned off.

`SENSe#:DMODe <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable the LED, 0 to disable.

Returns

None

SENSe#:DMODe?

Query the state of the dark mode for the specified module.

`SENSe#:DMODe?`

Returns

1 if dark mode is enabled, 0 if not.

Return type is *Boolean*

SENSe#:FILTer:HPASs:ATTenuation

Set the input filter high-pass rolloff for the specified module.

```
SENSe#:FILTer:HPASs:ATTenuation <rollOff>
```

Parameters

Name	Type	Description
rollOff	*Character*	Rolloff. R6 or R12. R6 = 6 dB/octave, R12 = 12 dB/octave.

Returns

None

SENSe#:FILTer:HPASs:ATTenuation?

Query the input filter high-pass rolloff for the specified module.

```
SENSe#:FILTer:HPASs:ATTenuation?
```

Returns

Rolloff. R6 or R12. R6 = 6 dB/octave, R12 = 12 dB/octave.

Return type is *Character*

SENSe#:FILTer:HPASs:FREQuency

Set the input filter high-pass cutoff frequency for the specified module.

```
SENSe#:FILTer:HPASs:FREQuency <cutoffFrequency>
```

Parameters

Name	Type	Description
cutoffFrequency	*Character*	Cutoff frequency. NONE, F10, F30, F100, F300, F1000, F3000, or F10000. F10 = 10 Hz, etc.

Returns

None

SENSe#:FILTer:HPASs:FREQuency?

Query the input filter high-pass cutoff frequency for the specified module.

```
SENSe#:FILTer:HPASs:FREQuency?
```

Returns

Cutoff frequency. NONE, F10, F30, F100, F300, F1000, F3000, or F10000. F10 = 10 Hz, etc.

Return type is *Character*

SENSe#:FILTer:LPASs:ATTenuation

Set the input filter low-pass rolloff for the specified module.

```
SENSe#:FILTer:LPASs:ATTenuation <rollOff>
```

Parameters

Name	Type	Description
rollOff	*Character*	Rolloff. R6 or R12. R6 = 6 dB/octave, R12 = 12 dB/octave.

Returns

None

SENSe#:FILTer:LPASs:ATTenuation?

Query the input filter low-pass rolloff for the specified module.

```
SENSe#:FILTer:LPASs:ATTenuation?
```

Returns

Rolloff. R6 or R12. R6 = 6 dB/octave, R12 = 12 dB/octave.

Return type is *Character*

SENSe#:FILTer:LPASs:FREQuency

Set the input filter low-pass cutoff frequency for the specified module.

```
SENSe#:FILTer:LPASs:FREQuency <cutoffFrequency>
```

Parameters

Name	Type	Description
cutoffFrequency	*Character*	Cutoff frequency. NONE, F10, F30, F100, F300, F1000, F3000, or F10000. F10 = 10 Hz, etc.

Returns

None

SENSe#:FILTer:LPASs:FREQuency?

Query the input filter low-pass cutoff frequency for the specified module.

```
SENSe#:FILTer:LPASs:FREQuency?
```

Returns

Cutoff frequency. NONE, F10, F30, F100, F300, F1000, F3000, or F10000. F10 = 10 Hz, etc.

Return type is *Character*

SENSe#:FILTer:OPTimization

Set the input filter optimization mode for the specified module.

```
SENSe#:FILTer:OPTimization <mode>
```

Parameters

Name	Type	Description
mode	*Character*	Optimization mode. NOISe or REServe.

Returns

None

SENSe#:FILTer:OPTimization?

Query the input filter optimization mode for the specified module.

```
SENSe#:FILTer:OPTimization?
```

Returns

Optimization mode. NOISe or REServe.

Return type is *Character*

SENSe#:FILTer[:STATe]

Set the input filter state for the specified module.

`SENSe#:FILTer[:STATe] <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable filter, 0 to disable filter.

Returns

None

SENSe#:FILTer[:STATe]?

Query the input filter state for the specified module.

`SENSe#:FILTer[:STATe]?`

Returns

1 if filter is enabled, 0 if filter is disabled.

Return type is *Boolean*

SENSe#:FRTHreshold

Sets the frequency range threshold for the specified module. When the modules range is set to Auto, a range such that the frequency of the signal does not exceed the given percentage of the bandwidth of the range will be chosen.

```
SENSe#:FRTHreshold <value>
```

Parameters

Name	Type	Description
value	*NRf*	Frequency range threshold normalized to the -3 db point with a valid range of 0.0 to 1.0. For example, a value of 0.1 means 10 % of the -3 db point.

Returns

None

SENSe#:FRTHreshold?

Query the frequency range threshold for the specified module.

```
SENSe#:FRTHreshold?
```

Returns

Frequency range threshold normalized to the -3 db point. For example, a value of 0.1 means 10 % of the -3 db point.

Return type is *NRf*

SENSe#:FUNCtion

Set the measure function of the specified module.

```
SENSe#:FUNCtion <value>
```

Parameters

Name	Type	Description
value	*Character*	Measure function. CURRent or VOLTage.

Returns

None

SENSe#:FUNCtion?

Query the measure function of the specified module.

```
SENSe#:FUNCtion?
```

Returns

Measure function. CURRent or VOLTage.

Return type is *Character*

SENSe#:HWVersion?

Query the hardware version of the specified module.

```
SENSe#:HWVersion?
```

Returns

Module hardware version.

Return type is *NR1*

SENSe#:IDENTify

Set the pod identification status for the specified module.

`SENSe#:IDENTify <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable identification, 0 to disable identification.

Returns

None

SENSe#:IDENTify?

Get the state of the pod identification for the specified module.

`SENSe#:IDENTify?`

Returns

None

SENSe#:LIA:DHARmonic

Set the lock-in *reference harmonic* (page 37) for the specified module. 1 is the fundamental frequency, 2 is twice the fundamental frequency, etc.

`SENSe#:LIA:DHARmonic <harmonic>`

Parameters

Name	Type	Description
harmonic	*NR1*	Reference harmonic. The minimum is 1. The harmonic times the reference frequency may not exceed 100 kHz.

Returns

None

SENSe#:LIA:DHARmonic?

Query the lock-in *reference harmonic* (page 37) for the specified module. 1 is the fundamental frequency, 2 is twice the fundamental frequency, etc.

```
SENSe#:LIA:DHARmonic?
```

Returns

Reference harmonic.

Return type is *NR1*

SENSe#:LIA:DPHase

Set the lock-in *reference phase shift* (page 37) for the specified module.

```
SENSe#:LIA:DPHase <phase>
```

Parameters

Name	Type	Description
phase	*NRf*	Phase shift in degrees between -360.0 and +360.0.

Returns

None

SENSe#:LIA:DPHase:AUTO

Automatically calculates demodulated phase for the specified module and sets its lock-in *reference phase shift* (page 37).

```
SENSe#:LIA:DPHase:AUTO
```

Returns

None

SENSe#:LIA:DPHase?

Query the lock-in *reference phase shift* (page 37) for the specified module.

```
SENSe#:LIA:DPHase?
```

Returns

Phase shift in degrees.

Return type is *NRf*

SENSe#:LIA:ENBW?

Query the lock-in *equivalent noise bandwidth* (page 38) (ENBW) for the specified module.

```
SENSe#:LIA:ENBW?
```

Returns

Equivalent noise bandwidth (ENBW) in Hz.

Return type is *NRf*

SENSe#:LIA:FIR:CYCLes

Set the number of lock-in *PSD output FIR filter cycles* (page 43) for the specified module.

```
SENSe#:LIA:FIR:CYCLes <cycles>
```

Parameters

Name	Type	Description
cycles	*Character*	The desired number of FIR cycles.

Returns

None

SENSe#:LIA:FIR:CYCLes?

Query the number of lock-in *PSD output FIR filter cycles* (page 43) for the specified module.

```
SENSe#:LIA:FIR:CYCLes?
```

Returns

The number of FIR cycles.

Return type is *Character*

SENSe#:LIA:FIR[:STATe]

Set the lock-in *PSD output FIR filter* (page 43) status for the specified module.

```
SENSe#:LIA:FIR[:STATe] <state>
```

Parameters

Name	Type	Description
state	*Boolean*	1 to enable the FIR filter, 0 to disable the FIR filter.

Returns

None

SENSe#:LIA:FIR[:STATe]?

Query the lock-in *PSD output FIR filter* (page 43) status for the specified module.

```
SENSe#:LIA:FIR[:STATe]?
```

Returns

1 if the FIR filter is enabled, 0 if the FIR filter is disabled.

Return type is *Boolean*

SENSe#:LIA:IIR[:STATe]

Set the lock-in *PSD output IIR filter* (page 43) status for the specified module.

```
SENSe#:LIA:IIR[:STATe] <state>
```

Parameters

Name	Type	Description
state	*Boolean*	1 to enable the IIR filter, 0 to disable the IIR filter.

Returns

None

SENSe#:LIA:IIR[:STATe]?

Query the lock-in *PSD output IIR filter* (page 43) status for the specified module.

```
SENSe#:LIA:IIR[:STATe]?
```

Returns

1 if the IIR filter is enabled, 0 if the IIR filter is disabled.

Return type is *Boolean*

SENSe#:LIA:ROLLoff

Set the lock-in *PSD output filter rolloff* (page 38) for the specified module.

```
SENSe#:LIA:ROLLoff <rollOff>
```

Parameters

Name	Type	Description
rollOff	*Character*	Rolloff. R6, R12, R18, or R24. R6 = 6 dB/octave, etc.

Returns

None

SENSe#:LIA:ROLLoff?

Query the lock-in *PSD output filter rolloff* (page 38) for the specified module.

```
SENSe#:LIA:ROLLoff?
```

Returns

Rolloff. R6, R12, R18, or R24. R6 = 6 dB/octave, etc.

Return type is *Character*

SENSe#:LIA:RSOurce

Set the lock-in *reference source* (page 37) for the specified module.

```
SENSe#:LIA:RSOurce <reference>
```

Parameters

Name	Type	Description
reference	*Character*	Reference source. S1, S2, S3, or RIN for M81-6. S1, S2, or RIN for M81-4. S1 or RIN for M81-2.

Returns

None

SENSe#:LIA:RSOurce?

Query the lock-in *reference source* (page 37) for the specified module.

```
SENSe#:LIA:RSOurce?
```

Returns

Reference source. S1, S2, S3, or RIN for M81-6. S1, S2, or RIN for M81-4. S1 or RIN for M81-2.

Return type is *Character*

SENSe#:LIA:STIMe?

Query the lock-in *settle time* (page 38) for the specified module.

```
SENSe#:LIA:STIMe? <settlePercent>
```

Parameters

Name	Type	Description
settlePercent	*NRf*	Desired percent signal has settled to in percent. An entered value of *0.1* is interpreted as 0.1 %. Defaults to 0.1 if omitted.

Returns

Settle time in seconds.

Return type is *NRf*

SENSe#:LIA:TIMEconstant

Set the lock-in *time constant* (page 38) for the specified module.

```
SENSe#:LIA:TIMEconstant <timeConstant>
```

Parameters

Name	Type	Description
timeConstant	*NRf*	Time constant in seconds between 0.0001 and 10000.0.

Returns

None

SENSe#:LIA:TIMEconstant?

Query the lock-in *time constant* (page 38) for the specified module.

```
SENSe#:LIA:TIMEconstant?
```

Returns

Time constant in seconds.

Return type is *NRf*

SENSe#:LOAD

Loads all connected modules.

`SENSe#:LOAD`

Returns

None

Remarks

This command is available for backwards compatibility and is equivalent to the SYSTem:LOAD command.

SENSe#:LOAD?

Gets the loaded state of the specified module.

`SENSe#:LOAD?`

Returns

True if the module is loaded; else false

Return type is *Boolean*

SENSe#:MODE

Set the mode of the specified module.

`SENSe#:MODE <mode>`

Parameters

Name	Type	Description
mode	*Character*	Sense module mode. DC, AC, or LIA.

Returns

None

SENSe#:MODE?

Query the mode of the specified module.

`SENSe#:MODE?`

Returns

Sense module mode. DC, AC, or LIA.

Return type is *Character*

SENSe#:MODel?

Query what type of module is connected to the given source slot.

`SENSe#:MODel?`

Returns

The name of the connected module, or None if nothing is connected.

Return type is *String*

SENSe#:NAME

Sets the name of the specified module.

`SENSe#:NAME <value>`

Parameters

Name	Type	Description
value	*String*	The name of the module with a maximum of 100 characters.

Returns

None

SENSe#:NAME?

Query the name of the specified module.

```
SENSe#:NAME?
```

Returns

The name.

Return type is *String*

SENSe#:NOTes

Sets the notes of the specified module.

```
SENSe#:NOTes <value>
```

Parameters

Name	Type	Description
value	*String*	The notes to be stored, limited to 500 characters.

Returns

None

SENSe#:NOTes?

Query the notes of the specified module.

```
SENSe#:NOTes?
```

Returns

The notes.

Return type is *String*

SENSe#:NPLCycles

Set the averaging time in number of power line cycles (NPLC) of the specified module. Only relevant in DC or AC mode. For lock-in mode, see *time constant* (page 208).

```
SENSe#:NPLCycles <value>
```

Parameters

Name	Type	Description
value	*NRf*	Averaging time in number of power line cycles (NPLC) between 0.01 and 600.00.

Returns

None

SENSe#:NPLCycles?

Query the averaging time in number of power line cycles (NPLC) of the specified module.

```
SENSe#:NPLCycles?
```

Returns

Averaging time in number of power line cycles (NPLC).

Return type is *NRf*

SENSe#:PRESet

Resets the module settings to their power on defaults.

`SENSe#:PRESet`

Returns

None

SENSe#:RELative:BASEline

Sets the baseline value to the user provided value for calculating relative readings.

`SENSe#:RELative:BASEline <baseline>`

Parameters

Name	Type	Description
baseline	*NRf*	The baseline to calculate the relative reading against.

Returns

None

SENSe#:RELative:BASEline?

Returns the baseline value being used to calculate the relative readings.

`SENSe#:RELative:BASEline?`

Returns

The baseline value used to calculate the relative readings, in system units.

Return type is *NRf*

SENSe#:RELative:ZERO

Sets the present measurement as the baseline value for calculating relative readings. Only available in *lock-in mode* (page 33) or *lock-in mode* (page 34).

```
SENSe#:RELative:ZERO
```

Returns

None

SENSe#:SCALibration:DATE?

Queries the specified module's self-calibration date.

```
SENSe#:SCALibration:DATE?
```

Returns

Date in the form of year, month, day, hour, minute, second

Return type is *Character*

SENSe#:SCALibration:RESet

Resets the self-calibration status for the specified module.

```
SENSe#:SCALibration:RESet
```

Returns

None

SENSe#:SCALibration:RUN

Initiates the self-calibration process on the specified module.

```
SENSe#:SCALibration:RUN
```

Returns

None

SENSe#:SCALibration:STATus?

Query the status of the self-calibration of the specified module.

```
SENSe#:SCALibration:STATus?
```

Returns

The self-calibration state.

Return type is *String*

SENSe#:SCALibration:TEMPerature?

Queries the specified module's self-calibration temperature.

```
SENSe#:SCALibration:TEMPerature?
```

Returns

Temperature in °C

Return type is *Character*

SENSe#:SERial?

Query the serial number of the specified module.

```
SENSe#:SERial?
```

Returns

Module serial number.

Return type is *String*

SENSe#:UNLoad

Unloads the specified module. A module must be unloaded before it is physically disconnected.

```
SENSe#:UNLoad
```

Returns

None

SENSe#:VOLTage:RANGe

Set the present voltage range of the specified module.

```
SENSe#:VOLTage:RANGe <range>
```

Parameters

Name	Type	Description
range	*NRf*	Voltage range in V.

Returns

None

Remarks

If autorange is enabled, then setting the range will disable it.

SENSe#:VOLTage:RANGe:AUTO

Enables or disables autorange for the specified module.

```
SENSe#:VOLTage:RANGe:AUTO <state>
```

Parameters

Name	Type	Description
state	*Boolean*	1 to enable autorange, 0 to disable autorange.

Returns

None

SENSe#:VOLTage:RANGe:AUTO?

Query the autorange state of the specified module.

`SENSe#:VOLTage:RANGe:AUTO?`

Returns

1 if autorange is enabled, 0 if autorange is disabled.

Return type is *Boolean*

SENSe#:VOLTage:RANGe?

Query the present voltage range of the specified module.

`SENSe#:VOLTage:RANGe?`

Returns

Voltage range in V.

Return type is *NRf*

SENSe#:VOLTage:SMODe

Set the voltage sense mode of the specified module.

```
SENSe#:VOLTage:SMODe <value>
```

Parameters

Name	Type	Description
value	*Character*	Voltage sense mode. TWIRe for 2-wire local. FWIRe for 4-wire remote.

Returns

None

SENSe#:VOLTage:SMODe?

Query the voltage sense mode of the specified module.

```
SENSe#:VOLTage:SMODe?
```

Returns

Voltage sense mode. TWIRE for 2-wire local. FWIRE for 4-wire remote.

Return type is *Character*

SOURce:NCHannels?

Query the number of source channels available on this instrument.

```
SOURce:NCHannels?
```

Returns

The number of source channels available on this instrument.

Return type is *Character*

SOURce#:CMR:SOURce

Set the common mode reduction (CMR) feedback source of the specified module.

`SOURce#:CMR:SOURce <cmrFeedbackSource>`

Parameters

Name	Type	Description
cmrFeedbackSource	*Character*	Feedback source. INTernal or EXTernal.

Returns

None

SOURce#:CMR:SOURce?

Query the common mode reduction (CMR) feedback source of the specified module.

`SOURce#:CMR:SOURce?`

Returns

Feedback source. INTernal or EXTernal.

Return type is *Character*

SOURce#:CMR[:STATe]

Set the common mode reduction (CMR) state of the specified module.

```
SOURce#:CMR[:STATe] <cmrState>
```

Parameters

Name	Type	Description
cmrState	*Boolean*	1 to enable CMR, 0 to disable CMR.

Returns

None

SOURce#:CMR[:STATe]?

Query the common mode reduction (CMR) state of the specified module.

```
SOURce#:CMR[:STATe]?
```

Returns

1 if CMR is enabled, 0 if CMR is disabled.

Return type is *Boolean*

SOURce#:COUPling

Set the output coupling type of the specified module.

```
SOURce#:COUPling <coupling>
```

Parameters

Name	Type	Description
coupling	*Character*	Coupling type. AC or DC.

Returns

None

SOURce#:COUPling:AUTO

Set the automatic coupling state of the specified module.

`SOURce#:COUPling:AUTO <auto>`

Parameters

Name	Type	Description
auto	*Boolean*	1 to enable automatic coupling, 0 to disable automatic coupling.

Returns

None

SOURce#:COUPling:AUTO?

Query the automatic coupling state of the specified module.

`SOURce#:COUPling:AUTO?`

Returns

1 if automatic coupling is enabled; else 0

Return type is *Boolean*

SOURce#:COUPling?

Query the output coupling type of the specified module.

`SOURce#:COUPling?`

Returns

Coupling type. AC or DC.

Return type is *Character*

SOURce#:CURRent:LIMit:HIGH

Set the high current output limit. The current output limits are software-defined limits preventing the user from entering an output that could potentially damage the module's load. When the shape is not DC, the limit is applied to the sum of the offset and amplitude. The high current output limit is bounded between -10 V and 10 V, and must be greater than the low current output limit.

`SOURce#:CURRent:LIMit:HIGH <limit>`

Parameters

Name	Type	Description
limit	*NRf*	Current output high limit in A.

Returns

None

SOURce#:CURRent:LIMit:HIGH?

Query the present high current output limit.

`SOURce#:CURRent:LIMit:HIGH?`

Returns

Current output upper limit in A.

Return type is *NRf*

SOURce#:CURRent:LIMit:LOW

Set the low current output limit. The current output limits are software-defined limits preventing the user from entering an output that could potentially damage the module's load. When the shape is not DC, the limit is applied to the sum of the offset and amplitude. The low current output limit is bounded between -100 mA and 100 mA, and must be less than the high current output limit.

`SOURce#:CURRent:LIMit:LOW <limit>`

Parameters

Name	Type	Description
limit	*NRf*	Current output lower limit in A.

Returns

None

SOURce#:CURRent:LIMit:LOW?

Query the present low current output limit.

`SOURce#:CURRent:LIMit:LOW?`

Returns

Current output low limit in A.

Return type is *NRf*

SOURce#:CURRent:MODE

Set the source mode of the specified module.

`SOURce#:CURRent:MODE <mode>`

Parameters

Name	Type	Description
mode	*Character*	desired mode, FIXed or SWEep

Returns

None

SOURce#:CURRent:MODE?

Query the source mode of the specified module.

`SOURce#:CURRent:MODE?`

Returns

The source mode, either FIXed or SWEep.

Return type is *Character*

SOURce#:CURRent:PROTection

Set the current limit of the specified module.

`SOURce#:CURRent:PROTection <currentLimit>`

Parameters

Name	Type	Description
currentLimit	*NRf*	Current limit in A.

Returns

None

SOURce#:CURRent:PROTection:TRIPped?

Query the status of the current limit for the specified module.

`SOURce#:CURRent:PROTection:TRIPped?`

Returns

1 if limit is active, 0 if limit is inactive.

Return type is *Boolean*

SOURce#:CURRent:PROTection?

Query the current limit of the specified module.

`SOURce#:CURRent:PROTection?`

Returns

Current limit in A.

Return type is *NRf*

SOURce#:CURRent:RANGe

Set the current range for the specified module.

`SOURce#:CURRent:RANGe <range>`

Parameters

Name	Type	Description
range	*NRf*	Current range in A.

Returns

None

Remarks

For standard modules, this is the output range regardless of the shape setting. For a module with a hybrid DAC, this defaults to the AC range. If autorange is enabled, then setting the range will disable it.

SOURce#:CURRent:RANGe:AC

Set the current range for the specified module.

`SOURce#:CURRent:RANGe:AC <range>`

Parameters

Name	Type	Description
range	*NRf*	Current range in A.

Returns

None

Remarks

This is only applicable on modules with a hybrid DAC. If autorange is enabled, then setting the range will disable it.

SOURce#:CURRent:RANGe:AC?

Query the present AC current range for the specified module.

`SOURce#:CURRent:RANGe:AC?`

Returns

Current range in A.

Return type is *NRf*

Remarks

This

SOURce#:CURRent:RANGe:AUTO

Set the autorange state for the specified module.

`SOURce#:CURRent:RANGe:AUTO <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable autorange, 0 to disable autorange.

Returns

None

SOURce#:CURRent:RANGe:AUTO?

Query the autorange state for the specified module.

```
SOURce#:CURRent:RANGe:AUTO?
```

Returns

1 if autorange is enabled, 0 if autorange is disabled.

Return type is *Boolean*

SOURce#:CURRent:RANGe:DC

Set the DC current range for the specified module.

```
SOURce#:CURRent:RANGe:DC <range>
```

Parameters

Name	Type	Description
range	*NRf*	Current range in A.

Returns

None

Remarks

This is only applicable on modules with a hybrid DAC. If autorange is enabled, then setting the range will disable it.

SOURce#:CURRent:RANGe:DC?

Query the present DC current range for the specified module.

`SOURce#:CURRent:RANGe:DC?`

Returns

Current range in A.

Return type is *NRf*

Remarks

This is only applicable on modules with a hybrid DAC.

SOURce#:CURRent:RANGe?

Query the present current range for the specified module.

`SOURce#:CURRent:RANGe?`

Returns

Current range in A.

Return type is *NRf*

Remarks

For standard modules, this is the output range regardless of the shape setting. For a module with a hybrid DAC, this defaults to the AC range.

SOURce#:CURRent:STARt

Set the starting current of the source sweep for the specified module.

```
SOURce#:CURRent:STARt <current>
```

Parameters

Name	Type	Description
current	*NRf*	The starting current in A.

Returns

None

SOURce#:CURRent:STARt?

Query the starting current of the source sweep for the specified module.

```
SOURce#:CURRent:STARt?
```

Returns

The starting current in A.

Return type is *NRf*

SOURce#:CURRent:STEP

Set the step size of the source sweep for the specified module.

```
SOURce#:CURRent:STEP <steps>
```

Parameters

Name	Type	Description
steps	*NRf*	The step size in A.

Returns

None

SOURce#:CURRent:STEP?

Query the step size of the source sweep for the specified module.

```
SOURce#:CURRent:STEP?
```

Returns

The step size in A.

Return type is *NRf*

SOURce#:CURRent:STOP

Set the ending current of the source sweep for the specified module.

```
SOURce#:CURRent:STOP <current>
```

Parameters

Name	Type	Description
current	*NRf*	The ending current in A.

Returns

None

SOURce#:CURRent:STOP?

Query the ending current of the source sweep for the specified module.

`SOURce#:CURRent:STOP?`

Returns

The ending current in A.

Return type is *NRf*

SOURce#:CURRent[:LEVel]:OFFSet

Set the excitation offset of the specified module. Ignored if shape is DC (use amplitude for DC level).

If range autoselection is enabled, the instrument will also select a new range if needed. If range autoselection is not enabled, the sum of the offset and amplitude must be less than the range.

`SOURce#:CURRent[:LEVel]:OFFSet <offset>`

Parameters

Name	Type	Description
offset	*NRf*	Excitation offset in A.

Returns

None

SOURce#:CURRent[:LEVel]:OFFSet?

Query the excitation offset of the specified module.

```
SOURce#:CURRent[:LEVel]:OFFSet?
```

Returns

Excitation offset in A.

Return type is *NRf*

SOURce#:CURRent[:LEVel][:AMPLitude]:RMS

Set the excitation amplitude of the specified module.

If range autoselection is enabled, the instrument will also select a new range if needed. If range autoselection is not enabled, the sum of the offset and amplitude must be less than the range.

```
SOURce#:CURRent[:LEVel][:AMPLitude]:RMS <amplitude>
```

Parameters

Name	Type	Description
amplitude	*NRf*	Excitation amplitude in A RMS.

Returns

None

SOURce#:CURRent[:LEVel][:AMPLitude]:RMS?

Query the excitation amplitude of the specified module.

```
SOURce#:CURRent[:LEVel][:AMPLitude]:RMS?
```

Returns

Excitation amplitude in A RMS.

Return type is *NRf*

SOURce#:CURRent[:LEVel][:AMPLitude][:PEAK]

Set the excitation amplitude of the specified module.

When shape is DC, the amplitude directly sets the total excitation level. When shape is not DC, the amplitude sets the peak level of the excitation waveform.

If range autoselection is enabled, the instrument will also select a new range if needed. If range autoselection is not enabled, the sum of the offset and amplitude must be less than the range.

```
SOURce#:CURRent[:LEVel][:AMPLitude][:PEAK] <amplitude>
```

Parameters

Name	Type	Description
amplitude	*NRf*	Excitation amplitude in A.

Returns

None

SOURce#:CURRent[:LEVel][:AMPLitude][:PEAK]?

Query the excitation amplitude of the specified module.

When shape is DC, the amplitude is the total excitation level. When shape is not DC, the amplitude is the peak level of the excitation waveform.

```
SOURce#:CURRent[:LEVel][:AMPLitude][:PEAK]?
```

Returns

Excitation amplitude in A.

Return type is *NRf*

SOURce#:DCYCle

Set the *duty factor* (page 29) of the specified module.

```
SOURce#:DCYCle <dutyCycle>
```

Parameters

Name	Type	Description
dutyCycle	*NRf*	Duty factor between 0.0 and 1.0.

Returns

None

SOURce#:DCYCle?

Query the *duty factor* (page 29) of the specified module.

```
SOURce#:DCYCle?
```

Returns

Duty factor.

Return type is *NRf*

SOURce#:DISPlay:AMPLitude[:TYPE]

Sets the displayed source amplitude type for the specified module.

```
SOURce#:DISPlay:AMPLitude[:TYPE] <amplitudeType>
```

Parameters

Name	Type	Description
amplitudeType	*Character*	PEAK or RMS

Returns

None

SOURce#:DISPlay:AMPLitude[:TYPE]?

Query the source amplitude type for the specified module.

```
SOURce#:DISPlay:AMPLitude[:TYPE]?
```

Returns

PEAK or RMS

Return type is *Character*

SOURce#:DMODe

Set the dark mode for the specified module. If dark mode is enabled, the indicator LEDs on the front of the module will be turned off.

```
SOURce#:DMODe <state>
```

Parameters

Name	Type	Description
state	*Boolean*	1 to enabled dark mode, 0 to disable.

Returns

None

SOURce#:DMODe?

Query the state of the dark mode for the specified module.

```
SOURce#:DMODe?
```

Returns

1 if dark mode is enabled, 0 if not.

Return type is *Boolean*

SOURce#:DOCompliance

Sets the state of the disable on compliance setting for the specified module.

```
SOURce#:DOCompliance <state>
```

Parameters

Name	Type	Description
state	*Boolean*	1 for the module to disable when in compliance; 0 for the module to remain enabled, even in compliance.

Returns

None

SOURce#:DOCompliance?

Gets the state of the disable on compliance setting for the specified module.

```
SOURce#:DOCompliance?
```

Returns

True if the output will be disabled on compliance; else false

Return type is *Boolean*

SOURce#:FREQuency:MODE

Set the frequency mode of the specified module.

```
SOURce#:FREQuency:MODE <mode>
```

Parameters

Name	Type	Description
mode	*Character*	The frequency mode, either FIXed|CW or SWEep.

Returns

None

SOURce#:FREQuency:MODE?

Query the frequency mode of the specified module.

```
SOURce#:FREQuency:MODE?
```

Returns

The frequency mode, either FIXed or SWEep.

Return type is *Character*

SOURce#:FREQuency:STARt

Set the starting frequency of the source sweep for the specified module.

`SOURce#:FREQuency:STARt <frequency>`

Parameters

Name	Type	Description
frequency	*NRf*	The starting frequency in Hz.

Returns

None

SOURce#:FREQuency:STARt?

Query the starting frequency of the source sweep for the specified module.

`SOURce#:FREQuency:STARt?`

Returns

The starting frequency in Hz.

Return type is *NRf*

SOURce#:FREQuency:STEP

Set the step size of the source sweep for the specified module.

`SOURce#:FREQuency:STEP <steps>`

Parameters

Name	Type	Description
steps	*NRf*	The step size in Hz.

Returns

None

SOURce#:FREQuency:STEP?

Query the step size of the source sweep for the specified module.

`SOURce#:FREQuency:STEP?`

Returns

The step size in Hz.

Return type is *NRf*

SOURce#:FREQuency:STOP

Set the ending frequency of the source sweep for the specified module.

`SOURce#:FREQuency:STOP <frequency>`

Parameters

Name	Type	Description
frequency	*NRf*	The ending frequency in Hz.

Returns

None

SOURce#:FREQuency:STOP?

Query the ending frequency of the source sweep for the specified module.

`SOURce#:FREQuency:STOP?`

Returns

The ending frequency in Hz.

Return type is *NRf*

SOURce#:FREQuency[:CW]

Set the excitation frequency of the specified module.

`SOURce#:FREQuency[:CW] <frequency>`

Parameters

Name	Type	Description
frequency	*NRf*	Excitation frequency in Hz. Frequency bounds are module and range dependent, see module specifications.

Returns

None

Remarks

This command is aliased to SOURce#:FREQuency[:FIXed]

SOURce#:FREQuency[:CW]?

Query the excitation frequency of the specified module.

`SOURce#:FREQuency[:CW]?`

Returns

Excitation frequency in Hz.

Return type is *NRf*

Remarks

This command is aliased to SOURce#:FREQuency[:FIXed]?

SOURce#:FREQuency[:FIXed]

Set the excitation frequency of the specified module.

`SOURce#:FREQuency[:FIXed] <frequency>`

Parameters

Name	Type	Description
frequency	*NRf*	Excitation frequency in Hz. Frequency bounds are module and range dependent, see module specifications.

Returns

None

SOURce#:FREQuency[:FIXed]?

Query the excitation frequency of the specified module.

```
SOURce#:FREQuency[:FIXed]?
```

Returns

Excitation frequency in Hz.

Return type is *NRf*

SOURce#:FUNCtion:MODE

Set the excitation mode for the specified module.

```
SOURce#:FUNCtion:MODE <excitationMode>
```

Parameters

Name	Type	Description
excitationMode	*Character*	Excitation mode. CURRent, or VOLTage.

Returns

None

SOURce#:FUNCtion:MODE?

Query the excitation mode for the specified module.

```
SOURce#:FUNCtion:MODE?
```

Returns

Excitation mode. CURRent, or VOLTage.

Return type is *Character*

SOURce#:FUNCtion[:SHAPe]

Set the signal shape for the specified module.

`SOURce#:FUNCtion[:SHAPe] <shape>`

Parameters

Name	Type	Description
shape	*Character*	Signal shape. DC, SINusoid, TRIangle, or SQUAre.

Returns

None

SOURce#:FUNCtion[:SHAPe]?

Query the signal shape for the specified module.

`SOURce#:FUNCtion[:SHAPe]?`

Returns

Signal shape. DC, SINusoid, TRIangle, or SQUAre.

Return type is *Character*

SOURce#:GUARd

Set the guard state of the specified module.

`SOURce#:GUARd <guardState>`

Parameters

Name	Type	Description
guardState	*Boolean*	1 to enable guards, 0 to disable guards.

Returns

None

SOURce#:GUARd?

Query the guard state of the specified module.

`SOURce#:GUARd?`

Returns

1 if guards are enabled, 0 if guards are disabled.

Return type is *Boolean*

SOURce#:HWVersion?

Query the hardware version of the specified module.

`SOURce#:HWVersion?`

Returns

Module hardware version.

Return type is *NR1*

SOURce#:IDENtify

Set the pod identification status for the specified module.

```
SOURce#:IDENtify <state>
```

Parameters

Name	Type	Description
state	*Boolean*	1 to enable identification, 0 to disable identificaiton.

Returns

None

SOURce#:IDENtify?

Query the state of the pod identification for the specified module.

```
SOURce#:IDENtify?
```

Returns

None

SOURce#:LOAD

Loads all connected modules. This command has been replaced by the SYSTem:LOAD command.

```
SOURce#:LOAD
```

Returns

None

Remarks

This command is available for backwards compatibility and is equivalent to the SYSTem:LOAD command.

SOURce#:LOAD?

Gets the loaded state of the specified module.

```
SOURce#:LOAD?
```

Returns

True if the module is loaded; else false

Return type is *Boolean*

SOURce#:MODel?

Query what type of module is connected to the given source slot.

```
SOURce#:MODel?
```

Returns

The type of module connected, or None if nothing is connected.

Return type is *String*

SOURce#:NAME

Set the name of the specified module.

```
SOURce#:NAME <value>
```

Parameters

Name	Type	Description
value	*String*	The name of the module with a maximum of 100 characters.

Returns

None

SOURce#:NAME?

Query the name of the specified module.

```
SOURce#:NAME?
```

Returns

The name.

Return type is *String*

SOURce#:NOTes

Set the notes of the specified module.

```
SOURce#:NOTes <value>
```

Parameters

Name	Type	Description
value	*String*	The notes to be stored, limited to 500 characters.

Returns

None

SOURce#:NOTes?

Query the notes of the specified module.

```
SOURce#:NOTes?
```

Returns

The notes.

Return type is *String*

SOURce#:OFFSet:MODE

Set the source mode of the specified module.

```
SOURce#:OFFSet:MODE <mode>
```

Parameters

Name	Type	Description
mode	*Character*	The desired mode, FIXed of SWEep

Returns

None

SOURce#:OFFSet:MODE?

Query the source mode of the specified module.

```
SOURce#:OFFSet:MODE?
```

Returns

The source mode, either FIXed or SWEep.

Return type is *Character*

SOURce#:OFFSet:STARt

Set the starting offset of the source sweep for the specified module.

`SOURce#:OFFSet:STARt <current>`

Parameters

Name	Type	Description
current	*NRf*	The starting offset in base units.

Returns

None

SOURce#:OFFSet:STARt?

Query the starting offset of the source sweep for the specified module.

`SOURce#:OFFSet:STARt?`

Returns

The starting offset in base units.

Return type is *NRf*

SOURce#:OFFSet:STEP

Set the step size of the source sweep for the specified module.

```
SOURce#:OFFSet:STEP <steps>
```

Parameters

Name	Type	Description
steps	*NRf*	The step size in base units.

Returns

None

SOURce#:OFFSet:STEP?

Query the step size of the source sweep for the specified module.

```
SOURce#:OFFSet:STEP?
```

Returns

The step size in base units.

Return type is *NRf*

SOURce#:OFFSet:STOP

Set the ending offset of the source sweep for the specified module.

```
SOURce#:OFFSet:STOP <current>
```

Parameters

Name	Type	Description
current	*NRf*	The ending offset in base units.

Returns

None

SOURce#:OFFSet:STOP?

Query the ending offset of the source sweep for the specified module.

```
SOURce#:OFFSet:STOP?
```

Returns

The ending offset in base units.

Return type is *NRf*

SOURce#:PRESet

Resets the module settings to their power on defaults.

```
SOURce#:PRESet
```

Returns

None

SOURce#:READback:NPLCycles

Set the averaging time in number of power line cycles (NPLC) of the specified module.

```
SOURce#:READback:NPLCycles <value>
```

Parameters

Name	Type	Description
value	*NRf*	Averaging time in number of power line cycles (NPLC) between 0.01 and 600.00.

Returns

None

SOURce#:READback:NPLCycles?

Query the averaging time in number of power line cycles (NPLC) of the specified module.

`SOURce#:READback:NPLCycles?`

Returns

Averaging time in number of power line cycles (NPLC).

Return type is *NRf*

SOURce#:SCALibration:DATE?

Queries the specified module's self-calibration date.

`SOURce#:SCALibration:DATE?`

Returns

Date in the form of year, month, day, hour, minute, second

Return type is *Character*

SOURce#:SCALibration:RESet

Reset the self-calibration status for the specified module.

`SOURce#:SCALibration:RESet`

Returns

None

SOURce#:SCALibration:RUN

Initiate the self-calibration process on the specified module.

`SOURce#:SCALibration:RUN`

Returns

None

SOURce#:SCALibration:STATus?

Query the status of the self-calibration of the specified module.

`SOURce#:SCALibration:STATus?`

Returns

The self-calibration state.

Return type is *String*

SOURce#:SCALibration:TEMPerature?

Queries the specified module's self-calibration temperature.

`SOURce#:SCALibration:TEMPerature?`

Returns

Temperature in °C

Return type is *Character*

SOURce#:SERial?

Query the serial number of the specified module.

`SOURce#:SERial?`

Returns

Module serial number.

Return type is *String*

SOURce#:STATe

Set the output state of the specified module.

`SOURce#:STATe <outputState>`

Parameters

Name	Type	Description
outputState	*Boolean*	1 to enable output, 0 to disable output.

Returns

None

SOURce#:STATe?

Query the output state of the specified module.

```
SOURce#:STATe?
```

Returns

1 if output is enabled, 0 if output is disabled.

Return type is *Boolean*

SOURce#:SWEep:DIRection

Set the direction of the source sweep for the specified module. UP will sweep from start to stop; DOWN will sweep from stop to start.

```
SOURce#:SWEep:DIRection <direction>
```

Parameters

Name	Type	Description
direction	*Character*	The direction of the sweep, UP or DOWN.

Returns

None

SOURce#:SWEep:DIRection:RTRip[:STATe]

Set the state of the round trip of the source sweep for the specified module.

```
SOURce#:SWEep:DIRection:RTRip[:STATe] <state>
```

Parameters

Name	Type	Description
state	*Boolean*	True if the sweep returns to the first point; else false.

Returns

None

SOURce#:SWEep:DIRection:RTRip[:STATe]?

Query the round trip state of the source sweep for the specified module.

`SOURce#:SWEep:DIRection:RTRip[:STATe]?`

Returns

True if the sweep should return to the first point; else false.

Return type is *Boolean*

SOURce#:SWEep:DIRection?

Query the direction of the source sweep for the specified module.

`SOURce#:SWEep:DIRection?`

Returns

UP when sweeping start to stop; DOWN when sweeping stop to start.

Return type is *Character*

SOURce#:SWEep:DWELl

Set the dwell time of the source sweep for the specified module.

```
SOURce#:SWEep:DWELl <dwell>
```

Parameters

Name	Type	Description
dwell	*NRf*	The dwell time in s.

Returns

None

SOURce#:SWEep:DWELl?

Query the dwell time of the source sweep for the specified module.

```
SOURce#:SWEep:DWELl?
```

Returns

The dwell time in s.

Return type is *NRf*

SOURce#:SWEep:HOLDoff

Set the holdoff time of the source sweep for the specified module.

```
SOURce#:SWEep:HOLDoff <holdoff>
```

Parameters

Name	Type	Description
holdoff	*NRf*	The holdoff time in s.

Returns

None

SOURce#:SWEep:HOLDoff?

Query the holdoff time of the source sweep for the specified module.

```
SOURce#:SWEep:HOLDoff?
```

Returns

The holdoff time in s.

Return type is *NRf*

SOURce#:SWEep:POINts

Set the number of points to sweep for the specified module.

```
SOURce#:SWEep:POINts <points>
```

Parameters

Name	Type	Description
points	*NR1*	The number of points.

Returns

None

SOURce#:SWEep:POINts?

Query the number of points ot sweep for the specified module.

```
SOURce#:SWEep:POINts?
```

Returns

The number of points.

Return type is *NR1*

SOURce#:SWEep:SPACing

Set the spacing type of the sweep for the specified module.

```
SOURce#:SWEep:SPACing <spacing>
```

Parameters

Name	Type	Description
spacing	*Character*	The spacing type of the sweep, LINear or LOGarithmic.

Returns

None

SOURce#:SWEep:SPACing?

Query the spacing type of the sweep on the specified module.

```
SOURce#:SWEep:SPACing?
```

Returns

The spacing type of the sweep, LINear or LOGarithmic.

Return type is *Character*

SOURce#:SWEep:TIME

Set the overall time that the source sweep should take for the specified module.

```
SOURce#:SWEep:TIME <time>
```

Parameters

Name	Type	Description
time	*NRf*	The time in s.

Returns

None

SOURce#:SWEep:TIME?

Query the overall time that the source sweep should take for the specified module.

```
SOURce#:SWEep:TIME?
```

Returns

None

SOURce#:SWEep[:STATus]?

Query the status of the sweep of the specified module.

```
SOURce#:SWEep[:STATus]?
```

Returns

1 if sweep is in progress, 0 if sweep is not in progress.

Return type is *Boolean*

SOURce#:SYNChronize:PHASe

Set the phase shift applied between this module and the synchronization source. See *source channel synchronization* (page 41).

```
SOURce#:SYNChronize:PHASe <phase>
```

Parameters

Name	Type	Description
phase	*NRf*	Phase shift in degrees between -360.0 and +360.0.

Returns

None

SOURce#:SYNChronize:PHASe?

Query the phase shift applied between this module and the synchronization source. See *source channel synchronization* (page 41).

```
SOURce#:SYNChronize:PHASe?
```

Returns

Phase shift in degrees.

Return type is *NRf*

SOURce#:SYNChronize:SOURce

Set the frequency source to which this module is synchronized. See *source channel synchronization* (page 41).

`SOURce#:SYNChronize:SOURce <source>`

Parameters

Name	Type	Description
source	*Character*	Frequency source. S1, S2, S3, or RIN for M81-6. S1, S2, or RIN for M81-4. S1 or RIN for M81-2.

Returns

None

SOURce#:SYNChronize:SOURce?

Query the frequency source to which this module is synchronized. See *source channel synchronization* (page 41).

`SOURce#:SYNChronize:SOURce?`

Returns

Frequency source. S1, S2, S3, or RIN for M81-6. S1, S2, or RIN for M81-4. S1 or RIN for M81-2.

Return type is *Character*

SOURce#:SYNChronize[:STATe]

Set the state of *source channel synchronization* (page 41) for the specified module.

`SOURce#:SYNChronize[:STATe] <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable sync, 0 to disable sync.

Returns

None

SOURce#:SYNChronize[:STATe]?

Query the state of *source channel synchronization* (page 41) for the specified module.

```
SOURce#:SYNChronize[:STATe]?
```

Returns

1 if sync is enabled, 0 if disabled.

Return type is *Boolean*

SOURce#:UNLoad

Unloads the specified module. A module must be unloaded before it is physically disconnected.

```
SOURce#:UNLoad
```

Returns

None

SOURce#:VOLTage:LIMit:HIGH

Set the high voltage output limit. The voltage output limits are software-defined limits preventing the user from entering an output that could potentially damage the module's load. When the shape is not DC, the limit is applied to the sum of the offset and amplitude. The high voltage output limit is bounded between -10 V and 10 V, and must be greater than the low voltage output limit.

```
SOURce#:VOLTage:LIMit:HIGH <limit>
```

Parameters

Name	Type	Description
limit	*NRf*	Voltage output high limit in V.

Returns

None

SOURce#:VOLTage:LIMit:HIGH?

Query the present high voltage output limit.

```
SOURce#:VOLTage:LIMit:HIGH?
```

Returns

Voltage output upper limit in V.

Return type is *NRf*

SOURce#:VOLTage:LIMit:LOW

Set the low voltage output limit. The voltage output limits are software-defined limits preventing the user from entering an output that could potentially damage the module's load. When the shape is not DC, the limit is applied to the sum of the offset and amplitude. The low voltage output limit is bounded between -10 V and 10 V, and must be less than the high voltage output limit.

```
SOURce#:VOLTage:LIMit:LOW <limit>
```

Parameters

Name	Type	Description
limit	*NRf*	Voltage output low limit in V.

Returns

None

SOURce#:VOLTage:LIMit:LOW?

Query the present low voltage output limit

```
SOURce#:VOLTage:LIMit:LOW?
```

Returns

Voltage output lower limit in V.

Return type is *NRf*

SOURce#:VOLTage:MODE

Set the source mode of the specified module.

```
SOURce#:VOLTage:MODE <mode>
```

Parameters

Name	Type	Description
mode	*Character*	desired mode, FIXed or SWEep

Returns

None

SOURce#:VOLTage:MODE?

Query the source mode of the specified module.

```
SOURce#:VOLTage:MODE?
```

Returns

The source mode, either FIXed or SWEep.

Return type is *Character*

SOURce#:VOLTage:PROTection

Set the compliance voltage limit of the specified module. The high compliance limit is set to absolute value of the specified voltage and the low compliance limit is set to the negated absolute value.

```
SOURce#:VOLTage:PROTection <complianceVoltage>
```

Parameters

Name	Type	Description
complianceVoltage	*NRf*	Compliance voltage limit in V.

Returns

None

SOURce#:VOLTage:PROTection:HIGH

Set the high compliance voltage limit of the specified module.

`SOURce#:VOLTage:PROTection:HIGH <complianceVoltage>`

Parameters

Name	Type	Description
complianceVoltage	*NRf*	Compliance voltage limit in V.

Returns

None

SOURce#:VOLTage:PROTection:LOW

Set the low compliance voltage limit of the specified module.

`SOURce#:VOLTage:PROTection:LOW <complianceVoltage>`

Parameters

Name	Type	Description
complianceVoltage	*NRf*	Compliance voltage limit in V.

Returns

None

SOURce#:VOLTage:PROTection:LOW?

Query the low compliance voltage limit of the specified module.

`SOURce#:VOLTage:PROTection:LOW?`

Returns

Compliance voltage limit in V.

Return type is *NRf*

SOURce#:VOLTage:PROTection:TRIPped?

Query the status of the compliance voltage limit for the specified module.

```
SOURce#:VOLTage:PROTection:TRIPped?
```

Returns

1 if limit is active, 0 if limit is inactive.

Return type is *Boolean*

SOURce#:VOLTage:PROTection[:HIGH]?

Query the high compliance voltage limit of the specified module.

```
SOURce#:VOLTage:PROTection[:HIGH]?
```

Returns

Compliance voltage limit in V.

Return type is *NRf*

SOURce#:VOLTage:RANGe

Set the voltage range for the specified module.

```
SOURce#:VOLTage:RANGe <range>
```

Parameters

Name	Type	Description
range	*NRf*	Voltage range in V.

Returns

None

Remarks

For standard modules, this is the output range regardless of the shape setting. For a module with a hybrid DAC, this defaults to the AC range. If autorange is enabled, then setting the range will disable it.

SOURce#:VOLTage:RANGe:AC

Set the AC voltage range for the specified module.

```
SOURce#:VOLTage:RANGe:AC <range>
```

Parameters

Name	Type	Description
range	*NRf*	Voltage range in V.

Returns

None

Remarks

This is only applicable on modules with a hybrid DAC. If autorange is enabled, then setting the range will disable it.

SOURce#:VOLTage:RANGe:AC?

Query the present AC voltage range for the specified module.

`SOURce#:VOLTage:RANGe:AC?`

Returns

Voltage range in V.

Return type is *NRf*

Remarks

This is only applicable on modules with a hybrid DAC.

SOURce#:VOLTage:RANGe:AUTO

Set the autorange status for the specified module.

`SOURce#:VOLTage:RANGe:AUTO <state>`

Parameters

Name	Type	Description
state	*Boolean*	1 to enable autorange, 0 to disable autorange.

Returns

None

SOURce#:VOLTage:RANGe:AUTO?

Query the autorange status for the specified module.

`SOURce#:VOLTage:RANGe:AUTO?`

Returns

1 if autorange is enabled, 0 if autorange is disabled.

Return type is *Boolean*

SOURce#:VOLTage:RANGe:DC

Set the DC current range for the specified module.

`SOURce#:VOLTage:RANGe:DC <range>`

Parameters

Name	Type	Description
range	*NRf*	Voltage range in V.

Returns

None

Remarks

This is only applicable on modules with a hybrid DAC. If autorange is enabled, then setting the range will disable it.

SOURce#:VOLTage:RANGe:DC?

Query the present DC current range for the specified module.

`SOURce#:VOLTage:RANGe:DC?`

Returns

Voltage range in V.

Return type is *NRf*

Remarks

This is only applicable on modules with a hybrid DAC.

SOURce#:VOLTage:RANGe?

Query the present voltage range for the specified module.

```
SOURce#:VOLTage:RANGe?
```

Returns

Voltage range in V.

Return type is *NRf*

Remarks

For standard modules, this is the output range regardless of the shape setting. For a module with a hybrid DAC, this defaults to the AC range.

SOURce#:VOLTage:SMODe

Set the source voltage sense mode for the specified module.

```
SOURce#:VOLTage:SMODe <state>
```

Parameters

Name	Type	Description
state	*Character*	LOCal or REMote.

Returns

None

SOURce#:VOLTage:SMODe?

Query the source voltage sense mode for the specified module.

```
SOURce#:VOLTage:SMODe?
```

Returns

LOCal or REMote.

Return type is *Character*

SOURce#:VOLTage:STARt

Set the starting voltage of the source sweep for the specified module.

```
SOURce#:VOLTage:STARt <voltage>
```

Parameters

Name	Type	Description
voltage	*NRf*	The starting voltage in V.

Returns

None

SOURce#:VOLTage:STARt?

Query the starting voltage of the source sweep for the specified module.

```
SOURce#:VOLTage:STARt?
```

Returns

The starting voltage in V.

Return type is *NRf*

SOURce#:VOLTage:STEP

Set the step size of the source sweep for the specified module.

`SOURce#:VOLTage:STEP <steps>`

Parameters

Name	Type	Description
steps	*NRf*	The step size in V.

Returns

None

SOURce#:VOLTage:STEP?

Query the step size of the source sweep for the specified module.

`SOURce#:VOLTage:STEP?`

Returns

The step size in V.

Return type is *NRf*

SOURce#:VOLTage:STOP

Set the ending voltage of the source sweep for the specified module.

```
SOURce#:VOLTage:STOP <voltage>
```

Parameters

Name	Type	Description
voltage	*NRf*	The ending voltage in V.

Returns

None

SOURce#:VOLTage:STOP?

Query the ending voltage of the source sweep for the specified module.

```
SOURce#:VOLTage:STOP?
```

Returns

The ending voltage in V.

Return type is *NRf*

SOURce#:VOLTage[:LEVel]:OFFSet

Set the excitation offset of the specified module.

If range autoselection is enabled, the instrument will also select a new range if needed. If range autoselection is not enabled, the sum of the offset and amplitude must be less than the range.

```
SOURce#:VOLTage[:LEVel]:OFFSet <offset>
```

Parameters

Name	Type	Description
offset	*NRf*	Excitation offset in V.

Returns

None

SOURce#:VOLTage[:LEVel]:OFFSet?

Query the excitation offset of the specified module.

```
SOURce#:VOLTage[:LEVel]:OFFSet?
```

Returns

Excitation offset in V.

Return type is *NRf*

SOURce#:VOLTage[:LEVel][:AMPLitude]:RMS

Set the excitation amplitude of the specified module.

If range autoselection is enabled, the instrument will also select a new range if needed. If range autoselection is not enabled, the sum of the offset and amplitude must be less than the range.

```
SOURce#:VOLTage[:LEVel][:AMPLitude]:RMS <amplitude>
```

Parameters

Name	Type	Description
amplitude	*NRf*	Excitation amplitude in V RMS.

Returns

None

SOURce#:VOLTage[:LEVel][:AMPLitude]:RMS?

Query the excitation amplitude of the specified module.

```
SOURce#:VOLTage[:LEVel][:AMPLitude]:RMS?
```

Returns

Excitation amplitude in V RMS.

Return type is *NRf*

SOURce#:VOLTage[:LEVel][:AMPLitude][:PEAK]

Set the excitation amplitude of the specified module.

When shape is DC, the amplitude directly sets the total excitation level. When shape is not DC, the amplitude sets the peak level of the excitation waveform

If range autoselection is enabled, the instrument will also select a new range if needed. If range autoselection is not enabled, the sum of the offset and amplitude must be less than the range.

```
SOURce#:VOLTage[:LEVel][:AMPLitude][:PEAK] <amplitude>
```

Parameters

Name	Type	Description
amplitude	*NRf*	Excitation amplitude in V.

Returns

None

SOURce#:VOLTage[:LEVel][:AMPLitude][:PEAK]?

Query the excitation amplitude of the specified module.

When shape is DC, the amplitude is the total excitation level. When shape is not DC, the amplitude is the peak level of the excitation waveform.

```
SOURce#:VOLTage[:LEVel][:AMPLitude][:PEAK]?
```

Returns

Excitation amplitude in V.

Return type is *NRf*

STATistic:MAXimum:RESet

Resets the maximum statistic for all modules.

```
STATistic:MAXimum:RESet
```

Returns

None

STATistic:MAXimum:SENSe#:DC?

Returns the maximum DC measurement for the specified module. The module must be in DC or AC mode.

```
STATistic:MAXimum:SENSe#:DC?
```

Returns

Maximum DC measurement in module units.

Return type is *NRf*

STATistic:MAXimum:SENSe#:LIA:R?

Returns the maximum R measurement from the lock-in for the specified module. The module must be in LIA mode.

```
STATistic:MAXimum:SENSe#:LIA:R?
```

Returns

Maximum R measurement in module units.

Return type is *NRf*

STATistic:MAXimum:SENSe#:LIA:THETa?

Returns the maximum theta measurement from the lock-in for the specified module. The module must be in LIA mode.

```
STATistic:MAXimum:SENSe#:LIA:THETa?
```

Returns

Maximum theta measurement in degrees.

Return type is *NRf*

STATistic:MAXimum:SENSe#:LIA:X?

Returns the maximum X measurement from the lock-in for the specified module. The module must be in LIA mode.

```
STATistic:MAXimum:SENSe#:LIA:X?
```

Returns

Maximum X measurement in module units.

Return type is *NRf*

STATistic:MAXimum:SENSe#:LIA:Y?

Returns the maximum Y measurement from the lock-in for the specified module. The module must be in LIA mode.

```
STATistic:MAXimum:SENSe#:LIA:Y?
```

Returns

Maximum Y measurement in module units.

Return type is *NRf*

STATistic:MAXimum:SENSe#:NPEak?

Returns the maximum *negative peak* (page 34) measurement for the specified module.

```
STATistic:MAXimum:SENSe#:NPEak?
```

Returns

Maximum negative peak measurement in module units.

Return type is *NRf*

STATistic:MAXimum:SENSe#:PPEak?

Returns the maximum *positive peak* (page 34) measurement for the specified module. The module must be in AC mode.

```
STATistic:MAXimum:SENSe#:PPEak?
```

Returns

Maximum positive peak measurement in module units.

Return type is *NRf*

STATistic:MAXimum:SENSe#:PTPeak?

Returns the maximum *peak to peak* (page 34) measurement for the specified module.

```
STATistic:MAXimum:SENSe#:PTPeak?
```

Returns

Maximum peak to peak measurement in module units.

Return type is *NRf*

STATistic:MAXimum:SENSe#:RESet

Resets the maximum statistics for the specified sense module.

```
STATistic:MAXimum:SENSe#:RESet
```

Returns

None

STATistic:MAXimum:SENSe#:RMS?

Returns the maximum RMS measurement for the specified module. The module must be in DC or AC mode.

```
STATistic:MAXimum:SENSe#:RMS?
```

Returns

Maximum RMS measurement in module units.

Return type is *NRf*

STATistic:MINimum:RESet

Resets the minimum statistic for all modules.

```
STATistic:MINimum:RESet
```

Returns

None

STATistic:MINimum:SENSe#:DC?

Returns the minimum DC measurement for the specified module. The module must be in DC or AC mode.

```
STATistic:MINimum:SENSe#:DC?
```

Returns

Minimum DC measurement in module units.

Return type is *NRf*

STATistic:MINimum:SENSe#:LIA:R?

Returns the minimum R measurement from the lock-in for the specified module. The module must be in LIA mode.

```
STATistic:MINimum:SENSe#:LIA:R?
```

Returns

Minimum R measurement in module units.

Return type is *NRf*

STATistic:MINimum:SENSe#:LIA:THETa?

Returns the minimum theta measurement from the lock-in for the specified module. The module must be in LIA mode.

```
STATistic:MINimum:SENSe#:LIA:THETa?
```

Returns

Minimum theta measurement in degrees.

Return type is *NRf*

STATistic:MINimum:SENSe#:LIA:X?

Returns the minimum X measurement from the lock-in for the specified module. The module must be in LIA mode.

```
STATistic:MINimum:SENSe#:LIA:X?
```

Returns

Minimum X measurement in module units.

Return type is *NRf*

STATistic:MINimum:SENSe#:LIA:Y?

Returns the minimum Y measurement from the lock-in for the specified module. The module must be in LIA mode.

```
STATistic:MINimum:SENSe#:LIA:Y?
```

Returns

Minimum Y measurement in module units.

Return type is *NRf*

STATistic:MINimum:SENSe#:NPEak?

Returns the minimum *negative peak* (page 34) measurement for the specified module.

```
STATistic:MINimum:SENSe#:NPEak?
```

Returns

Minimum negative peak measurement in module units.

Return type is *NRf*

STATistic:MINimum:SENSe#:PPEak?

Returns the minimum *positive peak* (page 34) measurement for the specified module.

```
STATistic:MINimum:SENSe#:PPEak?
```

Returns

Minimum positive peak measurement in module units.

Return type is *NRf*

STATistic:MINimum:SENSe#:PTPeak?

Returns the minimum *peak to peak* (page 34) measurement for the specified module.

```
STATistic:MINimum:SENSe#:PTPeak?
```

Returns

Minimum peak to peak measurement in module units.

Return type is *NRf*

STATistic:MINimum:SENSe#:RESet

Resets the minimum statistics for the specified sense module.

```
STATistic:MINimum:SENSe#:RESet
```

Returns

None

STATistic:MINimum:SENSe#:RMS?

Returns the minimum RMS measurement for the specified module. The module must be in DC or AC mode.

```
STATistic:MINimum:SENSe#:RMS?
```

Returns

Minimum RMS measurement in module units.

Return type is *NRf*

STATistic:MMAXimum[:MULTiple]?

Query the minimum and maximum values for multiple statistics sampled at the same moment in time.

```
STATistic:MMAXimum[:MULTiple]?
```

Returns

Comma separated list of values corresponding to the input parameters and channels.

Return type is *Character*

Examples

```
To query DC from measure module 1, RMS from measure module 2, and Peak-to-
↪Peak from measure module 3:
  FETCh? MDC,1,MRMS,2,MPTPeak,3
Example result:
  1.23, 1.34, 2.45, 2.56, 3.67, 3.78
```

STATistic:SENSe#:RESet

Resets all statistics for the specified sense module.

`STATistic:SENSe#:RESet`

Returns

None

SWEep:ABORt

Aborts all running sweeps leaving the output enabled and at the last sweep point applied.

`SWEep:ABORt`

Returns

None

SWEep:INITiate

Initiates all configured sweeps.

```
SWEep:INITiate
```

Returns

None

SWEep:INITiate:TRACe[:STATe]

Set if sweeps should be initiated when streaming is started via the TRACe:STARt command.

```
SWEep:INITiate:TRACe[:STATe] <traceInitiate>
```

Parameters

Name	Type	Description
traceInitiate	*Boolean*	1 to initiate sweeps when data streaming is started; 0 to not initiate sweeps when data streaming is started

Returns

None

SWEep:INITiate:TRACe[:STATe]?

Query if sweeps are configured to be initiated when streaming is started via the TRACe:STARt command.

```
SWEep:INITiate:TRACe[:STATe]?
```

Returns

1 if sweeps will be initiated when streaming is started; 0 if sweeps will not be stared when streaming is started

Return type is *Boolean*

SYSTem:LFRequency:DETected?

Gets the line frequency in Hz detected on startup.

`SYSTem:LFRequency:DETected?`

Returns

The line frequency detected.

Return type is *NRf*

SYSTem:LFRequency:ERRor?

Gets a value indicating whether the detected line frequency in Hz is out of normal operating bounds bounds.

`SYSTem:LFRequency:ERRor?`

Returns

True if the detected line frequency is out of bounds, otherwise false.

Return type is *Boolean*

SYSTem:LFRequency?

Gets the applied line frequency in Hz.

`SYSTem:LFRequency?`

Returns

The applied line frequency.

Return type is *NRf*

SYSTem:LOAD

Loads all connected modules.

`SYSTem:LOAD`

Returns

None

TRACe:ACTive?

Query the active status of the data stream.

`TRACe:ACTive?`

Returns

1 when the stream is started and 0 when the stream is reset or the last point is read from the stream buffer.

Return type is *Boolean*

TRACe:DATA:ALL?

Query all available in the data stream: For CSV encoding, comma separated ASCII values with each row separated by a semicolon. For B64 encoding raw binary data. Little endian, no padding.

`TRACe:DATA:ALL?`

Returns

None

Examples

```
CSV Encoding. False,-3.23649412656182E-05;False,-3.23649412656182E-05;
↪False,-3.23649412656182E-05;
B64 Encoding. ALpzhMLgofm+ALpzhMLgofm+ALpzhMLgofm+
```

TRACe:DATA:COUNt?

Query the number of unread rows in the data stream buffer.

`TRACe:DATA:COUNt?`

Returns

Number of unread rows in the data stream buffer.

Return type is *NR1*

TRACe:DATA:OVERflow?

Query the data stream overflow status.

`TRACe:DATA:OVERflow?`

Returns

1 if data stream has overflowed, 0 if data stream has not overflowed.

Return type is *Boolean*

TRACe:DATA[:SINGle]?

Query the oldest available row of data stream data.

`TRACe:DATA[:SINGle]?`

Returns

None

TRACe:FORMat:ELEMents

Set the *elements* (page 116) to include in the data stream. Takes a list of pairs of data source mnemonic and channel number. Up to 10 pairs.

`TRACe:FORMat:ELEMents`

Returns

None

Examples

```
TRACe:FORMat:ELEMents SAMPLITUDE,1,MX,2,MOVERLOAD,2
```

TRACe:FORMat:ELEMents?

Query the elements currently in the data stream.

`TRACe:FORMat:ELEMents?`

Returns

None

Examples

```
MRANge,1,MDC,1
```

TRACe:FORMat:ENCOding

Set the data stream encoding.

`TRACe:FORMat:ENCOding <encoding>`

Parameters

Name	Type	Description
encoding	*Character*	Encoding. CSV or B64.

Returns

None

TRACe:FORMat:ENCOding:B64:BCOunt?

Query the total number of bytes in a row for the present B64 encoded elements.

`TRACe:FORMat:ENCOding:B64:BCOunt?`

Returns

The total number of bytes.

Return type is *NR1*

TRACe:FORMat:ENCOding:B64:BFORmat?

Query the byte format.

`TRACe:FORMat:ENCOding:B64:BFORmat?`

Returns

Byte format. d, ?, or B.

Return type is *String*

Examples

```
If TRACe:FORMat:ELEMents is set to:
  ('MRANge',1),('MX',1),('MOVerload',1)
TRACe:FORMat:ENCOding:B64:BFORmat? will return:
  "fd?"
```

TRACe:FORMat:ENCOding?

Query the data stream encoding.

`TRACe:FORMat:ENCOding?`

Returns

Data stream encoding.

Return type is *Character*

TRACe:FORMat:HEADer?

Query the data stream header.

`TRACe:FORMat:HEADer?`

Returns

Comma separated header for present data stream elements.

Return type is *String*

TRACe:RATE

Set the data stream rate.

```
TRACe:RATE <rate>
```

Parameters

Name	Type	Description
rate	*NRf*	Data stream rate in samples/s. The maximum stream rate is 5000 samples/s.

Returns

None

TRACe:RATE?

Query the data stream rate.

```
TRACe:RATE?
```

Returns

Data stream rate in samples/second.

Return type is *NRf*

TRACe:RESet

Reset the data stream.

```
TRACe:RESet
```

Returns

None

TRACe:STARt

Start streaming data.

```
TRACe:STARt <count>
```

Parameters

Name	Type	Description
count	*Decimal Numeric*	Number of data points to stream. If param is blank or INFINITY then stream will continue indefinitely.

Returns

None

DIGital:IDATa?

Returns the integer representation of the digital input states (0 = 0b00, 1 = 0b01, 2 = 0b10, 3 = 0b11).

```
DIGital:IDATa?
```

DIGital:OSETting

Sets the integer representation of the digital output states (0 = 0b00, 1 = 0b01, 2 = 0b10, 3 = 0b11).

```
DIGital:OSETting
```

DIGital:OSETting?

Returns the integer representation of the digital output states (0 = 0b00, 1 = 0b01, 2 = 0b10, 3 = 0b11).

`DIGital:OSETting?`

DISPlay:BRIGhtness

This command controls the intensity of the display on the device. 0 (OFF), 25, 50, 75, or 100.

`DISPlay:BRIGhtness`

DISPlay:BRIGhtness?

Returns the present intensity of the display on the device. 0 (OFF), 25, 50, 75, or 100.

`DISPlay:BRIGhtness?`

DISPlay:ENABle

This command controls whether the display is turned on or off. 0 = OFF, 1 = ON.

`DISPlay:ENABle`

DISPlay:ENABle?

Returns the present state of the display. 0 = OFF, 1 = ON.

`DISPlay:ENABle?`

SYSTem:AUTODATETIME

Sets whether the instrument automatically sets the date and time when connected to the internet. 0 = OFF, 1 = ON.

`SYSTem:AUTODATETIME`

SYSTem:AUTODATETIME?

Returns whether the instrument automatically sets the date and time when connected to the internet.

`SYSTem:AUTODATETIME?`

SYSTem:BEEPer:VOLume

Sets the volume of the instrument speaker as a percentage of maximum (for example 75).

`SYSTem:BEEPer:VOLume`

SYSTem:BEEPer:VOLume?

Returns the volume setting as a percentage of maximum.

`SYSTem:BEEPer:VOLume?`

SYSTem:DATE

Returns the present date (`<year>,<month>,<day>`).

`SYSTem:DATE`

SYSTem:DATE?

Sets the date (`<year>,<month>,<day>`).

`SYSTem:DATE?`

SYSTem:ERRor:ALL?

Returns all SCPI errors in the queue and clears the queue.

`SYSTem:ERRor:ALL?`

SYSTem:ERRor:CLEar

Clears the SCPI error queue.

```
SYSTem:ERRor:CLEar
```

SYSTem:ERRor:COUNt?

Returns the number of errors in the SCPI error queue.

```
SYSTem:ERRor:COUNt?
```

SYSTem:ERRor[:NEXT]?

Returns the most recent error in the queue and removes it.

```
SYSTem:ERRor[:NEXT]?
```

SYSTem:FACTORYRESET

Resets all settings to their factory defaults. This includes System settings.

```
SYSTem:FACTORYRESET <restart>
```

Parameters

Name	Type	Description
restart	*bool*	1 to restart after reset, 0 to not restart. Defaults to 1 if omitted.

SYSTem:KLOCk

Locks and unlocks the front panel touch screen. 0 = OFF, 1 = ON.

```
SYSTem:KLOCk
```

SYSTem:KLOCk?

Returns whether the front panel touch screen is locked. 0 = OFF, 1 = ON.

`SYSTem:KLOCk?`

SYSTem:PRESet

Resets the measurement settings to their default values.

`SYSTem:PRESet`

SYSTem:TIME

Returns the present time in 24 hour format (`<hour>,<minute>,<second>`).

`SYSTem:TIME`

SYSTem:TIME?

Sets the time in 24 hour format (`<hour>,<minute>,<second>`).

`SYSTem:TIME?`

SYSTem:TZONe

Sets the timezone offset hour.

`SYSTem:TZONe`

SYSTem:TZONe?

Returns the timezone offset hour.

`SYSTem:TZONe?`

SYSTem:VERSion?

Returns the SCPI definition version that the instrument complies with.

```
SYSTem:VERSion?
```

CHAPTER

SEVEN

SERVICE

This chapter provides basic service information for the MeasureReady™ M81-SSM Synchronous Source Measure System. Customer service of the product is limited to the information presented in this chapter. Lake Shore service personnel should be consulted if the instrument requires repair.

Note: The purpose of this chapter is to help determine if there is a simple solution to your problem, or if something is out of working order. If you verify that there is something wrong with your instrument, do not attempt to open the instrument as there are no serviceable parts inside; contact Lake Shore for service.

7.1 Troubleshooting

7.1.1 General Troubleshooting

If the MeasureReady™ M81-SSM is not operating properly, try the following solutions.

1. Refer to the *list of error messages* (page 306)
2. Cycle the power
3. Reset the instrument to factory defaults
4. To determine whether the issue is with the module or the instrument, test the instrument with a different module, or try disconnecting and reconnecting the module
5. If none of these solutions work, contact Lake Shore Service

7.1.2 General Remote Interface Troubleshooting

Timeouts during range changes

The M81-SSM will not return a measurement while settling between range changes. If a module is configured with a long averaging time (large number of NPLCs) and/or is transitioning through multiple ranges, it may take multiple or even tens of seconds before a measurement is returned. Either increase the remote interface timeout setting or poll the *settling status bit* (page 122) to avoid querying during settling.

7.1.3 USB Troubleshooting

This section provides USB interface troubleshooting for issues that arise with new installations, existing installations, and intermittent lockups.

New Installation

1. Check that the instrument’s interface is set to USB.
2. Check that the USB driver is installed properly and that the device is functioning. In Microsoft Windows®, the device status can be checked using Device Manager by right-clicking Lake Shore Instrument under Ports (COM & LPT) or Other Devices and then clicking Properties.
3. Check that the correct com port is being used. In Microsoft Windows®, the com port number can be checked using Device Manager under Ports (COM & LPT).
4. Check that the correct settings are being used for communication.
5. Check cable connections and length.
6. Send the message terminator.
7. Send the entire message string at one time including the terminator. (Many terminal emulation programs do not.)
8. Send only one simple command at a time until communication is established.
9. Be sure to spell commands correctly and use proper syntax.

Existing Installation No Longer Working

1. Power the instrument off, then on again to see if it is a soft failure.
2. Power the computer off, then on again to see if communication port is locked up.
3. Check all cable connections.
4. Check that the com port assignment has not been changed. In Microsoft Windows®, the com port number can be checked in the Device Manager window, under the Ports (COM & LPT) device type.
5. Check that the USB driver is installed properly and that the device is functioning. In Microsoft Windows®, the device status can be checked using Device Manager by right-clicking **Lake Shore Instrument** under Ports (COM & LPT) or Other Devices and then clicking Properties.

Intermittent Lockups

1. Check cable connections and length.
2. Increase delay between all commands to 100 ms to make sure instrument is not being overloaded.
3. Ensure that the USB cable is not unplugged and that the M81-SSM is not powered down while the com port is open. The USB driver creates a com port when the USB connection is detected, and removes the com port when the USB connection is no longer detected. Removing the com port while in use by software can cause the software to lock up or crash.

7.1.4 IEEE Interface Troubleshooting

This section provides IEEE interface troubleshooting for issues that arise with new installations, old installations, and intermittent lockups.

New Installation

1. Check the instrument address.
2. Always send a message terminator.
3. Send the entire message string at one time including the terminator.
4. Send only one simple command at a time until communication is established.
5. Be sure to spell commands correctly and use proper syntax.

6. Attempt both Talk and Listen functions. If one works but not the other, the hardware connection is working, so look at syntax, terminator, and command format.

Existing Installation No Longer Working

1. Power the instrument off, then on again to see if it is a soft failure.
2. Power the computer off then on again to see if the IEEE card is locked up.
3. Verify that the address has not been changed on the instrument during a memory reset.
4. Check all cable connections.

Intermittent Lockups

1. Check cable connections and length.
2. Increase the delay between all commands to 50 ms to make sure the instrument is not being overloaded.

7.2 Error and Status Messages

The following are error and status messages that may be displayed by the M81-SSM during operation.

7.2.1 SCPI Error Messages

Critical startup errors:

The following are errors that may be displayed during startup. These errors are critical and will prevent the M81-SSM from functioning properly. Please contact Lake Shore Service for support see *Technical Inquiries* (page 317) for more information.

Error	SCPI error code	Description
Invalid calibration	-340	The calibration is not valid. The data is either corrupt, uninitialized or default values.
NOVRAM not communicat-ing	-320	Indicates that a NOVRAM device failed to initialize. See *Summary of Internal Memory Devices* (page 306) for more information about memory devices.
Model ID not communicat-ing	-300	Indicates that the device used to store the model ID failed to initialize.
Rev ID not communicat-ing	-300	Indicates that the device used to store the revision ID failed to initialize.
Line frequency out of bounds	-300	Indicates the detected line frequency is out of normal operating range. A default value of 60 Hz is used.
ADC config corrupt	-300	Indicates that an invalid internal ADC configuration was detected.
Invalid channel card configuration	-300	Indicates that an invalid channel card configuration was detected. This typically indicates that an expected channel card was not detected.

Runtime errors:

The following are errors that may be displayed during operation. These errors are critical and will prevent the M81-SSM from functioning properly. Please contact Lake Shore Service for support see *Technical Inquiries* (page 317) for more information.

Error	SCPI error code	Description
Overflow	-240	Internal data streaming buffer overflow. You can continue, but note that data loss has occurred during a data stream.
Measure module I2C corrupt	-240	Failed readback on the specified measure module I2C write.
Source module I2C corrupt	-240	Failed readback on the specified source module I2C write.
Head temp sensor failure	-240	Failed to communicate with the head board temperature sensor.
Measure module temp sensor failure	-240	Failed to communicate with the specified measure module temperature sensor.
Source module temp sensor failure	-240	Failed to communicate with the specified source module temperature sensor.
Master reference error	-240	The measured master reference voltage is out of bounds.
Source gain max correction	-240	Computed gain factor is out of bounds on the specified source module.
Source offset max correction	-240	Measured offset is too large on the specified source module.
Source DC gain max correction	-240	Computed gain factor is out of bounds on the specified source module's DC chain.
Source DC offset max correction	-240	Measured offset is too large on the specified source module's DC chain.
Measure gain max correction	-240	Computed gain factor is out of bounds on the specified measure module.
Measure offset max correction	-240	Measured offset is too large on the specified measure module.
Measure voltage nulling limit	-240	Input offset voltage is too large to correct on the specified measure module.
Measure current nulling limit	-240	Input bias current is too large to correct on the specified measure module.
Measure module JSON checksum	-230	Checksum validation of the specified measure module's JSON NOVRAM data has failed.

Continued on next page

Table 7.1 – continued from previous page

Error	SCPI error code	Description
Source module JSON checksum	-230	Checksum validation of the specified source module's JSON NOVRAM data has failed.
Internal communication error	N/A	Internal communication between the instrument's microcontrollers or hardware was interrupted. This typically occurs when a module is unplugged without first unloading it. See *Module Loading* (page 24) for details on how to properly load and unload modules. When this error is encountered the instrument will attempt to re-estabilish communication. If it is able to do so, an option to continue will be presented ortherwise the instrument will need to be restarted.

Module connect errors:

The following are errors that may be displayed when connected modules are loaded. The module will not function until the error is corrected.

Error	Description
Read error	There was an error identifying the connected module.
Unrecognized module error	The connected module was identified, but is not supported with this firmware version. This can typically be resolved by updating the firmware.
Port direction error	The connected module was plugged into the wrong port. For example, a source module was plugged into a measure port. This can typically be resolved by plugging the module into the correct port type.

Measurement status messages:

The following are status messages that may be displayed during measurement. Readings may not be available when these messages are displayed.

Status	Description
Settling	The measurement is settling due to a hardware change.
Overload	The module is in an active overload.
Unlocked	The module's lock-in amplifier is unlocked.
V Limit	The module is at its voltage limit.
I Limit	The module is at its current limit.

Self-calibration status messages:

The following are status messages that may be displayed during self-calibration.

Status	Description
In progress	The calibration is in progress. See specific step details.
Initializing step	Getting ready to run calibration.
Learning reference step	Measuring internal reference against external calibrator.
Measuring references step	Measuring internal reference voltage.
Calibrating source step	Calibrating specified source channel.
Calibrating measure step	Calibrating specified measure channel.
Voltage offsets step	Calibrating offset for specified voltage range.
Nulling current step	Calibrating offsets.
Nulling voltage step	Adjusting input offset voltage.
Self-calibration successful	Calibration completed successfully.

Resistance calculation status messages:

The following are status messages that may be displayed during self-calibration.

Status	Description
S MISS	The selected source module is not connected.
S UNL	The selected source module is connected but not loaded.
INVALID	The source module is not compatible with the measure module for calculating resistance. This occurs when the two modules use the same units, such as a VM-10 and VS-10.
E TYPE	The excitation type is not valid. This occurs when the measure mode and source shape do not create a supported resistance calculation.
DIV BY 0	The sourced or measured current is 0.
M ERR	There is an active error on the measure module.
S ERR	There is an active error on the source module.

7.3 Fuse and Line Voltage Selection

The fuse drawer supplied with the M81-SSM holds the instrument line fuses and line voltage selection module. The drawer holds two 5 mm × 20 mm (0.2 in × 0.79 in) time delay fuses. It requires two good fuses of the same rating to operate safely.

Figure 7.1: Line input assembly

7.3.1 Fuse Replacement

Use this procedure to remove and replace a line fuse.

> **Warning:** To avoid potentially lethal shocks, turn off controller and disconnect it from AC power before performing these procedures.

Caution: For continued protection against fire hazard, replace only with the same fuse type and rating specified for the line voltage selected.

Note: Test fuse with an ohmmeter. Do not rely on visual inspection of fuse.

1. Locate the line input assembly on the instrument rear panel.
2. Remove the instrument power cord.
3. With a small screwdriver, release the drawer holding the line voltage selector and fuse.
4. Remove existing fuse(s). Replace with proper slow-blow (time-delay) fuse ratings.
5. Re-assemble the line input assembly in reverse order.
6. Verify voltage indicator in the line input assembly window.
7. Connect the instrument power cord.

7.3.2 Line Voltage Selection

Use the following procedure to change the instrument line voltage selector.

Warning: To avoid potentially lethal shocks, turn off controller and disconnect it from AC power before performing these procedures.

1. Identify the line input assembly on the instrument rear panel.
2. Remove the instrument power cord.
3. With a small screwdriver, release the drawer holding the line voltage selector and fuse.
4. Slide out the voltage selector from the drawer.
5. Rotate the voltage selector until the proper voltage indicator shows through the window.
6. Re-assemble the line input assembly in the reverse order.
7. Verify the voltage indicator in the window of the line input assembly.
8. Connect the instrument power cord.

7.4 Rear Panel Connector Definitions

7.4.1 Digital I/O

This termial block provides configurable digital inputs and outputs. See *Digital I/O* (page 53) for more information.

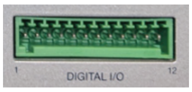


Figure 7.2: Digital I/O connector

Pin	Description
1	Input 1
2	I/O ground
3	Input 2
4	I/O ground
5	+5 V
6	I/O ground
7	Output 1
8	I/O ground
9	Output 2
10	I/O ground
11	Measurement ground
12	Earth ground

The mating connector for the digital I/O is a Phoenix 1923432 terminal block.

7.4.2 USB

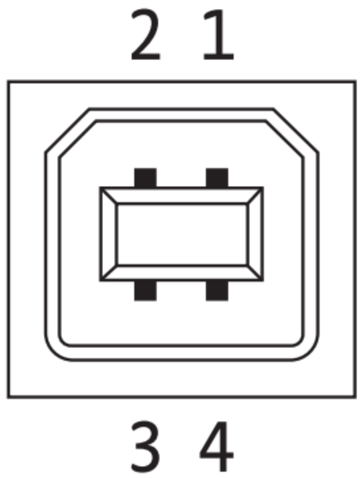


Figure 7.3: USB connector

Pin	Symbol	Description
1	VCC	+5 VDC
2	D-	Data-
3	D+	Data+
4	GND	Ground

7.4.3 USB Type-C™

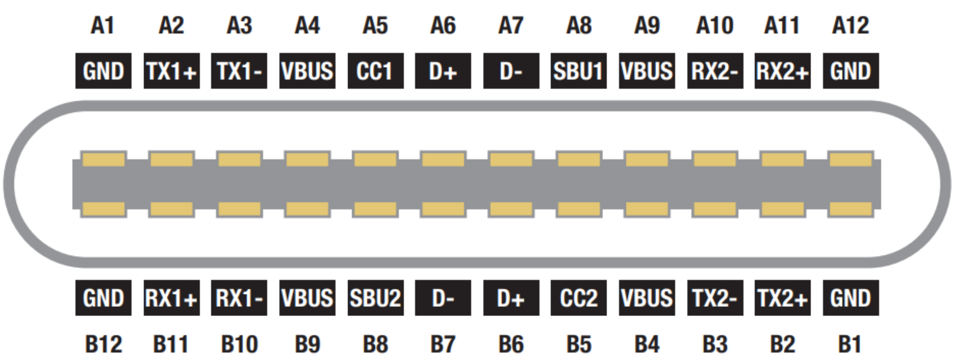


Figure 7.4: USB Type-C™ connector

7.4.4 Ethernet

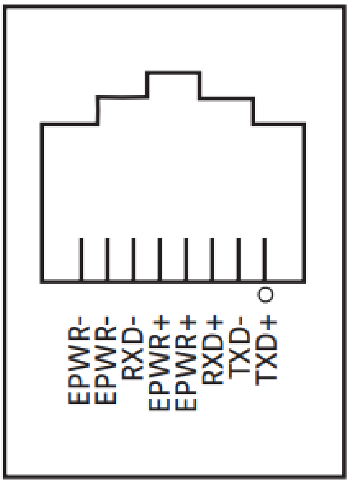


Figure 7.5: Ethernet connector

Pin	Symbol	Description
1	TXD+	Transmit data+
2	TXD-	Transmit data-
3	RXD+	Receive data+
4	EPWR+	Power from switch+ (not used)
5	EPWR+	Power from switch+ (not used)
6	RXD-	Receive data-
7	EPWR-	Power from switch- (not used)
8	EPWR-	Power from switch- (not used)

7.4.5 Module Connectors

Lake Shore M81-SSM Synchronous Source Measure System source and measure modules plug into these connectors. Source modules must be plugged into "S" ports and measure modules must be plugged into "M" ports.

Connect only Lake Shore M81-SSM Synchronous Source Measure System modules to these connectors.

7.4.6 Ref In

The reference input allows the M81-SSM to track an external phase reference. See *Reference In* (page 54) for more information.

BNC pinout:

Center	Input voltage
Shell	I/O ground

7.4.7 Ref Out

The reference input enables the M81-SSM to provide a phase reference to external equipment. See *Reference Out* (page 55) for more information.

BNC pinout:

Center	Output voltage
Shell	I/O ground

7.4.8 Monitor Out

The monitor output is a configurable analog output. See *Monitor Out* (page 56) for more information.

BNC pinout:

Center	Output voltage
Shell	I/O ground

7.5 Summary of Internal Memory Devices

This section outlines the internal memory devices used inside the M81-SSM and provides an explanation of the types of data they contain.

Table 7.2: Internal memory devices

Part	Location	Function	Volatility	Field-modifiable
Microchip ATSAME70Q21A Microprocessor	Main U1	Microprocessor code	Non-volatile	No
Cypress FM24V10 1 Mbit serial F-RAM	Main U2	Calibration, settlings, license files	Non-volatile	Yes
Kingston EMMC04G-S100-A08U 4 GB eMMC Flash	Digital U31	Operating system, firmware repository, app service, and instrument settings	Non-volatile	No
SK Hynix H5TC2G83GFR–PBA 1 GB (256 MB x 4) DDR3L SRAM	Digital U2, U3, U4, U5	Run-time data storage	Volatile	No
Microchip SST25VF016B-50-4C 16 Mbit serial flash	Digital U17	Bootloader	Non-volatile	No
Microchip 24LC1025T-I/SN 1 Mb serial EEPROM	Module U28	Module calibration and information	Non-volatile	Yes

1. The Atmel/Microchip ATSAME70Q21A is a microprocessor with 2048 KB of onboard programmable flash memory. The code is loaded by initiating a firmware update over the remote interface.
2. The Cypress FM24V10 is a 128 KB I2C FRAM. It is used to store calibration constants, instrument settings, and feature license files. The memory is programmed at Lake Shore and is modified when changing user settings and running self calibrations.
3. The Kingston EMMC04G-S100-A08U is a 4 GB eMMC flash memory used to store the instrument operating system. It also contains the local firmware repository and app service, and instrument settings.
4. The SK Hynix H5TC2G63GFR-PBA memories are composed of four 256 MBDDR3L SRAM chips, totaling 1 GB of volatile memory. This memory is used by the operating system and applications during runtime as temporary data storage. Data is lost on a power cycle.
5. The Microchip SST25VF016B-50-4C is a 16 Mbit SPI flash. It is used to store bootloader firmware. The memory is programmed at Lake Shore.
6. The 24LC1025T is a 1 Mbit I2C EEPROM. It is used on the module to store calibration

and other module information. The memory is programmed at Lake Shore and is modified when changing user settable module information and running module self calibrations.

7.6 Technical Inquiries

Refer to the following sections when contacting Lake Shore for application assistance or product service. Questions regarding product applications, price, availability and shipments should be directed to sales. Questions regarding instrument calibration or repair should be directed to instrument service. Do not return a product to Lake Shore without a Return Material Authorization (RMA) number.

7.6.1 Contacting Lake Shore

The Lake Shore Service Department is staffed Monday through Friday between the hours of 8:00 AM and 5:00 PM EST, excluding holidays and company shut down days.

The Lake Shore Forum is also a great place to look for solutions, to post issues, and to share successes: forums.lakeshore.com.

Contact Lake Shore Service through any of the means listed below. However, the most direct and efficient means of contacting is to complete the online service request form at www.lakeshore.com/service. Provide a detailed description of the problem and the required contact information. You will receive a response within 24 hours or the next business day in the event of weekends or holidays.

If you wish to contact Service or Sales, use the following:

Lake Shore Cryotronics, Inc.

575 McCorkle Blvd.

Westerville, Ohio

43082-8888 USA

Sales

sales@lakeshore.com

Telephone: (614) 891-2244

Fax: (614) 818-1600

Service

www.lakeshore.com/service

support@lakeshore.com

Telephone: (614) 891-2243, option 4

Fax: (614) 818-1609

7.6.2 Return of Equipment

The M81-SSM is packaged to protect it during shipment.

Note: The user should retain any shipping carton(s) in which equipment is originally received, in the event that any equipment needs to be returned.

If the original packaging is not available, a minimum of 76.2 mm (3 in) of shock absorbent packing material should be placed snugly on all sides of the instrument in a sturdy corrugated cardboard box. Please use reasonable care when removing the product from its protective packaging and inspect it carefully for damage. If it shows any sign of damage, please file a claim with the carrier immediately. Do not destroy the shipping container; it will be required by the carrier as evidence to support claims. Call Lake Shore for return and repair instructions.

All equipment returns must be approved by a member of the Lake Shore Service Department. The service engineer will use the information provided in the service request form and will issue an RMA. This number is necessary for all returned equipment. It must be clearly indicated on both the shipping carton(s) and any correspondence relating to the shipment. Once the RMA has been approved, you will receive appropriate documents and instructions for shipping the equipment to Lake Shore.

RMA Valid Period

RMAs are valid for 60 days from issuance; however, we suggest that equipment needing repair be shipped to Lake Shore within 30 days after the RMA has been issued. You will be contacted if we do not receive the equipment within 30 days after the RMA is issued. The RMA will be cancelled if we do not receive the equipment after 60 days.

Shipping Charges

All shipments to Lake Shore are to be made prepaid by the customer. Equipment serviced under warranty will be returned prepaid by Lake Shore. Equipment serviced out-of-warranty will be returned FOB Lake Shore.

Restocking Fee

Lake Shore reserves the right to charge a restocking fee for items returned for exchange or reimbursement.