



Model: 9833

## PROGRAMMING MANUAL

Version 1



## Notations

**TEXT** – Denotes a softkey.

**TEXT** – Denotes a front panel button.

# Table of Contents

|                                                                        |          |
|------------------------------------------------------------------------|----------|
| Notations .....                                                        | i        |
| <b>1. Remote Interfaces .....</b>                                      | <b>1</b> |
| A. RS488.2 - GPIB .....                                                | 1        |
| B. USB CDC .....                                                       | 1        |
| C. USB TMC .....                                                       | 1        |
| D. RS-232C .....                                                       | 1        |
| <b>2. SCPI Commands .....</b>                                          | <b>2</b> |
| E. Example query command – Read the serial number .....                | 2        |
| F. Example value set command .....                                     | 2        |
| G. Example value set command .....                                     | 3        |
| H. FETCh   MEASure .....                                               | 3        |
| 1. Command Summary .....                                               | 3        |
| 2. FETCh:VOLTage:AC? - MEASure:VOLTage:AC? .....                       | 3        |
| 3. FETCh:VOLTage:DC? – MEASure:VOLTage:DC? .....                       | 3        |
| 4. FETCh:CURREnt:AC? – MEASure:CURREnt:AC? .....                       | 4        |
| 5. FETCh:CURREnt:DC? – MEASure:CURREnt:DC? .....                       | 4        |
| 6. FETCh:CURREnt:CREStfactor? – MEASure:CURREnt:CREStfactor? .....     | 4        |
| 7. FETCh:CURREnt:INRush? – MEASure:CURREnt:INRush? .....               | 5        |
| 8. FETCh:CURREnt:PEAK:POSitive? – MEASure:CURREnt:PEAK:POSitive? ..... | 5        |
| 9. FETCh:CURREnt:PEAK:NEGative? – MEASure:CURREnt:PEAK:NEGative? ..... | 5        |
| 10. FETCh:FREQuency? – MEASure:FREQuency? .....                        | 6        |
| 11. FETCh:POWer:AC:REAL? – MEASure:POWer:AC:REAL? .....                | 6        |
| 12. FETCh:POWer:AC:APParent? – MEASure:POWer:AC:APParent? .....        | 6        |
| 13. FETCh:POWer:AC:REACtive? – MEASure:POWer:AC:REACtive? .....        | 6        |
| 14. FETCh:POWer:AC:PFACtor? – MEASure:POWer:AC:PFACtor? .....          | 6        |
| I. OUTPut .....                                                        | 7        |
| 1. Command Summary .....                                               | 7        |
| 2. OUTPut:[STATe] .....                                                | 7        |
| 3. OUTPut:MODE .....                                                   | 7        |
| 4. OUTPut:PROTection:CLEar .....                                       | 8        |
| 5. OUTPut:LIMit:VOLTage:AC .....                                       | 8        |
| 6. OUTPut:LIMit:VOLTage:DC:POSitive .....                              | 8        |
| 7. OUTPut:LIMit:VOLTage:DC:NEGative .....                              | 9        |
| 8. OUTPut:LIMit:CURREnt .....                                          | 9        |
| 9. OUTPut:LIMit:CURREnt:DELay .....                                    | 9        |
| 10. OUTPut:LIMit:POWer .....                                           | 10       |

|     |                                           |    |
|-----|-------------------------------------------|----|
| J.  | [SOURce] .....                            | 10 |
| 1.  | Command Summary .....                     | 10 |
| 2.  | [SOURce]:VOLTage:RANGe.....               | 11 |
| 3.  | [SOURce]:VOLTage:AC.....                  | 12 |
| 4.  | [SOURce]:VOLTage:DC.....                  | 12 |
| 5.  | [SOURce]:CURRent:INRush:START.....        | 12 |
| 6.  | [SOURce]:CURRent:INRush:INTerval .....    | 13 |
| 7.  | [SOURce]:FREQuency .....                  | 13 |
| 8.  | [SOURce]:SYNChronous.....                 | 13 |
| 9.  | [SOURce]:PHASe .....                      | 14 |
| 10. | [SOURce]:CONFigure:COUPling .....         | 14 |
| 11. | [SOURce]:CONFigure:TIMer[:STATe].....     | 15 |
| 12. | [SOURce]:CONFigure:TIMer:COUNT.....       | 15 |
| 13. | [SOURce]:CONFigure:EXTernal.....          | 16 |
| 14. | [SOURce]:CONFigure:INHibit.....           | 16 |
| 15. | [SOURce]:CONFigure:TRANSient.....         | 17 |
| 16. | [SOURce]:STEP:VOLTage:AC.....             | 17 |
| 17. | [SOURce]:STEP:VOLTage:DC.....             | 18 |
| 18. | [SOURce]:STEP:DVOLTage:AC .....           | 18 |
| 19. | [SOURce]:STEP:DVOLTage:DC.....            | 19 |
| 20. | [SOURce]:STEP:FREQuency .....             | 19 |
| 21. | [SOURce]:STEP:DFREQuency .....            | 19 |
| 22. | [SOURce]:STEP:TIME .....                  | 20 |
| 23. | [SOURce]:STEP:COUNT .....                 | 20 |
| 24. | [SOURce]:STEP:SYNChronous.....            | 21 |
| 25. | [SOURce]:STEP:PHASe .....                 | 21 |
| 26. | List mode programming .....               | 21 |
| 27. | [SOURce]:LIST:NUMber .....                | 22 |
| 28. | LIST:CONF:WRITE.....                      | 23 |
| 29. | [SOURce]:LIST:BASE.....                   | 23 |
| 30. | [SOURce]:LIST:SYNChronous .....           | 24 |
| 31. | [SOURce]:LIST:PHASe .....                 | 24 |
| 32. | [SOURce]:LIST:COUNT.....                  | 24 |
| 33. | LIST:WRITE.....                           | 25 |
| 34. | LIST:CONF:STEP?.....                      | 26 |
| 35. | LIST:CONF:READ? .....                     | 26 |
| 36. | LIST:READ?.....                           | 26 |
| 37. | LIST:START:QUERY .....                    | 27 |
| 38. | [SOURce]:LIST:STEPno .....                | 28 |
| 39. | [SOURce]:LIST:VOLTage:LEVel:AC:START..... | 28 |
| 40. | [SOURce]:LIST:VOLTage:LEVel:AC:END .....  | 28 |

|     |                                                      |    |
|-----|------------------------------------------------------|----|
| 41. | [SOURce]:LIST:VOLTage:LEVel:DC:START .....           | 29 |
| 42. | [SOURce]:LIST:VOLTage:LEVel:DC:END .....             | 29 |
| 43. | [SOURce]:LIST:FREQuency:LEVel:START .....            | 29 |
| 44. | [SOURce]:LIST:FREQuency:LEVel:END .....              | 30 |
| 45. | [SOURce]:LIST:CLear .....                            | 30 |
| 46. | [SOURce]:PULSe:VOLTage:AC .....                      | 30 |
| 47. | [SOURce]:PULSe:VOLTage:DC .....                      | 31 |
| 48. | [SOURce]:PULSe:FREQuency .....                       | 31 |
| 49. | [SOURce]:PULSe:SYNChronous .....                     | 31 |
| 50. | [SOURce]:PULSe:PHASe .....                           | 32 |
| 51. | [SOURce]:PULSe:DUTY .....                            | 32 |
| 52. | [SOURce]:PULSe:PERIOd .....                          | 33 |
| 53. | [SOURce]:PULSe:COUNt .....                           | 33 |
| 54. | [SOURce]:FUNCTion:SHAPE .....                        | 33 |
| 55. | [SOURce]:FUNCTion:SHAPE:A .....                      | 34 |
| 56. | [SOURce]:FUNCTion:SHAPE:B .....                      | 34 |
| 57. | [SOURce]:FUNC:SHAP:USER:TABLE:READ .....             | 35 |
| 58. | [SOURce]:FUNCTion:SHAPE:USER:TABLE:DATA .....        | 35 |
| 59. | [SOURce]:FUNCTion:SHAPE:USER:TABLE:DATA? <NR1> ..... | 36 |
| K.  | SYStem .....                                         | 37 |
| 1.  | Command Summary .....                                | 37 |
| 2.  | SYStem:BEEP .....                                    | 37 |
| 3.  | SYStem:BRIGhtness .....                              | 38 |
| 4.  | SYStem:KEY:LOCK .....                                | 38 |
| 5.  | SYStem:TIME .....                                    | 38 |
| 6.  | SYStem:DATE .....                                    | 39 |
| 7.  | SYStem:RECall:DEFault .....                          | 39 |
| 8.  | SYStem:SERial? .....                                 | 39 |
| 9.  | SYStem:MODEl? .....                                  | 39 |
| 10. | SYStem:MANUFacture? .....                            | 40 |
| 11. | SYStem:INTERFACE .....                               | 40 |
| 12. | SYStem:ERRor? .....                                  | 40 |
| 13. | SYStem:VERSion? .....                                | 40 |
| 14. | SYStem:GPIB:ADDRes .....                             | 41 |
| 15. | SYStem:IP:CONFig .....                               | 41 |
| 16. | SYStem:IP:ADDRes .....                               | 41 |
| 17. | SYStem:IP:GATEway .....                              | 41 |
| 18. | SYStem:IP:MASK .....                                 | 42 |
| L.  | USBFlash .....                                       | 42 |
| 1.  | Command Summary .....                                | 42 |
| 2.  | USBFlash:SAVE:CONFiguration .....                    | 42 |

|                                        |    |
|----------------------------------------|----|
| 3. USBFlash:RECall:CONFIguration ..... | 43 |
| 4. USBFlash:SAVE:SCReen .....          | 43 |
| 5. USBFlash:SAVE:COMPlete? .....       | 43 |

## **1. Remote Interfaces**

- A. RS488.2 - GPIB**
- B. USB CDC**
- C. USB TMC**
- D. RS-232C**

## 2. SCPI Commands

General syntax for SCPI commands or query is a “:” (colon) separated string with either a “?” or an argument following the command string separated by a SPACE character. Commands are terminated by the linefeed character (0x10, “\n”). Arguments are listed using “< ARG | ARG | ... >” in the following descriptions. The “ARG” will be from the following:

| Symbol    | Format                                                    |
|-----------|-----------------------------------------------------------|
| <NR1>     | Number with an implicit decimal point at the end. Ex. 100 |
| <NRf>     | Number with an explicit decimal point. Ex. 100.5          |
| <Boolean> | Boolean value. Ex. 0   OFF   1   ON                       |
| <STR>     | Named value. Ex. “phase”                                  |

Most commands have a long and short format, the capitalized portion of the name is the short form, and the complete name is the long form. For example, the FETch can be sent as either “fet” or fetch. Short and long forms may be mixed in a command string. An optional portion of a command is shown within “[ ]”.

### E. Example query command – Read the serial number

“

*XX.XX SYStem*

*XX.XX.YY SERial?*

“

This command is in the SYStem section of the SCPI command set. The command is a “query” command and will return a string containing the serial number. As a “query” it must end with a “?” character. The value returned will be terminated by 2 characters “\r” “\n” (0x0D 0x0A).

Command string: “sys:ser?\n” OR “system:serial?\n” OR “sys:serial?\n” etc...

### F. Example value set command

“

*XX.XX SYStem*

*XX.XX.YY SERial?*

“



## G. Example value set command

“

XX.XX SYStem

XX.XX.YY SERial?

“

## H. FETCh | MEASure

### 1. Command Summary

The following commands use either FETCH or MEASURE as the root. For example, “FETCH:FREQ?”, and “MEASURE:FREQ?” are equivalent. Fetch and Measure function identically, they trigger measurement and return the value measured.

[MEASure:VOLTage:AC?](#)  
[MEASure:VOLTage:DC?](#)  
[MEASure:CURRent:AC?](#)  
[MEASure:CURRent:DC?](#)  
[MEASure:CURRent:CREStfactor?](#)  
[MEASure:CURRent:INRush?](#)  
[MEASure:CURRent:PEAK:POSitive?](#)  
[MEASure:CURRent:PEAK:NEGative?](#)  
[MEASure:POWer:AC:REAL?](#)  
[MEASure:POWer:AC:APParent?](#)  
[MEASure:POWer:AC:REACTive?](#)  
[MEASure:POWer:AC:PFACTOR?](#)

### 2. *FETCH:VOLTage:AC? - MEASure:VOLTage:AC?*

**Description** Query the RMS AC voltage.

**Query Syntax** FETCH:VOLTage:AC? - MEASure:VOLTage:AC?

**Returned Parameters** <NRf>

Value: 0-300V or 0-150V depending on the range setting.

**Related Commands** [\[SOURce\]:VOLTage:AC](#)

### 3. *FETCH:VOLTage:DC? – MEASure:VOLTage:DC?*

**Description** Query the DC voltage.

**Query Syntax** `FEtCh:VOLTage:DC? – MEASure:VOLTage:DC?`

**Returned Parameters** `<NR1>`

Value: -424 to 424 or -212 to 212 depending on the range setting.

**Related Commands** [\[SOURce\]:VOLTage:DC](#)

4. *FEtCh:CURRent:AC? – MEASure:CURRent:AC?*

**Description** Query the AC current (Amps).

**Query Syntax** `FEtCh:CURRent:AC? – MEASure:CURRent:AC?`

**Returned Parameters** `<NRf>`

Value: Returns a floating point number.

**Related Commands** [OUTPut:LIMit:CURRent](#)

5. *FEtCh:CURRent:DC? – MEASure:CURRent:DC?*

**Description** Query the DC current being delivered.

**Query Syntax** `FEtCh:CURRent:DC? – MEASure:CURRent:DC?`

**Returned Parameters** `<NRf>`

Value: A floating point

**Related Commands** [\[SOURce\]:VOLTage:DC](#),  
[OUTPut:LIMit:VOLTage:DC:POSitive](#),  
[OUTPut:LIMit:VOLTage:DC:NEGative](#)

6. *FEtCh:CURRent:CREStfactor? – MEASure:CURRent:CREStfactor?*

**Description** Query the Crest Factor measured by the unit.

**Query Syntax** `FEtCh:CURRent:CREStfactor?`  
`MEASure:CURRent:CREStfactor?`

**Returned Parameters** `<NRf>`

7. *FETCh:CURRent:INRush? – MEASure:CURRent:INRush?*

**Description** Query the measured inrush current delivered to the load.

**Note** Value valid only when the supply output is on. Disabling output clears the value and returns 0.000000.

**Query Syntax** FETCh:CURRent:INRush?  
MEASure:CURRent:INRush?

**Returned Parameters** <NRf>

**Related Commands** [source:current:inrush:start](#)  
[source:current:inrush:interval](#)

8. *FETCh:CURRent:PEAK:POSitive? – MEASure:CURRent:PEAK:POSitive?*

**Description** Query the positive peak current delivered to the load.

**Note** This is the present value at the time the command is issued.

**Query Syntax** FETCh:CURRent:PEAK:POSitive?  
MEASure:CURRent:PEAK:POSitive?

**Returned Parameters** <NRf>

9. *FETCh:CURRent:PEAK:NEGative? – MEASure:CURRent:PEAK:NEGative?*

**Description** Query the negative peak current delivered to the load.

**Note** This is the present value at the time the command is issued.

**Query Syntax** FETCh:CURRent:PEAK:NEGative?  
MEASure:CURRent:PEAK:NEGative?

**Returned Parameters** <NRf>

10. *FETCh:FREQuency? – MEASure:FREQuency?*

**Description** Query the frequency at the output.

**Query Syntax** FETCh:FREQuency?  
MEASure:FREQuency?

**Returned Parameters** <NRf>

**Related Commands** [\[SOURce\]:FREQuency](#)

11. *FETCh:POWer:AC:REAL? – MEASure:POWer:AC:REAL?*

**Description** Query the real component of the power consumed by the load.

**Query Syntax** FETCh:POWer:AC:REAL?  
MEASure:POWer:AC:REAL?

**Returned Parameters** <NRf>

12. *FETCh:POWer:AC:APParent? – MEASure:POWer:AC:APParent?*

**Description** Query the apparent power consumed by the load.

**Query Syntax** FETCh:POWer:AC:APParent?  
MEASure:POWer:AC:APParent?

**Returned Parameters** <NRf>

13. *FETCh:POWer:AC:REACtive? – MEASure:POWer:AC:REACtive?*

**Description** Query the reactive power consumed by the load.

**Query Syntax** FETCh:POWer:AC:REACtive?  
MEASure:POWer:AC:REACtive?

**Returned Parameters** <NRf>

14. *FETCh:POWer:AC:PFACtor? – MEASure:POWer:AC:PFACtor?*

**Description** Query the power factor of the output.

**Query Syntax**    FETCh:POWer:AC:PFACtor?  
MEASure:POWer:AC:PFACtor?

**Returned Parameters**    <NRf>

## I. OUTPut

This is some text that describes the section.

### 1. *Command Summary*

[OUTPut\[:STATe\]](#)  
[OUTPut:MODE](#)  
[OUTPut:PROTection:CLear](#)  
[OUTPut:LIMit:VOLTage:AC](#)  
[OUTPut:LIMit:VOLTage:DC:POSitive](#)  
[OUTPut:LIMit:VOLTage:DC:NEGative](#)  
[OUTPut:LIMit:CURREnt](#)  
[OUTPut:LIMit:CURREnt:DELay](#)  
[OUTPut:LIMit:POWer](#)

### 2. *OUTPut:[STATe]*

**Description**    Turn the OUTPut on or off, or query the present state.

**Command Syntax**    OUTPut[:STATe] <Bool>

**Parameters**    0 | OFF | 1 | ON

**Examples**    OUTP on

**Query Syntax**    OUTPut:[STATe]?

**Returned Parameters**    <STR>

Returns the string "ON" or "OFF"

### 3. *OUTPut:MODE*

**Description**    Set or query the waveform operation to output.

Use this to switch between normal output mode or to run one of the other programmatic modes.

**Command Syntax**    OUTPut:MODE <STR>

**Parameters** NORMal | STEP | LIST | PULSE

**Examples** OUTP:MOD step

**Query Syntax** OUTPut:MODE?

**Returned Parameters** <STR>

"NORMal", "STEP", "LIST", "PULSE"

#### 4. *OUTPut:PROTection:CLEar*

**Description** Reset the protection state in the event of a protection fault. When a limit is reached, the output will turn off and the alarm will sound. This command clears the alarm.

**Command Syntax** OUTPut:PROTection:CLEar

**Parameters** None

**Examples** output:prot:cle

#### 5. *OUTPut:LIMit:VOLTage:AC*

**Description** Set or query the AC voltage limit.

**Note** This limit applies to AC and AC+DC modes and limits based on the RMS output voltage.

**Command Syntax** OUTPut:LIMit:VOLTage:AC <NR2>

**Parameters** Output voltage limit as a floating point number. 310V for example.

**Examples** outp:lim:volt:ac 310

**Query Syntax** OUTPut:LIMit:VOLTage:AC?

**Returned Parameters** <NR2>

Example: 310.000000

#### 6. *OUTPut:LIMit:VOLTage:DC:POSitive*

**Description** Set or query the upper DC voltage limit.

**Command Syntax** OUTPut:LIMit:VOLTage:DC:POSitive <NR2>

**Parameters** Positive voltage limit

**Examples** output:limit:voltage:dc:positive 125

**Query Syntax** OUTPut:LIMit:VOLTage:DC:POSitive?

**Returned Parameters** <NR2>

Example: 125.000000

7. *OUTPut:LIMit:VOLTage:DC:NEGative*

**Description** Set or query the lower DC voltage limit.

**Command Syntax** OUTPut:LIMit:VOLTage:DC:NEGative <NR2>

**Parameters** Negative voltage limit

**Examples** OUTP:LIM:VOLT:DC:NEG -100

**Query Syntax** OUTPut:LIMit:VOLTage:DC:NEGative?

**Returned Parameters** <NR2>

8. *OUTPut:LIMit:CURREnt*

**Description** Set or query the current limit in amps.

**Command Syntax** OUTPut:LIMit:CURREnt <NR2>

**Examples** OUTP:LIM:CURR 12.34

**Query Syntax** OUTPut:LIMit:CURREnt?

**Returned Parameters** <NR2>

9. *OUTPut:LIMit:CURREnt:DELay*

**Description** Set the delay, in milliseconds, for triggering output protection.

**Command Syntax** OUTPut:LIMit:CURREnt:DELay <NR2>

**Parameters** <NR2>

0-10000ms

**Examples** OUTPut:LIMit:CURRent:DEL 133

**Query Syntax** OUTPut:LIMit:CURRent:DElay?

**Returned Parameters** <NR1>

**Related Commands** [OUTPut:LIMit:CURRent](#)

## 10. *OUTPut:LIMit:POWer*

**Description** Set or query the OUTPut power limit in watts.

**Command Syntax** OUTPut:LIMit:POWer <NR2>

**Parameters** 0 to 3300 depending on unit.

**Examples** OUTP:LIM:POW 3300

**Query Syntax** OUTPut:LIMit:POWer?

**Returned Parameters** <NR2>

## J. [SOURce]

### 1. *Command Summary*

[\[SOURce\]:VOLTage:RANGe](#)

[\[SOURce\]:VOLTage:AC](#)

[\[SOURce\]:VOLTage:DC](#)

[\[SOURce\]:CURRent:INRush:STARt](#)

[\[SOURce\]:CURRent:INRush:INTerval](#)

[\[SOURce\]:FREQuency](#)

[\[SOURce\]:SYNChronous](#)

[\[SOURce\]:PHASe](#)

[\[SOURce\]:CONFigure:COUPling](#)

[\[SOURce\]:CONFigure:TIMer\[:STATe\]](#)

[\[SOURce\]:CONFigure:TIMer:COUNt](#)

[\[SOURce\]:CONFigure:EXTernal](#)

[\[SOURce\]:CONFigure:INHibit](#)

[\[SOURce\]:CONFigure:TRANSient](#)

[\[SOURce\]:STEP:VOLTage:AC](#)

[\[SOURce\]:STEP:VOLTage:DC](#)

[\[SOURce\]:STEP:DVOLTage:AC](#)

[\[SOURce\]:STEP:DVOLTage:DC](#)



[\[SOURce\]:STEP:FREQuency](#)  
[\[SOURce\]:STEP:DFREquency](#)  
[\[SOURce\]:STEP:TIME](#)  
[\[SOURce\]:STEP:COUNT](#)  
[\[SOURce\]:STEP:SYNChronous](#)  
[\[SOURce\]:STEP:PHASe](#)  
[\[SOURce\]:LIST:NUMber](#)  
[\[SOURce\]:LIST:BASE](#)  
[\[SOURce\]:LIST:SYNChronous](#)  
[\[SOURce\]:LIST:PHASe](#)  
[\[SOURce\]:LIST:COUNT](#)  
[\[SOURce\]:LIST:STEPno](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:AC:START](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:AC:END](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:DC:START](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:DC:END](#)  
[\[SOURce\]:LIST:FREQuency:LEVel:START](#)  
[\[SOURce\]:LIST:FREQuency:LEVel:END](#)  
[\[SOURce\]:LIST:DWEL](#)  
[\[SOURce\]:LIST:STEP:COUNT](#)  
[\[SOURce\]:LIST:CLEar](#)  
[\[SOURce\]:LIST:SAVE](#)  
[\[SOURce\]:PULSe:VOLTage:AC](#)  
[\[SOURce\]:PULSe:VOLTage:DC](#)  
[\[SOURce\]:PULSe:FREQuency](#)  
[\[SOURce\]:PULSe:SYNChronous](#)  
[\[SOURce\]:PULSe:PHASe](#)  
[\[SOURce\]:PULSe:DUTY](#)  
[\[SOURce\]:PULSe:PERIOd](#)  
[\[SOURce\]:PULSe:COUNT](#)  
[\[SOURce\]:FUNCtion:SHAPE](#)  
[\[SOURce\]:FUNCtion:SHAPE:A](#)  
[\[SOURce\]:FUNCtion:SHAPE:B](#)

## 2. *[SOURce]:VOLTage:RANGe*

**Description** Set the voltage range to HIGH (300V) or LOW (150V).

**Command Syntax** [SOURce]:VOLTage:RANGe <STR>

**Parameters** <HIGH | LOW>

**Examples** sour:volt:rang high

**Query Syntax** [SOURce]:VOLTage:RANGe?

**Returned Parameters** <STR>  
"HIGH" or "LOW"

3. *[SOURce]:VOLTage:AC*

**Description** Set or query the AC voltage OUTPut.

**Command Syntax** [SOURce]:VOLTage:AC <NR2>

**Parameters** 0-150, 0-300

**Examples** SOUR:VOLT:AC 110.0

**Query Syntax** [SOURce]:VOLTage:AC?

**Returned Parameters** <NR2>

**Related Commands** [OUTPut:LIMit:VOLTage:AC](#)

4. *[SOURce]:VOLTage:DC*

**Description** Set or query the DC OUTPut voltage setting in Volts.

**Command Syntax** [SOURce]:VOLTage:DC <NR2>

**Parameters** 0 to  $\pm 212$ , 0 to  $\pm 424$

**Examples** SOUR:VOLT:DC -100

**Query Syntax** SOUR:VOLT:DC?

**Returned Parameters** <NR2>

**Related Commands** [OUTPut:LIMit:VOLTage:DC:NEGative](#)

[OUTPut:LIMit:VOLTage:DC:POSitive](#)

5. *[SOURce]:CURRent:INRush:START*

**Description** Set or query the start time for measuring the inrush current being supplied. Parameter is the delay in milliseconds.

**Command Syntax** [SOURce]:CURRent:INRush:START <NR1>

**Parameters** 0-10000

**Examples** SOUR:CURR:INR:STARt 300

**Query Syntax** [SOURce]:CURRent:INRush:STARt?

**Returned Parameters** <NR1>

6. *[SOURce]:CURRent:INRush:INTerval*

**Description** This sets the duration of the inrush current measurement.

**Command Syntax** [SOURce]:CURRent:INRush:INTerval <NR1>

**Parameters** 0-10000

**Examples** SOUR:CURR:INR:INT 200

**Query Syntax** [SOURce]:CURRent:INRush:INTerval?

**Returned Parameters** <NR1>

7. *[SOURce]:FREQuency*

**Description** Set or query the AC OUTPut frequency.

**Command Syntax** [SOURce]:FREQuency <NR2>

**Parameters** 43-1200.000000

**Examples** Sour:frequency 60

**Query Syntax** [SOURce]:FREQuency?

**Returned Parameters** <NR2>

8. *[SOURce]:SYNChronous*

**Description** Set or query the synchronization [SOURce] for starting waveform OUTPut. Phase mode starts the waveform starts at a specified point in a cycle with the positive going zero cross being 0 degrees.

**Command Syntax** [SOURce]:SYNChronous <STR>

**Parameters** <IMMediate | PHASe>

**Examples** sync phase

**Query Syntax** [SOURce]:SYNChronous?

**Returned Parameters** <STR>

“PHASe” or “IMMediate”

**Related Commands** [\[SOURce\]:PHASe](#)

## 9. *[SOURce]:PHASe*

**Description** When “[SOURce]:SYNChronous” is set to phase, this command sets or queries the phase angle to use. Value is in degrees.

**Command Syntax** [SOURce]:PHASe <NR2>

**Parameters** 0-359.7

**Examples** sour:phase 45.0

**Query Syntax** [SOURce]:PHASe?

**Returned Parameters** <NR2>

**Related Commands** [\[SOURce\]:SYNChronous](#)

## 10. *[SOURce]:CONFigure:COUPling*

**Description** When using the “Digital IO” and the “Level” function, this sets or queries which mode to use. “AC” will vary the OUTPut AC amplitude according to the “Level” voltage supplied. When in “DC”, this varies the DC voltage of the OUTPut via the “Level” voltage supplied. “AC/DC” is not used in “Level” mode.

When not in “Level” mode, this changes the set of values settable by the front panel for user input. AC mode allows setting of the AC amplitude, DC, the DC voltage and AC/DC allows for both.

**Command Syntax** [SOURce]:CONFigure:COUPling <STR>

**Parameters** <AC | DC | ACDC>

**Examples** source:configure:coup AC

**Query Syntax** [SOURce]:CONFigure:COUPling?

**Returned Parameters** <STR>

"AC", "DC", "ACDC"

#### 11. [SOURce]:CONFigure:TIMer[:STATe]

**Description** This enables or disables the OUTPut countdown timer.

**Note** When this is active, the display of the timer on the unit's display changes from a count up timer to a countdown timer.

**Command Syntax** [SOURce]:CONFigure:TIMer[:STATe] <BOOL>

**Parameters** 0, "OFF", 1, "ON"

Parameter may be sent as any of the above forms.

**Examples** sour:conf:tim on

**Query Syntax** [SOURce]:CONFigure:TIMer[:STATe]?

**Returned Parameters** "OFF" or "ON"

**Related Commands** [\[SOURce\]:CONFigure:TIMer:COUNt](#)

#### 12. [SOURce]:CONFigure:TIMer:COUNt

**Description** This sets or queries the value of the countdown timer. The format is in Hours,Minutes,Seconds.

**Note** The return value of this command has a bug currently and drops the 1's digit of the seconds field.

**Command Syntax** [SOURce]:CONFigure:TIMer:COUNt <NR1,NR1,NR1>

**Parameters** The countdown timer time in hours, minutes and seconds.

**Examples** sour:conf:tim:count 0,0,30

This sets the output to turn off 30 seconds after it is turned on.

**Query Syntax** [SOURce]:CONFigure:TIMer:COUNT?

**Returned Parameters** "hours,mins,seconds"

example: "00,00,3" for 0 hours, 0 mins and 33 seconds

**Related Commands** [\[SOURce\]:CONFigure:TIMer\[:STATe\]](#)

### 13. [SOURce]:CONFigure:EXternal

**Description** Set or query the use of the external signal inputs. Off will disable external input, and Level or AMP modes use the Digital IO and BNC connectors respectively.

**Command Syntax** [SOURce]:CONFigure:EXternal <NR1/STR>

**Parameters** <0 | OFF | 1 | LEVEL | 2 | AMP>

0 or Off to disable external control.

1 or Level to use the level control signal on the Digital IO connector to control the AC or DC level, depending on the output coupling mode.

2 or AMP to use the Amplifier mode via the BNC connector.

**Examples** sour:conf:ext amp

**Query Syntax** [SOURce]:CONFigure:EXternal?

**Returned Parameters** <Bool>

"OFF" or "ON"

**Related Commands** [\[SOURce\]:CONFigure:INHibit](#)

[\[SOURce\]:CONFigure:TRANSient](#)

### 14. [SOURce]:CONFigure:INHibit

**Description** Set or query the state of the "inhibit" function. When enabled, the state of the "inhibit" pin of the Digital IO is used. See the user manual for more information as this

function is rather complex.

**Command Syntax** [SOURce]:CONFigure:INHibit <Bool>

**Parameters** <ON | OFF>

**Examples** sour:conf:inh off

**Query Syntax** [SOURce]:CONFigure:INHibit?

**Returned Parameters** <Bool>

“OFF” or “ON”

**Related Commands** [\[SOURce\]:CONFigure:EXTernal](#)

#### 15. [SOURce]:CONFigure:TRANSient

**Description** Enable or disable the “Trans” pin OUTPut of the Digital IO, or query whether it is enabled.

**Command Syntax** [SOURce]:CONFigure:TRANSient <Bool>

**Parameters** <ON | OFF>

**Examples** [SOURce]:configure:transient on

**Query Syntax** [SOURce]:CONFigure:TRANSient?

**Returned Parameters** <Bool>

“ON” or “OFF”

**Related Commands** [\[SOURce\]:CONFigure:EXTernal](#)

#### 16. [SOURce]:STEP:VOLTage:AC

**Description** Set or query the starting AC voltage amplitude in STEP mode.

**Command Syntax** [SOURce]:STEP:VOLTage:AC <NR2>

**Parameters** 0 to 150, 0 to 300

**Examples** sour:step:volt:ac 125

**Query Syntax** [SOURce]:STEP:VOLTage:AC?

**Returned Parameters** <NR2>

**Related Commands** [\[SOURce\]:STEP:DVOLtage:AC](#)

17. *[SOURce]:STEP:VOLTage:DC*

**Description** Set or query the starting DC voltage value in STEP mode.

**Command Syntax** [SOURce]:STEP:VOLTage:DC <NR2>

**Parameters** 0 to  $\pm 212$ , 0 to  $\pm 424$

**Examples** [SOURce]:step:volt:dc 35

**Query Syntax** [SOURce]:STEP:VOLTage:DC?

**Returned Parameters** <NR2>

**Related Commands** [\[SOURce\]:STEP:DVOLtage:DC](#)

18. *[SOURce]:STEP:DVOLtage:AC*

**Description** Set or query the change in AC voltage magnitude per individual step in a step sequence.

**Note** This is the change in AC output voltage between steps. The change can be increasing (positive), decreasing (negative) or zero.

**Command Syntax** [SOURce]:STEP:DVOLtage:AC <NR2>

**Parameters** 0 to 150, 0 to 300

**Examples** Step the AC voltage up 20V per step:

step:dvol:ac 20

Step the AC voltage down 11.5V per step:

step:dvol:ac -11.5

**Query Syntax** [SOURce]:STEP:DVOLtage:AC?

**Returned Parameters** <NR2>

**Related Commands** [\[SOURce\]:STEP:VOLTage:AC](#)



### 19. *[SOURce]:STEP:DVOLtage:DC*

**Description** Set or query the change in DC voltage per individual step in a step sequence.

**Note** This is the change in DC output voltage between steps. The change can be increasing (positive), decreasing (negative) or zero.

**Command Syntax** *[SOURce]:STEP:DVOLtage:DC <NR2>*

**Parameters** 0 to  $\pm 212$ , 0 to  $\pm 424$

**Examples** *step:dvol:dc 20*

**Query Syntax** *[SOURce]:STEP:DVOLtage:DC?*

**Returned Parameters** *<NR2>*

**Related Commands** [\[SOURce\]:STEP:VOLTage:DC](#)

### 20. *[SOURce]:STEP:FREQuency*

**Description** Set or query the starting AC voltage frequency for the step sequence.

**Command Syntax** *[SOURce]:STEP:FREQuency <NR2>*

**Parameters** 43-1200

**Examples** *sour:step:freq 50*

**Query Syntax** *[SOURce]:STEP:FREQuency?*

**Returned Parameters** *<NR2>*

**Related Commands** [\[SOURce\]:STEP:DFREQuency](#)

### 21. *[SOURce]:STEP:DFREQuency*

**Description** Set or query the change in AC frequency per individual step in a step sequence.

**Note** This is the change in AC output frequency between steps. The change can be increasing (positive), decreasing (negative) or zero.

**Command Syntax** [SOURce]:STEP:DFREQuency <NR2>

**Parameters** 43-1200

**Examples** step:drfreq 10

**Query Syntax** [SOURce]:STEP:DFREQuency?

**Returned Parameters** <NR2>

**Related Commands** [\[SOURce\]:STEP:FREQuency](#)

## 22. [SOURce]:STEP:TIMe

**Description** Set or query the total time for the step sequence. The individual step duration is a division of the step time and step number (time/number=(time per individual change)).

**Command Syntax** [SOURce]:STEP:TIMe <NR2>

**Parameters** 0-100000

**Examples** sour:step:tim 2000

**Query Syntax** [SOURce]:STEP:TIMe?

**Returned Parameters** <NR1>

## 23. [SOURce]:STEP:COUNt

**Description** Set or query the total number or steps to run in a step sequence.

**Note** Using many steps can help to simulate a sweep.

**Command Syntax** [SOURce]:STEP:COUNt <NR1>

**Parameters** 1-99

**Examples** step:count 45

**Query Syntax** [SOURce]:STEP:COUNt?

**Returned Parameters** <NR1>

**Related Commands** [\[SOURce\]:STEP:TIME](#)

#### 24. *[SOURce]:STEP:SYNChronous*

**Description** Set or query the phase synchronization mode of the individual steps within the step sequence.

**Command Syntax** [SOURce]:STEP:SYNChronous <Bool/Str>

**Parameters** <0 | IMMEDIATE | 1 | PHASe>

**Examples** sour:step:sync phase

**Query Syntax** [SOURce]:STEP:SYNChronous?

**Returned Parameters** <STR>

“IMMEDIATE” or “PHASe”

**Related Commands** [\[SOURce\]:STEP:PHASe](#)

#### 25. *[SOURce]:STEP:PHASe*

**Description** Set or query the phase of the individual steps within the step sequence when using phase synchronization mode.

**Note** When a step occurs, the angle of the waveform relative to the positive going zero cross is the start point.

**Command Syntax** [SOURce]:STEP:PHASe <NR2>

**Parameters** 0-359.7

**Examples** source:step:phase 60

**Query Syntax** [SOURce]:STEP:PHASe?

**Returned Parameters** <NR2>

**Related Commands** [\[SOURce\]:STEP:SYNChronous](#)

#### 26. *List mode programming*

Programming and reading list programs entails a sequence of steps to first pick the list, then write or read values to the list.

Writing list programs to the machine:

1. Write - [\[SOURce\]:LIST:NUMBER 0](#)

2. Write – [SOURce]:LIST:CONF:WRITE 0,5,0,0,0,0
  - a. Alternatively, the individual fields may be written
    - i. [\[SOURce\]:LIST:BASE](#)
    - ii. [SOURce]:LIST:SYNChronous
    - iii. [SOURce]:LIST:PHASe
    - iv. [SOURce]:LIST:COUNT
3. Write – [SOURce]:LIST:WRITE 1,111,0,0,43,53,1,2,0
  - a. Alternatively, the individual fields may be written  
 To write the fields, the command sequence must start with a [SOURce]:LIST:STEPno command, enter every parameter, and end with a [SOURce]:LIST:SAVE command. Only then will the data be entered. This is much like the “ADD STEP” button that must be used when entering list steps manually via the front panel.
    - i. [SOURce]:LIST:STEPno
    - ii. [SOURce]:LIST:VOLTage:LEVel:AC:START
    - iii. [SOURce]:LIST:VOLTage:LEVel:AC:END
    - iv. [SOURce]:LIST:VOLTage:LEVel:DC:START
    - v. [SOURce]:LIST:VOLTage:LEVel:DC:END
    - vi. [SOURce]:LIST:FREQuency:LEVel:START
    - vii. [SOURce]:LIST:FREQuency:LEVel:END
    - viii. [SOURce]:LIST:SAVE
4. Write – [SOURce]:LIST:WRITE 0,0,0,0,43,43,1,2,1

Reading list programs from the machine:

1. Write LIST:NUM
2. Query LIST:CONF:STEP?
3. Query LIST:CONF:READ?
4. Write LIST:START:QUERY
5. Query LIST:READ? for each element in the list. Once the list is completely read, further LIST:READ? commands will start at the beginning and repeat the sequence.

## 27. *[SOURce]:LIST:NUMBER*

**Description** Of the 0-9 lists available, this sets or queries which list is being edited.

**Command Syntax** [SOURce]:LIST:NUMBER <NR1>

**Parameters** 0 through 9

**Examples** list:number 2

**Query Syntax** [SOURce]:LIST:NUMBER?

**Returned Parameters** <NR1>

## 28. *LIST:CONF:WRITE*

**Description** Write the configuration values of a list program. This sets the values shown on the main List screen. The following values are set: "Infinite", "Repeat", "Base", "Sync Source", and "Phase".

**Note** SOURce is not used for this command!

The list number should be set before issuing this command.

**Command Syntax** LIST:CONF:WRITE <NR1, NR1, NR1, 0, NR1, NR2, 0>

**Parameters** Parameter number:

1 – Infinite – "0" for off and "1" for on.

2 – Repeat – 0 through 99 times.

3 – Base – "0" for time and "1" for cycle

4 – Not used, set to 0

5 – Sync Source – 0 for "Immediate" and 1 for "Phase"

6 – Phase(deg) floating point number.

7 – Not used, set to 0.

**Examples** list:conf:write 0,5,0,0,0,0,0

**Related Commands** [\[SOURce\]:LIST:NUMber](#)

## 29. *[SOURce]:LIST:BASE*

**Description** Set or query the units to use for timing each list entry, either time or cycles.

**Command Syntax** [SOURce]:LIST:BASE <STR>

**Parameters** <TIME | CYCLE>

**Examples** list:base cycle

**Query Syntax** [SOURce]:LIST:BASE?

**Returned Parameters** <STR>

“TIME” or “CYCLE”

### 30. [SOURce]:LIST:SYNChronous

**Description** Set or query the synchronization to use for the start point of the waveform

**Command Syntax** [SOURce]:LIST:SYNChronous <STR>

**Parameters** <IMMEDIATE | PHASE>

**Examples** sour:list:sync phase

**Query Syntax** [SOURce]:LIST:SYNChronous?

**Returned Parameters** <STR>

“IMMEDIATE” or “PHASE”

### 31. [SOURce]:LIST:PHASE

**Description** When the “SYNChronous” mode is set to phase, this setting determines where in the cycle the step within the list is started. For example, 90 degrees will be at the positive maximum of the AC waveform.

**Command Syntax** [SOURce]:LIST:PHASE <NR2>

**Parameters** 0-359.7

**Examples** list:phase 90

**Query Syntax** [SOURce]:LIST:PHASE?

**Returned Parameters** <NR2>

### 32. [SOURce]:LIST:COUNT

**Description** Set or query the number of times to repeat the current list.

**Command Syntax** [SOURce]:LIST:COUNT <NR1>

**Parameters** 0-99

**Examples** list:count 11

**Query Syntax** [SOURce]:LIST:COUNT?

**Returned Parameters** <NR1>

### 33. *LIST:WRITE*

**Description** This command writes all fields for an individual step in a list program.

**Note** SOURce is not used for this command!

**Command Syntax** LIST:WRITE <NR2,NR2,NR1,NR1,NR1,NR1,NR1,NR1,NR1,>

- Parameters**
1. Volts(rms) Start – 0 to 300
  2. Volts(rms) End – 0 to 300
  3. Volts (DC) Start – 0 to  $\pm 212$  or 0 to  $\pm 424$
  4. Volts (DC) End - 0 to  $\pm 212$  or 0 to  $\pm 424$
  5. Freq. (Hz) Start – 43 to 1200
  6. Freq. (Hz) End – 43 to 1200
  7. Time (ms) or Cycle – 0 to 999999
  8. Steps – 0 to 200
  9. Not Used

**Examples** LIST:WRITE 1,111,0,0,43,53,1,2,0

**Related Commands** [\[SOURce\]:LIST:VOLTage:LEVel:AC:START](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:AC:END](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:DC:START](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:DC:END](#)

[\[SOURce\]:LIST:FREQuency:LEVel:START](#)

[\[SOURce\]:LIST:FREQuency:LEVel:END](#)

[\[SOURce\]:LIST:STEPno](#)

34. *LIST:CONF:STEP?*

**Description** Read the number of steps in the current list.

**Note** Issue a [SOURce]:LIST:NUM first, to set the list ID to be read.

**Query Syntax** LIST:CONF:STEP?

**Returned Parameters** <NR1> The number of steps in the list.

35. *LIST:CONF:READ?*

**Description** Read the list configuration for a given list ID. This is the information on the “LIST” menu page set before entering data for individual list program steps.

**Query Syntax** LIST:CONF:READ?

**Returned Parameters** <NR1, NR1, NR1, 1, NR1, NR2>

1 – Infinite – “0” for off and “1” for on.

2 – Repeat – 0 through 99 times.

3 – Base – “0” for time and “1” for cycle

4 – No data, always “1”

5 – Sync Source – 0 for “Immediate” and 1 for “Phase”

6 – Phase(deg) floating point number.

**Related Commands**

36. *LIST:READ?*

**Description** Read the individual program steps in the list. Successive issuance of this command returns then next list program entry. When the end of the list program is reached, the



returned list program steps restart at the beginning.

**Query Syntax** LIST:READ?

**Returned Parameters** <NR2, NR2, NR2, NR2, NR2, NR2, NR2, NR2, 0>

1. Volts(rms) Start – 0 to 300
2. Volts(rms) End – 0 to 300
3. Volts (DC) Start – 0 to  $\pm 212$  or 0 to  $\pm 424$
4. Volts (DC) End - 0 to  $\pm 212$  or 0 to  $\pm 424$
5. Freq. (Hz) Start – 43 to 1200
6. Freq. (Hz) End – 43 to 1200
7. Time (ms) or Cycle – 0 to 999999
8. Steps – 0 to 200
9. Not Used

**Related Commands** [\[SOURce\]:LIST:VOLTage:LEVel:AC:START](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:AC:END](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:DC:START](#)  
[\[SOURce\]:LIST:VOLTage:LEVel:DC:END](#)  
[\[SOURce\]:LIST:FREQuency:LEVel:START](#)  
[\[SOURce\]:LIST:FREQuency:LEVel:END](#)  
[\[SOURce\]:LIST:STEPno](#)

### 37. LIST:START:QUERY

**Description** Enable read-back of the list program steps and set pointer to the head of the list program.

**Note** This command resets the location in the list to read back to the head of the list program. Without this command, the return values are invalid and all zero's.

[SOURce] cannot be added to this command.

**Command Syntax** LIST:START:QUERY

**Parameters** None

**Examples** LIST:START:QUERY

**Related Commands** [LIST:READ?](#)

38. *[SOURce]:LIST:STEPno*

**Description** Set or query the current step within the current list being edited. Remember that the total number of steps defined in all lists is limited.

**Command Syntax** [SOURce]:LIST:STEPno <NR1>

**Parameters** 0-99

**Examples** list:stepno 3

**Query Syntax** [SOURce]:LIST:STEPno?

**Returned Parameters** <NR1>

39. *[SOURce]:LIST:VOLTage:LEVel:AC:START*

**Description** Set or query the starting AC voltage of the current step within the current list. Value is limited by the range setting of the machine.

**Command Syntax** [SOURce]:LIST:VOLTage:LEVel:AC:START <NR2>

**Parameters** 0-300

**Examples** sour:list:volt:lev:ac:start 140

**Query Syntax** [SOURce]:LIST:VOLTage:LEVel:AC:START?

**Returned Parameters** <NR2>

40. *[SOURce]:LIST:VOLTage:LEVel:AC:END*

**Description** Set or query the ending AC voltage of the current step within the current list. Value is limited by the range

setting of the machine.

**Command Syntax** [SOURce]:LIST:VOLTage:LEVel:AC:END <NR2>

**Parameters** 0-300

**Examples** sour:list:volt:lev:ac:end 110

**Query Syntax** [SOURce]:LIST:VOLTage:LEVel:AC:END?

**Returned Parameters** <NR2>

41. *[SOURce]:LIST:VOLTage:LEVel:DC:START*

**Description** Set or query the starting DC voltage of the current step within the current list. Value is limited by the range setting of the machine.

**Command Syntax** [SOURce]:LIST:VOLTage:LEVel:DC:START <NR2>

**Parameters** 0 to  $\pm 212$ , 0 to  $\pm 424$

**Examples** [SOURce]:LIST:VOLTage:LEVel:DC:START?

**Query Syntax** [SOURce]:LIST:VOLTage:LEVel:DC:START?

**Returned Parameters** <NR2>

42. *[SOURce]:LIST:VOLTage:LEVel:DC:END*

**Description** Set or query the ending DC voltage of the current step within the current list. Value is limited by the range setting of the machine.

**Command Syntax** [SOURce]:LIST:VOLTage:LEVel:DC:END <NR2>

**Parameters** 0 to  $\pm 212$ , 0 to  $\pm 424$

**Examples** sour:list:volt:lev:dc:end 60

**Query Syntax** [SOURce]:LIST:VOLTage:LEVel:DC:END?

**Returned Parameters** <NR2>

43. *[SOURce]:LIST:FREQuency:LEVel:START*

**Description** Set or query the starting AC frequency of the current step

within the current list. Value is limited by the range setting of the machine.

**Command Syntax** [SOURce]:LIST:FREQuency:LEVel:START <NR2>

**Parameters** 43-1200

**Examples** sour:list:freq:lev:start 50

**Query Syntax** [SOURce]:LIST:FREQuency:LEVel:START?

**Returned Parameters** <NR2>

#### 44. *[SOURce]:LIST:FREQuency:LEVel:END*

**Description** Set or query the ending AC frequency of the current step within the current list. Value is limited by the range setting of the machine.

**Command Syntax** [SOURce]:LIST:FREQuency:LEVel:END <NR2>

**Parameters** 43-1200

**Examples** sour:list:freq:lev:end 60

**Query Syntax** [SOURce]:LIST:FREQuency:LEVel:END?

**Returned Parameters** <NR2>

#### 45. *[SOURce]:LIST:CLEar*

**Description** Clear the current list. This deletes the contents of the current list.

**Command Syntax** [SOURce]:LIST:CLEar <NR1>

**Parameters** 0 through 9

**Examples** list:clear 0

**Related Commands** [\[SOURce\]:LIST:NUMber](#)

#### 46. *[SOURce]:PULSe:VOLTage:AC*

**Description** Set or query the AC voltage magnitude or the pulse.

**Command Syntax** [SOURce]:PULSe:VOLTage:AC <NR2>

**Parameters** 0 to 150 or 0 to 300

**Examples** source:pulse:voltage:ac 140

**Query Syntax** [SOURce]:PULSe:VOLTage:AC?

**Returned Parameters** <NR2>

**Related Commands**

47. *[SOURce]:PULSe:VOLTage:DC*

**Description** Set or query the DC voltage level of the pulse.

**Command Syntax** [SOURce]:PULSe:VOLTage:DC <NR2>

**Parameters** 0 to  $\pm 212$ , 0 to  $\pm 424$

**Examples** sour:puls:volt:dc -50

**Query Syntax** [SOURce]:PULSe:VOLTage:DC?

**Returned Parameters** <NR2>

**Related Commands**

48. *[SOURce]:PULSe:FREQuency*

**Description** Set or query the frequency of the AC portion of the pulse.

**Command Syntax** [SOURce]:PULSe:FREQuency <NR2>

**Parameters** 43 to 1200

**Examples** pulse:freq 300

**Query Syntax** [SOURce]:PULSe:FREQuency?

**Returned Parameters** <NR2>

49. *[SOURce]:PULSe:SYNChronous*

**Description** Set or query the phase of the pulse start as either “immediate” (random phase), or at a specific “phase”

angle of the steady state waveform.

**Command Syntax** [SOURce]:PULSe:SYNChronous <Bool/STR>

**Parameters** <0 | IMMEDIATE | 1 | PHASE>

**Examples** sour:pulse:sync phas

**Query Syntax** [SOURce]:PULSe:SYNChronous?

**Returned Parameters** "IMMEDIATE" or "PHASE"

50. *[SOURce]:PULSe:PHASE*

**Description** Set or query the phase angle to start the pulse from.

**Command Syntax** [SOURce]:PULSe:PHASE <NR2>

**Parameters** Phase angle to start pulse at, relative to positive going zero cross of steady state waveform.

**Examples** pulse:phase 60

**Query Syntax** [SOURce]:PULSe:PHASE?

**Returned Parameters** <NR2>

51. *[SOURce]:PULSe:DUTY*

**Description** Set or query the duty cycle of the pulse. This is the percentage of the pulse period to OUTPUT the pulse. The minimum pulse duration is limited and will not go below 1.5ms typically.

**Note**

**Command Syntax** [SOURce]:PULSe:DUTY <NR2>

**Parameters** 0-100.0

**Examples** source:pulse:duty 15

**Query Syntax** [SOURce]:PULSe:DUTY?

**Returned Parameters** <NR2>

52. *[SOURce]:PULSe:PERIOd*

**Description** Set or query the length, or time between pulses, in milliseconds.

**Command Syntax** *[SOURce]:PULSe:PERIOd <NR2>*

**Parameters** 0-100000

**Examples** *source:pulse:period 5000*

**Query Syntax** *[SOURce]:PULSe:PERIOd?*

**Returned Parameters** *<NR1>*

53. *[SOURce]:PULSe:COUNt*

**Description** Set or query how many pulse intervals to perform.

**Command Syntax** *[SOURce]:PULSe:COUNt <NR1>*

**Parameters** 0-99

**Examples** *pulse:count 10*

**Query Syntax** *[SOURce]:PULSe:COUNt?*

**Returned Parameters** *<NR1>*

54. *[SOURce]:FUNCTion:SHAPE*

**Description** Set or query which waveform configuration to OUTPUT.

**Command Syntax** *[SOURce]:FUNCTion:SHAPE <Bool/STR>*

**Parameters** *<0 | A | 1 | B>*

**Examples** *function:shape A*

**Query Syntax** *[SOURce]:FUNCTion:SHAPE?*

**Returned Parameters** *"A" or "B"*

**Related Commands** [\[SOURce\]:FUNCTion:SHAPE:A](#)

[\[SOURce\]:FUNCTion:SHAPE:B](#)

## 55. *[SOURce]:FUNction:SHApe:A*

**Description** Set or query the waveform defined for setting A.

**Command Syntax** *[SOURce]:FUNction:SHApe:A <STR>*

**Parameters** <SINE | CSIN,NR2 | SQUA | THD,<NR1> | USR,<NR1>>

When setting the shape, the “SINE” and “SQUA” functions

**Examples** func:shap:a sine

sour:func:shap:a csin,40.5

source:function:shape:a squa

sour:function:shap:a thd,1

func:shap:a user,4

**Query Syntax** *[SOURce]:FUNction:SHApe:A?*

**Returned Parameters** “SINE”, “CSIN,<NR2>”, “SQUA”, “USR,<NR1>”

The return values for CSIN, THD and USER waveforms also return the amount of clipping, the THD waveform number or the user waveform number respectively.

**Related Commands** [\[SOURce\]:FUNction:SHApe](#)

## 56. *[SOURce]:FUNction:SHApe:B*

**Description** Set or query the waveform defined for setting A.

**Command Syntax** *[SOURce]:FUNction:SHApe:B <STR>*

**Parameters** <SINE | CSIN,NR2 | SQUA | THD,<NR1> | USR,<NR1>>

When setting the shape, the “SINE” and “SQUA” functions

**Examples** func:shap:b sine

sour:func:shap:b csin,40.5

source:function:shape:b squa



sour:function:shap:b thd,1

func:shap:b user,4

**Query Syntax** [SOURce]:FUNCtion:SHAPE:A?

**Returned Parameters** "SINE", "CSIN,<NR2>", "SQUA", "USER,<NR1>"

The return values for CSIN, THD and USER waveforms also return the amount of clipping, the THD waveform number or the user waveform number respectively.

**Related Commands** [\[SOURce\]:FUNCtion:SHAPE](#)

#### 57. [SOURce]:FUNC:SHAP:USER:TABLE:READ

**Description** The ID number of the USER defined waveform to read.

**Note** Use this function to set the user defined waveform to retrieve with the

**Command Syntax** FUNC:SHAP:USER:READ <NR1>

**Parameters** <NR1>

**Examples** func:shap:user:read 1

**Related Commands** [\[SOURce\]:FUNCtion:SHAPE:USER:TABLE:DATA? <NR1>](#)

#### 58. [SOURce]:FUNCtion:SHAPE:USER:TABLE:DATA

**Description** Write the individual harmonic elements that compose the user defined waveform. The waveform is the sum of the harmonics. For example:

60Hz fundamental and 100% of the 3<sup>rd</sup> harmonic.

$$\text{out}(t) = \sin(2 \cdot \pi \cdot 60 \cdot t) + \sin(2 \cdot \pi \cdot 180 \cdot t)$$

**Note** The complete set (0 through 39) should be sent at one time.

Only integer multiples of the fundamental are to be specified. The set of which are the first 39 multiples.

60Hz fundamental:

Field 0 = 60Hz

Field 1 = 120Hz

Field 2 = 180Hz

...

Field 39 = 2400Hz

Magnitude of harmonic: This is the “%” field in the PC software.

Phase – This is the phase shift from the fundamental waveform of the specified harmonic.

**Command Syntax** [SOURce]:FUNCtion:SHAPE:USER:TABLE:DATA <NR1, NR2, NR1, NR2>

**Parameters** Parameter 1 – element number (0 through 39)

Parameter 2 – Weighting factor (0 to 1)

Parameter 3 – Frequency

Parameter 4 - Phase

**Examples** func:shap:user:table:data 0,1,50,0

func:shap:user:table:data 1, 0.02, 100, 0

...

func:shap:user:table:data 39,0,2000,0

59. [SOURce]:FUNCtion:SHAPE:USER:TABLE:DATA? <NR1>

**Description** Query the data table for the previously specified waveform table.

This returns a portion of the user defined waveform parameters. The argument of this function defines which set of values to return. This command returns 5 sets at a time.

0 – fields 0 through 4

1 – fields 5 through 9

...

7 – fields 35 through 39

**Query Syntax** func:shap:user:tabl:data? <NR1>

**Returned Parameters** ID,magnitude,frequence,phase;(next entry)

Example (50Hz fundamental):

Query command: "source:function:user:table:data? 0"

"0,1.000000,50.000000,0.000000;1,0.020000,100.000000  
0,0.000000;2,0.030000,150.000000,0.000000;3,0.040000  
,200.000000,0.000000;4,0.050000,250.000000,0.000000  
\n\r"

## K. SYStem

### 1. *Command Summary*

[SYStem:BEEP](#)  
[SYStem:BRIGhtness](#)  
[SYStem:KEY:LOCK](#)  
[SYStem:TIME](#)  
[SYStem:DATE](#)  
[SYStem:RECall:DEFault](#)  
[SYStem:SERial?](#)  
[SYStem:MODEl?](#)  
[SYStem:MANUFACTure?](#)  
[SYStem:INTERFACE](#)  
[SYStem:ERRor?](#)  
[SYStem:VERSion?](#)  
[SYStem:GPIB:ADDReSS](#)  
[SYStem:IP:CONFIg](#)  
[SYStem:IP:ADDReSS](#)  
[SYStem:IP:GATEWay](#)  
[SYStem:IP:MASK](#)

### 2. *SYStem:BEEP*

**Description** Set or query the system beep setting.

**Command Syntax** SYStem:BEEP <Bool>

**Parameters** <0 | OFF 1 | ON>

**Examples** sys:beep off

**Query Syntax** SYStem:BEEP?

**Returned Parameters** "OFF" or "ON"

### 3. *SYStem:BRIGhtness*

**Description** Set or query the LCD brightness. 0 – min, 9 – max

**Command Syntax** SYStem:BRIGhtness <NR1>

**Parameters** 0 through 9

**Examples** sys:brig 9

**Query Syntax** SYStem:BRIGhtness?

**Returned Parameters** <NR1>

### 4. *SYStem:KEY:LOCK*

**Description** Set or query the keyboard lock state.

**Command Syntax** SYStem:KEY:LOCK <Bool>

**Parameters** <0 | OFF 1 | ON>

**Examples** sys:key:lock on

**Query Syntax** SYStem:KEY:LOCK?

**Returned Parameters** "OFF" or "ON"

### 5. *SYStem:TIME*

**Description** Set or query the time setting on the machine. The format is in Hours, Minutes and Seconds, in that order.

**Command Syntax** SYStem:TIME <NR1, NR1, NR1>

**Parameters** 0-24, 0-59, 0-59

Hours, Minutes, Seconds

**Examples** sys:time 16,10,00

**Query Syntax** SYStem:TIME?

**Returned Parameters** <NR1,NR1,NR1>

## 6. *SYStem:DATE*

**Description** Set or query the date setting on the machine. The format is Year, Month, Day.

**Command Syntax** SYStem:DATE <NR1, NR1, NR1>

**Parameters** 0 to 99, 0 to 12, 0 to 31

Year, Month, Day

**Examples** sys:date 85,11,5

**Query Syntax** SYStem:DATE?

**Returned Parameters** <NR1,NR1,NR1>

## 7. *SYStem:RECall:DEFault*

**Description** Recall the default settings for the machine.

**Command Syntax** SYStem:RECall:DEFault

**Examples** sys:rec:def

## 8. *SYStem:SERial?*

**Description** Query the machine's serial number.

**Query Syntax** SYStem:SERial?

**Returned Parameters** <STR>

## 9. *SYStem:MODEl?*

**Description** Query the machine's model number.

**Query Syntax** SYStem:MODEl?

**Returned Parameters** <NR1>  
"9833"

10. *SYStem:MANUFACTure?*

**Description** Query the machine manufacturer's name.

**Query Syntax** SYStem:MANUFACTure?

**Returned Parameters** "B&K PRECISION"

11. *SYStem:INTERFACE*

**Description** Set or query the remote interface to use.

**Note** Currently only returning "TMC" and "VCP" correctly.

**Command Syntax** SYStem:INTERFACE <NR1/STR>

**Parameters** <0 | VCP | 1 | TMC | 2 | GPIB | 3 | LAN | 4 | RS232C>

**Examples** sys:interface gpib

**Query Syntax** sys:interface?

**Returned Parameters** "VCP", "TMC", "GPIB", "LAN", "RS232C"

12. *SYStem:ERRor?*

**Description** Query the current error code, if any. See the user manual for a listing of the error codes returned.

**Query Syntax** SYStem:ERRor?

**Returned Parameters** <NR1>, <STR>

13. *SYStem:VERSion?*

**Description** Query the firmware version of the machine.

**Query Syntax** SYStem:VERSion?

**Returned Parameters** <STR>

#### 14. *SYStem:GPIB:ADDRess*

**Description** Set or query the GPIB address of the machine.

**Command Syntax** SYStem:GPIB:ADDRess <NR1>

**Parameters** GPIB address

**Examples** sys:gpiB:addr 5

**Query Syntax** SYStem:GPIB:ADDRess?

**Returned Parameters** <NR1>

#### 15. *SYStem:IP:CONFig*

**Description** Set or query the network configuration mode, static or dynamic.

**Command Syntax** SYStem:IP:CONFig <NR1/STR>

**Parameters** <0 | DHCP | 1 | STATic>

**Examples** SYS:IP:CONF DHCP

**Query Syntax** SYStem:IP:CONFig?

**Returned Parameters** "DHCP" or "STATic"

#### 16. *SYStem:IP:ADDRess*

**Description** Set or query the IP address of the machine.

**Command Syntax** SYStem:IP:ADDRess <NR1, NR1, NR1, NR1>

**Parameters** 0 to 255 for each entry.

**Examples** sys:ip:addr 192,168,2,96

**Query Syntax** SYStem:IP:ADDRess?

**Returned Parameters** <STR>

#### 17. *SYStem:IP:GATEway*

**Description** Set or query the network gateway.

**Command Syntax** SYStem:IP:GATEway <NR1, NR1, NR1, NR1>

**Parameters** 0 through 255 for each entry.

**Examples** SYStem:IP:GATEway 255,255,255,0

**Query Syntax** SYStem:IP:GATEway?

**Returned Parameters** <STR>

## 18. *SYStem:IP:MASK*

**Description** Set or query the network mask.

**Command Syntax** SYStem:IP:MASK <NR1, NR1, NR1, NR1>

**Parameters** 0 through 255 for each entry.

**Examples** SYStem:IP:MASK 1,2,3,4

**Query Syntax** SYStem:IP:MASK?

**Returned Parameters** <STR>

## L. USBFlash

### 1. *Command Summary*

[USBFlash:SAVE:CONFiguration](#)  
[USBFlash:RECall:CONFiguration](#)  
[USBFlash:SAVE:SCReen](#)  
[USBFlash:SAVE:COMPlite?](#)

### 2. *USBFlash:SAVE:CONFiguration*

**Description** Store the current settings to internal memory or a USB storage device.

**Command Syntax** USBFlash:SAVE:CONFiguration <NR1>

**Parameters** 0 through 9, internal  
10 through 99, external

**Examples** usbf:save:conf 10



3. *USBFlash:RECall:CONFiguration*

**Description** Recall a configuration setup from either internal storage or a USB device.

**Command Syntax** USBFlash:RECall:CONFiguration

**Parameters** 0 through 9, internal  
10 through 99, external

**Examples** usbf:rec:conf 0

4. *USBFlash:SAVE:SCReen*

**Description** Save an image of the LCD screen.

**Command Syntax** USBFlash:SAVE:SCReen <NR1>

**Parameters** 0 through 999

**Examples** usbf:save:scr 0

5. *USBFlash:SAVE:COMPlete?*

**Description** Query whether the save action has completed.

**Query Syntax** USBFlash:SAVE:COMPlete?

**Returned Parameters** 1 for complete and 0 for all other conditions including busy and no previous save issued.



22820 Savi Ranch Parkway  
Yorba Linda, CA 92887  
[www.bkprecision.com](http://www.bkprecision.com)

© 2017 B&K Precision Corp.

Printed in China

v053017