NHR 9400 Series
AC/DC Power Module

Programmer's Reference Manual

Part No. 09-0335

REVISION H

March 28, 2016

Document History

Rev A		**5/29/2015**	**R. W.**
Rev B	**Add Macro Subsection**	**6/26/2015**	**R.W.**
Rev C	**Add CF (crest factor) and PF (power factor measurements)**	**10/5/2015**	**R.W.**
Rev D	**ECO 17364, Remove INT:SENS references**	**10/15/2015**	**R.W.**
Rev E	**Add absolute peak to SAFety command**	**10/27/2015**	**R.W.**
Rev F	**Include Load Mode**	**11/24/2015**	**R.W.**
Rev G	**Define for separate models**	**12/2/2015**	**R.W.**
Rev H	**ECO 17423, Corrections for separate models and macro ability**	**3/28/2016**	**P.B.**

Proprietary Rights Notice

The material in this publication is current as of the release date on the cover page and is subject to change without notice. Any questions concerning the product or any questions or comments concerning this manual should be directed to the NH Research Field Service Department during normal business hours Monday through Friday at (949) 474-3900 or FAX (949) 474-7062 or www.nhresearch.com.

TABLE OF CONTENTS

SAFETY 13

WARNING: LIVE CIRCUITS 13

PROGRAMMING OVERVIEW 14

SCPI PROGRAMMING 15

Getting Started with SCPI 15

Command Reference NOTES 20

Command or Query Scope 20

Units 20

Common Commands 21

*IDN? (Identify) 21

*OPC (Operation Complete) 21

*RST (Reset) 21

*SRE (Service Request Enable) 22

*TRG (Trigger) 22

*TST? (Selftest) 22

*WAI 22

Status Commands 24

Status Registers 24

*CLS (Clear Status) 27

*ESE (Event Status Enable) 27

*ESR? (Event Status Register) 27

*STB? (Status Byte Register) 27

STATus:OPERation[:EVENt]? 27

STATus:OPERation:CONDition? 28

STATus:OPERation:ENABle 28

STATus:PRESet 28

STATus:QUEStionable[:EVENt]? 28

STATus:QUEStionable:CONDition? 28

STATus:QUEStionable:ENABle 29
SYSTem:ERRor? 29
CALibration Subsection 30
CALibration:DATE? 30
CALibration:RANGe 30
CALibration:CURRent:RAM 30
CALibration:CURRent:FLASh? 31
CALibration:CURRent:RESet 31
CALibration:VOLTage:RAM 31
CALibration:VOLTage:FLASh? 31
CALibration:VOLTage:RESet 32
CALibration:SAVE 32
CONFigure Subsection 33
CONFigure:HW:MODE 33
CONFigure:HW:MODE:VALid 34
CONFigure:INSTrument:ATRansformer 34
CONFigure:INSTrument:SYNChronous 35
DATA Subsection 37
DATA[:DATA] 37
DATA:LOAD[:ALL] 37
DATA:LOAD:APHase, BPHase, or CPHase 37
DIAGnostic Subsection 38
DIAGnostic:ERRor:COMMand? 38
DIAGnostic:NHR:COMMand 38
DIAGnostic:NHR:MEMory:READ? 38
DIAGnostic:NHR:SIMulate? 38
DIGital Subsystem 39
DIGital:INPut? 39
DIGital:INPut:COUNt? 39
DIGital:OUTPut 39
DIGital:OUTPut:COUNt? 39
INSTrument Subsystem 40

INSTrument:CAPabilities: 40
INSTrument:CAPabilities:APERture:MAXimum? 40
INSTrument:CAPabilities:CURRent:MAXimum? 40
INSTrument:CAPabilities:CURRent:MEASurement:MAXimum? 40
INSTrument:CAPabilities:CURRent:MEASurement:RANGe:MAXimum? 40
INSTrument:CAPabilities:CURRent:RANGe:LIST:MINimum? 40
INSTrument:CAPabilities:CURRent:RANGe:MAXimum? 41
INSTrument:CAPabilities:CURRent:SLEW:MAXimum? 41
INSTrument:CAPabilities:CURRent:SLEW:RANGe:MAXimum? 41
INSTrument:CAPabilities:FREQuency:MAXimum? 41
INSTrument:CAPabilities:FREQuency:RANGe:MAXimum? 41
INSTrument:CAPabilities:FREQuency:SLEW:MAXimum? 41
INSTrument:CAPabilities:FREQuency:SLEW:RANGe:MAXimum? 41
INSTrument:CAPabilities:MACRo:DELay:MAXimum? 42
INSTrument:CAPabilities:PHASe:SLEW:MAXimum? 42
INSTrument:CAPabilities:PHASe:SLEW:RANGe:MAXimum? 42
INSTrument:CAPabilities:POWer:APHase, BPHase or CPHase:MINimum? 42
INSTrument:CAPabilities:POWer[:ALL]:MAXimum? 42
INSTrument:CAPabilities:POWer:APHase, BPHase or CPHase:MAXimum? 42
INSTrument:CAPabilities:POWer:APHase, BPHase or CPHase:RANGe: MINimum? 42
INSTrument:CAPabilities:POWer[:ALL]:RANGe:MAXimum? 42
INSTrument:CAPabilities:POWer: APHase, BPHase or CPHase:RANGe:MAXimum? 42
INSTrument:CAPabilities:POWer:SLEW:MAXimum? 42
INSTrument:CAPabilities:POWer:SLEW:RANGe:MAXimum? 43
INSTrument:CAPabilities:RESistance:MAXimum? 43
INSTrument:CAPabilities:RESistance:RANGe:MAXimum? 43
INSTrument:CAPabilities:RESistance:SLEW:MAXimum? 43
INSTrument:CAPabilities:RESistance:SLEW:RANGe:MAXimum? 43
INSTrument:CAPabilities:RGAin:NOMinal? 43
INSTrument:CAPabilities:RGAin:MAXimum? 43
INSTrument:CAPabilities:SAMPle:FREQuency:MAXimum? 43
INSTrument:CAPabilities:TRIP:MAXimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase: MINimum? 44

INSTrument:CAPabilities:VOLT[:ALL]:MAXimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MAXimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MEASurement: MINimum? 44

INSTrument:CAPabilities:VOLT[:ALL]:MEASurement:MAXimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MEASurement:MAXimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MEASurement:RANGe: MINimum? 44

INSTrument:CAPabilities:VOLT[:ALL]:MEASurement:RANGe:MAXimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MEASurement:RANGe:MAXimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:RANGe:LIST[:MAXimum]? 44

INSTrument:CAPabilities:VOLT[:ALL]:RANGe:LIST:MINimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:RANGe:LIST:MINimum? 44

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:RANGe: MINimum? 45

INSTrument:CAPabilities:VOLT[:ALL]:RANGe:MAXimum? 45

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:RANGe:MAXimum? 45

INSTrument:CAPabilities:VOLT:SLEW:MAXimum? 45

INSTrument:CAPabilities:VOLT:SLEW:RANGe:MAXimum? 45

INSTrument:CAPabilities:CF:MAXimum? 45

INSTrument:CAPabilities:PF:MAXimum? 45

INSTrument:CAPabilities:R:MAXimum? 45

INSTrument:CAPabilities:L:MAXimum? 45

INSTrument:DEFine[:NAME] 45

INSTrument:DELete[:NAME] 46

INSTrument:NAME? 46

INSTrument:NSELect 46

INSTrument:SELect 46

MACRo Subsection 48

MACRo:LEARn 48

MACRo:ABORt 48

MACRo:RUN[:ONCE] 48

MACRo:RUN:CONTinuous 48

MACRo:WAIT[:TIME] 48

MACRo:WAIT:CYCLes:INSTrument1 (or INSTrument2, or INSTrument3)..49

MACRo:WAIT:SLEW 49

MACRo:WAIT:TRIGger..49

Measurement Commands 51

ABORt..51

INITiate[:IMMediate] 51

MEASure:ARRay:CURRent?..52

MEASure:ARRay:CURRent:APHase? (or BPHase, or CPHase) 52

FETCh:ARRay:CURRent?..52

FETCh:ARRay:CURRent:APHase? (or BPHase, or CPHase) 52

MEASure:ARRay:VOLTage?..52

MEASure:ARRay:VOLTage:APHase? (or BPHase, or CPHase) 52

FETCh:ARRay:VOLTage? 52

FETCh:ARRay:VOLTage:APHase? (or BPHase, or CPHase) 52

FETCh:BACKground? 53

FETCh:BACKground:ALL?..54

FETCh:BACKground:DC?..54

FETCh:BACKground:TEMPerature?..54

MEASure:CF?..55

MEASure:CF:APHase? (or BPHase, or CPHase) 55

FETCh:CF?..55

FETCh:CF:APHase? (or BPHase, or CPHase) 55

MEASure:CURRent? 55

MEASure:CURRent:APHase? (or BPHase, or CPHase)..55

FETCh:CURRent? 55

FETCh:CURRent:APHase? (or BPHase, or CPHase)..55

MEASure:CURRent:PEAK:MAXimum? 55

MEASure:CURRent:PEAK:MAXimum:APHase? (or BPHase, or CPHase)..55

FETCh:CURRent:PEAK:MAXimum?..55

FETCh:CURRent:PEAK:MAXimum:APHase? (or BPHase, or CPHase)..55

MEASure:CURRent:PEAK:MINimum? 55

MEASure:CURRent:PEAK: MINimum:APHase? (or BPHase, or CPHase)..55

FETCh:CURRent:PEAK:MINimum? 55
FETCh:CURRent:PEAK: MINimum:APHase? (or BPHase, or CPHase) 55
MEASure:RESet[:ALL] 55
MEASure:RESet:ENERgy 56
MEASure:RESet:PEAKs 56
MEASure:PF? 56
MEASure:PF:APHase? (or BPHase, or CPHase) 56
FETCh:PF? 56
FETCh:PF:APHase? (or BPHase, or CPHase) 56
MEASure:POWer[:TRUE]? 56
MEASure:POWer[:TRUE]:APHase? (or BPHase, or CPHase) 56
FETCh:POWer[:TRUE]? 56
FETCh:POWer[:TRUE]:APHase? (or BPHase, or CPHase) 57
MEASure:POWer:APParent? 57
MEASure:POWer:APParent:APHase? (or BPHase, or CPHase) 57
FETCh:POWer:APParent? 57
FETCh:POWer:APParent:APHase? (or BPHase, or CPHase) 57
MEASure:POWer:APParent:PEAK:MAXimum? 57
MEASure:POWer:APParent:PEAK:MAXimum:APHase? (or BPHase, or CPHase) 57
FETCh:POWer:APParent:PEAK:MAXimum? 57
FETCh:POWer:APParent:PEAK:MAXimum:APHase? (or BPHase, or CPHase) 57
MEASure:POWer:APParent:PEAK:MINimum? 57
MEASure:POWer:APParent:PEAK:MINimum:APHase? (or BPHase, or CPHase) 57
FETCh:POWer:APParent:PEAK:MINimum? 57
FETCh:POWer:APParent:PEAK:MINimum:APHase? (or BPHase, or CPHase) 57
MEASure:POWer:PEAK:MAXimum? 57
MEASure:POWer:PEAK:MAXimum:APHase? (or BPHase, or CPHase) 57
FETCh:POWer:PEAK:MAXimum? 58
FETCh:POWer:PEAK:MAXimum:APHase? (or BPHase, or CPHase) 58
MEASure:POWer:PEAK:MINimum? 58
MEASure:POWer:PEAK:MINimum:APHase? (or BPHase, or CPHase) 58
FETCh:POWer:PEAK:MINimum? 58

FETCh:POWer:PEAK:MINimum:APHase? (or BPHase, or CPHase) 58
MEASure:VOLTage? 58
MEASure:VOLTage:APHase? (or BPHase, or CPHase) 58
FETCh:VOLTage? 58
FETCh:VOLTage:APHase? (or BPHase, or CPHase) 58
MEASure:VOLTage:PEAK:MAXimum? 58
MEASure:VOLTage:PEAK:MAXimum:APHase? (or BPHase, or CPHase) 58
FETCh:VOLTage:PEAK:MAXimum? 58
FETCh:VOLTage:PEAK:MAXimum:APHase? (or BPHase, or CPHase) 58
MEASure:VOLTage:PEAK:MINimum? 59
MEASure:VOLTage:PEAK:MINimum:APHase? (or BPHase, or CPHase) 59
FETCh:VOLTage:PEAK:MINimum? 59
FETCh:VOLTage:PEAK:MINimum:APHase? (or BPHase, or CPHase) 59
SENSe:SWEep:APERture 59
SENSe:SWEep:POINts 59
SENSe:SWEep:TINTerval? 59
SOURce Subsystem 60
[SOURCE:]CURRent[:ALL] 60
[SOURCE:]CURRent:APHase, BPHase, or CPHase 60
[SOURCE:]CURRent:RANGe 60
[SOURCE:]CURRent:SLEW 61
[SOURCE:]CURRent:ANGLe 61
[SOURCE:]CURRent:CF 61
[SOURCE:]CURRent:PF 61
[SOURCE:]FREQuency 62
[SOURCE:]FREQuency:SLEW 62
[SOURCE:]FUNCtion[:SHAPe][:ALL] 62
[SOURCE:]FUNCtion[:SHAPe]:APHase, BPHase, or CPHase 62
[SOURCE:]LOAD:MODE 62
[SOURCE:]LOAD:SEND 63
[SOURCE:]LOAD:APPLy 63
[SOURCE:]OUTPut[:ON] 63

[SOURCE:]OUTPut[:ON]:ALL 63
[SOURCE:]OUTPut:ABORt 63
[SOURCE:]OUTPut:SYNC:ANGLe[:ALL] 63
[SOURCE:]OUTPut:SYNC:ANGLe:APHase, BPHase, or CPHase 63
[SOURCE:]OUTPut:RGAin 64
[SOURCE:]PHASe[:ANGLe] 64
[SOURCE:]PHASe:ANGLe:BPHase 64
[SOURCE:]PHASe:ANGLe:CPHase 64
[SOURCE:]PHASe:SLEW 65
[SOURCE:]POWer[:ALL] 65
[SOURCE:]POWer:APHase, BPHase, or CPHase 65
[SOURCE:]POWer:SLEW 65
[SOURCE:]CURRent:PRIority 66
[SOURCE:]RESistance 66
[SOURCE:]RL:RESistance 66
[SOURCE:]RL:INDuctance 66
[SOURCE:]SAFety 66
[SOURCE:]VA 67
[SOURCE:]VOLTage[:ALL] 67
[SOURCE:]VOLTage:APHase, BPHase, or CPHase 67
[SOURCE:]VOLTage:RANGe 67
[SOURCE:]VOLTage:SLEW 68
System Commands 69
SYSTem:ERRor? 69
SYSTem:LED 69
SYSTem:LOCal 69
SYSTem:REMote 69
SYSTem:RWLock 69
SYSTem:VERSion? 69
SYSTem:WATChdog:INTerval 70
SYSTem:WATChdog:ROBust 70
SYSTem:WATChdog:SERVice 70

Trigger Commands 71

ABORt 71

<Future>TRIGger:AGENerate 71

<Future>TRIGger:SOURce 71

LABVIEW™ PROGRAMMING 73

Getting Started with LabVIEW™ 73

Introduction 73

Preparing the System 73

Referencing the Driver 73

Initializing the Driver 73

Accessing Methods and Properties 73

Destroying the Driver Instance 73

Programming Example 73

IVI PROGRAMMING 75

Getting Started with IVI 75

Introduction 75

Types of IVI Drivers 75

Getting Help 75

Getting Started with Visual Studio (.NET) 77

Introduction 77

Referencing the Driver 77

Creating an Instance of the Driver 77

Connecting 77

Accessing Methods and Properties 78

Hardware Mode 78

Accessing Instrument Channels 79

Destroying the Driver Instance 79

Programming Examples 80

Getting Started with LabWindows™/CVI™ 81

Introduction 81

Referencing the Driver 81
Initializing the Driver 81
Accessing Methods and Properties 81
Hardware Mode 82
Accessing Instrument Channels 82
Destroying the Driver Instance 83
Programming Example 83

SAFETY

WARNING: LIVE CIRCUITS

NO INTERNAL ADJUSTMENT OR COMPONENT REPLACEMENT IS ALLOWED BY NON-NH RESEARCH QUALIFIED PERSONNEL. COMPLETE THE FOLLOWING BEFORE ACCESSING THE INSTRUMENT INTERNALS:

- REMOVE ALL EXTERNAL VOLTAGE SOURCES
- DISCONNECT POWER CORD
- WAIT A MINIMUM OF 1 MINUTE TO DISCHARGE CIRCUITS
- VERIFY CIRCUITS ARE DISCHARGED

Programming Overview

SCPI Programming

Getting Started with SCPI

Introduction

The AC/DC Power Module is a highly flexible piece of equipment. It can act as one or more AC sources, one or more DC sources, or nearly any combination. The configuration commands allow you to set up the power module to function as you desire.

Important Terminology

Output: Output is the generic term applied to the connection from the 9400 to the UUT (unit under test). Output, however, should not be construed to imply power flow direction. The output when connected to a UUT can be used to source or sink current.

Power Module: Power module is the generic name for the 9400. Since the 9400 can act as a DC source, an AC source, and is bi-directional with energy flowing to or from the grid it cannot simply be referred to as a source or a load.

Output Channel: The 9400 has one to three internal power sections per chassis known as channels which can be configured for AC or DC operation. Each chassis channel is 4 kW so a 12 kW 9400 will have three channels. Also, channels 2 & 3 can be paralleled with channel 1. When in AC mode, the channels are also referred to as Phase A, B, and C. Channels in multiple chassis are always paralleled.

Hardware Mode: The organization of the channels is done by setting a “mode” with the CONFigureMODE command. Setting the mode will allow you to determine if it is a three output AC source, or a single output DC source, or anything in-between. There are a total of 13 hardware modes available for a three-channel 9400 power module, six for a two-channel 9400, and two for a one-channel 9400.

Instrument: A configured power module will have from one to three logical instruments. The logical instruments have both an instrument number and an instrument name. For example, in hardware mode 3 (3 ac source outputs) there are three logical instruments named AC1, AC2, and AC3. Many commands will be directed at an instrument. That will require using INST:SEL or INST:NSEL prior to sending the command. Note that a command or query that applies to the currently selected instrument will be noted as an “instrument command” or “instrument query” in the description. Commands and queries that do NOT apply exclusively to the selected instrument will be noted as a “system command” or “system query” in the description.

Query: A query is a command sent with a trailing ‘?’, like “*IDN?” ***IMPORTANT***: Queries ALWAYS return a string even if the called failed. A response string of “<ERROR xxx> is returned if an error occurred where xxx is the error number. A response string of “<NO RESPONSE>” is returned if there is nothing to return.

Connecting

When using SCPI, you will connect to the IP address of the Module and use a port based on how you will be interacting with it.

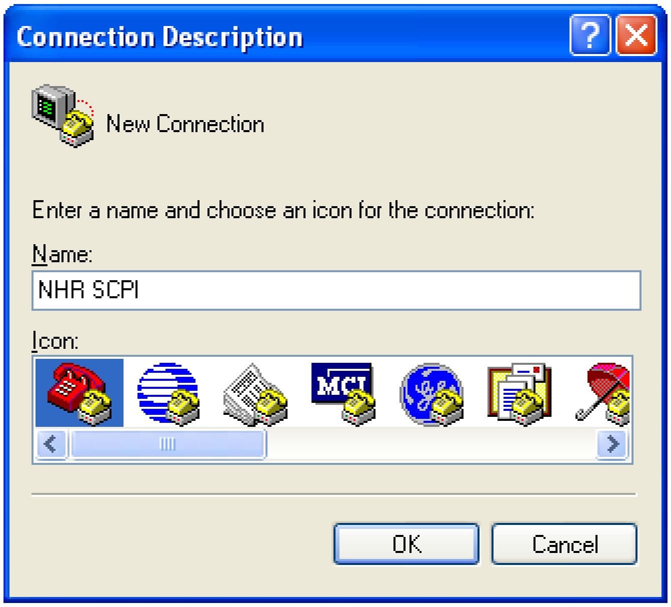
- When using a terminal (like HyperTerminal) where you will be interacting with the Module directly, open a connection to the static IP address of the Module on port 5024 (Telnet SCPI).

The Telnet SCPI port is expected to interact with a human. It will display a welcome message and prompt queues.

- When controlling the Module from a program, open a connection to the static IP address of the Module on port 5025 (Raw SCPI). The Raw SCPI port expects to interact with a computer and does not have any superfluous communication.

Here is how to set up HyperTerminal:

Launch HyperTerminal and create a new connection.



Connection Description

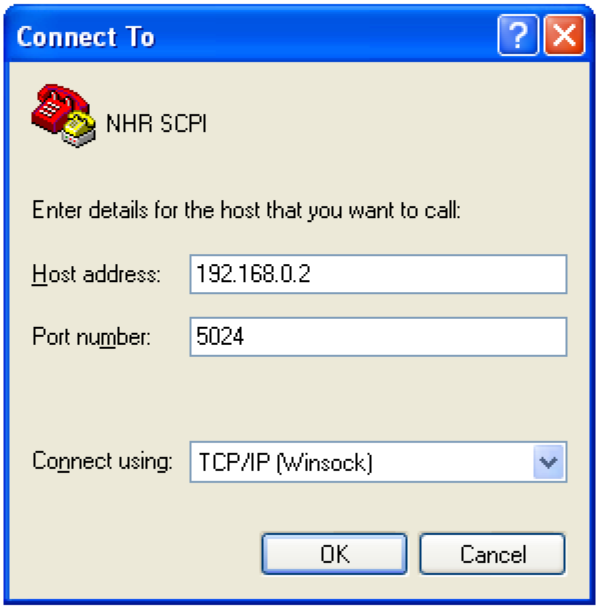
New Connection

Enter a name and choose an icon for the connection:

Name: NHR SCPI

Icon:

Configure for NHR Telnet SCPI.



Connect To

NHR SCPI

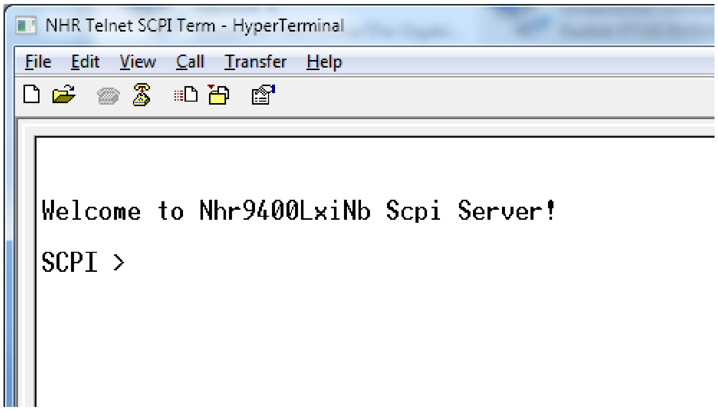
Enter details for the host that you want to call:

Host address: 192.168.0.2

Port number: 5024

Connect using: TCP/IP (Winsock)

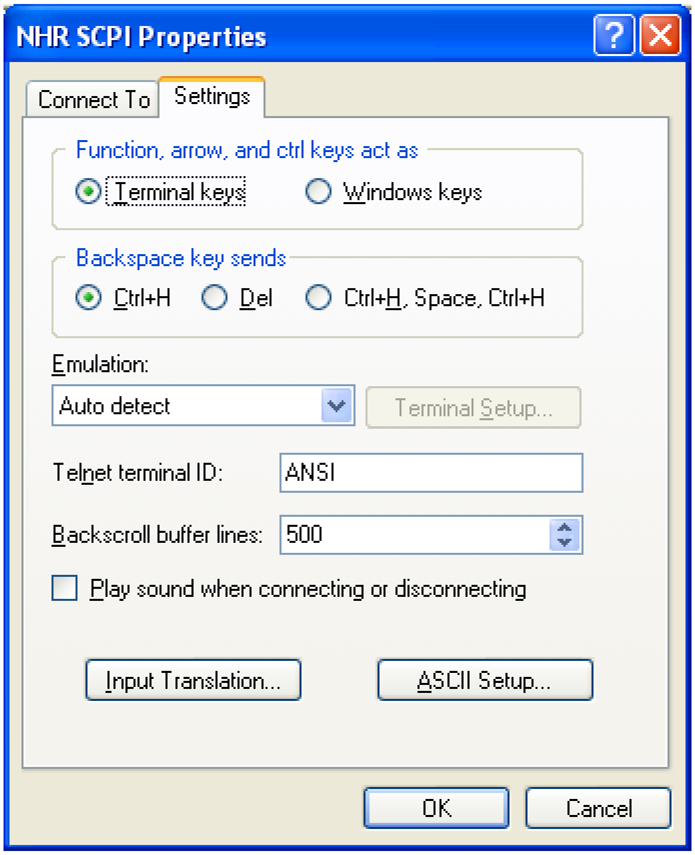
The Module should send a welcome message when you connect.


NHR Telnet SCPI Term - HyperTerminal
File Edit View Call Transfer Help

```
Welcome to Nhr9400LxiNb Scpi Server!
SCPI >
```

HOWEVER, you need to do some more configurations before you are ready to go.

Open the File/Properties menu and view the "Settings" tab.



Function, arrow, and ctrl keys act as
● Terminal keys ○ Windows keys

Backspace key sends
● Ctrl+H ○ Del ○ Ctrl+H, Space, Ctrl+H

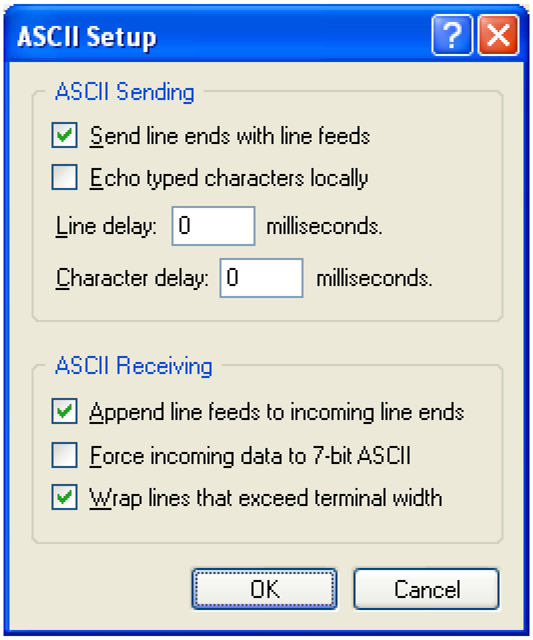
Emulation: Auto detect

Telnet terminal ID: ANSI

Backscroll buffer lines: 500

□ Play sound when connecting or disconnecting

Click on "ASCII Setup…" and setup as shown below.



ASCII Sending

☑ Send line ends with line feeds

□ Echo typed characters locally

Line delay: 0 milliseconds.

Character delay: 0 milliseconds.

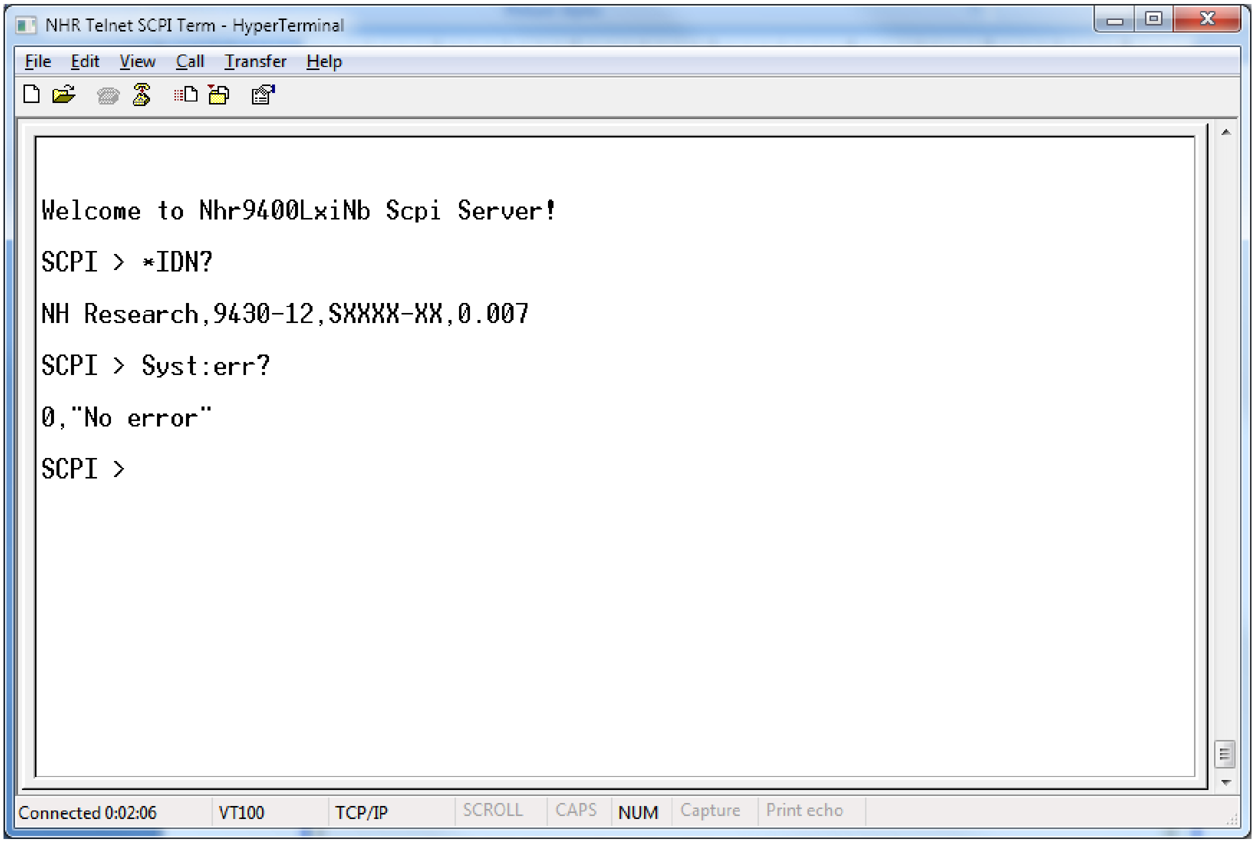
ASCII Receiving

☑ Append line feeds to incoming line ends

□ Force incoming data to 7-bit ASCII

☑ Wrap lines that exceed terminal width

Now you are ready to interact with the Module.


NHR Telnet SCPI Term - HyperTerminal
File Edit View Call Transfer Help
Connected 0:02:06
VT100
TCP/IP
SCROLL
CAPS
NUM
Capture
Print echo

```
Welcome to Nhr9400LxiNb Scpi Server!
SCPI > *IDN?
NH Research,9430-12,SXXXX-XX,0.007
SCPI > Syst:err?
0,"No error"
SCPI >
```

Basic Programming Sequence

Here is a typical programming sequence.

1. Confirm the hardware is communicating.
    a. see *IDN?
2. Consider resetting the module to put the hardware in a known state.
    a. see *RST
3. Configure the hardware according to your requirements (number of phases, AC or DC, front or rear connectors).
    a. see CONF:MODE
    b. see CONF:HW:CONN
4. Select the logical instrument to control.
    a. see INST:NSEL or INST:SEL
5. Configure the instrument according to your requirements (auto transformer and operating mode)
    a. see CONF:INST:ATR
    b. see CONF:INST:OPER
6. Set the output to ON.
    a. see [SOUR:]OUTP[:ON]
7. Set the current limit as desired.
    a. see [SOUR:]CURR
8. Set the power limit as desired.
    a. see [SOUR:]POW
9. Set the output voltage as desired. Note: If the current or power exceeds the limits set, the voltage will reduce as necessary (perhaps all the way to zero).
    a. see [SOUR:]VOLT
10. Measure the current.
    a. see MEAS:CURR?
11. Monitor the status.
    a. see *STB? and all status commands.

Programming Examples

TBD

Command Reference NOTES

Command or Query Scope

This device can be configured in many hardware modes. Without physically changing the hardware, it can be configured as a single DC source, a single 3-phase AC source, three individual AC sources, or 10 other configurations (see CONF:MODE). Most of the SCPI commands apply ONLY to the currently selected (INST:NSEL) logical instrument. HOWEVER, some of the commands apply to the entire system.

Not only can this device be configured as multiple logical instruments, there can be multiple simultaneous network connections to the device.

In order to understand what a command or query applies to, every command or query reference will start with one of the following statements:

- SYSTEM command (or query, or both).
  - A SYSTEM command or query applies to the entire 9400 system. That is, it does not matter what logical instrument is selected and it doesn't matter through which connection the command is issued.
  - SYSTEM commands are always LAST COMMAND WINS. That is, if there are multiple connections are being used and connection 1 sends a command and then connection 2 sends the same command, the hardware will end up set per connection 2.
- INSTRUMENT command (or query, or both).
  - An INSTRUMENT command or query applies to the currently selected logical instrument and it doesn't matter through which connection the command is issued.
  - INSTRUMENT commands are always LAST COMMAND WINS. That is, if there are multiple connections are being used and connection 1 sends a command and then connection 2 sends the same command to the SAME logical instrument, the hardware will end up set per connection 2.
- PER CONNECTION command (or query, or both).
  - A PER CONNECTION command or query is unique to the individual LAN connection. The error queue is kept for each connection so one connection cannot steal the errors from another connection.

Units

Unless otherwise stated, units are as follows:

Voltage	Volts (V)
Current	Amperes (A)
Resistance	Ohms (Ω)
True power	Watts (W)
Apparent power	Volt-Amperes (VA)
Frequency	Hertz (Hz)
Energy	Ampere-Hours (Ah), Kilowatt-Hours (kWh), Kilovolt-Ampere-Hours (kVAh)
Angles	degrees (°)
Time	seconds (s)

Common Commands

Common commands begin with an * and consist of three letters (command) IEEE 488.2 standard to perform some common interface functions. The module responds to the required common commands that control status reporting, synchronization, and internal operations. The module also responds to optional common commands that control triggers, power-on conditions, and stored operating parameters.

Common commands that are part of the status system are in the following section "Status Commands". Common commands and queries are listed alphabetically. If a command has a corresponding query that simply returns the data specified by the command, then both command and query are included under the explanation for the command. If a query does not have a corresponding command or is functionally different from the command, then the query is listed separately.

*IDN? (Identify)

SYSTEM query

This query requests the module to identify itself. It returns the data in four fields separated by commas.

Query Parameters None

Query Returns <mfg>,<model>,<sn>,<rev>

Field	**Information**
<mfg>	NH Research
<model>	4350-kw
<sn>	serial#
<rev>	firmware rev of network controller

*OPC (Operation Complete)

SYSTEM command and query

This command causes the interface to set the OPC bit (bit 0) of the Standard Event Status register when the module has completed all pending operations. (See *ESE for the bit configuration of the Standard Event Status registers.) Pending operations are complete when:

- All commands sent before *OPC have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed. Overlapped commands are executed in parallel with other commands. Commands that affect trigger actions are overlapped with subsequent commands sent to the module. The *OPC command provides notification that all overlapped commands have been completed.
- All triggered actions are completed and the trigger system returns to the idle state. *OPC does not prevent processing of subsequent commands but Bit 0 will not be set until all pending operations are completed. The query causes the interface to place an ASCII "1" in the Output Queue when all pending operations are completed.

Command Parameters None

Query Parameters None

Query Returns <NR1>

*RST (Reset)

SYSTEM command

This command resets the module to the factory-defined states. All variables on the device are returned to their default values. All errors are cleared if possible.

Command Parameters None

*SRE (Service Request Enable)

SYSTEM command and query

This command sets the condition of the Service Request Enable Register. This register determines which bits from the Status Byte Register (see *STB for its bit configuration) are allowed to set the Master Status Summary (MSS) bit and the Request for Service (RQS) summary bit. A 1 in any Service Request Enable Register bit position enables the corresponding Status Byte Register bit and all such enabled bits then are logically ORed to cause Bit 6 of the Status Byte Register to be set.

When the controller conducts a serial poll in response to SRQ, the RQS bit is cleared, but the MSS bit is not. When *SRE is cleared (by programming it with 0), the module cannot generate an SRQ to the controller. The query returns the current state of *SRE.

Command Parameters	0 to 255
Query Parameters	None
Query Returns	<NR1> (register binary value)

*TRG (Trigger)

SYSTEM command

This command generates a trigger to any system that has BUS selected as its source (for example, TRIG:SOUR BUS). The command has the same effect as the Group Execute Trigger (<GET>) command.

Command Parameters None

*TST? (Selftest)

SYSTEM query

This query causes the module to reset all channels and run a selftest. All errors are cleared if possible except those generated during selftest. This command will generate PROM, Firmware, Configuration and Calibration Errors as required. If there is no Calibration Error, calibration data of the specific channel is copied from FLASH to RAM and used. Returns a 0 if no errors, a 1 if any errors. Use SYST:ERR? to retrieve errors.

Query Parameters	None
Query Returns	<NR1> 0 indicates the module has passed selftest. Non-zero indicates an error.

*WAI

SYSTEM command

This command instructs the module not to process any further commands until all pending operations are completed. Pending operations are complete when:

- All commands sent before *WAI have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed.

Overlapped commands are executed in parallel with other commands. Commands that affect input voltage or state, relays, and trigger actions are overlapped with subsequent commands sent to the module. The *WAI command prevents subsequent commands from being executed before any overlapped commands have been completed.

- All triggered actions are completed and the trigger system returns to the idle state.

*WAI can be aborted only by sending the module a GPIB DCL (Device Clear) command.

Command Parameters None

Status Commands

These commands program allow control and monitoring of the status system. The status system consists of a number of registers as well as an error queue.

An application may select the events that shall cause a service request at the mandatory SCPI status data structure level, rather than excluding those that are of no interest to the application. Further, there is no requirement on the application to set up the device-dependent structures for normal device operation. For this concept to be extended to encompass the IEEE 488.2-mandated status data structures, the following sequence of commands is required when initializing the device:

*CLS – Clears all event status registers and queues

*SRE 0 – Clears the IEEE 488.2-mandated service request enable register

*ESE 0 – Clears the IEEE 488.2-mandated standard event status enable register

STATus:PRESet – Presets all other registers and queues

The table below indicates the effects of various commands upon the status data structures in a device.

	SCPI Enable Registers	SCPI Event Registers	SCPI Error/Event Queue Enable	SCPI Error/Event Queue	IEEE 488.2 Registers ESE SRE	IEEE 488.2 Registers SESR STB
*RST	none	none	none	none	none	none
*CLS	none	clear	none	clear	none	clear
power-on	preset#	clear#	preset#	clear#	clear#	clear#
STATus:PRESet	preset	none	preset	none	none	none

# Occurs if the power-on state clear flag is true. Effect changes to none if the power-on state clear flag is false.

The following table defines the effect of STATus:PRESet

Register	Filter/enable	PRESet value =
OPERational	ENABle	0's
QUEStionable	ENABle	0's

# Requires either PTR, NTR, or both to be set.

Status Registers

If the bit is marked with:

* indicates not defined in SCPI standard (available to designer).
** indicates different from SCPI standard.

Status Byte Register

(see *STB?)

Bit	Status	Description	Hex Value	Dec Value

0*	Busy	Module is busy and NOT able to process any command	0x01	1
1*	Remote	Module is in remote mode	0x02	2
2*	Error Queue	Error in queue, use SYST:ERR?	0x04	4
3	QUES	Questionable status summary. See questionable status register.	0x08	8
4	MAV	Message available	0x10	16
5	ESB	Event status byte summary. See event status register.	0x20	32
6	RQS	Request for service.	0x40	64
7	OPER	Operation event summary. See operation event register. Set for the selected instrument.	0x80	128

Event Status Register

(see *ESR?)

Bit	Status	Description	Hex Value	Dec Value
0	OPC	Operation Complete	0x01	1
1	RQC	Request Control	0x02	2
2	QYE	Query Error	0x04	4
3	DDE	Device Dependent Error	0x08	8
4	EXE	Execution Error	0x10	16
5	CME	Command Error	0x20	32
6	URQ	User Request	0x40	64
7	PON	Power On	0x80	128

Operation Register (instrument specific)

(seeSTATus:OPERation[:EVENt]?

Bit	Status	Description	Hex Value	Dec Value
0**	Ready	Module is powered and ready	0x0001	1
1**	Execution	Waiting for phase angle	0x0002	2
2	Ranging	Channel range change in progress	0x0004	4
3	Sweeping	Channel slew in progress	0x0008	8
4	Measuring	Channel measurement in progress	0x0010	16

5	Waiting For Trig	Module waiting for trigger			0x0020	32
6**	Trigger Detected	Module Trigger was being waited on and was received			0x0040	64
7**	Phase Locked	Channel is phase locked to phase A			0x0080	128
8*	Output Enabled	Channel output relays are closed			0x0100	256
9*	Active Mode bit0	Off 0	V Control 1	I Control 0	0x0200	512
10*	Active Mode bit1	0	0	1	0x0400	1024
11*	Active Mode bit2	0	0	0	0x0800	2048
12*	Sinking	Channel is sinking current (energy going to grid)			0x1000	4096
13**	Macro Ready	Macro loaded and ready to execute			0x2000	8192
14	Program Running	Module macro in progress			0x4000	16384
15	Reserved	Reserved			0x8000	32768

Questionable Status Register

(see STATus:QUEStionable[:EVENt]?)

Bit	Status	Description	Hex Value	Dec Value
0			0x0001	1
1			0x0002	2
2			0x0004	4
3			0x0008	8
4			0x0010	16
5			0x0020	32
6			0x0040	64
7			0x0080	128
8	Calibration	Calibration checksum error	0x0100	256
9*	Firmware	Firmware checksum error	0x0200	512
10*	Program Exec	Program executing from flash	0x0400	1024
11*			0x0800	2048
12*			0x1000	4096
13			0x2000	8192
14			0x4000	16384
15	Reserved		0x8000	32768

*CLS (Clear Status)

SYSTEM command

This command clears the error message queue and the following registers: Status Byte, Event Status, and Questionable Status.

Command Parameters	None

*ESE (Event Status Enable)

SYSTEM command and query

This command programs the Standard Event Status Enable register bits. The programming determines which events of the Standard Event Status Event register (see *ESR?) are allowed to set the ESB (Event Summary Bit) of the Status Byte register. A "1" in the bit position enables the corresponding event. All of the enabled events of the Standard Event Status Event Register are logically ORed to cause the Event Summary Bit (ESB) of the Status Byte Register to be set. The query reads the Standard Event Status Enable register.

Command Parameters	<NR1> 0 to 255
Query Parameters	None
Query Returns	<NR1>

*ESR? (Event Status Register)

SYSTEM query

This query reads the Standard Event Status Event register. Reading the register clears it. The bit configuration of this register is the same as the Standard Event Status Enable register (see *ESE).

Query Parameters	None
Query Returns	<NR1> (register value)

*STB? (Status Byte Register)

SYSTEM query

This query reads the Status Byte register, which contains the status summary bits and the Output Queue MAV bit. Reading the Status Byte register does not clear it. The input summary bits are cleared when the appropriate event registers are read (see chapter 3 under "Programming the Status Registers" for more information). A serial poll also returns the value of the Status Byte register, except that bit 6 returns Request for Service (RQS) instead of Master Status Summary (MSS). A serial poll clears RQS, but not MSS. When MSS is set, it indicates that the module has one or more reasons for requesting service.

Query Parameters	None
Query Returns	<NR1> (register value)

STATus:OPERation[:EVENt]?

INSTRUMENT query

This query returns the value of the Operation Event register. The Event register is a read-only register that holds (latches) all operation status events that pass into it. Reading the Operation Event register clears it.

Query Parameters	None
Query Returns	<NR1> (register value)

STATus:OPERation:CONDition?

SYSTEM query

This query returns the value of the Operation Condition register. That is a read-only register that holds the real-time (unlatched) operational status.

Query Parameters	None
Query Returns	<NR1> (register value)

STATus:OPERation:ENABle

INSTRUMENT command and query

This command and its query set and read the value of the Operation Enable register. This register is a mask for enabling specific bits from the Operation Event register to set the operation summary bit (OPER) of the Status Byte register. The operation summary bit is the logical OR of all enabled Operation Event register bits.

Command Parameters	0 to 32767
Query Parameters	None
Query Returns	<NR1> (register value)

STATus:PRESet

SYSTEM command

This command sets the value of all the enable registers (operation and questionable) to 0 (ALL disabled).

The PRESet command affects only the enable registers, PRESet does not affect either the “status byte” or the “standard event status” as defined by IEEE 488.2. PRESet does not clear any of the event registers or any item from the error/event queue. The *CLS command is used to clear all event registers and queues in the device status-reporting mechanism. For the SCPI mandatory status data structures, the PRESet command sets the enable registers to 0’s.

Command Parameters	None

STATus:QUEStionable[:EVENt]?

SYSTEM query

This query returns the value of the Questionable Event register. The Event register is a read-only register that holds (latches) all events that pass into it. Reading the Questionable Event register clears it.

Query Parameters	None
Query Returns	<NR1> (register value)

STATus:QUEStionable:CONDition?

SYSTEM query

This query returns the value of the Questionable Condition register. That is a read-only register that holds the real-time (unlatched) questionable status.

Query Parameters	None
Query Returns	<NR1> (register value)

STATus:QUEStionable:ENABle

SYSTEM command and query

This command sets or reads the value of the Questionable Enable register. This register is a mask for enabling specific bits from the Questionable Event register to set the questionable summary (QUES) bit of the Status Byte register. This bit (bit 3) is the logical OR of all the Questionable Event register bits that are enabled by the Questionable Status Enable register.

Command Parameters	0 to 32767
Query Parameters	None
Query Returns	<NR1> (register value)

SYSTem:ERRor?

PER-CONNECTION query

This query returns the next error number followed by its corresponding error message string from the remote programming error queue. The queue is a FIFO (first-in, first-out) buffer that stores errors as they occur. As it is read, each error is removed from the queue. When all errors have been read, the query returns “0, No Error”. If more errors are accumulated than the queue can hold, the last error in the queue is “-350, Too Many Errors”.

Query Parameters	None
Query Returns	<NR1>, <SRD>

CALibration Subsection

CAUTION: Calibration should only be attempted by qualified personnel.

CALibration:DATE?

SYSTEM query

This query returns the last date the calibration factors were saved to flash.

Query Parameters	None
Query Returns	<NR1>,<NR1>,<NR1>,<NR1>,<NR1>,<NR1> representing year, month, day, hour, minute, second

CALibration:RANGe

INSTRUMENT command and query

This command sets the voltage and current range of the module. The difference is that this allows setting the extreme low current ranges for the purposes of calibration.

Range values are set as an integer with the sum of the current bit value and the voltage bit value.

Voltage Range	Bit Value
Low Range	0
High Range	1
Current Range	
Mid Range 6A	0
High Range 30A	256
Low Range 1A	512
Extra Low Range 0.01A	768

Command Parameters	<NR1>
Query Parameters	None
Query Returns	<NR1>

CALibration:CURRent:RAM

INSTRUMENT command and query

This command sets or gets the current calibration factors for the currently selected instrument for the currently selected mode (AC or DC).

<range> is 0 = high-range, 1 = mid-range, 2 = low-range, 3 = extra-low range
Note: "INSTrument:CAPabilities:CURRent:RANGe:LIST?" will return the high-range and mid-range. Low range is 1A * "INSTrument:CAPabilities:SYSTem:CHASsis?" and extra-low range is 0.01A * "INSTrument:CAPabilities:SYSTem:CHASsis?".

Command Parameters	<range>,<MeasOffset>,<MeasGain>,<SetOffset>,<SetGain>
Query Parameters	<range>
Query Returns	<MeasOffset>,<MeasGain>,<SetOffset>,<SetGain>

CALibration:CURRent:FLASh?

INSTRUMENT query

This query gets the current calibration factors stored in flash for the currently selected instrument for the currently selected mode (AC or DC).

<range> is 0 = high-range, 1 = mid-range, 2 = low-range, 3 = extra-low range
Note: “INSTrument:CAPabilities:CURRent:RANGe:LIST?” will return the high-range and mid-range. Low range is 1A * “INSTrument:CAPabilities:SYSTem:CHASsis?” and extra-low range is 0.01A * “INSTrument:CAPabilities:SYSTem:CHASsis?”.

Query Parameters	<range>
Query Returns	<MeasOffset>,<MeasGain>,<SetOffset>,<SetGain>

CALibration:CURRent:RESet

INSTRUMENT command

This command resets the current calibration factors to the values stored in flash for the currently selected instrument for the currently selected mode (AC or DC).

<range> is 0 = high-range, 1 = mid-range, 2 = low-range, 3 = extra-low range
Note: “INSTrument:CAPabilities:CURRent:RANGe:LIST?” will return the high-range and mid-range. Low range is 1A * “INSTrument:CAPabilities:SYSTem:CHASsis?” and extra-low range is 0.01A * “INSTrument:CAPabilities:SYSTem:CHASsis?”.

Command Parameters	<range>

CALibration:VOLTage:RAM

INSTRUMENT command and query

This command sets or gets the voltage calibration factors for the currently selected instrument for the currently selected mode (AC or DC).

<range> is 0 = low-range, 1 = high-range
Note: “INSTrument:CAPabilities:VOLTage:RANGe:LIST?” will return the high-range and low-range.

Command Parameters	<range>,<MeasOffset>,<MeasGain>,<SetOffset>,<SetGain>
Query Parameters	<range>
Query Returns	<MeasOffset>,<MeasGain>,<SetOffset>,<SetGain>

CALibration:VOLTage:FLASh?

INSTRUMENT query

This query gets the voltage calibration factors stored in flash for the currently selected instrument for the currently selected mode (AC or DC).

<range> is 0 = low-range, 1 = high-range
Note: “INSTrument:CAPabilities:VOLTage:RANGe:LIST?” will return the high-range and low-range.

Query Parameters	<range>
Query Returns	<MeasOffset>,<MeasGain>,<SetOffset>,<SetGain>

CALibration:VOLTage:RESet

INSTRUMENT command

This command resets the voltage calibration factors to the values stored in flash for the currently selected instrument for the currently selected mode (AC or DC).

<range> is 0 = low-range, 1 = high-range
Note: "INSTrument:CAPabilities:VOLTage:RANGe:LIST?" will return the high-range and low-range.

Command Parameters	<range>

CALibration:SAVE

SYSTEM command

This command saves all the calibration factors in RAM to flash.

Command Parameters	<NR1>,<NR1>,<NR1>[,<NR1>[,<NR1>[,<NR1>]]] representing year, month, day, hour, minute, second at the time the command was called.

CONFigure Subsection

The AC/DC Power Modules is a highly flexible piece of equipment. It can act as one or more AC sources, one or more DC sources, or nearly any combination

Perform the following as required to create a configuration of your choice.

1. Set the hardware configuration mode. This will determine if a channel is in AC or DC mode and whether it is paralleled with other channels. See CONFigure:HW:MODE
2. If you are configuring a channel in AC mode and you wish to use an Auto Transformer to boost the output/input voltage set it to use an autotransformer. See CONFigure:INSTrument:ATRansformer

NOTE:

is:

[:GRID] if Model 9410
SOURce if Model 9420
LOAD if Model 9430

CONFigure:HW:MODE

SYSTEM command and query

This command will set the hardware mode.

Command Parameters	0 through 12
Query Parameters	None
Query Returns	<NR1>

An AC/DC Power Module has one to three power channels installed. Each channel can operate independently or in parallel with others. In addition, the channels can operate in AC or in DC mode.

To select the appropriate hardware mode for your application, first go to the subsection for the number of channels in your unit. Then review the hardware modes appropriate to your size unit to find the mode appropriate to your application.

Unit must be OFF to send this command.

NOTE: This command will reset the hardware and may take a number of seconds. It is suggested to do a “SYST:ERR?” with at least a 5 second timeout to know when the mode change is complete.

CAUTION: switching modes may require wiring changes to your fixturing.

3 Chan		*Ch A*		*Ch B*					*Ch C*				
Mode	*Desc*	*Out*	*NSEL*	*Out*	*Parallel with A*	*Sync with*	*Phase Angle*	*NSEL*	*Out*	*Parallel with A*	*Sync with*	*Phase Angle*	*NSEL*
0	AC1, AC1, AC1 (3ph)	AC	1	AC	0	A (1)	240		AC	0	A (1)	120	
1	AC1, AC1, AC1	AC	1	AC	1	A (1)			AC	1	A (1)		
2	DC1, DC1, DC1	DC	1	DC	1	A (1)			DC	1	A (1)		
3	AC1, AC2, AC3	AC	1	AC	0	B (0)	0	2	AC	0	C (0)	0	3
4	DC1, DC2, DC3	DC	1	DC	0	B (0)		2	DC	0	C (0)		3
5	AC1, AC1, AC3 (2ph)	AC	1	AC	0	A (1)	180		AC	0	C (0)	0	3
6	AC1, AC1, DC3 (2ph)	AC	1	AC	0	A (1)	180		DC	0	C (0)		3
7	AC1, AC1, AC3	AC	1	AC	1	A (1)			AC	0	C (0)	0	3
8	AC1, AC1, DC3	AC	1	AC	1	A (1)			DC	0	C (0)		3
9	AC1, AC2, DC3	AC	1	AC	0	B (0)	0	2	DC	0	C (0)		3
10	AC1, DC2, DC3	AC	1	DC	0	B (0)		2	DC	0	C (0)		3
11	DC1, DC1, AC3	DC	1	DC	1	A (1)			AC	0	C (0)	0	3
12	DC1, DC1, DC3	DC	1	DC	1	A (1)			DC	0	C (0)		3

2 Chan		*Ch A*		*Ch B*				
Mode	*Desc*	*Out*	*NSEL*	*Out*	*Parallel with A*	*Sync with*	*Phase Angle*	*NSEL*
0	AC1, AC1 (2ph)	AC	1	AC	0	A (1)	180	
1	AC1, AC1	AC	1	AC	1	A (1)		
2	DC1, DC1	DC	1	DC	1	A (1)		
3	AC1, AC2	AC	1	AC	0	B (0)	0	2
4	DC1, DC2	DC	1	DC	0	B (0)		2
5	AC1, DC2	AC	1	DC	0	B (0)		2

1 Chan		*Ch A*	
Mode	*Desc*	*Out*	*NSEL*
0	AC1	AC	1
1	DC1	DC	1

CONFigure:HW:MODE:VALid

SYSTEM query

This query returns a valid flag for the hardware mode parameter. The query returns a 1 to indicate the mode is valid and returns a 0 to indicate it is not a valid mode. Hardware mode is not changed with this query command. Use CONFigure:HW:MODE to set a mode.

Query Parameters 0 through 12
Query Returns 0 | 1

CONFigure:INSTrument:ATRansformer

INSTRUMENT command and query

This command will set the selected instrument for connection to an auto transformer if a non-zero ratio is set, or disables auto-transformer operation if a ratio of 0.0 is set. The auto transformer will automatically be switched in for AC operation, and bypassed for DC operation.

IMPORTANT: Execute CONFigure:MODE command BEFORE this command.

IMPORTANT: If two or more channels are paralleled, each MUST have the same auto transformer configuration. For example, if you are running in mode 5 (channel 1 & 2 paralled in AC mode), you must have an autotransformer on channel #1 AND an identical auto transformer on channel #2.

Command Parameters	<ratio> is an <NR2>: 0.0 if NO auto transformer connected, otherwise the ratio SEC:PRI as a float. If it is a 1.2:1 enter 1.2.
Query Parameters	None
Query Returns	<NR2>

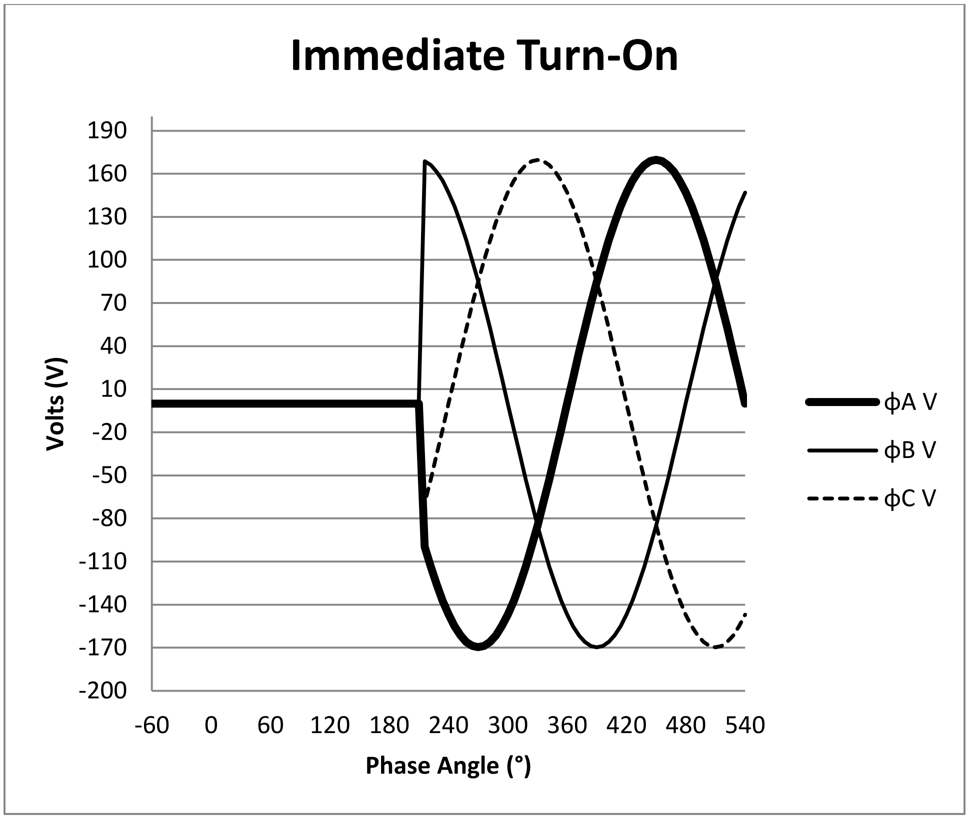
CONFigure:INSTrument:SYNChronous

INSTRUMENT command and query

If false, all settings are applied immediately.

Changes to the selected logical instrument are applied when the command is received. There is no regard to the phase angle. If the selected instrument is a multi-phase logical instrument, the outputs will be turned on or off simultaneously at the point the command is received.

For example if the command was received at about 215 degrees (on A phase) the output would look like this:

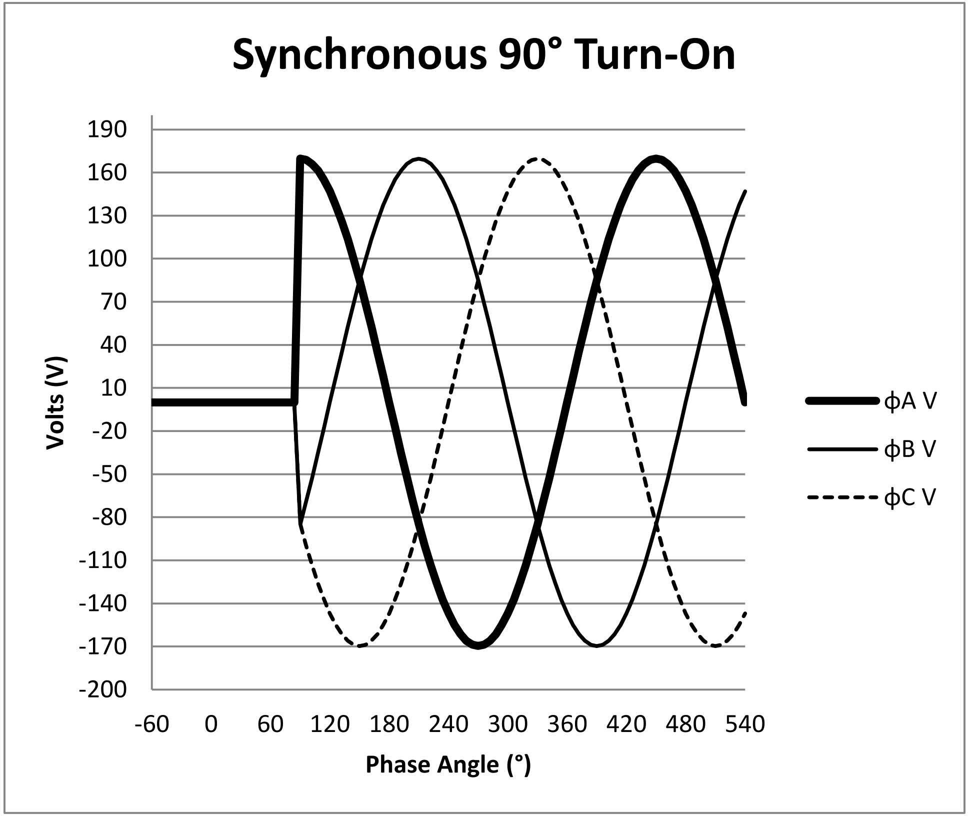


If true, settings are applied as follows:

In **DC mode**, this command is ignored and the instrument will apply changes when the command is received.

In **AC mode**: If true, and the selected instrument is a single phase, the changes will apply at the next phase angle. If the selected instrument is a multi-phase logical instrument, the changes will apply synchronous with phase A according to phase A's next phase angle as appropriate.

For example, with CONF:INST:SYNC 1 and a three-phase configuration: if AC1 is set to turn on at 90 degrees, the command "VOLT 120" will look like this:



Command Parameters	0 | 1 | YES | NO | TRUE | FALSE | ON | OFF
Query Parameters	None
Query Returns	0 | 1

DATA Subsection

DATA[:DATA]

SYSTEM command and query

This command sets data values in the controller for use in defining an arbitrary waveform shape. To define an arbitrary waveform shape 360 values representing the waveform at each degree must be defined. Values must be between -1 and +1. The waveform is divided into four quadrants: Q1 is from 1°to 90°, Q2 is from 91° to 180°, Q3 is from 181° to 270°, Q4 is from 271° to 360°. Each quadrant’s set of values is set individually. This command overwrites the previous waveform quadrant data in memory (no error is generated). Use the DATA:LOAD command to write the complete full cycle waveform data memory to the controller. Use the DATA:LOAD:APHase, BPHase, or CPHase query of desired waveform data from the controller prior to query for quadrant data values.

Command Parameters	Q1 | Q2 | Q3 |Q4 ,<NR2>,<NR2>,<NR2> … ,<NR2> (90 float values)
Query Parameters	Q1 | Q2 | Q3 | Q4
Query Returns	<NR2>,<NR2>,<NR2> … ,<NR2> (90 float values)

DATA:LOAD[:ALL]
DATA:LOAD:APHase, BPHase, or CPHase

INSTRUMENT command and query

This command writes the volatile data values to the controller for use in defining an arbitrary waveform shape. A query will retrieve the waveform shape from the user memory location for reading with the DATA:DATA query. A return value of 1 indicates success in reading the waveform shape data from the controller and placing it in the DATA:DATA buffer. A return value of 0 indicates waveform shape data was not read from the controller. The query form requires the use of the APHase, BPHase, or CPHase command.

Command Parameters	USER[1] | USER2 | USER3
Query Parameters	USER[1] | USER2 | USER3
Query Returns	0 | 1

DIAGnostic Subsection

DIAGnostic:ERRor:COMMand?

SYSTEM query

This query will retrieve a saved copy of the DSP Common Purpose Error Registers and the Channel Specified Error Registers at the time of the first error since the last Reset or Clear Errors command. If no errors have occurred since the last Reset or Clear Errors command, all 0's are returned.

Query Parameters None

Query Returns <error command>,<DSP common error register>,<DSP error register for chan 1>,<DSP error register for chan 2>, >,<DSP error register for chan 3>

DIAGnostic:NHR:COMMand

SYSTEM command and query

This command allows sending a low-level command to the DSP controller. The query form sends a command and retrieves the response.

Command Parameters <command> is the command DWORD in the form 0xnnnn
<param> is the parameter: 0xnnnn for DWORD, n.nnn for float

Examples DIAG:NHR:COMM 0x20,0x1,0.123,0.0,0.0

Query Parameters See command parameters

Query Returns <num params returned> [,<DWORD form> (<float form>)]...

Examples DIAG:NHR:COMM ? 0x21,0x1
returns "3,0x3F9D70A4 (1.230000),0x00000000 (0.000000),0x00000000 (0.000000)"

DIAGnostic:NHR:MEMory:READ?

SYSTEM query

This query will retrieves the contents at the specified memory location.

Query Parameters <offset>,<count>
where
<offset> is the offset in bytes into the dual-port memory.
<count> is number of bytes to read
--- OR ---
OPERBUF,<chan#>
where
<chan#> determines which OPERBUF channel to return.

Query Returns list of values

DIAGnostic:NHR:SIMulate?

SYSTEM query

This query will return a 0 if running on real hardware, 1 if running on the Netburner development board.

Query Parameters None

Query Returns 0 | 1

DIGital Subsystem

These commands control the general purpose digital port on the module.

DIGital:INPut?

SYSTEM query

This query gets the state of the general purpose digital input port on the specified module.

Query Parameters	None
Query Returns	<NR1>

DIGital:INPut:COUNt?

SYSTEM query

This query gets the number of general purpose digital input bits on the input port.

Query Parameters	None
Query Returns	<NR1>

DIGital:OUTPut

SYSTEM command and query

This command sets the state of the general purpose digital output port on the specified module.

Command Parameters	Value to write to the port
Query Parameters	None
Query Returns	<NR1>

DIGital:OUTPut:COUNt?

SYSTEM query

This query gets the number of general purpose digital output bits on the output port.

Query Parameters	None
Query Returns	<NR1>

INSTrument Subsystem

INSTrument:CAPabilities:

INSTRUMENT query

INSTrument:CAPabilities:SYSTem:CHANnels?

This query will retrieve the number of channels detected. There are at most 3 channels.

Query Parameters	None
Query Returns	<NR1>

INSTrument:CAPabilities:SYSTem:CHASsis?

This query will retrieve the number of chassis detected. There are at most 6 chassis.

Query Parameters	None
Query Returns	<NR1>

INSTrument:CAPabilities:APERture:MINimum?
INSTrument:CAPabilities:APERture:MAXimum?

These queries will retrieve the absolute minimum and maximum aperture setting for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:CURRent:MINimum?
INSTrument:CAPabilities:CURRent:MAXimum?

These queries will retrieve the absolute minimum and maximum current setting for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:CURRent:MEASurement:MINimum?
INSTrument:CAPabilities:CURRent:MEASurement:MAXimum?

These queries will retrieve the absolute minimum and maximum currents for measuring.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:CURRent:MEASurement:RANGe:MINimum?
INSTrument:CAPabilities:CURRent:MEASurement:RANGe:MAXimum?

These queries will retrieve the active range's minimum and maximum currents for measuring.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:CURRent:RANGe:LIST[:MAXimum]?
INSTrument:CAPabilities:CURRent:RANGe:LIST:MINimum?

This query will retrieve a list of minimum or maximum currents for each range of the Module.

Query Parameters	None
Query Returns	<NR2>{,<NR2>}

INSTrument:CAPabilities:CURRent:RANGe:MINimum?
INSTrument:CAPabilities:CURRent:RANGe:MAXimum?

These queries will retrieve the active range’s minimum and maximum current setting for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:CURRent:SLEW:MINimum?
INSTrument:CAPabilities:CURRent:SLEW:MAXimum?

These queries will retrieve the absolute minimum and maximum current slew rate for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:CURRent:SLEW:RANGe:MINimum?
INSTrument:CAPabilities:CURRent:SLEW:RANGe:MAXimum?

These queries will retrieve the active range’s minimum and maximum current slew rate for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:FREQuency:MINimum?
INSTrument:CAPabilities:FREQuency:MAXimum?

These queries will retrieve the absolute minimum and maximum frequency setting for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:FREQuency:RANGe:MINimum?
INSTrument:CAPabilities:FREQuency:RANGe:MAXimum?

These queries will retrieve the active range’s minimum and maximum frequency setting for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:FREQuency:SLEW:MINimum?
INSTrument:CAPabilities:FREQuency:SLEW:MAXimum?

These queries will retrieve the absolute minimum and maximum frequency slew rate for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:FREQuency:SLEW:RANGe:MINimum?
INSTrument:CAPabilities:FREQuency:SLEW:RANGe:MAXimum?

These queries will retrieve the active range’s minimum and maximum frequency slew rate for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:MACRo:COMMand:MAXimum?

This query will retrieve the maximum number of commands in a macro.

Query Parameters None
Query Returns <NR1>

INSTrument:CAPabilities:MACRo:DELay:MINimum?
INSTrument:CAPabilities:MACRo:DELay:MAXimum?

These queries will retrieve the absolute minimum and maximum legal values for the time interval for macros.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:PHASe:SLEW:MINimum?
INSTrument:CAPabilities:PHASe:SLEW:MAXimum?

These queries will retrieve the absolute minimum and maximum phase angle slew rate for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:PHASe:SLEW:RANGe:MINimum?
INSTrument:CAPabilities:PHASe:SLEW:RANGe:MAXimum?

These queries will retrieve the active range's minimum and maximum phase angle slew rate for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:POWer[:ALL]:MINimum?
INSTrument:CAPabilities:POWer:APHase, BPHase or CPHase:MINimum?
INSTrument:CAPabilities:POWer[:ALL]:MAXimum?
INSTrument:CAPabilities:POWer:APHase, BPHase or CPHase:MAXimum?

These queries will retrieve the absolute minimum and maximum power setting for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:POWer[:ALL]:RANGe:MINimum?
INSTrument:CAPabilities:POWer:APHase, BPHase or CPHase:RANGe: MINimum?
INSTrument:CAPabilities:POWer[:ALL]:RANGe:MAXimum?
INSTrument:CAPabilities:POWer: APHase, BPHase or CPHase:RANGe:MAXimum?

These queries will retrieve the active range's minimum and maximum power setting for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:POWer:SLEW:MINimum?
INSTrument:CAPabilities:POWer:SLEW:MAXimum?

These queries will retrieve the absolute minimum and maximum power slew rate for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:POWer:SLEW:RANGe:MINimum?

INSTrument:CAPabilities:POWer:SLEW:RANGe:MAXimum?

These queries will retrieve the active range's minimum and maximum power slew rate for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:RESistance:MINimum?
INSTrument:CAPabilities:RESistance:MAXimum?

These queries will retrieve the absolute minimum and maximum resistance setting for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:RESistance:RANGe:MINimum?
INSTrument:CAPabilities:RESistance:RANGe:MAXimum?

These queries will retrieve the active range's minimum and maximum resistance setting for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:RESistance:SLEW:MINimum?
INSTrument:CAPabilities:RESistance:SLEW:MAXimum?

These queries will retrieve the absolute minimum and maximum resistance slew rate for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:RESistance:SLEW:RANGe:MINimum?
INSTrument:CAPabilities:RESistance:SLEW:RANGe:MAXimum?

These queries will retrieve the active range's minimum and maximum resistance slew rate for the Module.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:RGAin:MINimum?
INSTrument:CAPabilities:RGAin:NOMinal?
INSTrument:CAPabilities:RGAin:MAXimum?

These queries will retrieve the absolute minimum and maximum legal values for regulation gain and the nominal (default) value.

Query Parameters	None
Query Returns	<NR2>

INSTrument:CAPabilities:SAMPle:POINts:MAXimum?

This query will retrieve the absolute maximum legal values for the number of samples.

Query Parameters	None
Query Returns	<NR1>

INSTrument:CAPabilities:SAMPle:FREQuency:MAXimum?
INSTrument:CAPabilities:SAMPle:FREQuency:MAXimum?

This query will retrieve the absolute minimum and maximum legal for the sample rate in Hz

Query Parameters None
Query Returns <NR1>

INSTrument:CAPabilities:TRIP:MINimum?
INSTrument:CAPabilities:TRIP:MAXimum?
These queries will retrieve the absolute minimum and maximum trip time setting for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:VOLT[:ALL]:MINimum?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase: MINimum?
INSTrument:CAPabilities:VOLT[:ALL]:MAXimum?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MAXimum?
These queries will retrieve the absolute minimum and maximum voltage setting for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:VOLT[:ALL]:MEASurement:MINimum?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MEASurement: MINimum?
INSTrument:CAPabilities:VOLT[:ALL]:MEASurement:MAXimum?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MEASurement:MAXimum?
These queries will retrieve the absolute minimum and maximum voltages for measuring.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:VOLT[:ALL]:MEASurement:RANGe:MINimum?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MEASurement:RANGe: MINimum?
INSTrument:CAPabilities:VOLT[:ALL]:MEASurement:RANGe:MAXimum?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:MEASurement:RANGe:MAXimum?
These queries will retrieve the active range's minimum and maximum voltages for measuring.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:VOLT[:ALL]:RANGe:LIST[:MAXimum]?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:RANGe:LIST[:MAXimum]?
INSTrument:CAPabilities:VOLT[:ALL]:RANGe:LIST:MINimum?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:RANGe:LIST:MINimum?
This query will retrieve a list of minimum or maximum voltages for each range of the Module.

Query Parameters None
Query Returns <NR2>{,<NR2>}

INSTrument:CAPabilities:VOLT[:ALL]:RANGe:MINimum?

INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:RANGe: MINimum?
INSTrument:CAPabilities:VOLT[:ALL]:RANGe:MAXimum?
INSTrument:CAPabilities:VOLT:APHase, BPHase or CPHase:RANGe:MAXimum?
These queries will retrieve the active range’s minimum and maximum voltage setting for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:VOLT:SLEW:MINimum?
INSTrument:CAPabilities:VOLT:SLEW:MAXimum?
These queries will retrieve the absolute minimum and maximum voltage slew rate for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:VOLT:SLEW:RANGe:MINimum?
INSTrument:CAPabilities:VOLT:SLEW:RANGe:MAXimum?
These queries will retrieve the active range’s minimum and maximum voltage slew rate for the Module.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:CF:MINimum?
INSTrument:CAPabilities:CF:MAXimum?
These queries will retrieve the absolute minimum and maximum crest factor that can be set.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:PF:MINimum?
INSTrument:CAPabilities:PF:MAXimum?
These queries will retrieve the absolute minimum and maximum power factor that can be set.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:R:MINimum?
INSTrument:CAPabilities:R:MAXimum?
These queries will retrieve the absolute minimum and maximum resistance that can be set.

Query Parameters None
Query Returns <NR2>

INSTrument:CAPabilities:L:MINimum?
INSTrument:CAPabilities:L:MAXimum?
These queries will retrieve the absolute minimum and maximum inductance that can be set.

Query Parameters None
Query Returns <NR2>

INSTrument:DEFine[:NAME]

INSTRUMENT command and query

The command form defines a logical instrument name, <identifier>, and associates it with a logical instrument number, <numeric_value>. The <identifier> is in character data format. The query form takes only <identifier> as a parameter and returns the logical instrument number. There can only be one logical instrument name associated with a given instrument number.

Command Parameters	<identifier> is the new logical instrument name in string form <numeric_value> is the associated logical instrument number in <NR1> form. Logical instrument number must be from 1 to <num channels>.
Query Parameters	<identifier> is the logical instrument name in string form
Query Returns	<NR1> the logical instrument number

INSTrument:DELete[:NAME]

INSTRUMENT command

This command disassociates <identifier>'s from the logical instrument number. After the DELete command executes, the default identifier is re-associated with the instrument number. NOTE: The default identifier cannot be deleted.

Command Parameters	<identifier> is the logical instrument name in string form

INSTrument:NAME?

INSTRUMENT query

This query returns the name of the currently selected instrument or the instrument passed in.

Query Parameters	None (for currently selected instrument) OR <NR1> (the logical instrument number)
Query Returns	the instrument name

INSTrument:NSELect

INSTRUMENT command and query

This command selects an instrument as the default. This command is used in conjunction with the INSTrument:SELect command. It serves the same purpose, except that it uses a numeric value instead of the identifier used in the INSTrument:SELect command. When queried it shall return the logical instrument number. By selecting an instrument with either INSTrument:SELect or INSTrument:NSELect and querying the selected instrument with the other command, the equivalence between the instrument number and identifier can be determined. For example, by selecting a logical instrument using the INSTrument:SELect command with its name identifier and then querying which logical instrument is selected with the INSTrument:NSELect? query, the corresponding numeric value shall be returned.

Command Parameters	<NR1> is the logical instrument number
Query Parameters	None
Query Returns	<NR1> the logical instrument number

INSTrument:SELect

INSTRUMENT command and query

This command selects an instrument as the default. This command is used in conjunction with the INSTrument:NSELect command. It serves the same purpose, except that it uses a string identifier instead of the value used in the INSTrument:NSELect command. When queried it shall return the logical instrument identifier. By selecting an instrument with either INSTrument:SELect or INSTrument:NSELect and querying the selected instrument with the other command, the equivalence between the instrument number and identifier can be determined. For example, by selecting a logical instrument using the INSTrument:SELect command with its name identifier and then querying which logical instrument is selected with the INSTrument:NSELect? query, the corresponding numeric value shall be returned.

Command Parameters	<identifier> is the logical instrument identifier in string form
Query Parameters	None
Query Returns	the logical instrument identifier

MACRo Subsection

MACRo:LEARn

SYSTEM command and query

This command sets or gets the macro learn mode state. If learn mode is TRUE, the following commands will be recorded and NOT executed:

- MEASurement:RESet commands
- INITiate
- INITiate:TRIGgered
- MACRo:WAIT[:TIME]
- MACRo:WAIT:CYCles
- MACRo:WAIT:SLEW
- MACRo:WAIT:TRIGger
- [SOURce:] Subsection Commands
- GRID: Subsection Commands
- LOAD: Subsection Commands

Set learn mode to FALSE to stop recording commands (normal operation).

Command Parameters	<boolean>
Query Parameters	None
Query Returns	<boolean>

MACRo:ABORt

SYSTEM command

Stop playback all recorded commands .

Command Parameters	None

MACRo:RUN[:ONCE]

SYSTEM command

Playback all recorded commands sent between MACRo:LEARn TRUE and MACRo:LEARn FALSE.

Command Parameters	None

MACRo:RUN:CONTinuous

SYSTEM command

Playback all recorded commands sent between MACRo:LEARn TRUE and MACRo:LEARn FALSE. When complete, start the playback again. The macro will loop continuously.

Command Parameters	None

MACRo:WAIT[:TIME]

SYSTEM command

This command sets a time delay that will apply during macro playback.

Command Parameters <seconds>

MACRo:WAIT:CYCLes:INSTrument1 (or INSTrument2, or INSTrument3)

INST command

NOTE: Only valid in AC mode. This command will pause a macro during playback until the specified number of cycles have occurred.

Command Parameters <NR1> number of cycles

MACRo:WAIT:SLEW

SYSTEM command

This command will pause the macro during playback until any active slew is complete.

Command Parameters None

MACRo:WAIT:TRIGger

SYSTEM command

This command will wait for a trigger during macro playback.

Command Parameters None

TBD:

MACRo:MEASure:INITiate:ALL
MACRo:MEASure:INITiate:INST1, 2, or 3
MACRo:MEASure:RESet:ALL:ALL
MACRo:MEASure:RESet:ALL:INST1, 2, or 3
MACRo:MEASure:RESet:ENERgy:ALL
MACRo:MEASure:RESet:ENERgy:INST1, 2, or 3
MACRo:MEASure:RESet:PEAKs:ALL
MACRo:MEASure:RESet:PEAKs:INST1, 2, or 3
MACRo:NOTCh:STARt:ANGLe:ALL
MACRo:NOTCh:STARt:ANGLe:INST1, 2, or 3
MACRo:NOTCh:STARt:ANGLe:CHAN
MACRo:NOTCh:STARt:VOLTage:ALL
MACRo:NOTCh:STARt:VOLTage:INST1, 2, or 3
MACRo:NOTCh:STARt:VOLTage:CHAN
MACRo:NOTCh:STOP:ANGLe:ALL
MACRo:NOTCh:STOP:ANGLe:INST1, 2, or 3
MACRo:NOTCh:STOP:ANGLe:CHAN
MACRo:NOTCh:STOP:VOLTage:ALL
MACRo:NOTCh:STOP:VOLTage:INST1, 2, or 3
MACRo:NOTCh:STOP:VOLTage:CHAN
MACRo:OPERation:WAVeshape:ALL
MACRo:OPERation:WAVeshape:INST1, 2, or 3

MACRo:OPERation:WAVeshape:CHAN
MACRo:OPERation:SYNC:ALL
MACRo:OPERation:SYNC:INST1, 2, or 3
MACRo:OPERation:VOLTage:ALL
MACRo:OPERation:VOLTage:INST1, 2, or 3
MACRo:OPERation:VOLTage:CHAN1, 2, or 3
MACRo:OPERation:CURRent:ALL
MACRo:OPERation:CURRent:INST1, 2, or 3
MACRo:OPERation:CURRent:CHAN1, 2, or 3
MACRo:OPERation:POWer:ALL
MACRo:OPERation:POWer:INST1, 2, or 3
MACRo:OPERation:POWer:CHAN1, 2, or 3
MACRo:OPERation:RESistance:ALL
MACRo:OPERation:RESistance:INST1, 2, or 3
MACRo:OPERation:RESistance:CHAN1, 2, or 3
MACRo:OPERation:FREQuency:ALL
MACRo:OPERation:FREQuency:INST1, 2, or 3
MACRo:OPERation:VA:ALL
MACRo:OPERation:VA:INST1, 2, or 3
MACRo:OPERation:VA:CHAN1, 2, or 3
MACRo:OUTPut:ALL
MACRo:OUTPut:INST1, 2, or 3
MACRo:PHASe:ANGLe:BPHase
MACRo:PHASe:ANGLe:CPHase
MACRo:SYNC:ANGLe:ALL
MACRo:SYNC:ANGLe:INST1, 2, or 3
MACRo:SYNC:ANGLe:CHAN1, 2, or 3
MACRo:RANGe:CURRent:ALL
MACRo:RANGe:CURRent:INST1, 2, or 3
MACRo:RANGe:VOLTage:ALL
MACRo:RANGe:VOLTage:INST1, 2, or 3
MACRo:SLEW:CURRent:ALL
MACRo:SLEW:CURRent:INST1, 2, or 3
MACRo:SLEW:FREQuency:ALL
MACRo:SLEW:FREQuency:INST1, 2, or 3
MACRo:SLEW:POWer:ALL
MACRo:SLEW:POWer:INST1, 2, or 3
MACRo:SLEW:RESistance:ALL
MACRo:SLEW:RESistance:INST1, 2, or 3
MACRo:SLEW:PF:ALL
MACRo:SLEW:PF:INST1, 2, or 3
MACRo:SLEW:PSHift:ALL
MACRo:SLEW:PSHift:INST1, 2, or 3
MACRo:SLEW:VOLTage:ALL
MACRo:SLEW:VOLTage:INST1, 2, or 3

Measurement Commands

Measurement commands consist of measurement and sense commands. Two measurement commands are available: MEASure and FETCh. MEASure triggers the acquisition of new data before returning the readings. FETCh returns previously acquired data. The input voltage and current are digitized whenever a measure command is given or whenever an acquire trigger occurs. The capture aperture is set by SENSe:SWEep:APERture. If the instrument is in AC mode, the actual aperture used will be a whole multiple of cycles where the number of cycles = aperture * Hz rounded UP to the next whole number.

Important: The hardware makes all measurements when an acquisition sequence is performed. An acquisition sequence is performed when:

- A MEAS:<function>? is sent
- An INITiate command is sent

After the acquisition sequence, all measurements are available. For example:

```
MEAS:VOLT?     // starts acquisition sequence then returns the voltage calculated during the acquisition.
FETCH:CURR?    // returns the current calculated during the SAME acquisition as voltage.
FETCH:POW?     // returns the power calculated during the SAME acquisition as voltage.
```

Sense commands control the measurement range, the acquisition sequence, and the measurement window of the module.

Prior to executing any command in the subsection, the appropriate logical instrument should be selected with INST:NSEL or INST:SEL.

Important: Units may change based on the instrument selected. Most importantly, a multi-phase instrument selected (INST:SEL AC1) is different than a single phase instrument:

Set/Measure	DC	AC Single-Phase	AC Multi-Phase
Voltage	Volts (V)	L-N Volts RMS (Vrms)	L-L Volts RMS (Vrms)
Current	Amperes (A)	L-N Amperes RMS (Arms)	L-N Amperes RMS (Arms)
Power	Watts (W)	Watts (W)	Total Watts (W)

ABORt

SYSTEM command

This command resets the list and measurement trigger systems to the Idle state. Any list or measurement that is in progress is immediately aborted. ABORt also resets the WTG bit in the Operation Condition Status register. ABORt is executed at power turn-on and upon execution of *RCL, RST, or any implied abort command (see List Commands).

Command Parameters None

INITiate[:IMMediate]

INSTRUMENT command

This command starts the measurement aperture.

Command Parameters None

MEASure:ARRay:CURRent?
MEASure:ARRay:CURRent:APHase? (or BPHase, or CPHase)
FETCh:ARRay:CURRent?
FETCh:ARRay:CURRent:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return an array containing the instantaneous input current. The array starts at index 0 if none is specified, otherwise it starts at the start index.

Waveforms are captured with every initiated measurement based on the SENSe:SWEep:APERture, SENSe:SWEep:POINts, and SENSe:SWEep:TINTerval commands.

MEAS:ARR:CURR? will start a new acquisition, FETC:ARR:CURR? will retrieve the data from a previous capture.

IMPORTANT: Any measurement initiate will capture the waveform data. To gather measurements and waveform data from the same set of samples, be sure to only initiate once. For example, the following steps would get voltage and current measurements and waveforms from the SAME data.

1. Send query "MEAS:VOLT?" which initiates a new measurement, waits for the measurement to complete and then returns the measured value.
2. Send query "FETC:CURR?" which returns the measured value for the previously initiated measurement.
3. Repeat the query "FETC:ARR:VOLT? index" to get the samples of the captured voltage waveform until all samples have been retrieved. Index should be zero for the first call. Keep adding the total number of samples to the index to retrieve the remaining data.
4. Repeat the query "FETC:ARR:CURR? index" to get the samples of the captured current waveform until all samples have been retrieved. Index should be zero for the first call. Keep adding the total number of samples to the index to retrieve the remaining data.

Query Parameters [<start index>[,<index increment>]]
<start index> is the zero-based index of the first point to fetch. Defaults to 0.
<index increment> is added to index to determine the next sample returned. Defaults to 1.

Query Returns <sample 1>{... ,<sample n>}
NOTE: returns 3.40282347e+38 if <start index> is past the end of available data.

MEASure:ARRay:VOLTage?
MEASure:ARRay:VOLTage:APHase? (or BPHase, or CPHase)
FETCh:ARRay:VOLTage?
FETCh:ARRay:VOLTage:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return an array containing the instantaneous input voltage. The array starts at index 0 if none is specified, otherwise it starts at the start index.

Waveforms are captured with every initiated measurement based on the SENSe:SWEep:APERture, SENSe:SWEep:POINts, and SENSe:SWEep:TINTerval commands.

MEAS:ARR:VOLT? will start a new acquisition, FETC:ARR:VOLT? will retrieve the data from a previous capture.

IMPORTANT: Any measurement initiate will capture the waveform data. To gather measurements and waveform data from the same set of samples, be sure to only initiate once. For example, the following steps would get voltage and current measurements and waveforms from the SAME data.

1. Send query "MEAS:VOLT?" which initiates a new measurement, waits for the measurement to complete and then returns the measured value.
2. Send query "FETC:CURR?" which returns the measured value for the previously initiated measurement.
3. Send query "SENS:SWE:POIN?" to determine the number of available samples.
4. Send query "SENS:SWE:TINT?" to determine the sample rate of the waveform data.
5. Repeat the query "FETC:ARR:VOLT? index" to get the samples of the captured voltage waveform until all samples have been retrieved. Index should be zero for the first call. Keep adding the total number of samples to the index to retrieve the remaining data.
6. Repeat the query "FETC:ARR:CURR? index" to get the samples of the captured current waveform until all samples have been retrieved. Index should be zero for the first call. Keep adding the total number of samples to the index to retrieve the remaining data.

Query Parameters [<start index>[,<index increment>]]
<start index> is the zero-based index of the first point to fetch. Defaults to 0.
<index increment> is added to index to determine the next sample returned. Defaults to 1.

Query Returns <sample 1>{... ,<sample n>}
NOTE: returns 3.40282347e+38 if <start index> is past the end of available data.

FETCh:BACKground?

INSTRUMENT query

This query returns the most recent measurements that are always being made by the hardware in the background (except while a waveform capture is being made).

Query Parameters [<instrument number>]
<instrument number> to force a specific instrument for measurement without doing an INST:NSEL, or raw measurements from a specific channel. Default is the currently selected instrument.
1 | 2 | 3 | CH1 | CH2 | CH3

Query Returns <NR2>,<NR2>,... representing:
Voltage, Current, Power, Frequency, Ampere Hour, Kilowatt Hour, Voltage Peak Min, Voltage Peak Max, Current Peak Min, Current Peak Max, True Power Peak Min, True Power Peak Max, Apparent Power

FETCh:BACKground:ALL?

INSTRUMENT query

This query returns the most recent measurements that are always being made by the hardware in the background (except while a waveform capture is being made).

Query Parameters [<instrument number>]
<instrument number> to force a specific instrument for measurement without doing an INST:NSEL, or raw measurements from a specific channel. Default is the currently selected instrument.
1 | 2 | 3 | CH1 | CH2 | CH3

Query Returns <NR2>,<NR2>,... representing:
Voltage, Current, Power, Frequency, Ampere Hour, Kilowatt Hour, Voltage Peak Min, Voltage Peak Max, Current Peak Min, Current Peak Max, True Power Peak Min, True Power Peak Max, Apparent Power, KiloVolt AmpHour, Voltage AC only RMS Min one cycle, Voltage AC only RMS Max one cycle, Current AC only RMS Min one cycle, Current AC only RMS Max one cycle, True Power Min one cycle, True Power Max one cycle, Apparent Power Min one cycle, Apparent Power Max one cycle, Time Stamp,Power Factor, Crest Factor

FETCh:BACKground:DC?

INSTRUMENT query

This query returns the most recent DC measurements that are always being made by the hardware in the background (except while a waveform capture is being made).

Query Parameters [<instrument number>]
<instrument number> to force a specific instrument for measurement without doing an INST:NSEL, or raw measurements from a specific channel. Default is the currently selected instrument.
1 | 2 | 3 | CH1 | CH2 | CH3

Query Returns <NR2>,<NR2> representing:
DC Voltage, DC Current

FETCh:BACKground:TEMPerature?

SYSTEM query

This query returns the most recent heatsink temperatures that are always being monitored by the hardware in the background. NOTE: this is NOT instrument specific. That is, it doesn't matter which logical instrument is selected (INST:SEL).

Query Parameters None
Query Returns <NR2>,<NR2>,<NR2>,<NR2>
representing: Grid heatsink temperature, channel 1's hottest heatsink temperature, channel 2's hottest heatsink temperature, channel 3's hottest heatsink temperature.

MEASure:CF?
MEASure:CF:APHase? (or BPHase, or CPHase)
FETCh:CF?
FETCh:CF:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return theCREST FACTOR during the aperture. Note: This is a measurement that only makes sense in AC mode, however, the calculated value will be returned regardless of the mode.

Calculated as (larger amplitude of CurrentPeakMax and CurrentPeakMin) / Amps AC RMS. (returns 1000000 if Amps AC RMS is < 0.001).

Query Parameters	None
Query Returns	<NR2>

MEASure:CURRent?
MEASure:CURRent:APHase? (or BPHase, or CPHase)
FETCh:CURRent?
FETCh:CURRent:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return the average current (RMS for AC, average for DC) during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:CURRent:PEAK:MAXimum?
MEASure:CURRent:PEAK:MAXimum:APHase? (or BPHase, or CPHase)
FETCh:CURRent:PEAK:MAXimum?
FETCh:CURRent:PEAK:MAXimum:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return the maximum current during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:CURRent:PEAK:MINimum?
MEASure:CURRent:PEAK: MINimum:APHase? (or BPHase, or CPHase)
FETCh:CURRent:PEAK:MINimum?
FETCh:CURRent:PEAK: MINimum:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return the minimum current during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:RESet[:ALL]

INSTRUMENT command

This command resets all accumulated measurements:

Ampere Hour, Kilowatt Hour, Voltage Peak Min, Voltage Peak Max, Current Peak Min, Current Peak Max, True Power Peak Min, True Power Peak Max, KiloVolt AmpHour, Voltage AC only RMS Min one cycle, Voltage AC only RMS Max one cycle, Current AC only RMS Min one cycle, Current AC only RMS Max one cycle, True Power Min one cycle, True Power Max one cycle, Apparent Power Min one cycle, Apparent Power Max one cycle

Command Parameters None

MEASure:RESet:ENERgy

INSTRUMENT command

This command resets all accumulated energy measurements:

Ampere Hour, Kilowatt Hour, KiloVolt AmpHour

Command Parameters None

MEASure:RESet:PEAKs

INSTRUMENT command

This command resets all accumulated peak measurements:

Voltage Peak Min, Voltage Peak Max, Current Peak Min, Current Peak Max, True Power Peak Min, True Power Peak Max, Voltage AC only RMS Min one cycle, Voltage AC only RMS Max one cycle, Current AC only RMS Min one cycle, Current AC only RMS Max one cycle, True Power Min one cycle, True Power Max one cycle, Apparent Power Min one cycle, Apparent Power Max one cycle

Command Parameters None

MEASure:PF?
MEASure:PF:APHase? (or BPHase, or CPHase)
FETCh:PF?
FETCh:PF:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return thePOWER FACTOR during the aperture. Note: This is a measurement that only makes sense in AC mode, however, the calculated value will be returned regardless of the mode.

Calculated as True Power / Apparent Power. (returns 1000000 if Apparent Power is < 0.001).

Query Parameters None
Query Returns <NR2>

MEASure:POWer[:TRUE]?
MEASure:POWer[:TRUE]:APHase? (or BPHase, or CPHase)
FETCh:POWer[:TRUE]?

FETCh:POWer[:TRUE]:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return the average true power during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:POWer:APParent?
MEASure:POWer:APParent:APHase? (or BPHase, or CPHase)
FETCh:POWer:APParent?
FETCh:POWer:APParent:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return the average apparent power during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:POWer:APParent:PEAK:MAXimum?
MEASure:POWer:APParent:PEAK:MAXimum:APHase? (or BPHase, or CPHase)
FETCh:POWer:APParent:PEAK:MAXimum?
FETCh:POWer:APParent:PEAK:MAXimum:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return:

- For AC: the maximum apparent power for any cycle during the aperture.
- For DC: the absolute maximum instantaneous power during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:POWer:APParent:PEAK:MINimum?
MEASure:POWer:APParent:PEAK:MINimum:APHase? (or BPHase, or CPHase)
FETCh:POWer:APParent:PEAK:MINimum?
FETCh:POWer:APParent:PEAK:MINimum:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return:

- For AC: the minimum apparent power for any cycle during the aperture.
- For DC: the absolute minimum instantaneous power during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:POWer:PEAK:MAXimum?
MEASure:POWer:PEAK:MAXimum:APHase? (or BPHase, or CPHase)

FETCh:POWer:PEAK:MAXimum?
FETCh:POWer:PEAK:MAXimum:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return:

- For AC: the maximum power for any cycle during the aperture.
- For DC: the absolute maximum instantaneous power during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:POWer:PEAK:MINimum?
MEASure:POWer:PEAK:MINimum:APHase? (or BPHase, or CPHase)
FETCh:POWer:PEAK:MINimum?
FETCh:POWer:PEAK:MINimum:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return:

- For AC: the minimum power for any cycle during the aperture.
- For DC: the absolute minimum instantaneous power during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:VOLTage?
MEASure:VOLTage:APHase? (or BPHase, or CPHase)
FETCh:VOLTage?
FETCh:VOLTage:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return the average voltage (RMS for AC, average for DC) during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:VOLTage:PEAK:MAXimum?
MEASure:VOLTage:PEAK:MAXimum:APHase? (or BPHase, or CPHase)
FETCh:VOLTage:PEAK:MAXimum?
FETCh:VOLTage:PEAK:MAXimum:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return:

- For AC: the maximum RMS voltage for any cycle during the aperture.
- For DC: the absolute maximum instantaneous voltage during the aperture.

Query Parameters	None
Query Returns	<NR2>

MEASure:VOLTage:PEAK:MINimum?
MEASure:VOLTage:PEAK:MINimum:APHase? (or BPHase, or CPHase)
FETCh:VOLTage:PEAK:MINimum?
FETCh:VOLTage:PEAK:MINimum:APHase? (or BPHase, or CPHase)

INSTRUMENT query

These queries return:

- For AC: the minimum RMS voltage for any cycle during the aperture.
- For DC: the absolute minimum instantaneous voltage during the aperture.

Query Parameters	None
Query Returns	<NR2>

SENSe:SWEep:APERture

INSTRUMENT command and query

This command specifies how long a measurement aperture is desired to be. The Module will pick the sample rate. The actual number of samples may be less than that specified in SENS:SWE:POINTS. Applies to both voltage and current measurements.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

SENSe:SWEep:POINts

INSTRUMENT command and query

This command specifies the maximum number of samples. Applies to both voltage and current measurements. The query will return the number set in the command if a capture has not been initiated. After the capture, the actual number of samples captured will be returned (may be less than the number specified).

Command Parameters	<NR1>
Query Parameters	None
Query Returns	<NR1>

SENSe:SWEep:TINTerval?

INSTRUMENT query

This query returns the time interval between samples (1 / sample rate). Applies to both voltage and current measurements.

Query Parameters	None
Query Returns	<NR2>

SOURce Subsystem

These commands control the operation of the instrument.

Prior to executing any command in the subsection, the appropriate logical instrument should be selected with INSTrument:NSELect or INSTrument:SELect. The VOLTage command sets the desired voltage of the source which is limited by the CURRent and POWer commands.

Important: Units may change based on the instrument selected. Most importantly, a multi-phase instrument selected (INST:SEL AC1) is different than a single phase instrument:

Set/Measure	DC	AC Single-Phase	AC Multi-Phase
Voltage	Volts (V)	L-N Volts RMS (Vrms)	L-L Volts RMS (Vrms)
Current	Amperes (A)	L-N Amperes RMS (Arms)	L-N Amperes RMS (Arms)
Power	Watts (W)	Watts (W)	Total Watts (W)

[SOURCE:]CURRent[:ALL]

INSTRUMENT command and query

This command sets the output current that the Module will limit to.

IMPORTANT: Current is always specified as A L-N (line-to-neutral) regardless of whether the selected instrument is a multi-phase logical instrument (INST:SEL AC1) or a single phase.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]CURRent:APHase, BPHase, or CPHase

INSTRUMENT command and query

This command sets the output current that the channel will limit to.

IMPORTANT: Current is always specified as A L-N (line-to-neutral) regardless of whether the selected instrument is a multi-phase logical instrument (INST:SEL AC1) or a single phase.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]CURRent:RANGe

INSTRUMENT command and query

This command sets the current range of the module. Use "INST:CAP:CURR:RANG:LIST?" to determine available range values.

IMPORTANT: Current is always specified as A L-N (line-to-neutral) regardless of whether the selected instrument is a multi-phase logical instrument (INST:SEL AC1) or a single phase.

When you program a range value, the Module automatically selects the range that corresponds to the value that you program. If the value falls in a region where ranges overlap, the Module selects the range with the highest resolution.

NOTE: When this command is executed, the IMMediate, TRANsient, TRIGgered, and SLEW current settings are adjusted as follows:

If the existing settings are within the new range: No adjustment is made.

If the existing settings are outside the new range: The levels are set to the maximum value of the new range.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]CURRent:SLEW

INSTRUMENT command and query

This command sets the slew rate for all programmed changes in the output current level of the module. This command programs both positive and negative going slew rates. Although any slew rate value may be entered, the module selects a slew rate that is closest to the programmed value.

Command Parameters	0 to 9.9E37
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]CURRent:ANGLe

INSTRUMENT command and query

This command sets the angle if in AC mode.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]CURRent:CF

INSTRUMENT command and query

This command sets the crest factor if in AC mode.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]CURRent:PF

INSTRUMENT command and query

This command sets the power factor if in AC mode.

Command Parameters	<NR2>
Query Parameters	None

Query Returns	<NR2>

[SOURCE:]FREQuency

INSTRUMENT command and query

This command sets the frequency if in AC mode.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]FREQuency:SLEW

INSTRUMENT command and query

This command sets the slew rate for all programmed changes in the frequency. This command programs both positive and negative going slew rates. Although any slew rate value may be entered, the module selects a slew rate that is closest to the programmed value.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>
Query Returns	<NR2>

[SOURCE:]FUNCtion[:SHAPe][:ALL]
[SOURCE:]FUNCtion[:SHAPe]:APHase, BPHase, or CPHase

INSTRUMENT command and query

If command includes ":ALL" it applies to the entire device. This command sets the voltage waveform shape of the output signal. STANDARD is the default voltage waveform shape. User waveform shape must be defined prior to use (see DATA Subsystem). The selected waveform shape is output using the previously selected frequency and amplitude settings.

Command Parameters	STANDARD| USER[1] | USER2 | USER3
Query Parameters	None
Query Returns*	STANDARD | USER1 | USER2 | USER3

***Note:** SOURCE:FUNC:SHAP:ALL? returns values for all phases of a multi-phase instrument separated by commas, e.g. USER1, USER1, USER1 for a 3-phase AC Source with all three phases set for USER1 waveform shape.

[SOURCE:]LOAD:MODE

INSTRUMENT command and query

This command sets the load mode.

Command Parameters	NORMal | CR | RL
Query Parameters	None
Query Returns	NORMal | CR |RL

[SOURCE:]LOAD:SEND

INSTRUMENT command and query

This command sets the send mode.

Command Parameters	IMMediate | APPLy
Query Parameters	None
Query Returns	IMMediate | APPLy

[SOURCE:]LOAD:APPLy

INSTRUMENT command

This command applies any pending settings .

Command Parameters	None

[SOURCE:]OUTPut[:ON]
[SOURCE:]OUTPut[:ON]:ALL

INSTRUMENT command and query

If command does not include ":ALL" it applies to the selected instrument only. With ":ALL" it applies to the entire device. This command turns on or off the unit ON is defined as the output relay closed with the unit at zero volts. OFF the output relay will be open.

Command Parameters	0 | 1 | YES | NO | TRUE | FALSE | ON | OFF
Query Parameters	None
Query Returns	0 | 1

[SOURCE:]OUTPut:ABORt

INSTRUMENT command

This command aborts any slewing of voltage, current, power, or frequency.

Command Parameters	None

[SOURCE:]OUTPut:SYNC:ANGLe[:ALL]

INSTRUMENT command and query

This command ONLY applies to an instrument in AC mode (see CONFigure:MODE).

Sets the selected logical instrument to apply changes at the set phase angle when operating in CONF:INST:SYNC is 1.

Command Parameters	<NR2> 0.0 to 359.999 degrees
Query Parameters	None
Query Returns	<angle>

[SOURCE:]OUTPut:SYNC:ANGLe:APHase, BPHase, or CPHase

INSTRUMENT command and query

This command ONLY applies to an instrument in AC mode (see CONFigure:MODE).

Sets the selected logical instrument to apply changes at the set phase angle when operating in CONF:INST:SYNC is 1.

Command Parameters	<NR2> 0.0 to 359.999 degrees
Query Parameters	None
Query Returns	<angle>

[SOURCE:]OUTPut:RGAin

INSTRUMENT command and query

This command (regulation gain) sets the percent correction applied during each control loop to regulate the output. A value of 0 will inhibit the regulation feedback. The larger the value the bigger the adjustment. The value is a percent entered as a floating point where 5% is 0.05.

Command Parameters	0 through 0.30
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]PHASe[:ANGLe]

INSTRUMENT command and query

This command ONLY applies to an instrument in AC multi-phase mode (see CONFigure:MODE).

Sets the phase angle of B and C phase relative to A

Note: This setting is included in the saved hardware configuration that can be set as a power-up and reset default using the CONFigure:SAVE command.

Command Parameters	<angle A to B>, <angle A to C> (0.0 to 359.999 degrees each)
Query Parameters	None
Query Returns	<angle A to B>, <angle A to C>

[SOURCE:]PHASe:ANGLe:BPHase

INSTRUMENT command and query

This command ONLY applies to an instrument in AC multi-phase mode (see CONFigure:MODE).

Sets the phase angle of B phase relative to A

Note: This setting is included in the saved hardware configuration that can be set as a power-up and reset default using the CONFigure:SAVE command.

Command Parameters	<angle A to B>,(0.0 to 359.999 degrees each)
Query Parameters	None
Query Returns	<angle A to B>

[SOURCE:]PHASe:ANGLe:CPHase

INSTRUMENT command and query

This command ONLY applies to an instrument in AC multi-phase mode (see CONFigure:MODE).

Sets the phase angle of C phase relative to A

Note: This setting is included in the saved hardware configuration that can be set as a power-up and reset default using the CONFigure:SAVE command.

Command Parameters	<angle A to C> (0.0 to 359.999 degrees each)
Query Parameters	None
Query Returns	<angle A to C>

[SOURCE:]PHASe:SLEW

INSTRUMENT command and query

This command ONLY applies to an instrument in AC multi-phase mode (see CONFigure:MODE).

Sets the slew rate when changing the phase angle of B and C phase relative to A ([SOURce:]PHAS[:ANGLe]).

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<degrees per second>

[SOURCE:]POWer[:ALL]

INSTRUMENT command and query

This command sets the output power that the Module will limit to.

IMPORTANT: Power is always specified as total watts for the logical instrument regardless of whether the selected instrument is a multi-phase logical instrument (INST:SEL AC1) or a single phase.

Command Parameters	<NR2> 0 through maximum for range
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]POWer:APHase, BPHase, or CPHase

INSTRUMENT command and query

This command sets the output power that the channel will limit to.

Command Parameters	<NR2> 0 through maximum for range
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]POWer:SLEW

INSTRUMENT command and query

This command sets the slew rate for all programmed changes in the output power level of the module. This command programs both positive and negative going slew rates. Although any slew rate value may be entered, the module selects a slew rate that is closest to the programmed value.

Command Parameters	0 to 9.9E37
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]CURRent:PRIority

INSTRUMENT command and query

This command sets the leading or lagging priority if in AC mode.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]RESistance

INSTRUMENT command and query

This command sets the resistance if in AC mode.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]RL:RESistance

INSTRUMENT command and query

This command sets the resistance if in RL mode.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]RL:INDuctance

INSTRUMENT command and query

This command sets the inductance to be used in RL mode.

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]SAFety

INSTRUMENT command and query

This command is used to set the maximum allowable time and value, which, if exceeded, will cause the Module to shut off. Time values of <0 disable that parameter. Peak values are the maximum instantaneous absolute value and peak enable is a boolean (0 = disabled, 1 = enabled).

Command Parameters	<Min V>,<Min V Time>,<Max V>,<Max V Time>,<Max Source A>,<Max Source A Time>,<Max Sink A>,<Max Sink A Time>,<Max Source W>,<Max Source W Time>,<Max Sink W>,<Max Sink W Time>,<Peak V>,<Peak V Enable>,<Peak A>,<Peak A Enable>
Query Parameters	None
Query Returns	<Min V>,<Min V Time>,<Max V>,<Max V Time>,<Max Source A>,<Max Source A Time>,<Max Sink A>,<Max Sink A Time>,<Max Source W>,<Max

Source W Time>,<Max Sink W>,<Max Sink W Time>,<Peak V>,<Peak V Enable>,<Peak A>,<Peak A Enable>

[SOURCE:]VA

INSTRUMENT command and query

This command sets the VA .

Command Parameters	<NR2>
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]VOLTage[:ALL]

INSTRUMENT command and query

This command sets the voltage that the Module will attempt to regulate to unless limited by current or power settings.

IMPORTANT: Voltage is specified as V for DC outputs, Vrms L-N for single-phase AC outputs and Vrms L-L for a multi-phase logical instrument (INST:SEL AC1).

Command Parameters	0 through max
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]VOLTage:APHase, BPHase, or CPHase

INSTRUMENT command and query

This command sets the voltage that the phase will attempt to regulate to unless limited by current or power settings.

Command Parameters	
Query Parameters	None
Query Returns	

[SOURCE:]VOLTage:RANGe

INSTRUMENT command and query

This command sets the voltage range of the module.

IMPORTANT: Voltage is specified as V for DC outputs, Vrms L-N for single-phase AC outputs and Vrms L-L for a multi-phase logical instrument (INST:SEL AC1).

When you program a range value, the Module automatically selects the range that corresponds to the value that you program. If the value falls in a region where ranges overlap, the Module selects the range with the highest resolution.

NOTE: When this command is executed, the IMMediate, TRANsient, TRIGgered, and SLEW voltage settings are adjusted as follows:

If the existing settings are within the new range: No adjustment is made.

If the existing settings are outside the new range: The levels are set to the maximum value of the new range.

Command Parameters	0 through MAX
Query Parameters	None
Query Returns	<NR2>

[SOURCE:]VOLTage:SLEW

INSTRUMENT command and query

This command sets the slew rate for all programmed changes in the input voltage level of the module. This command programs both positive and negative going slew rates. Although any slew rate value may be entered, the module selects a slew rate that is closest to the programmed value.

IMPORTANT: Voltage/second is specified as V for DC outputs, Vrms L-N for single-phase AC outputs and Vrms L-L for a multi-phase logical instrument (INST:SEL AC1).

Command Parameters	0 to 9.9E37
Query Parameters	None
Query Returns	<NR2>

System Commands

System commands control the system-level functions of the module that are not directly related to input control or measurement functions.

SYSTem:ERRor?

PER-CONNECTION query

See Status Commands

SYSTem:LED

SYSTEM command and query

This command causes the LED on the Module to blink.

Command Parameters	0 | 1 | OFF | ON
Query Parameters	None
Query Returns	0 | 1

SYSTem:LOCal

SYSTEM command

This command places the module in local mode during SCPI operation. The front panel keys are functional.

Command Parameters None

SYSTem:REMote

SYSTEM command

This command places the module in remote mode during SCPI operation. Pressing any key on the front panel will return the front panel to the local state. This command is not necessary as ANY SCPI command (except SYSTem:LOCal) will put the front panel in remote mode.

Command Parameters None

SYSTem:RWLock

SYSTEM command

This command places the module in remote mode during SCPI operation. All front panel keys are disabled. Use SYSTem:LOCal to return the front panel to the local state.

Command Parameters None

SYSTem:VERSion?

SYSTEM query

This query returns the SCPI version number to which the module complies. The value is of the form YYYY.V, where YYYY is the year and V is the revision number for that year. Query Syntax SYSTem:VERSion?

Command Parameters	None
Query Parameters	None
Query Returns	<NR2>

SYSTem:WATChdog:INTerval

SYSTEM command and query

This command specifies how often communications must take place before the Module shuts itself off. The communication required to reset the watchdog timer (and keep the Module working) depends on the SYSTem:WATChdog:ROBust command. If robust is set to ON, only the SYSTem:WATChdog:SERVice command will reset the timer. If robust is set to OFF (the default), any communication will reset the timer.

Command Parameters	<NR2> (note: 0.0 disables the watchdog function)
Query Parameters	None
Query Returns	<NR2>

SYSTem:WATChdog:ROBust

SYSTEM command and query

This command determines the type of communication required to reset the watchdog timer (and keep the Module working). If robust is set to ON, only the SYSTem:WATChdog:SERVice command will reset the timer. If robust is set to OFF (the default), any communication will reset the timer.

Command Parameters	0 | 1 | OFF | ON
Query Parameters	None
Query Returns	0 | 1

SYSTem:WATChdog:SERVice

SYSTEM command

This command satisfies the required communication when SYSTem:WATChdog:ROBust command is ON.

Command Parameters	none

Trigger Commands

Trigger commands controls the triggering of the module. Chapter 3 under “Triggering Changes” provides an explanation of the Trigger System.

NOTE: The list and measurement commands must first be enabled using the INITiate commands or no action due to triggering will occur. This does not apply to transient triggers.

ABORt

SYSTEM command

This command resets the list and measurement trigger systems to the Idle state. Any list or measurement that is in progress is immediately aborted. ABORt also resets the WTG bit in the Operation Condition Status register (see chapter 3 under “Programming the Status Registers”). ABORt is executed at power turn-on and upon execution of *RCL, RST, or any implied abort command (see List Commands).

Command Parameters	None

<Future>TRIGger:AGENerate

SYSTEM command and query

This command puts the Module in auto generate trigger mode. When set, the Module will generate a trigger on each operating change in value (new current, resistance, power, or voltage).

Command Parameters	0 | 1 | OFF | ON
Query Parameters	None
Query Returns	0 | 1

<Future>TRIGger:SOURce

SYSTEM command and query

This command selects the trigger source.

- IMMediate: Does not wait for a trigger.
- BUS: Accepts a *TRG command as the trigger source.
- EXTernal: Selects the module’s trigger input as the trigger source. This trigger is processed as soon as it is received.
- D0Rise: Selects the module’s digital input bit 0 as the trigger source. This trigger is processed on the rising edge.
- D0Fall: Selects the module’s digital input bit 0 as the trigger source. This trigger is processed on the falling edge.

Command Parameters	IMMediate |BUS | EXTernal | D0Rise | D0Fall
Query Parameters	None
Query Returns	<CRD>

LabVIEW™ Programming

Getting Started with LabVIEW™

Introduction

The AC/DC Power Module is a highly flexible piece of equipment. It can act as one or more AC sources, one or more DC sources, or nearly any combination. The configuration commands allow you to set up the power module to function as you desire.

Preparing the System

Copy the files

Referencing the Driver

Traverse the menu

Initializing the Driver

Open the I/O resource

Accessing Methods and Properties

View the vi collection

Destroying the Driver Instance

Call close

Programming Example

Reference sample code

IVI Programming

Getting Started with IVI

Introduction

The AC/DC Power Module is a highly flexible piece of equipment. It can act as one or more AC sources, one or more DC sources, or nearly any combination. The configuration commands allow you to set up the power module to function as you desire.

The NHR9400 IVI driver implements the IVI Foundation IviACPwr Class Specification (IVI-4.5, Rev. 1.2), along with extensions for the unique capabilities of the NHR9400 power module.

Types of IVI Drivers

The IVI Foundation defines two fundamental types of IVI drivers – IVI-COM drivers and IVI-C drivers. IVI-COM drivers are implemented using standard COM technology. Specifically, IVI-COM drivers support what are known as *custom* COM interfaces. To ensure compatibility with the broadest possible set of development environments, the IVI specifications restrict data types that can be used in an IVI-COM driver to a special set called *oleautomation* data types (also known simply as *automation* data types. Automation types were originally designed for Visual Basic but have since gained broad acceptance. Consequently, IVI-COM drivers can be used in a wide variety of development environments, namely, those that support custom oleautomation COM interfaces.

IVI-C drivers are simple DLLs that export C-based entry points. They use standard include (.h) files to define function signatures and attribute values.

Getting Help

The NHR IVI driver provides documentation in several formats for numerous integrated development environments (IDEs):

- HTML 1.0 help file (.chm) – This is a standalone help file that is used by nearly all IDEs except for Visual Studio. This file can be located in the IVI Foundation folders, usually located in:

  ```
  C:\Program Files\IVI Foundation\IVI\Drivers\NHR9400
  ```

- Microsoft Help Viewer help file (.cab) – This is a help file that is used by Visual Studio 2010 and later. It integrates with the Visual Studio local help so that driver help content appears alongside Visual Studio's own help content as well as the help content of other products that integrate with Visual Studio.

- IntelliSense – IVI drivers used within .NET programming languages, such as Visual C#, require a specially formatted XML file to present IntelliSense help as end users type code. In addition, type libraries are embedded into every IVI-COM driver. Object browsers and many COM-based environments, such as Visual Basic 6 and Visual C++, rely upon information in the type library for IntelliSense help-based features.

- IVI-C function panel help – IVI-C drivers use function panel files (.fp) and attribute files (.sub) to represent the hierarchy of functions and attributes. They also use these files to store help information for the IVI-C drivers. The help information embedded in .fp and .sub files is used by environments such as National Instruments® LabWindows™/CVI and LabVIEW®.

Getting Started with Visual Studio (.NET)

Introduction

Interop assemblies are used in .NET environments, such as C# and VB.NET, to access COM components. When installing an IVI-COM driver on a computer with the .NET Framework, the IVI specifications require an interop assembly be deployed and properly registered. The full path to the interop assemblies for NHR drivers is usually:

```
C:\Program Files\IVI Foundation\IVI\Bin\Primary Interop Assemblies
```

The actual assembly name is:

```
NHR.NHR9400.Interop.dll
```

The topics in this section provide detailed instructions on how to access and use IVI-COM drivers in C# and VB.NET.

Referencing the Driver

In order to access any of the instrument specific interfaces, a reference to the driver DLL must be created.

1. In the Solution Explorer, right-click on References and select "Add Reference."
2. Click the COM tab.
3. Select the following type library: `IVI NHR9400 1.0 Type Library`
   (Note: the version field may differ from the above example)
4. Press OK.

The selected driver should now appear under the References node in Solution Explorer. There will also be entries named `IviDriverLib` and `IviACPwrLib`.

All data types (interfaces and enumerations) are located under namespaces. Usually the namespace-qualified name must be used, but the "using" statement (C#) or "imports" statement (VB) allows the type name to be used directly.

C#: `using NHR.NHR9400.Interop;`

VB: `Imports NHR.NHR9400.Interop`

It is also quite likely that you will want to add entries for `IviDriverLib` and `IviACPwrLib` as well.

Creating an Instance of the Driver

To create a reference to the driver DLL, declare a variable of the driver type and create a new instance.

C#: `NHR9400Class driver = new NHR9400Class();`

VB: `Dim driver as new NHR9400`

Connecting

Calling "Initialize" will establish a connection to an instrument, or setup the driver to work in simulation mode. The first argument selects the desired instrument by "resource name" (stored in the IVI Configuration Store) or physical resource descriptor (like "TCPIP0::192.168.0.2::inst0::INSTR").

```
C#:     driver.Initialize("NHR9400", true, true, "QueryInstrStatus=true");

VB#:    driver.Initialize("NHR9400", True, True, "QueryInstrStatus=true")
```

Refer to the help documentation section titled "Initializing the IVI-COM Driver" for more information on the arguments for "Initialize."

Accessing Methods and Properties

Each set of capabilities is defined as a hierarchy of interfaces containing methods and properties.

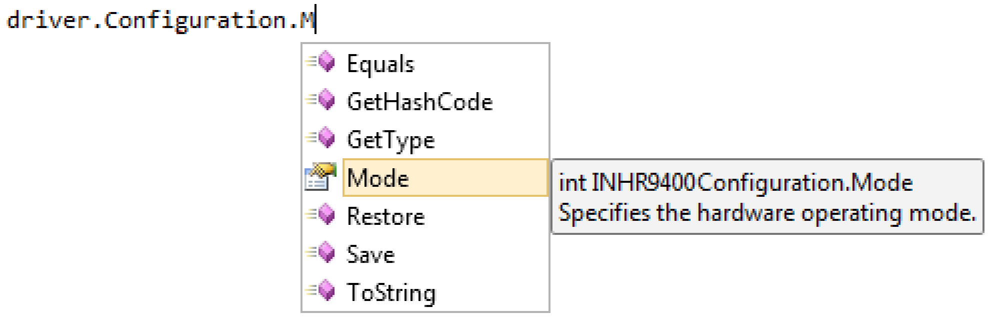
To call a method:

```
C#:     driver.Utility.Reset();

VB:     driver.Utility.Reset()
```

To set the value of a property:

```
C#:     driver.Configuration.Mode = 0;

VB:     driver.Configuration.Mode = 0
```

The IntelliSense system will prompt you for valid selections as you type:



```
driver.Configuration.M
```

int INHR9400Configuration.Mode
Specifies the hardware operating mode.

Hardware Mode

The organization of the channels is done by setting a "mode." Setting the mode will allow you to determine if it is a three output AC source, or a single output DC source, or anything in-between. There are a total of 13 hardware modes available for a three-channel 9400 power module, six for a two-channel 9400, and two for a one-channel 9400.

There are two ways to specify the operating mode of the power module. The first (and most flexible) is to set the property directly in code using the "Mode" property:

```
C#:     driver.Configuration.Mode = 6; // Split-phase AC + independent DC

VB:     driver.Configuration.Mode = 6 ' Split-phase AC + independent DC
```

The second method is to specify the desired operation using the "DriverSetup" string in the call to "Initialize." The IVI Foundation IviACPwr section 2.3.4 states that the DriverSetup field can include a

"NumPhases:n" option. When this method is used, the driver automatically selects AC operation in the proper "mode" based on the number of phases requested. To set three-phase operation:

C#:
```
driver.Initialize("NHR9400", true, true, "QueryInstrStatus=true,
DriverSetup=NumPhases:3");
```

VB#:
```
driver.Initialize("NHR9400", True, True, "QueryInstrStatus=true
DriverSetup=NumPhases:3")
```

The following table illustrates the mode selection based on the value of "NumPhases":

"NumPhases"	1-Channel 9400	2-Channel 9400	3-Channel 9400
1	Mode 0	Mode 1	Mode 1
2	N/A	Mode 0	Mode 5*
3	N/A	N/A	Mode 0

Refer to the help documentation section titled "INHR9400Configuration.Mode Property" for descriptions of the various mode settings.

Accessing Instrument Channels

The NHR9400 IVI instrument implements the IVI Foundation IviACPwr instrument class. This class exposes four "output" objects. There is one object per channel, and one final object used to program all channels to the same setting.

The driver stores these objects in an array named "OutputPhases," and they can be accessed individually by name. By convention, the names of the objects are "PhaseA", "PhaseB", "PhaseC" and "AllPhases."

If you configured a 3-phase AC source, you would program the voltage of AC1 as follows:

C#: `driver.OutputPhases.Item("PhaseA").Voltage = 120.0; // Line-Neutral`

VB#: `driver.OutputPhases.Item("PhaseA").Voltage = 120.0 ' Line-Neutral`

Alternately, you could program all phases at the same time:

C#: `driver.OutputPhases.Item("AllPhases").Voltage = 208.0; // Line-Line`

VB#: `driver.OutputPhases.Item("AllPhases").Voltage = 208.0 ' Line-Line`

Note the potentially different behavior of the commands when sent to all channels compared to values sent to individual channels.

If you are using the driver in a DC-only mode (such as Mode 4), the channels are still referred to as "phases," where "PhaseA" is DC1, "PhaseB" is DC2, etc.

Destroying the Driver Instance

When you are finished with the driver, calling "Close" is required; simply releasing the driver object (setting the variable to "null") is not sufficient.

C#: `driver.Close();`

VB: `driver.Close()`

* Channel 3 is unused in this state

Programming Examples

There are five C# programming examples provided with the IVI driver. Each example is written around a different configuration "mode." IVI sample code is normally located in:

```
C:\Program Files\IVI Foundation\IVI\Drivers\NHR9400\Examples
```

The samples assume you have an entry named "NHR9430" in your IVI Configuration Store (use NI-MAX or other utility to add an entry).

- "1-Ph AC" uses `NumPhases:1` to configure the power module. A multi-channel instrument will have all the channels synchronized together.
- "2-Ph AC" uses `NumPhases:2` to configure the power module. You must have a 2- or 3-channel instrument to run this sample. The instrument will operate in a split-phase mode.
- "3-Ph AC" uses `NumPhases:3` to configure the power module. You must have a 3-channel instrument to run this sample.
- "1-Ch DC" uses `Configuration.Mode` to configure the power module. A multi-channel instrument will have all the channels synchronized together.
- "1-Ph AC + 1-Ch DC" uses `Configuration.Mode` to configure the power module into two logical instruments, one in AC mode and another in DC mode. You must have a 2- or 3-channel instrument to run this sample.

Getting Started with LabWindows™/CVI™

Introduction

The topics in this section provide detailed instructions on how to access and use IVI-C drivers in LabWindows™/CVI™.

Referencing the Driver

In order to access any of the instrument specific interfaces, a reference to the driver DLL must be created.

1. In the Edit menu, select “Add Files to Project”, then “Instrument (*.fp).”
2. Browse to find the desired function panel (.fp), usually located in

   ```
   C:\Program Files\IVI Foundation\IVI\Drivers\NHR9400
   ```

   Select `NHR9400.fp`
3. Press “Add”, then “OK.”

The selected driver should now be added to your Project Tree, and appear under the Instruments folder in the Library Tree.

Initializing the Driver

Expand the NHR9400 node in the Instruments folder and drag an “init” function into the desired area of the source code. Calling “Init” will establish a connection to an instrument, or setup the driver to work in simulation mode. The first argument selects the desired instrument by “resource name” (stored in the IVI Configuration Store) or physical resource descriptor (like “TCPIP0::192.168.0.2::inst0::INSTR”).

```
ViSession vi = 0;
ViRsrc resourceName = "NHR9430";
NHR9400_InitWithOptions(resourceName, VI_TRUE, VI_TRUE,
                        "QueryInstrStatus=true", &vi);
```

Refer to the help documentation section titled “Initializing the IVI-C Driver” for more information on the arguments for “Initialize.”

After the Init call, the **vi** reference contains the instance of the driver, and is used as the **vi** input parameter of the remaining functions.

Accessing Methods and Properties

Each set of capabilities is defined as a hierarchy of interfaces containing methods and properties.

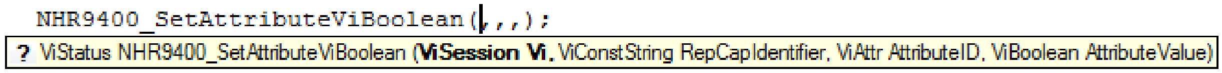
To call a “method”:

```
NHR9400_reset(vi);
```

To set the value of a “property”:

```
NHR9400_SetAttributeViBoolean(vi, "AllPhases", NHR9400_ATTR_OUTPUT_ENABLED, VI_TRUE);
```

The IntelliSense system will prompt you for valid selections as you type:



```
NHR9400_SetAttributeViBoolean(,,,);
? ViStatus NHR9400_SetAttributeViBoolean (ViSession Vi, ViConstString RepCapIdentifier, ViAttr AttributeID, ViBoolean AttributeValue)
```

Hardware Mode

The organization of the channels is done by setting a "mode." Setting the mode will allow you to determine if it is a three output AC source, or a single output DC source, or anything in-between. There are a total of 13 hardware modes available for a three-channel 9400 power module, six for a two-channel 9400, and two for a one-channel 9400.

There are two ways to specify the operating mode of the power module. The first (and most flexible) is to set the property directly in code using the "Mode" property:

```
// Split-phase AC + independent DC
NHR9400_SetAttributeViInt32(vi, "AllPhases", NHR9400_ATTR_CONFIGURATION_MODE, 6);
```

The second method is to specify the desired operation using the "DriverSetup" string in the call to "Initialize." The IVI Foundation IviACPwr section 2.3.4 states that the DriverSetup field can include a "NumPhases:n" option. When this method is used, the driver automatically selects AC operation in the proper "mode" based on the number of phases requested. To set three-phase operation:

```
NHR9400_InitWithOptions(resourceName, VI_TRUE, VI_TRUE,
                    "QueryInstrStatus=true, DriverSetup=NumPhases:3", &vi);
```

The following table illustrates the mode selection based on the value of "NumPhases":

"NumPhases"	1-Channel 9400	2-Channel 9400	3-Channel 9400
1	Mode 0	Mode 1	Mode 1
2	N/A	Mode 0	Mode 5[†]
3	N/A	N/A	Mode 0

Refer to the help documentation section titled "NHR9400_ATTR_CONFIGURATION_MODE Attribute" for descriptions of the various mode settings.

Accessing Instrument Channels

The NHR9400 IVI instrument implements the IVI Foundation IviACPwr instrument class. This class exposes four "output" objects. There is one object per channel, and one final object used to program all channels to the same setting.

The driver stores these objects in an array named "OutputPhases," and they can be accessed individually by name. By convention, the names of the objects are "PhaseA", "PhaseB", "PhaseC" and "AllPhases."

If you configured a 3-phase AC source, you would program the voltage of AC1 as follows:

```
// Line-Neutral
NHR9400_SetAttributeViReal64(vi, "PhaseA", NHR9400_ATTR_VOLTAGE_LEVEL, 120.0);
```

Alternately, you could program all phases at the same time:

[†] Channel 3 is unused in this state

```
// Line-Line
NHR9400_SetAttributeViReal64(vi, "AllPhases", NHR9400_ATTR_VOLTAGE_LEVEL, 208.0);
```

Note the potentially different behavior of the commands when sent to all channels compared to values sent to individual channels.

If you are using the driver in a DC-only mode (such as Mode 4), the channels are still referred to as “phases,” where “PhaseA” is DC1, “PhaseB” is DC2, etc.

Destroying the Driver Instance

When you are finished with the driver, calling “Close” is required; simply releasing the driver object (setting the variable to zero) is not sufficient.

```
/* clean up */
if (vi != 0)
      NHR9400_close(vi);
```

Programming Example

A complete sample of a program written in LabWindows/CVI is available in the examples folder. IVI sample code is normally located in:

```
C:\Program Files\IVI Foundation\IVI\Drivers\NHR9400\Examples
```

The sample assumes you have an entry named “NHR9430” in your IVI Configuration Store (use NI-MAX or other utility to add an entry), and requires a three-channel instrument to run.