FLUKE®

NORMA 6003/NORMA 6003+/ NORMA 6004/NORMA 6004+

Power Analyzers

Programmers Reference Guide

BC

June 2020

© 2020 Fluke Corporation. All rights reserved. Specifications are subject to change without notice.
All product names are trademarks of their respective companies.

LIMITED WARRANTY AND LIMITATION OF LIABILITY

Each Fluke product is warranted to be free from defects in material and workmanship under normal use and service. The warranty period is two years and begins on the date of shipment. Parts, product repairs, and services are warranted for 90 days. This warranty extends only to the original buyer or end-user customer of a Fluke authorized reseller, and does not apply to fuses, disposable batteries, or to any product which, in Fluke's opinion, has been misused, altered, neglected, contaminated, or damaged by accident or abnormal conditions of operation or handling. Fluke warrants that software will operate substantially in accordance with its functional specifications for 90 days and that it has been properly recorded on non-defective media. Fluke does not warrant that software will be error free or operate without interruption.

Fluke authorized resellers shall extend this warranty on new and unused products to end-user customers only but have no authority to extend a greater or different warranty on behalf of Fluke. Warranty support is available only if product is purchased through a Fluke authorized sales outlet or Buyer has paid the applicable international price. Fluke reserves the right to invoice Buyer for importation costs of repair/replacement parts when product purchased in one country is submitted for repair in another country.

Fluke's warranty obligation is limited, at Fluke's option, to refund of the purchase price, free of charge repair, or replacement of a defective product which is returned to a Fluke authorized service center within the warranty period.

To obtain warranty service, contact your nearest Fluke authorized service center to obtain return authorization information, then send the product to that service center, with a description of the difficulty, postage and insurance prepaid (FOB Destination). Fluke assumes no risk for damage in transit. Following warranty repair, the product will be returned to Buyer, transportation prepaid (FOB Destination). If Fluke determines that failure was caused by neglect, misuse, contamination, alteration, accident, or abnormal condition of operation or handling, including overvoltage failures caused by use outside the product’s specified rating, or normal wear and tear of mechanical components, Fluke will provide an estimate of repair costs and obtain authorization before commencing the work. Following repair, the product will be returned to the Buyer transportation prepaid and the Buyer will be billed for the repair and return transportation charges (FOB Shipping Point).

THIS WARRANTY IS BUYER'S SOLE AND EXCLUSIVE REMEDY AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FLUKE SHALL NOT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OR LOSSES, INCLUDING LOSS OF DATA, ARISING FROM ANY CAUSE OR THEORY.

Since some countries or states do not allow limitation of the term of an implied warranty, or exclusion or limitation of incidental or consequential damages, the limitations and exclusions of this warranty may not apply to every buyer. If any provision of this Warranty is held invalid or unenforceable by a court or other decision-maker of competent jurisdiction, such holding will not affect the validity or enforceability of any other provision.

Fluke Corporation
P.O. Box 9090
Everett, WA 98206-9090
U.S.A.

11/99

Table of Contents

Title	Page
Introduction	1
How to Contact Fluke	1
Safety Information	1
Symbols and Syntax	2
Hardware Interfaces	6
Interface Specification	7
Switch between Remote and Local Modes	7
Switch from Local to Remote Mode	7
Switch from Remote Mode to Local	7
Commands and Responses	8
Introduction to SCPI	8
Structure of Commands	8
Long and Short Form	9
RS485 Command's Prefix	9
Multiple Commands Per Line	9
HDLC Protocol (High-Level Data Link Control)	10
Formatted Resolution String Format	11
Communication Commands	13
FORMula Commands	18
[FORMula:]HVF:MOTOr:TYPE?	18
[FORMula:]HVF:MOTOr:TYPE {}	18
[FORMula:]HVF:NOMinal:TYPE?	18
[FORMula:]HVF:NOMinal:TYPE {}	18
[FORMula:]HVF:NOMinal:VALue?	18
[FORMula:]HVF:NOMinal:VALue <>	18
[FORMula:]HCF:NOMinal:TYPE?	19
[FORMula:]HCF:NOMinal:TYPE {}	19
[FORMula:]HCF:NOMinal:VALue?	19
[FORMula:]HCF:NOMinal:VALue <>	19
HARMonics Commands	19
HARMonics:ORDer? {""}	19
HARMonics:DATA? {""}	20
HARMonics:FFT:ITEM?	21
HARMonics:FFT:ITEM {""}	21
HARMonics:FFT:FREQuency?	21
HARMonics:FFT:FREQuency {}	21
HARMonics:FFT:DATA?	21
HOLD Commands	22
HOLD:STARt	22

HOLD:STOP 22
HOLD:STATus? 22
INPut Commands 22
INPut#:CURRent:AUTO? 22
INPut#:CURRent:AUTO {} 22
INPut#:CURRent:RANGe? 23
INPut#:CURRent:RANGe 23
INPut#:CURRent:RATIo? 23
INPut#:CURRent:RATIo<> 23
INPut#:CURRent:FILTer:LINE? 23
INPut#:CURRent:FILTer:LINE {} 23
INPut#:CURRent:EXTernal[:SENSor]? 24
INPut#:CURRent:EXTernal[:SENSor] {} 24
INPut#:CURRent:EXTernal[:SENSor]:RATIo? {} 24
INPut#:VOLTage:AUTO? 24
INPut#:VOLTage:AUTO {} 24
INPut#:VOLTage:RANGe? 25
INPut#:VOLTage:RANGe {} 25
INPut#:VOLTage:RATIo? 25
INPut#:VOLTage:RATIo<> 25
INPut#:VOLTage:FILTer:LINE? 25
INPut#:VOLTage:FILTer:LINE {} 25
INPUT:MS:MODE? 26
INPUT:MS:MODE {} 26
INPut:MS:RATIo? 26
INPut:MS:RATIo<> 26
INPut:MT:RATIo? 26
INPut:MT:RATIo<> 26
INPut:MS:MODE:RATIo? {} 27
INTEgral Commands 27
INTEgral:STARt 27
INTEgral:STOP 27
INTEgral:RESEt 27
INTEgral:STATus? 28
MEMory Commands 28
MEM:DOWNload {},<""> 28
MEM:UPLoad {},<""> 28
MEM:CATalog? {} 29
MEM:CATalog:HDLC? {} 29
RATE Commands 30
RATE? 30
RATE {} 30
SCOPe Commands 30
SCOPe? 30
SCOPe:TRIGger:METHod:TYPE? 30
SCOPe:TRIGger:METHod:TYPE {} 30
SCOPe:TRIGger:METHod:MODE? 31
SCOPe:TRIGger:METHod:MODE {} 31
SCOPe:TRIGger:SOURce? 31
SCOPe:TRIGger:SOURce {} 31
SCOPe:TRIGger:POSition:X? 31
SCOPe:TRIGger:POSition:X <> 31
SCOPe:TRIGger:POSition:Y? 32
SCOPe:TRIGger:POSition:Y <> 32

SCOPe:TIME:DIV? 32
SCOPe:TIME:DIV {} 32
SCOPe:AMPlitude? {} 32
SCOPe:AMPlitude {},{},{} 32
SCOPe:ADD:ITEM {} 33
SCOPe:REMOve:ITEM {} 33
SCOPe:DATA:ITEM? {} 34
SCOPe:HOLD {STOP} 34
SENSe Commands 34
[SENSe:][FORMatted:]DATA? {""} 34
[SENSe:]RAWData? {""} 35
SYNC Commands 35
SYNC#[:SOURce]? 35
SYNC#[:SOURce] {} 35
SYNC#[:SOURce]:STATus? {} 35
SYSTem Commands 35
SYSTem:REMote {} 35
SYSTem:ERRor[:NEXT]? 36
SYSTem:ERRor:ALL? 36
SYSTem:ERRor:CODE[:NEXT]? 36
SYSTem:ERRor:CODE:ALL? 37
SYSTem:ERRor:COUNt? 37
SYSTem:HOST? 37
SYSTem:HOST {} 37
SYSTem:COMMunicate:SERial:BAUD? 38
SYSTem:COMMunicate:SERial:BAUD {} 38
SYSTem:COMMunicate:TYPE? 38
SYSTem:COMMunicate:TYPE {} 38
SYSTem:DATE < >,< >,< > 38
SYSTem:DATE? 38
SYSTem:TIME < >,< >,< > 39
SYSTem:TIME? 39
SYSTem:LANGuage? 39
SYSTem:LANGuage {} 39
SYSTem:RS485:ADDRess <> 40
SYSTem:RS485:ADDRess? 40
SYSTem:REBOot 40
SYSTem:[:FACTory]:RESET 40
TRENd Commands 40
TRENd? 40
TRENd:ADD:ITEM {""} 41
TRENd:REMOve:ITEM {""} 41
TRENd:INTErval:TYPE? 41
TRENd:INTErval:TYPE {} 41
TRENd:INTErval? 42
TRENd:INTErval <> 42
TRENd:DATA:ITEM? {""} 42
WIRing Commands 42
WIRing:GROUp? 42
WIRing:GROUp {},},},} 42
WIRing:GROUp:STATus? {},},},} 43
ZERO Commands 43
ZERO? {“”} 43
Common Commands 44

*IDN? 44
*CLS 44
*RST 44
Error Message 45

Introduction

This document defines the remote interface commands for the Fluke NORMA 6003 / NORMA 6003+ / NORMA 6004 / NORMA 6004+ Power Analyzers (the Product or Analyzer). These commands may be used by a computer connected through the remote interface port to set settings, read measurement data, and control the operation of the Analyzer. Command syntax and names follow IEEE-488.2 and SCPI standards.

For operational information, see the printed *Getting Started Manual* that ships with the Product and the *Users Manual*, both online at www.fluke.com.

How to Contact Fluke

Fluke Corporation operates worldwide. For more contact information, go to our website: www.fluke.com.

To register your product, view, print, or download the latest manual or manual supplement, go to cn.fluke.com (Chinese) or www.fluke.com (English).

Fluke Corporation
6920 Seaway Blvd
Everett, WA 98203

+1-425-446-5500

fluke-info@fluke.com

Safety Information

General safety information is located in the printed *Getting Started Manual* that ships with the Product and online at www.fluke.com or cn.fluke.com. More specific safety information is listed where applicable.

Symbols and Syntax

The Analyzer accepts commands to set parameters, execute functions or respond with requested data. These commands are strings of ASCII-encoded characters. See *Introduction to SCPI* for standard SCPI command syntax. See Table 1 for the special symbols used in this guide for SCPI commands..

Table 1. Symbols for SCPI Syntax

Symbol	Meaning	Command Examples	
		Long form	Actual Entry
{|}	Select one value from the values given in { }	HVF:MOTOr:TYPE {3PHASE|NON3PHASE}	HVF:MOTOR:TYPE 3PHASE
{""}	Select an available value from the parameters	HARMonics:FFT:ITEM {""}	HARM:FFT:ITEM "VOLT:3"
< >	Required value	HVF:NOM:VAL <>	HVF:NOM:VAL 3.2
[ ]	Can be omitted	[SENSe:]DATA? "VOLT:RMS:1"	DATA? "VOLT:RMS:1"
#	Suffix for channel or wiring group	INPUT#:CURR:AUTO?	INPUT1:CURR:AUTO?

The Analyzers measures 3 to 4 channels and up to 2 wiring groups. See Table 2 for a list of abbreviations used in the command syntax.

Table 2. Abbreviations Used in Numeric Definitions

Abbreviation	Description
1	Channel 1
2	Channel 2
3	Channel 3
4	Channel 4
A1	Phase A in wiring group 1
B1	Phase B in wiring group 1
C1	Phase C in wiring group 1
AB1	Line A to B in wiring group 1
BC1	Line B to C in wiring group 1
CA1	Line C to A in wiring group 1
SGM1	Sum of wiring group 1
N1	Neutral-line in wiring group 1
A2	Phase A in wiring group 2
B2	Phase B in wiring group 2

Table 2. Abbreviations Used in Numeric Definitions (cont.)

Abbreviation	Description
C2	Phase C in wiring group 2
AB2	Line A to B in wiring group 2
BC2	Line B to C in wiring group 2
CA2	Line C to A in wiring group 2
SGM2	Sum of wiring group 2
N2	Neutral-line in wiring group 2

The Analyzer uses many symbols for measurements and the measurement results. See Table 3 for the available options for the measurements.

Table 3. Measurement Options

Symbol	Measurement Name	Example in Command
U_{rms}	RMS voltage	VOLT:RMS[:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|SGM1|A2|B2|C2|AB2|BC2|CA2|SGM2]
I_{rms}	RMS current	CURR:RMS[:1|2|3|4|A1|B1|C1|N1|SGM1|A2|B2|C2|N2|SGM2]
U_{H01}	Fundamental voltage	VOLT:H01[:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|SGM1|A2|B2|C2|AB2|BC2|CA2|SGM2]
I_{H01}	Fundamental current	CURR:H01[:1|2|3|4|A1|B1|C1|N1|SGM1|A2|B2|C2|N2|SGM2]
U_{DC}	DC component voltage	VOLT:DC[:1|2|3|4]
I_{DC}	DC component current	CURR:DC[:1|2|3|4]
U_{AC}	AC component voltage	VOLT:AC[:1|2|3|4]
I_{AC}	AC component current	CURR:AC[:1|2|3|4]
U_{rmc}	Corrected rectified mean voltage	VOLT:RMCORR[:1|2|3|4]
I_{rmc}	Corrected rectified mean current	CURR:RMCORR[:1|2|3|4]
U_{rm}	Rectified mean voltage	VOLT:RMEAN[:1|2|3|4]
I_{rm}	Rectified mean current	CURR:RMEAN[:1|2|3|4]
U_{uc}	Voltage unbalance coefficient	VOLT:UNCOEF[:SGM1|SGM2]
I_{uc}	Current unbalance coefficient	CURR:UNCOEF[:SGM1|SGM2]
P	Active power	POW[:1|2|3|4|SGM1|SGM2]
P_{H01}	Fundamental active power	POW:H01[:1|2|3|4|SGM1|SGM2]

Table 3. Measurement Options (cont.)

Symbol	Measurement Name	Example in Command
S	Apparent power	POW:APP[:1|2|3|4|SGM1|SGM2]
S_{H01}	Fundamental apparent power	POW:APP:H01[:1|2|3|4]
Q	Reactive power	POW:REACT[:1|2|3|4|SGM1|SGM2]
Q_{H01}	Fundamental reactive power	POW:REACT:H01[:1|2|3|4]
λ	Power factor	POW:FACT[:1|2|3|4|SGM1|SGM2]
$λ_{H01}$	Fundamental power factor	POW:FACT:H01[:1|2|3|4]
φ	Phase	PHAS[:1|2|3|4|SGM1|SGM2]
$φU_{H01}$	Fundamental phase for voltage (relative to the sync source)	PHAS:UH01[:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2 |AB2|BC2|CA2]
$φI_{H01}$	Fundamental phase for current (relative to the sync source)	PHAS:IH01[:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2]
$φ_{H01}$	Fundamental phase shift	PHAS:H01[:1|2|3|4|A1|B1|C1|A2|B2|C2]
U_{p+}	Peak+ voltage	VOLT:MAX[:1|2|3|4]
I_{p+}	Peak+ current	CURR:MAX[:1|2|3|4]
U_{p-}	Peak- voltage	VOLT:MIN[:1|2|3|4]
I_{p-}	Peak- current	CURR:MIN[:1|2|3|4]
U_{pp}	Peak-to-peak voltage	VOLT:PTP[:1|2|3|4]
I_{pp}	Peak-to-peak current	CURR:PTP[:1|2|3|4]
U_{cf}	Voltage crest factor	VOLT:CFAC[:1|2|3|4]
I_{cf}	Current crest factor	CURR:CFAC[:1|2|3|4]
Urip	Voltage ripple factor	VOLT:RIP[:1|2|3|4]
Irip	Current ripple factor	CURR:RIP[:1|2|3|4]
U_{ff}	Voltage form factor	VOLT:FFAC[:1|2|3|4]
I_{ff}	Current form factor	CURR:FFAC[:1|2|3|4]
U_{THD}	Voltage THD	VOLT:THD[:1|2|3|4|A1|B1|C1|AB1|BC1|CA1 |A2|B2|C2|AB2|BC2|CA2]
I_{THD}	Current THD	CURR:THD[:1|2|3|4|A1|B1|C1|N1 |A2|B2|C2|N2]

Table 3. Measurement Options (cont.)

Symbol	Measurement Name	Example in Command
U_{fc}	Voltage fundamental content	VOLT:FCONT[:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2 |C2|AB2|BC2|CA2]
I_{fc}	Current fundamental content	CURR:FCONT[:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2]
P_{fc}	Power fundamental content	POW:FCONT[:1|2|3|4]
U_{hc}	Voltage harmonic content	VOLT:HCONT[:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2 |AB2|BC2|CA2]
I_{hc}	Current harmonic content	CURR:HCONT[:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2]
hvf	Harmonic voltage factor	VOLT:HF[:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2|AB2| BC2|CA2]
hcf	Harmonic current factor	CURR:HF[:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2]
η	Efficiency	POW:EFF
P_c	Corrected power	POW:CORR[:1|2|3|4]
Z	Impedance	IMP[:1|2|3|4]
Z_{H01}	Fundamental impedance	IMP:H01[:1|2|3|4]
R_s	Equivalent serial resistance	RES:SER[:1|2|3|4]
R_{sH01}	Fundamental equivalent serial resistance	RES:SER:H01[:1|2|3|4]
X_s	Equivalent serial reactance	REACT:SER[:1|2|3|4]
X_{sH01}	Fundamental equivalent serial reactance	REACT:SER:H01[:1|2|3|4]
R_p	Equivalent parallel resistance	RES:PAR[:1|2|3|4]
R_{pH01}	Fundamental equivalent parallel resistance	RES:PAR:H01[:1|2|3|4]
X_p	Equivalent parallel reactance	REACT:PAR[:1|2|3|4]
X_{pH01}	Fundamental equivalent parallel reactance	REACT:PAR:H01[:1|2|3|4]
W	Electric energy	POW:INT[:1|2|3|4|SGM1|SGM2]
W+	Electric energy +	POW:INT:POS[:1|2|3|4]
W-	Electric energy -	POW:INT:NEG[:1|2|3|4]
q	Electric quantity	CURR:INT[:1|2|3|4]
q+	Electric quantity +	CURR:INT:POS[:1|2|3|4]

Table 3. Measurement Options (cont.)

Symbol	Measurement Name	Example in Command
q-	Electric quantity -	CURR:INT:NEG[:1|2|3|4]
f	Fundamental frequency	FREQ[:1|2|3|4]
$t_{\int}$	Integral time for both power integral and current integral	INT:TIME
T	Torque	TORQ
n	Rotation speed	SPE
P_m	Mechanical power	POW:MECH
E_m	Mechanical energy	POW:MECH:INT

Hardware Interfaces

The Analyzer has a mini USB port (serial port) and an RS-485 port. Use only one communication port at a time. Figure 1 shows the RS-485 port connections.

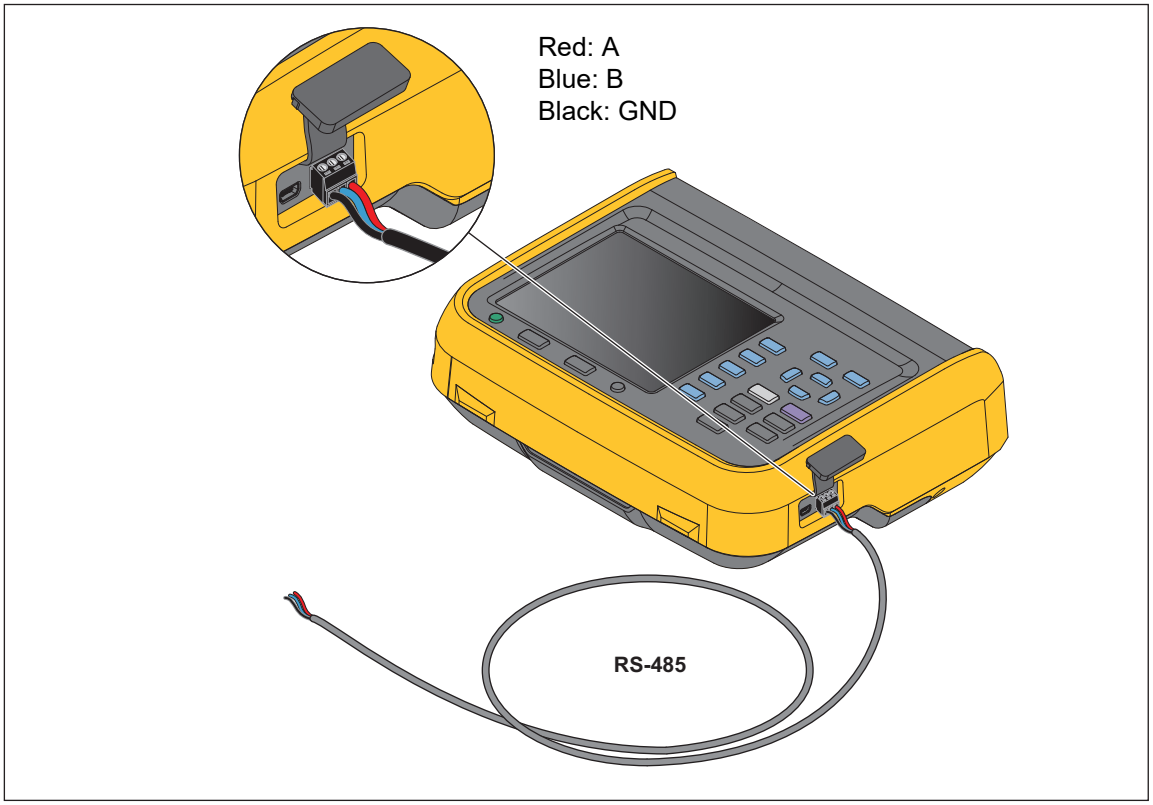


Figure 1. Communication Port

Interface Specification

See Table 4 for interface specifications

Table 4. Interface Specification

Interface	USB	RS485
Connector	Type B mini	3 wire terminal with 3.81 mm pitch
Number of Ports	1	1
Baud Rate	9600 bps to 921600 bps	9600 bps to 921600 bps
EOL	CR/LF/CRLF	CR/LF/CRLF
Data bits	7 or 8	7 or 8
Parity	None/Odd/Even	None/Odd/Even
Stop bits	1, 1.5, or 2	1, 1.5, or 2
Address	--	1 to 9

Switch between Remote and Local Modes

Use Local mode for direct operation of the Analyzer. Use Remote mode to program, read data, and send commands from a linked computer.

Switch from Local to Remote Mode

To enter Remote mode from Local mode: send the SYSTem:REMote ON command to the controller. See *SYSTem:REMote {}*.

In Remote mode,

- The Analyzer shows the communication icon: (USB) or RS485 (RS485)
- The Analyzer disables all keys except F1 (LOCAL).

Switch from Remote Mode to Local

To enter Local mode from Remote mode, choose a method:

- On the Analyzer, push F1 (LOCAL).

  OR

- From the connected computer, send the SYSTem:REMote OFF command.

In Local mode,

- The Analyzer turns off the communication icon: (USB) or RS485 (RS485)
- Key operations are enabled.

Commands and Responses

Introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) describes a standard command set for programming instruments, irrespective of the type of instrument or manufacturer. The objective of the SCPI consortium is to standardize the device-specific commands to a large extent. The model defines identical functions of a device or of different devices. Command systems are assigned to these functions and it is possible to address identical functions with identical commands. The command systems have a hierarchical structure.

Figure 2 shows an example of the tree structure using the command system INPut, which operates the input channel of the devices. The other examples concerning syntax and structure of the commands are derived from this command system.

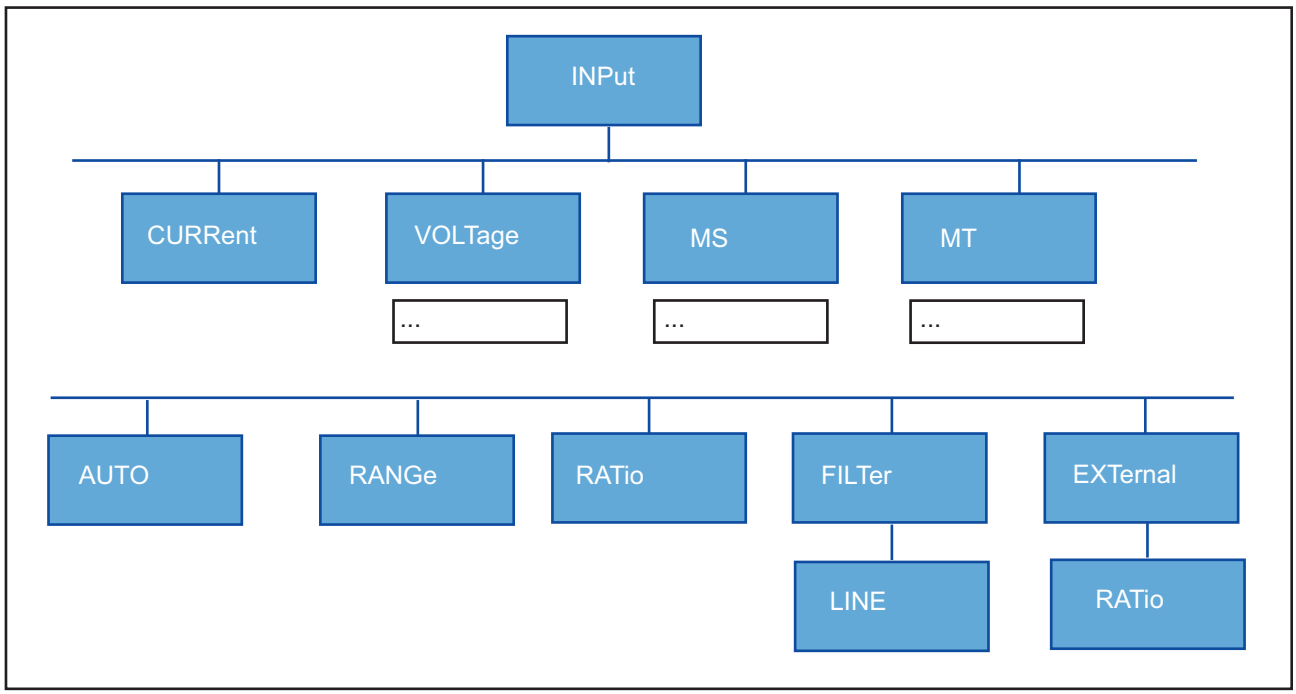


Figure 2. Command Tree Structure

SCPI is based on standard IEEE 488.2 and uses the same basic syntax elements and common commands defined in this standard. Part of the syntax of the device responses is defined in greater detail than in standard IEEE 488.2 (see *Commands and Responses*).

Structure of Commands

Commands consist of a command header and, if necessary, parameter data.

- The header and the parameters are separated by a “space” (0x20).
- Multiple parameters are separated by a comma (“,” 0x2C).
- Headers may consist of several key words.

- Query commands are commands that request data in response. Query commands have a question mark (“?”, 0x3F) immediately following the command header. Responses to query commands are generated, placed in the output buffer, and transmitted to the PC.
- If a device has several functions or features of the same kind, such as inputs, the desired function can be selected by appending a numerical suffix (“#” is used as a placeholder in a command description) to the Command.
- Commands are not case sensitive.
- Terminate commands with a carriage return (0x0D), new line character (0x0A), or carriage return (0x0D) and new line character (0x0A).

Long and Short Form

Command headers consist of one or more mnemonics. Most mnemonics have a long form that is more readable and a short form consisting of three or four characters that is more efficient. The long form is a complete word. The short form is shown by upper-case letters.

Note

This manual uses upper-case to indicate the short form and lower-case letters to indicate the long forms. The commands are not case sensitive.

Long form example

INPut1:CURRent:EXTernal:SENSor?

Short form example

INP1:CURR:EXT?

Both commands query channel 1’s current input using external sensor.

RS485 Command’s Prefix

For the RS485 interface append “:” + “Address” (1 digit code) to the front of the command.

Example:

TX: :1*idn?

RX: :1FLUKE,NORMA_6004+,000-000-00,v4.2.0-1-g1e3151-dev,v4.2.0,V1.2

Multiple Commands Per Line

Use a semicolon (;) to separate commands in a command line. If the next command belongs to a different command system, use a semicolon followed by a colon.

USB port examples:

TX: *idn?;:input1:current:range?

RX: FLUKE,NORMA_6004+,000-000-00,v4.2.0-1-g1e3151-dev,v4.2.0,V1.2

RX: 100mV

TX: :input1:current:range?;ratio?

RX: 100mA

RX: 1

RS485 port examples:

TX: :1*idn?;:1:input1:current:range?

RX: :1FLUKE,NORMA_6004+,000-000-00,v4.2.0-32-g0b7023-dev,v4.2.0-6-g6621c1,V1.2

RX: :1100Mv

TX: :1:input1:current:range?;:1ratio?

RX: :1100mA

RX: :11

HDLC Protocol (High-Level Data Link Control)

See Table 5 through Table 8 for control information.

Table 5. HDLC Data Frame

	STX	**Address Byte**	**Control Byte**	**Information Bytes**	**FCS Bytes**	**ETX**
Bytes Number	1 (0x7D)	1 (default is 0x00)	1 (default is 0x00)	Variable, only for iframe	2	1 (0x7E)

Table 6. Control Byte for iframe

	N(R)	**P/F**	**N(S)**	**0**
Bits Number	Bit 5 - Bit 7	Bit 4	Bit 1 - Bit 3	Bit 0
Information bytes for iframe: When information bytes are not ASCII code, bytes location follows little-endian.				

Table 7. Control Byte for sframe

	N(R)	**P/F**	**Type**	**0**	**1**
Bits Number	Bit 5 - Bit 7	Bit 4	Bit 2 - Bit 3	Bit 1	Bit 0

Table 8. Control Byte for uframe

	N(R)	**P/F**	**Code**	**1**	**1**
Bits Number	Bit 5~7	Bit 4	Bit 2~3	Bit 1	Bit 0

See Table 9 for escape characters

Table 9. Escape characters

HDLC_STX	0x7D
HDLC_ETX	0x7E
HDLC_DEL	0x10

Escape rule:

Insert HDLC_DEL before the escape character, and then XOR the escape character with 0x20. See Table 10 for an example.

Table 10. Escape Example

Byte Index	**1**	**2**	**3**	**4**	**5**
Raw Bytes	0x01	0x02	0x10	0x7D	0x05
Converted Bytes	0x7D,0x01	0x02	0x10,0x30	0x10,0x5D	0x05,0x7E

Formatted Resolution String Format

Each formatted resolution string consists of 8 bytes:

Table 11. Formated String

Byte Index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	Raw value (float, little-endian)				Reading status (See Table 12)	Unit (See Table 13)	Magnitude (See Table 14)	Decimal points
Byte 5, byte 6 and byte 7 are enumerated value, byte 8 is an integer value.								

Table 12. Reading Status

Enumerated Value	**Status**
0	Normal
1	Invalid
2	Over upper limit
3	Over lower limit

Table 13. Unit

Enumerated Value	**Unit**
0	None
1	V
2	A
3	Ω
4	jΩ
5	W
6	Hz
7	VA

Table 13. Unit (cont.)

Enumerated Value	Unit
8	var
9	degree
10	radian
11	%
12	Wh
13	Ah
14	Nm
15	rpm

Table 14. Magnitude

Enumerated Value	Magnitude
0	Normal
1	Tera
2	Giga
3	Mega
4	Kilo
5	Milli
6	Micro
7	Nano
8	Pico

Communication Commands

This section lists the available commands. See Table 15.

Table 15. Communication Commands

Command Groups	Function
FORMula Commands	
[FORMula:]HVF:MOTOr:TYPE? *[FORMula:]HVF:MOTOr:TYPE {}*	Queries and sets HVF motor type.
[FORMula:]HVF:NOMinal:TYPE? *[FORMula:]HVF:NOMinal:TYPE {}*	Queries and sets HVF voltage nominal value type.
[FORMula:]HVF:NOMinal:VALue? *[FORMula:]HVF:NOMinal:VALue <>*	Queries and sets HVF specified nominal value, float value.
[FORMula:]HCF:NOMinal:TYPE? *[FORMula:]HVF:NOMinal:TYPE {}*	Queries and sets HCF current nominal value type.
[FORMula:]HCF:NOMinal:VALue? *[FORMula:]HCF:NOMinal:VALue <>*	Queries and sets HCF specified nominal value, float value.
HARMonics Commands	
HARMonics:ORDer? {""}	Queries the maximum output harmonic order of the specified function's harmonic measurement numeric list data.
HARMonics:DATA? {""}	Queries harmonic data for the specified items.
HARMonics:FFT:ITEM? *HARMonics:FFT:ITEM {""}*	Queries and sets FFT item name.
HARMonics:FFT:FREQuency? *HARMonics:FFT:FREQuency {}*	Queries and sets the FFT's upper limit frequency for FFT item.
HARMonics:FFT:DATA?	Queries FFT data for the FFT item.
HOLD Commands	
HOLD:STARt	Starts HOLD for METER/TREND/PHASOR/HARMONIC.
HOLD:STOP	Stops HOLD for METER/TREND/PHASOR/HARMONIC.

Table 15. Communication Commands (cont.)

Command Groups	Function
HOLD:STATus?	Queries HOLD status for METER/TREND/PHASOR/HARMONIC.
INPut Commands	
INPut#:CURRent:AUTO? *INPut#:CURRent:AUTO {}*	Queries and sets the current auto range for the specified channel.
INPut#:CURRent:RANGe? *INPut#:CURRent:RANGe*	Queries and sets the current range for the specified channel.
INPut#:CURRent:RATIo? *INPut#:CURRent:RATIo<>*	Queries and sets the current ratio for the specified channel.
INPut#:CURRent:FILTer:LINE? *INPut#:CURRent:FILTer:LINE {}*	Queries and sets the current line filter for the specified channel.
INPut#:CURRent:EXTernal[:SENSor]? *INPut#:CURRent:EXTernal[:SENSor] {}*	Queries and sets the status of external sensor for the specified channel.
INPut#:CURRent:EXTernal[:SENSor]:RATIo? {}	Queries the current ratio for the specified channel when external sensor is enabled/disabled.
INPut#:VOLTage:AUTO? *INPut#:VOLTage:AUTO {}*	Queries and sets the voltage auto range for the specified channel.
INPut#:VOLTage:RANGe? *INPut#:VOLTage:RANGe {}*	Queries and sets the voltage range for the specified channel.
INPut#:VOLTage:RATIo? *INPut#:VOLTage:RATIo<>*	Queries and sets the voltage ratio for the specified channel.
INPut#:VOLTage:FILTer:LINE? *INPut#:VOLTage:FILTer:LINE {}*	Queries and sets the voltage line filter for the specified channel.
INPUT:MS:MODE? *INPUT:MS:MODE {}*	Queries and sets the motor speed mode.
INPut:MS:RATIo? *INPut:MS:RATIo<>*	Queries and sets the motor speed ratio.
INPut:MT:RATIo? *INPut:MT:RATIo<>*	Queries and sets the motor torque ratio.
INPut:MS:MODE:RATIo? {}	Queries the motor speed ratio for the specified mode.

Table 15. Communication Commands (cont.)

Command Groups	Function
INTEgral Commands	
INTEgral:STARt	Starts integral process.
INTEgral:STOP	Stops integral process.
INTEgral:RESEt	Resets integral process.
INTEgral:STATus?	Queries integral status.
MEMory Commands	
MEM:DOWNload {},<"">	Starts transfer protocol to download a file from host.
MEM:UPLoad {},<"">	Starts transfer protocol to upload a file to host.
MEM:CATalog? {}	Lists files in specified folder, response are raw ASCII codes.
MEM:CATalog:HDLC? {}	Lists files in specified folder, response are encoded as a HDLC iframe.
RATE Commands	
RATE? *RATE {}*	Queries and sets data update rate.
SCOPe Commands	
SCOPe?	Queries all scope items' name.
SCOPe:TRIGger:METHod:TYPE? *SCOPe:TRIGger:METHod:TYPE {}*	Queries and sets trigger method type.
SCOPe:TRIGger:METHod:MODE? *SCOPe:TRIGger:METHod:MODE {}*	Queries and sets trigger method mode.
SCOPe:TRIGger:SOURce? *SCOPe:TRIGger:SOURce {}*	Queries and sets trigger source.
SCOPe:TRIGger:POSition:X? *SCOPe:TRIGger:POSition:X <>*	Queries and sets trigger position (horizontal).
SCOPe:TRIGger:POSition:Y? *SCOPe:TRIGger:POSition:Y <>*	Queries and sets trigger position (vertical).
SCOPe:TIME:DIV? *SCOPe:TIME:DIV {}*	Queries and sets time division.

Table 15. Communication Commands (cont.)

Command Groups	Function
SCOPe:AMPlitude? {} *SCOPe:AMPlitude {},{},{}*	Queries and sets the specified item's amplitude_coefficient and amplitude_exponent.
SCOPe:ADD:ITEM {}	Adds a specified scope item.
SCOPe:REMOve:ITEM {}	Removes a specified scope item.
SCOPe:DATA:ITEM? {}	Queries specified item's data.
SCOPe:HOLD {STOP}	Stops hold while in HOLD status.
SENSe Commands	
[SENSe:][FORMatted:]DATA? {""}	Queries the specified numeric item's formatted value with unit/status/resolution.
[SENSe:]RAWData? {""}	Queries the specified numeric item's raw value without unit/status/resolution.
SYNC Commands	
SYNC#[:SOURce]? *SYNC#[:SOURce] {}*	Queries and sets the sync source for specified wiring group.
SYNC#[:SOURce]:STATus? {}	Sets the sync source for specified wiring group and returns whether setting is OK.
SYSTem Commands	
SYSTem:REMote {}	Enables/disables remote mode.
SYSTem:ERRor[:NEXT]?	Dequeues one error and returns its error code and error name.
SYSTem:ERRor:ALL?	Dequeues all the unread errors and returns their error code and error name.
SYSTem:ERRor:CODE[:NEXT]?	Dequeues one error and returns only its error code.
SYSTem:ERRor:CODE:ALL?	Dequeues all the unread errors and returns their error code.
SYSTem:ERRor:COUNt?	Queries the unread error count.
SYSTem:HOST? *SYSTem:HOST {}*	Queries and sets the host type.
SYSTem:COMMunicate:SERial:BAUD? *SYSTem:COMMunicate:SERial:BAUD {}*	Queries and sets baud rate for the serial port.
SYSTem:COMMunicate:TYPE? *SYSTem:COMMunicate:TYPE {}*	Queries and sets the type for the serial port.

Table 15. Communication Commands (cont.)

Command Groups	Function
SYSTem:DATE < >,< >,< > *SYSTem:DATE?*	Queries and sets system date.
SYSTem:TIME < >,< >,< > *SYSTem:TIME?*	Queries and sets system time.
SYSTem:LANGuage?SYSTem:LANGuage {}	Queries and sets system language.
SYSTem:RS485:ADDRess <>SYSTem:RS485:ADDRess?	Queries and sets RS485 address.
SYSTem:REBOot	Reboot device.
SYSTem:[:FACTory]:RESET	Factory reset device's settings.
TRENd Commands	
TRENd?	Queries all trend items' name.
TRENd:ADD:ITEM {""}	Adds a specified trend item.
TRENd:REMOve:ITEM {""}	Removes a specified trend item.
TRENd:INTErval:TYPE? *TRENd:INTErval:TYPE {}*	Queries and sets the interval type.
TRENd:INTErval? *TRENd:INTErval <>*	Queries and sets the interval between 2 data points.
TRENd:DATA:ITEM? {""}	Queries data for specified trend item.
WIRing Commands	
WIRing:GROUp? *WIRing:GROUp {}{,}{,}{,}*	Queries and sets the wiring groups.
WIRing:GROUp:STATus? {}{,}{,}{,}	Sets the wiring groups and return whether setting is OK.
ZERO Commands	
ZERO? {“”}	Zero specified ADC channels and return whether zeroing is successful.
Common Commands	
**IDN?*	Queries device identification.
**CLS*	Clears the standard event register, extended event register, and error queue.
**RST*	Reset the standard event register, extended event register, and error queue.

FORMula Commands

[FORMula:]HVF:MOTOr:TYPE?

[FORMula:]HVF:MOTOr:TYPE {}

Description: Queries and sets HVF motor type.

Parameter: For SET command: 3phase|non3phase

Response: For QUERY command: 3phase|non3phase

Example:

TX: HVF:MOTOR:TYPE 3phase

TX: HVF:MOTOR:TYPE?

RX: 3phase

[FORMula:]HVF:NOMinal:TYPE?

[FORMula:]HVF:NOMinal:TYPE {}

Description: Queries and sets HVF voltage nominal value type.

Parameter: For SET command: measured|specified

Response: For QUERY command: measured|specified

Example:

TX: HVF:NOM:TYPE specified

TX: HVF:NOM:TYPE?

RX: specified

[FORMula:]HVF:NOMinal:VALue?

[FORMula:]HVF:NOMinal:VALue <>

Description: Queries and sets HVF specified nominal value, float value.

Parameter: For SET command: Float value, min: 2.4, max: 2.3e+5

Response: For QUERY command: Float value, min: 2.4, max: 2.3e+5

Example:

TX: HVF:NOM:VAL 3.2

TX: HVF:NOM:VAL?

RX: 3.2

[FORMula:]HCF:NOMinal:TYPE?

[FORMula:]HCF:NOMinal:TYPE {}

Description: Queries and sets HCF current nominal value type.

Parameter: For SET command: measured|specified

Response: For QUERY command: measured|specified

Example:

TX: HCF:NOM:TYPE specified

TX: HCF:NOM:TYPE?

RX: specified

[FORMula:]HCF:NOMinal:VALue?

[FORMula:]HCF:NOMinal:VALue <>

Description: Queries and sets HCF specified nominal value, float value.

Parameter: For SET command: Float value, min: 1.0, max: 1.0e+8

Response: For QUERY command: Float value, min: 1.0, max: 1.0e+8

Example:

TX: HVF:NOM:VAL 3.2

TX: HVF:NOM:VAL?

RX: 3.2

HARMonics Commands

HARMonics:ORDer? {""}

Description: Queries the maximum output harmonic order of the specified function's harmonic measurement numeric list data.

Max measurable frequency is 6 kHz, for fundamental frequency < 60Hz, order is 100, otherwise: $order = 6000/(fundamentalfrequency)$

Parameter: For SET command:

"VOLT:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2|AB2|BC2|CA2"

or

"CURR:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2"

Response: For QUERY command:

"VOLT:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2|AB2|BC2|CA2"

or

"CURR:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2"

Example:

TX: HARM:ORDER? "VOLT:1"

RX: 100

HARMonics:DATA? {""}

Description: Queries the specified item's harmonic data.

Response data meets HDLC protocol's iframe, and information bytes consist of up to actual order data (HARMonics:ORDer? {""}).

See Table 16 for the order of data information as part of the iframe.

Table 16. Harmonic Data Information Order: iframe

Item Name	Type	Length
Actual harmonics orders	Integer	4
Amplitudes for every order	Formatted resolution string format	8 * actual_harmonics_orders
Ratio for every order	Formatted resolution string format	8 * actual_harmonics_orders
Relative phases for every order	Formatted resolution string format	8 * actual_harmonics_orders

Parameter: For SET command:

"VOLT:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2|AB2|BC2|CA2"

or

"CURR:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2"

Response: For QUERY command:

"VOLT:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2|AB2|BC2|CA2"

or

"CURR:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2"

Example:

TX: HARM:DATA? "VOLT:1"

RX: …

HARMonics:FFT:ITEM?

HARMonics:FFT:ITEM {""}

Description: Queries and sets FFT item name.

Parameter: For SET command:

"VOLT:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2|AB2|BC2|CA2"

or

"CURR:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2"

Response: For QUERY command:

"VOLT:1|2|3|4|A1|B1|C1|AB1|BC1|CA1|A2|B2|C2|AB2|BC2|CA2"

or

"CURR:1|2|3|4|A1|B1|C1|N1|A2|B2|C2|N2"

Example:

TX: HARM:FFT:ITEM "VOLT:3"

TX: HARM:FFT:ITEM?

RX: "VOLT:3"

HARMonics:FFT:FREQuency?

HARMonics:FFT:FREQuency {}

Description: Queries and sets FFT upper limit frequency for FFT item.

Parameter: For SET command: 100kHz|20kHz|4kHz|1kHz

Response: For QUERY command: 100kHz|20kHz|4kHz|1kHz

Example:

TX: HARM:FFT:FREQ 100kHz

TX: HARM:FFT:FREQ?

RX: 100kHz

HARMonics:FFT:DATA?

Description: Queries FFT item's FFT data.

Parameter: None

Response: Response data meets HDLC protocol's iframe, and information bytes consist of 1024 float value.

Example:

TX: HARM:FFT:DATA?

RX: …

HOLD Commands

HOLD:STARt

Description: Starts HOLD for METER/TREND/PHASOR/HARMONIC.

Parameter: None

Response: None

Example:

TX: HOLD:STARt

HOLD:STOP

Description: Stops HOLD for METER/TREND/PHASOR/HARMONIC.

Parameter: None

Response: None

Example:

TX: HOLD:STOP

HOLD:STATus?

Description: Queries HOLD status for METER/TREND/PHASOR/HARMONIC.

Parameter: None

Response: Started|Stopped

Example:

TX: HOLD:STAT?

RX: Stopped

INPut Commands

INPut#:CURRent:AUTO?

INPut#:CURRent:AUTO {}

Description: Queries and sets the current auto range for the specified channel.

Parameter: For SET command: ON|OFF

Response: For QUERY command: ON|OFF

Example:

TX:

TX: INPUT1:CURR:AUTO?

RX: ON

INPut#:CURRent:RANGe?

INPut#:CURRent:RANGe

Description: Queries and sets the current range for the specified channel.

Parameter: For SET command: 100mA|1A|10A|100mV|1V|10V

Response: For QUERY command: 100mA|1A|10A|100mV|1V|10V

Example:

TX: INPUT1:CURR:RANG 1A

TX: INPUT1:CURR:RANG?

RX: 1A

INPut#:CURRent:RATIo?

INPut#:CURRent:RATIo<>

Description: Queries and sets the current ratio for the specified channel.

Parameter: For SET command: max: 1.0e+4, min: 1.0e-5

Response: For QUERY command: max: 1.0e+4, min: 1.0e-5

Example:

TX: INPUT1:CURR:RATI 12.34

TX: INPUT1:CURR:RATI?

RX: 12.34

INPut#:CURRent:FILTer:LINE?

INPut#:CURRent:FILTer:LINE {}

Description: Queries and sets the current line filter for the specified channel.

Parameter: For SET command: OFF|650Hz|10kHz

Response: For QUERY command: OFF|650Hz|10kHz

Example:

TX: INPUT1:CURR:FILT:LINE 650Hz

TX: INPUT1:CURR:FILT:LINE?

RX: 650Hz

INPut#:CURRent:EXTernal[:SENSor]?

INPut#:CURRent:EXTernal[:SENSor] {}

Description: Queries and sets the status of external sensor for the specified channel.

Parameter: For SET command: ON|OFF

Response: For QUERY command: ON|OFF

Example:

TX: INPUT1:CURR:EXT ON

TX: INPUT1:CURR:EXT?

RX: ON

INPut#:CURRent:EXTernal[:SENSor]:RATIo? {}

Description: Queries the current ratio for the specified channel when external sensor is enabled/disabled.

Parameter: ON|OFF

Response: Max: 1.0e+4, min: 1.0e-5

Example:

TX: INPUT1:CURR:EXT:RATI? OFF

RX: 1000

TX: INPUT1:CURR:EXT:RATI? ON

RX: 1

INPut#:VOLTage:AUTO?

INPut#:VOLTage:AUTO {}

Description: Queries and sets the voltage auto range for the specified channel.

Parameter: For SET command: ON|OFF

Response: For QUERY command: ON|OFF

Example:

TX: INPUT1:VOLT:AUTO ON

TX: INPUT1:VOLT:AUTO?

RX: ON

INPut#:VOLTage:RANGe?

INPut#:VOLTage:RANGe {}

Description: Queries and sets the voltage range for the specified channel.

Parameter: For SET command: 10V|100V|1000V

Response: For QUERY command: 10V|100V|1000V

Example:

TX: INPUT1:VOLT:RANG 10V

TX: INPUT1:VOLT:RANG?

RX: 10V

INPut#:VOLTage:RATIo?

INPut#:VOLTage:RATIo<>

Description: Queries and sets the voltage ratio for the specified channel.

Parameter: For SET command: max: 1.0e+4, min: 1.0e-5

Response: For QUERY command: max: 1.0e+4, min: 1.0e-5

Example:

TX: INPUT1:VOLT:RATI 12.34

TX: INPUT1:VOLT:RATI?

RX: 12.34

INPut#:VOLTage:FILTer:LINE?

INPut#:VOLTage:FILTer:LINE {}

Description: Queries and sets the voltage line filter for the specified channel.

Parameter: For SET command: OFF|650Hz|10kHz

Response: For QUERY command: OFF|650Hz|10kHz

Example:

TX: INPUT1:VOLT:FILT:LINE 650Hz

TX: INPUT1:VOLT:FILT:LINE?

RX: 650Hz

INPUT:MS:MODE?

INPUT:MS:MODE {}

Description: Queries and sets the motor speed mode.

Parameter: For SET command: 10V|A|AB|ABZ

Response: For QUERY command: 10V|A|AB|ABZ

Example:

TX: INPUT:MS:MODE A

TX: INPUT:MS:MODE?

RX: A

INPut:MS:RATIo?

INPut:MS:RATIo<>

Description: Queries and sets the motor speed ratio.

Parameter: For SET command:

While MODE is 10V, RATIO is a float value, max: 1.0e+4, min: 1.0e-5;

While MODE is A|AB|ABZ, RATIO is an integer value, max: 10000, min: 1

Response: For QUERY command:

While MODE is 10V, RATIO is a float value, max: 1.0e+4, min: 1.0e-5;

While MODE is A|AB|ABZ, RATIO is an integer value, max: 10000, min: 1

Example:

TX: INPUT:MS:RATI 1.2

TX: INPUT:MS:RATI?

RX: 1.2

INPut:MT:RATIo?

INPut:MT:RATIo<>

Description: Queries and sets the motor torque ratio.

Parameter: For SET command, max: 1.0e+4, min: 1.0e-5

Response: For QUERY command, max: 1.0e+4, min: 1.0e-5

Example:

TX: INPUT:MT:RATI 1.4

TX: INPUT:MT:RATI?

RX: 1.4

INPut:MS:MODE:RATIo? {}

Description: Queries the motor speed ratio for specified mode.

Parameter: 10V|A|AB|ABZ

Response:

While PARAMETER is 10V, RATIO is a float value, max: 1.0e+4, min: 1.0e-5;

While PARAMETER is A|AB|ABZ, RATIO is an integer value, max: 10000, min: 1

Example:

TX: INPUT:MS:MODE:RATI? ABZ

RX: 1

INTEgral Commands

INTEgral:STARt

Description: Starts integral process.

Parameter: None

Response: None

Example:

TX: INTEgral:STARt

INTEgral:STOP

Description: Stops integral process.

Parameter: None

Response: None

Example:

TX: INTEgral:STOP

INTEgral:RESEt

Description: Resets integral process.

Parameter: None

Response: None

Example:

TX: INTEgral:RESEt

INTEgral:STATus?

Description: Queries INTEGRAL status.

Parameter: None

Response: Started|Stopped

Example:

TX: INTEgral:STAT?

RX: Stopped

MEMory Commands

MEM:DOWNload {},<"">

Description: Starts Kermit transfer protocol to download a file from host to meter. The communication terminal on PC should support Kermit protocol, otherwise file cannot be transferred.

Parameter:

1st parameter is meter's folder name: data|screen|settings|log

2nd parameter is file name (MEM:CAT? {})

Response: Kermit process data

Example:

TX: MEM:DOWNload data,"tt.txt"

RX: …

MEM:UPLoad {},<"">

Description: Starts Kermit transfer protocol to upload a file from meter to host. The communication terminal on PC should support Kermit protocol, otherwise the file cannot be transferred.

Parameter:

1st parameter is meter's folder name: data|screen|settings|log

2nd parameter is file name (MEM:CAT? {})

Response: Kermit process data

Example:

TX: MEM:UPload data,"tt.txt"

RX: …

MEM:CATalog? {}

Description: Lists files in specified folder, response are raw ASCII codes.

Parameter: Meter's folder name: data|screen|settings|log

Response:

Number of files

1st File name

…

Example:

TX: MEM:CAT? screen

RX: 2

Screen_20190902_080857.png

Screen_20190902_080907.png

MEM:CATalog:HDLC? {}

Description: Lists files in specified folder, response are encoded as a HDLC iframe.

Parameter: Meter's folder name: data|screen|settings|log

Response: Response data meets HDLC protocol's iframe. See Table 17 for data in information part of iframe.

Table 17. Memory Data Information: iframe

Item Name	Type
Number of files	String with EOL
1st File name	String with EOL
…	…

Example:

TX: MEM:CAT:HDLC? screen

RX: …

RATE Commands

RATE?

RATE {}

Description: Queries and sets data update rate.

Parameter: For SET command: 100ms|200ms|500ms|1s

Response: For QUERY command: 100ms|200ms|500ms|1s

Example:

TX: RATE 500ms

TX: RATE?

RX: 500ms

SCOPe Commands

SCOPe?

Description: Queries all scope items' name.

Parameter: None

Response: Items are selected from U1|U2|U3|U4|I1|I2|I3|I4, max number of items is 4.

Example:

TX: SCOPe?

RX: U1,U2

SCOPe:TRIGger:METHod:TYPE?

SCOPe:TRIGger:METHod:TYPE {}

Description: Queries and sets trigger method type.

Parameter: For SET command: Positive|Negative|Dual

Response: For QUERY command: Positive|Negative|Dual.

Example:

TX: SCOPe:TRIGger:METHod:TYPE Negative

TX: SCOPe:TRIGger:METHod:TYPE?

RX: Negative

SCOPe:TRIGger:METHod:MODE?

SCOPe:TRIGger:METHod:MODE {}

Description: Queries and sets trigger method mode.

Parameter: For SET command: Auto|Normal|Single

Response: For QUERY command: Auto|Normal|Single

Example:

TX: SCOPe:TRIGger:METHod:MODE Single

TX: SCOPe:TRIGger:METHod:MODE?

RX: Single

SCOPe:TRIGger:SOURce?

SCOPe:TRIGger:SOURce {}

Description: Queries and sets trigger source.

Parameter: For SET command: U1|U2|U3|U4|I1|I2|I3|I4

Response: For QUERY command: U1|U2|U3|U4|I1|I2|I3|I4

Example:

TX: SCOPe:TRIGger:SOURce I1

TX: SCOPe:TRIGger:SOURce?

RX: I1

SCOPe:TRIGger:POSition:X?

SCOPe:TRIGger:POSition:X <>

Description: Queries and sets trigger position (horizontal).

Parameter: For SET command: min: 0, max: 500

Response: For QUERY command: min: 0, max: 500

Example:

TX: SCOPe:TRIGger:POSition:X 200

TX: SCOPe:TRIGger:POSition:X?

RX: 200

SCOPe:TRIGger:POSition:Y?

SCOPe:TRIGger:POSition:Y <>

Description: Queries and sets trigger position (vertical).

Parameter: For SET command: min: 0, max: 500

Response: For QUERY command: min: 0, max: 500

Example:

TX: SCOPe:TRIGger:POSition:Y 300

TX: SCOPe:TRIGger:POSition:Y?

RX: 300

SCOPe:TIME:DIV?

SCOPe:TIME:DIV {}

Description: Queries and sets time division.

Parameter: For SET command: 500us|1ms|2ms|5ms|10ms|20ms|50ms

Response: For QUERY command: 500us|1ms|2ms|5ms|10ms|20ms|50ms

Example:

TX: SCOPe:TIME:DIV 20ms

TX: SCOPe:TIME:DIV?

RX: 20ms

SCOPe:AMPlitude? {}

SCOPe:AMPlitude {},{},{}

Description: Queries and sets specified item's amplitude_coefficient and amplitude_exponent.

Amplitude is calculated as a*pow(10, b), a is coefficient, and b is exponent.

Parameter:

For QUERY command: U1|U2|U3|U4|I1|I2|I3|I4

For SET command:

1st parameter (item name): U1|U2|U3|U4|I1|I2|I3|I4

2nd parameter (coefficient): 1|2|5

3rd parameter (exponent): -6|-5|-4|…|13|14|15

Response:

For QUERY command:

1st returned value: a*pow(10,b)

2nd returned value: a

3rd returned value: b

Example:

TX: SCOPe?

RX: I4

TX: SCOPe:AMP I4,2,-1

TX: SCOPe:AMP? I4

RX: 0.2,2,-1

SCOPe:ADD:ITEM {}

Description: Adds a specified scope item.

Parameter: U1|U2|U3|U4|I1|I2|I3|I4

Response: None

Example:

TX: SCOPe?

RX: U1,U2

TX: SCOPe:ADD:ITEM I1

TX: SCOPe?

RX: U1,U2,I1

SCOPe:REMOve:ITEM {}

Description: Removes a specified scope item.

Parameter: U1|U2|U3|U4|I1|I2|I3|I4

Response: None

Example:

TX: SCOPe?

RX: U1,U2

TX: SCOPe:REMOve:ITEM U1

TX: SCOPe?

RX: U2

SCOPe:DATA:ITEM? {}

Description: Queries specified item's data.

Parameter: U1|U2|U3|U4|I1|I2|I3|I4

Response: Response data meets HDLC protocol's iframe. See Table 18.

Table 18. Scope Data Information Order: iframe

Item Name					**Type**	**Length**
Frequency					float	4
Specified item's max amplitude					float	4
Trigger status					integer	4
Enumerated Value	**0**	**1**	**2**	**3**		
Status	Wait	Triggered	Finished	Hold		
Number of points is 501, every point include min, max and avg. Replied data are in float mode.					Float	4 * 3 * 501

Example:

TX: SCOPe:DATA:ITEM? U1

RX: …

SCOPe:HOLD {STOP}

Description: Stops hold while in HOLD status.

Parameter: STOP

Response: None

Example:

TX: SCOPe:HOLD STOP

SENSe Commands

[SENSe:][FORMatted:]DATA? {""}

Description: Queries the specified numeric item's formatted value with unit/status/resolution.

Parameter: See Table 3

Response: Formatted value with unit/status/resolution

Example:

TX: DATA? "CURR:DC:3"

RX: -0.25mA

[SENSe:]RAWData? {""}

Description: Queries the specified numeric item's raw value without unit/status/resolution.

Parameter: See Table 3

Response: Raw value with unit/status/resolution

Example:

TX: RAWData? "CURR:DC:3"

RX: -0.000256242

SYNC Commands

SYNC#[:SOURce]?

SYNC#[:SOURce] {}

Description: Queries and sets the sync source for specified wiring group. # may be 1|2|3|4.

Parameter: For SET command: U1|U2|U3|U4|I1|I2|I3|I4

Response: For QUERY command: U1|U2|U3|U4|I1|I2|I3|I4

Example:

TX: SYNC1 I1

TX: SYNC1?

RX: I1

SYNC#[:SOURce]:STATus? {}

Description: Sets the sync source for specified wiring group and returns whether setting is OK. # may be 1|2|3|4.

Some seconds may need to set the wiring group, so if Fail is returned, to retry the setting.

Parameter: U1|U2|U3|U4|I1|I2|I3|I4

Response: Ok|Fail

Example:

TX: SYNC1:STAT? I1

RX: Ok

SYSTem Commands

SYSTem:REMote {}

Description: Enables/disables remote mode.

Parameter: ON|OFF

Response: None

Example:

TX: SYSTem:REMote ON

SYSTem:ERRor[:NEXT]?

Description: Dequeues one error and returns its error code and error name.

Parameter: None

Response:

1st returned: error code

2nd returned: error name

Example:

TX: SYST:REMote

TX: SYST:ERR?

RX: -109,"Missing parameter"

SYSTem:ERRor:ALL?

Description: Dequeues all the unread errors and returns their error code and error name.

Parameter: None

Response:

1st returned: 1st error's code

2nd returned: 1st error's name

3rd returned: 2nd error's code

4th returned: 2nd error's name

...

Example:

TX: SYST:REMote

TX: SYNC5?

TX: SYST:ERR:ALL?

RX: -109,"Missing parameter",-102,"Syntax error"

SYSTem:ERRor:CODE[:NEXT]?

Description: Dequeues one error and returns its error code and error name.

Parameter: None

Response:

1st returned: error code

Example:

TX: SYST:REMote

TX: SYST:ERR:CODE?

RX: -109

SYSTem:ERRor:CODE:ALL?

Description: Dequeues all the unread errors and returns their error code and error name.

Parameter: None

Response:

1st returned: 1st error's code

2nd returned: 2nd error's code

…

Example:

TX: SYST:REMote

TX: SYNC5?

TX: SYST:ERR:CODE:ALL?

RX: -109,-102

SYSTem:ERRor:COUNt?

Description: Queries the unread error count.

Parameter: None

Response: Errors' count

Example:

TX: SYST:REMote

TX: SYNC5?

TX: SYST:ERR:COUNt?

RX: 2

SYSTem:HOST?

SYSTem:HOST {}

Description: Queries and sets the host type. When host type is COMPUTER, ECHO mode is enabled; otherwise, ECHO mode is disabled.

Parameter: For SET command: TERMINAL|COMPUTER

Response: For QUERY command: TERMINAL|COMPUTER

Example:

TX: SYSTem:HOST COMPUTER

TX: SYSTem:HOST?

RX: COMPUTER

SYSTem:COMMunicate:SERial:BAUD?

SYSTem:COMMunicate:SERial:BAUD {}

Description: Queries and sets serial port's baud rate. For USB and RS-485 ports, default baud rate is 115200.

Parameter: For SET command: 9600|9200|38400|57600|115200|230400|460800|576000|921600

Response: For QUERY command: 9600|9200|38400|57600|115200|230400|460800|576000|921600

Example:

TX: SYSTem:COMM:SER:BAUD 115200

TX: SYSTem:COMM:SER:BAUD?

RX: 115200

SYSTem:COMMunicate:TYPE?

SYSTem:COMMunicate:TYPE {}

Description: Queries and sets serial port's type. While converting from USB to RS485, baud rate will be changed to be 921600, and all commands need add a prefix ":RS485_ADDR".

Parameter: For SET command: USB|RS485

Response: For QUERY command: USB|RS485

Example:

TX: SYSTem:COMM:TYPE USB

TX: SYSTem:COMM:TYPE?

RX: USB

If meter's RS485 address is 1, then:

TX: SYSTem:COMM:TYPE RS485

TX: :1SYSTem:COMM:TYPE? *--Note: Use this command when baud rate is 921600*

RX: :1RS485

SYSTem:DATE < >,< >,< >

SYSTem:DATE?

Description: Queries and sets system date.

Parameter: For SET command:

1st parameter: year;

2nd parameter: month;

3rd parameter: day

Response: For QUERY command:

1st returned value: year;

2nd returned value: month;

3rd returned value: day

Example:

TX: SYSTem:DATE 2019,07,19

TX: SYSTem:DATE?

RX: 2019,07,19

SYSTem:TIME < >,< >,< >

SYSTem:TIME?

Description: Queries and sets system time.

Parameter: For SET command:

1st parameter: hour;

2nd parameter: minute;

3rd parameter: second

Response: For QUERY command:

1st returned value: hour;

2nd returned value: minute;

3rd returned value: day

Example:

TX: SYSTem:TIME 17,31,30

TX: SYSTem:TIME?

RX: 17,31,32

SYSTem:LANGuage?

SYSTem:LANGuage {}

Description: Queries and sets system language.

Parameter: For SET command: CHN|ENG

Response: For QUERY command: CHN|ENG

Example:

TX: SYSTem:LANG ENG

TX: SYSTem:LANG?

RX: ENG

SYSTem:RS485:ADDRess <>

SYSTem:RS485:ADDRess?

Description: Queries and sets RS485 address. Default RS485 address is 1.

Parameter: For SET command: 1|2|3|4|5|6|7|8|9

Response: For QUERY command: 1|2|3|4|5|6|7|8|9

Example:

TX: SYSTem:RS485:ADDR 2

TX: SYSTem:RS485:ADDR?

RX: 2

SYSTem:REBOot

Description: Reboot meter. Rebooting process needs about 80 seconds.

Parameter: None

Response: None

Example:

Example

TX: SYSTem:REBOot

SYSTem:[:FACTory]:RESET

Description: Factory reset the meter settings.

Parameter: None

Response: None

Example:

TX: SYSTem:RESET

TRENd Commands

TRENd?

Description: Queries all trend items' name.

Parameter: None

Response: Items refer to Table 3, max number of items is 4.

Example:

TX: TRENd?

RX: "volt:rms:2","volt:rms:4"

TRENd:ADD:ITEM {""}

Description: Adds a specified trend item.

Parameter: Items refer to Table 3

Response: None

Example:

TX: TRENd?

RX: "volt:rms:2","volt:rms:4"

TX: TRENd:ADD:ITEM "volt:rms:3"

TX: TRENd?

RX: "volt:rms:3","volt:rms:2","volt:rms:4"

TRENd:REMOve:ITEM {""}

Description: Removes a specified trend item.

Parameter: Items refer to Table 3

Response: None

Example:

TX: TRENd?

RX: "volt:rms:3","volt:rms:2","volt:rms:4"

TX: TRENd:REMOve:ITEM "volt:rms:3"

TX: TRENd?

RX: "volt:rms:2","volt:rms:4"

TRENd:INTErval:TYPE?

TRENd:INTErval:TYPE {}

Description: Queries and sets the interval type.

Parameter: For SET command: SPECIFIED|RATE

Response: For QUERY command: SPECIFIED|RATE

Example:

TX: TRENd:INTErval:TYPE RATE

TX: TRENd:INTErval:TYPE?

RX: RATE

TRENd:INTErval?

TRENd:INTErval <>

Description: Queries and sets the interval between 2 data points. Unit is second.

Parameter: For SET command: min: 1, max: 24*60*60 - 1

Response: For QUERY command: min: 1, max: 24*60*60 - 1

Example:

TX: TRENd:INTErval 3

TX: TRENd:INTErval?

RX: 3

TRENd:DATA:ITEM? {""}

Description: Queries specified item's data.

Parameter: Items refer to Table 3

Response: Response data meets HDLC protocol's iframe. See Table 19.

Table 19. Trend Data Information Order: iframe

Item Name	Type	Length
Final point’s sample timestamp since 1970-01-01T00:00:00.000	Long integer	8
Number of points	Integer	4
Number of points is 2002, replied data are in float mode.	Float	4 * 2002

Example:

TX: TRENd:DATA:ITEM? "volt:rms:4"

RX: …

WIRing Commands

WIRing:GROUp?

WIRing:GROUp {}{,}{,}{,}

Description: Queries and sets the wiring groups. If one channel is not grouped, then 1P2W1M is the default value.

Parameter: For SET command: 1P2W1M|3P3W2M|3P4W3M|3P3W3M,

1st parameter: wiring group 1’s name,

2nd parameter: wiring group 2’s name,

3rd parameter: wiring group 3’s name,

4th parameter: wiring group 4’s name

Response: For QUERY command: 1P2W1M|3P3W2M|3P4W3M|3P3W3M,

1st returned value: wiring group 1's name,

2nd returned value: wiring group 2's name,

3rd returned value: wiring group 3's name,

4th returned value: wiring group 4's nameData in information part of iframe are defined as following order:

Example:

TX: WIRing:GROUp 1P2W1M,3P4W3M

TX: WIRing:GROUp?

RX: 1P2W1M,3P4W3M

WIRing:GROUp:STATus? {}{,}{,}{,}

Description: Queries and sets the wiring groups. The wiring group takes several seconds to set up, so if Fail is returned, retry the setting.

Parameter: For SET command: 1P2W1M|3P3W2M|3P4W3M|3P3W3M,

1st parameter: wiring group 1's name,

2nd parameter: wiring group 2's name,

3rd parameter: wiring group 3's name,

4th parameter: wiring group 4's name

Response: Ok|Fail

Example:

TX: WIRing:GROUp:STATus? 1P2W1M,3P4W3M

RX: Ok

ZERO Commands

ZERO? {""}

Description: Zero the specified ADC channels and return whether or not zeroing is successful.

For N6003&N6003+, U4&I4 are not supported.

For N6004&N6003, MT&MS are not supported.

Note

1. *For MS, zero is allowed when speed mode is 10V.*
2. *ZERO command cannot complete when a zeroing is already in process.*
3. *ZERO is done for every range (each ADC channel has 3 ranges), each range requires about 4 seconds and has nothing to do with the ADC channel number.*

Parameter: U1|U2|U3|U4|I1|I2|I3|I4|MT|MS

Response: None

Example:

TX: ZERO? “U1,U2”

RX: Ok,Fail

Common Commands

**IDN?*

Description: Queries device identification.

Parameter: None

Response:

1st returned value: company name,

2nd returned value: model name,

3rd returned value: serial number,

4th returned value: ARM firmware version,

5th returned value: DSP firmware version,

6th returned value: FPGA firmware version

Example:

TX: *IDN?

RX: FLUKE,NORMA_6004+,12345678WS,v4.2.0,v4.2.0,V1.5

**CLS*

Description: Clears the standard event register, extended event register, and error queue.

Parameter: None

Response: None

Example:

TX: *CLS

**RST*

Description: Reset the standard event register, extended event register, and error queue.

Parameter: None

Response: None

Example:

TX: *RST

Error Message

See Table 20 for a list of error messages.

Table 20. Error Messages

Error Code	Error Name
0	No error
-100	Command error
-101	Invalid character
-102	Syntax error
-103	Invalid separator
-104	Data type error
-105	GET not allowed
-108	Parameter not allowed
-109	Missing parameter
-110	Command header error
-111	Header separator error
-112	Program mnemonic too long
-113	Undefined header
-114	Header suffix out of range
-115	Unexpected number of parameters
-120	Numeric data error
-121	Invalid character in number
-123	Exponent too large
-124	Too many digits
-128	Numeric data not allowed
-130	Suffix error
-131	Invalid suffix
-134	Suffix too long
-138	Suffix not allowed
-140	Character data error
-141	Invalid character data
-144	Character data too long
-148	Character data not allowed
-150	String data error
-151	Invalid string data

Table 20. Error Messages (cont.)

Error Code	Error Name
-158	String data not allowed
-160	Block data error
-161	Invalid block data
-168	Block data not allowed
-170	Expression error
-171	Invalid expression
-178	Expression data not allowed
-200	Execution error
-201	Invalid while in local
-202	Settings lost due to rtl
-203	Command protected
-210	Trigger error
-211	Trigger ignored
-212	Arm ignored
-213	Init ignored
-214	Trigger deadlock
-215	Arm deadlock
-220	Parameter error
-221	Settings conflict
-222	Data out of range
-223	Too much data
-224	Illegal parameter value
-225	Out of memory
-226	Lists not same length
-230	Data corrupt or stale
-231	Data questionable
-232	Invalid format
-233	Invalid version
-240	Hardware error
-241	Hardware missing
-250	Mass storage error
-251	Missing mass storage

Table 20. Error Messages (cont.)

Error Code	Error Name
-252	Missing media
-253	Corrupt media
-254	Media full
-255	Directory full
-256	File name not found
-257	File name error
-258	Media protected
-260	Expression error
-261	Math error in expression
-280	Program error
-281	Cannot create program
-282	Illegal program name
-283	Illegal variable name
-284	Program currently running
-285	Program syntax error
-286	Program runtime error
-290	Memory use error
-300	Device-specific error
-310	System error
-311	Memory error
-313	Calibration memory lost
-315	Configuration memory lost
-320	Storage fault
-330	Self-test failed
-350	Queue overflow
-360	Communication error
-363	Input buffer overrun
-365	Time out error