

# Communication Interface Manual

PCR-WE Series AC Power Supply

**PCR1000WE**

**PCR2000WE**

PCR-WE2 Series AC Power Supply

**PCR3000WE2**

**PCR6000WE2/PCR6000WE2R**

**PCR12000WE2/PCR12000WE2R**

**PCR18000WE2/PCR18000WE2R**

**PCR24000WE2/PCR24000WE2R**

**PCR30000WE2/PCR30000WE2R**

**PCR36000WE2/PCR36000WE2R**

# INDEX

|                          |          |
|--------------------------|----------|
| <b>Command List.....</b> | <b>7</b> |
|--------------------------|----------|

|                          |           |
|--------------------------|-----------|
| <b>Introduction.....</b> | <b>25</b> |
|--------------------------|-----------|

|   |    |
|---|----|
| Reading environment.....  | 25 |
| Intended readers.....   | 25 |
| Structure of the manual .....   | 26 |
| Trademarks.....   | 26 |
| Firmware version of the product to which<br>this manual applies ..... | 26 |
| Measuring instrument interface<br>standards.....                      | 27 |
| Copyright .....   | 27 |

|  |           |
|--|-----------|
| <b>Installing the VISA Library .....</b> | <b>28</b> |
|--|-----------|

|                                       |           |
|---------------------------------------|-----------|
| <b>Setting Up the Interface .....</b> | <b>29</b> |
|---------------------------------------|-----------|

|  |    |
|--|----|
| RS232C (standard).....   | 30 |
| USB (standard).....  | 33 |
| LAN (standard) .....   | 34 |
| Accessing and Operating the Product<br>from a Web Browser (LAN)..... | 40 |
| GPIO (option).....   | 49 |

|                            |           |
|----------------------------|-----------|
| <b>About Commands.....</b> | <b>51</b> |
|----------------------------|-----------|

|                        |    |
|------------------------|----|
| Command Hierarchy..... | 51 |
| Command Syntax .....   | 52 |

|                         |           |
|-------------------------|-----------|
| <b>Parameters .....</b> | <b>55</b> |
|-------------------------|-----------|

|                                       |    |
|---------------------------------------|----|
| Non-numeric parameters .....          | 55 |
| Numeric parameters .....              | 56 |
| Special form numeric parameters.....  | 57 |
| Phase designation (channel list)..... | 58 |
| Measurement units .....               | 58 |

|                                     |  |
|-------------------------------------|--|
| <b>IEEE 488.2 common commands .</b> |  |
|-------------------------------------|--|

## 59

|            |    |
|------------|----|
| *CLS ..... | 59 |
| *ESE ..... | 60 |
| *ESR.....  | 61 |
| *IDN ..... | 62 |
| *OPC.....  | 63 |

|            |    |
|------------|----|
| *OPT ..... | 64 |
| *PSC ..... | 65 |
| *RCL ..... | 66 |
| *RST ..... | 67 |
| *SAV ..... | 69 |
| *SRE.....  | 70 |
| *STB ..... | 71 |
| *TRG.....  | 72 |
| *TST.....  | 73 |
| *WAI.....  | 74 |

|                            |           |
|----------------------------|-----------|
| <b>ABORt Command .....</b> | <b>75</b> |
|----------------------------|-----------|

|                |    |
|----------------|----|
| ABOR.....      | 75 |
| ABOR:ACQ.....  | 76 |
| ABOR:PROG..... | 77 |
| ABOR:SIM ..... | 78 |
| ABOR:TRAN..... | 79 |

|                             |           |
|-----------------------------|-----------|
| <b>DISPlay Command.....</b> | <b>80</b> |
|-----------------------------|-----------|

|                          |    |
|--------------------------|----|
| DISP:CONT .....          | 80 |
| DISP:PHAS.....           | 81 |
| DISP:MMOD:CURR.....      | 82 |
| DISP:MMOD:VOLT .....     | 83 |
| DISP:MMOD:VOLT:LTL ..... | 84 |

|                           |           |
|---------------------------|-----------|
| <b>HCOPy Command.....</b> | <b>85</b> |
|---------------------------|-----------|

|                      |    |
|----------------------|----|
| HCOP:SDUM:DATA?..... | 85 |
|----------------------|----|

|                               |           |
|-------------------------------|-----------|
| <b>INITiate Command .....</b> | <b>86</b> |
|-------------------------------|-----------|

|                |    |
|----------------|----|
| INIT:ACQ.....  | 86 |
| INIT:PROG..... | 87 |
| INIT:SIM ..... | 88 |
| INIT:TRAN..... | 89 |

|                                |           |
|--------------------------------|-----------|
| <b>INSTrument Command.....</b> | <b>90</b> |
|--------------------------------|-----------|

|                       |    |
|-----------------------|----|
| INST/ INST:NSEL ..... | 90 |
| INST:COUP.....        | 91 |

|                          |           |
|--------------------------|-----------|
| <b>LXI Command .....</b> | <b>92</b> |
|--------------------------|-----------|

|               |    |
|---------------|----|
| LXI:IDEN..... | 92 |
|---------------|----|

**MEASure/FETCh Command .... 93**

|  |    |
|--|----|
| FETC:<meas-item>/ MEAS:<meas-item> .....           | 94 |
| FETC:ARR:<harm-item>?/ MEAS:ARR:<harm-item>? ..... | 96 |

**MEMory Command ..... 97**

|                    |     |
|--------------------|-----|
| MEM:REC .....      | 97  |
| MEM:REC:CONF ..... | 98  |
| MEM:REC:PREV ..... | 99  |
| MEM:SAVE .....     | 100 |

**OUTPut Command..... 101**

|                          |     |
|--------------------------|-----|
| OUTP .....               | 101 |
| OUTP:IMP .....           | 102 |
| OUTP:IMP:REAC .....      | 103 |
| OUTP:IMP:REAL .....      | 104 |
| OUTP:PHAS:OFF .....      | 105 |
| OUTP:PHAS:OFF:LEV .....  | 106 |
| OUTP:PHAS:ON .....       | 107 |
| OUTP:PHAS:ON:LEV .....   | 108 |
| OUTP:PON .....           | 109 |
| OUTP:PROT:CLE .....      | 110 |
| OUTP:PROT:WDOG .....     | 111 |
| OUTP:PROT:WDOG:DEL ..... | 112 |
| OUTP:SST .....           | 113 |
| OUTP:SST:FALL .....      | 114 |
| OUTP:SST:TIME .....      | 115 |
| OUTP:SST:TIME:FALL ..... | 116 |

**PROGrama Command..... 117**

|                           |     |
|---------------------------|-----|
| PROG:CLE .....            | 117 |
| PROG:EDIT .....           | 118 |
| PROG:EDIT:FUNC:BANK ..... | 120 |
| PROG:EDIT:JUMP .....      | 121 |
| PROG:EDIT:IMP .....       | 122 |
| PROG:EDIT:IMP:REAC .....  | 123 |
| PROG:EDIT:IMP:REAL .....  | 124 |
| PROG:EDIT:PHAS:RAMP ..... | 125 |
| PROG:EDIT:PHAS:STAR ..... | 126 |
| PROG:EDIT:PHAS:STOP ..... | 127 |
| PROG:EDIT:PHAS:UOFF ..... | 128 |
| PROG:EDIT:PHAS:UV .....   | 129 |
| PROG:EDIT:PHAS:UW .....   | 130 |

|                           |     |
|---------------------------|-----|
| PROG:EDIT:VOLT .....      | 131 |
| PROG:EDIT:VOLT:OFFS ..... | 132 |
| PROG:EXEC .....           | 133 |
| PROG:LOOP .....           | 134 |
| PROG:STAT .....           | 135 |
| PROG:STEP:END .....       | 136 |
| PROG:STEP:STAR .....      | 137 |

**SENSe Command ..... 138**

|                           |     |
|---------------------------|-----|
| SENS:AVER:COUN .....      | 138 |
| SENS:CURR:HOLD:CLE .....  | 139 |
| SENS:CURR:HOLD:TIME ..... | 140 |
| SENS:VOLT:AVER:COUN ..... | 141 |
| SENS:VOLT:HOLD:CLE .....  | 142 |
| SENS:VOLT:HOLD:TIME ..... | 143 |

**SIMulation Command..... 144**

|                        |     |
|------------------------|-----|
| SIM:EXEC .....         | 144 |
| SIM:POL .....          | 145 |
| SIM:REP:COUN .....     | 146 |
| SIM:STAT .....         | 147 |
| SIM:T1:PHAS .....      | 148 |
| SIM:T1:PHAS:STAT ..... | 149 |
| SIM:T1:TIME .....      | 150 |
| SIM:T2:TIME .....      | 151 |
| SIM:T3:TIME .....      | 152 |
| SIM:T3:VOLT .....      | 153 |
| SIM:T4:TIME .....      | 154 |
| SIM:T5:CYCL .....      | 155 |
| SIM:T5:CYCL:STAT ..... | 156 |
| SIM:T5:TIME .....      | 157 |

**[SOURce:]CURRent Command ... 158**

|                          |     |
|--------------------------|-----|
| CURR .....               | 158 |
| CURR:PEAK .....          | 159 |
| CURR:PEAK:LOW .....      | 160 |
| CURR:PROT:STAT .....     | 161 |
| CURR:PROT:TRIP:DEL ..... | 162 |

**[SOURce:]FREQuency Command 163**

|            |     |
|------------|-----|
| FREQ ..... | 163 |
|------------|-----|

|                          |     |
|--------------------------|-----|
| FREQ:LIM:LOW .....       | 164 |
| FREQ:LIM:UPP .....       | 165 |
| FREQ:SYNC .....          | 166 |
| FREQ:SYNC:MODE .....     | 167 |
| FREQ:SYNC:PHAS:DEL ..... | 168 |
| FREQ:TRIG .....          | 169 |

## **[SOURce:]FUNCTioN Command ..**

170

|                 |     |
|-----------------|-----|
| FUNC:BANK ..... | 170 |
|-----------------|-----|

## **[SOURce:]VOLTage Command....**

171

|                                    |     |
|------------------------------------|-----|
| VOLT .....                         | 171 |
| VOLT:COMP:MODE .....               | 172 |
| VOLT:COMP:RADJ .....               | 173 |
| VOLT:COMP:SOFT:CONT .....          | 174 |
| VOLT:COMP:SOFT:TERM .....          | 175 |
| VOLT:EXT:INP:EXTDC:ADJ:GAIN... ..  | 176 |
| VOLT:EXT:INP:EXTDC:ADJ:OFFS.. ..   | 177 |
| VOLT:EXT:INP:EXTDC:APER .....      | 178 |
| VOLT:EXT:INP:EXTDC:SIGN:POL .. ..  | 179 |
| VOLT:EXT:INP:EXTDC:SIGN:SOUR ..... | 180 |
| VOLT:EXT:INP:FUNC:MODE .....       | 181 |
| VOLT:EXT:INP:VPR:ADJ:GAIN .....    | 182 |
| VOLT:EXT:INP:VPR:ADJ:OFFS .....    | 183 |
| VOLT:EXT:INP:VPR:MAP .....         | 184 |
| VOLT:EXT:INP:VPR:STAT .....        | 185 |
| VOLT:LIM:LOW .....                 | 186 |
| VOLT:LIM:UPP .....                 | 187 |
| VOLT:LTL .....                     | 188 |
| VOLT:OFFS .....                    | 189 |
| VOLT:OFFS:LIM:LOW .....            | 190 |
| VOLT:OFFS:LIM:UPP .....            | 191 |
| VOLT:OFFS:LTL .....                | 192 |
| VOLT:OFFS:TRIG .....               | 193 |
| VOLT:PROT:LOW .....                | 194 |
| VOLT:PROT:PEAK:LOW .....           | 195 |
| VOLT:PROT:PEAK:UPP .....           | 196 |
| VOLT:PROT:LOW:STAT .....           | 197 |
| VOLT:PROT:UPP .....                | 198 |
| VOLT:RANG .....                    | 199 |
| VOLT:RESP .....                    | 200 |

|                 |     |
|-----------------|-----|
| VOLT:TRIG ..... | 201 |
|-----------------|-----|

## **STATus Command ..... 202**

|   |     |
|---|-----|
| Status Report Structure .....                             | 202 |
| Architecture .....  | 204 |
| Status byte register .....                                | 205 |
| Event status register .....                               | 206 |
| OPERation status register .....                           | 207 |
| STAT:OPER .....   | 208 |
| STAT:OPER:COND .....                                      | 209 |
| STAT:OPER:ENAB .....                                      | 210 |
| STAT:OPER:NTR .....                                       | 211 |
| STAT:OPER:PTR .....                                       | 212 |
| OPERation:INSTrument subregister .....                    | 213 |
| STAT:OPER:INST .....                                      | 214 |
| STAT:OPER:INST:COND .....                                 | 215 |
| STAT:OPER:INST:ENAB .....                                 | 216 |
| STAT:OPER:INST:NTR .....                                  | 217 |
| STAT:OPER:INST:PTR .....                                  | 218 |
| OPERation:INSTrument:ISUMmary{1 2 3} subregister .....    | 219 |
| STAT:OPER:INST:ISUM{1 2 3} .....                          | 220 |
| STAT:OPER:INST:ISUM{1 2 3}:COND... ..                     | 221 |
| STAT:OPER:INST:ISUM{1 2 3}:ENAB ... ..                    | 222 |
| STAT:OPER:INST:ISUM{1 2 3}:NTR .....                      | 223 |
| STAT:OPER:INST:ISUM{1 2 3}:PTR .....                      | 224 |
| QUESTionable status register .....                        | 225 |
| STAT:QUES .....   | 226 |
| STAT:QUES:COND .....                                      | 227 |
| STAT:QUES:ENAB .....                                      | 228 |
| STAT:QUES:NTR .....                                       | 229 |
| STAT:QUES:PTR .....                                       | 230 |
| QUESTionable:INSTrument subregister. ..                   | 231 |
| STAT:QUES:INST .....                                      | 232 |
| STAT:QUES:INST:COND .....                                 | 233 |
| STAT:QUES:INST:ENAB .....                                 | 234 |
| STAT:QUES:INST:NTR .....                                  | 235 |
| STAT:QUES:INST:PTR .....                                  | 236 |
| QUESTionable:INSTrument:ISUMmary{1 2 3} subregister ..... | 237 |

|                                     |            |                                      |            |
|-------------------------------------|------------|--------------------------------------|------------|
| STAT:QUES:INST:ISUM{1 2 3} .....    | 238        | SYST:EXT:SSIGIO:STAT .....           | 275        |
| STAT:QUES:INST:ISUM{1 2 3}:COND...  | 239        | SYST:KLOC .....                      | 276        |
| STAT:QUES:INST:ISUM{1 2 3}:ENAB ... | 240        | SYST:LOC/ SYST:REM/ SYST:RWL         | 277        |
| STAT:QUES:INST:ISUM{1 2 3}:NTR      | 241        | SYST:OPT .....                       | 278        |
| STAT:QUES:INST:ISUM{1 2 3}:PTR      | 242        | SYST:PASS .....                      | 279        |
| Preset status .....                 | 243        | SYST:PASS:CDIS .....                 | 280        |
| STAT:PRES .....                     | 243        | SYST:PASS:NEW .....                  | 281        |
|                                     |            | SYST:PASS:STAT .....                 | 282        |
| <b>SYSTem Command.....</b>          | <b>244</b> | SYST:SEC:IMM .....                   | 283        |
| SYST:COMM:RLST .....                | 244        | SYST:SLE .....                       | 284        |
| SYST:CONF:ACC .....                 | 245        | SYST:SLE:EXEC .....                  | 285        |
| SYST:CONF:ADJ:VOLT:FINE .....       | 246        | SYST:SLE:TIME .....                  | 286        |
| SYST:CONF:ADJ:VOLT:TERM:MODE ..     | 247        | SYST:TIME .....                      | 287        |
| SYST:CONF:FORM:FRAM .....           | 248        | SYST:TIME:ADJ .....                  | 288        |
| SYST:CONF:FORM:FRAM:INFO ....       | 249        | SYST:TZON .....                      | 289        |
| SYST:CONF:FORM:PMOD .....           | 250        | SYST:TZON:CAT .....                  | 290        |
| SYST:CONF:FORM:PMOD:INFO ....       | 251        | SYST:VERS .....                      | 291        |
| SYST:CONF:FORM:PSAV:MAX ....        | 252        |                                      |            |
| SYST:CONF:FORM:PSAV:MOD ....        | 253        | <b>TRIGger Command.....</b>          | <b>292</b> |
| SYST:CONF:FORM:PSAV:RES ....        | 254        | TRIG:ACQ .....                       | 292        |
| SYST:CONF:PHAS:UOFF .....           | 255        | TRIG:ACQ:SOUR .....                  | 293        |
| SYST:CONF:PHAS:UV .....             | 256        | TRIG:PROG .....                      | 294        |
| SYST:CONF:PHAS:UW .....             | 257        | TRIG:PROG:SOUR .....                 | 295        |
| SYST:CONF:PON:STAT .....            | 258        | TRIG:SIM .....                       | 296        |
| SYST:CONF:SSUP .....                | 259        | TRIG:SIM:SOUR .....                  | 297        |
| SYST:CONF:WIR/ SYST:CONF:NOUT .     | 260        | TRIG:TRAN .....                      | 298        |
| SYST:DATE .....                     | 261        | TRIG:TRAN:SOUR .....                 | 299        |
| SYST:ERR .....                      | 262        |                                      |            |
| SYST:ERR:COUN .....                 | 263        | <b>WAVE Command.....</b>             | <b>300</b> |
| SYST:EXT:DIG:READ .....             | 264        | WAVE:DATA:ARB .....                  | 300        |
| SYST:EXT:DIG:WRIT .....             | 265        | WAVE:DATA:CLIP .....                 | 301        |
| SYST:EXT:SIGIN:MAP .....            | 266        | WAVE:DATA:IECP .....                 | 302        |
| SYST:EXT:SIGIN:POL .....            | 267        | WAVE:DATA:POIN .....                 | 303        |
| SYST:EXT:SIGIN:STAT .....           | 268        | WAVE:DATA:SIN .....                  | 304        |
| SYST:EXT:SIGOUT:MAP .....           | 269        | WAVE:DATA:TYPE .....                 | 305        |
| SYST:EXT:SIGOUT:POL .....           | 270        |                                      |            |
| SYST:EXT:SIGOUT:STAT .....          | 271        | <b>List of Errors .....</b>          | <b>306</b> |
| SYST:EXT:SSIGIO:MAP .....           | 272        | Command errors .....                 | 306        |
| SYST:EXT:SSIGIO:POL .....           | 274        | Execution errors .....               | 307        |
|                                     |            | Product-specific errors .....        | 308        |
|                                     |            | Query errors .....                   | 308        |
|                                     |            | Operation complete event errors .... | 309        |
|                                     |            | Product-dependent errors .....       | 309        |

## Command processing time ... 312

### Tutorial..... 313

|  |     |
|--|-----|
| Programming AC Output.....                               | 313 |
| Programming DC Output .....                              | 315 |
| Trigger Subsystem.....                                   | 316 |
| Changing the output with triggers<br>(TRANsient).....    | 318 |
| Measurement (ACQuire).....                               | 321 |
| Power line abnormality simulations<br>(SIMulation) ..... | 326 |
| Sequence Operation (PROGram)....                         | 329 |
| Waiting for Operation Complete .....                     | 332 |
| Status Monitoring.....                                   | 333 |
| Error Checking.....                                      | 336 |
| Visual Basic 2017 .....                                  | 337 |

# Command List

## IEEE 488.2 common commands

### **\*CLS**

Clears all event registers including the status byte, event status, and error queue.

### **\*ESE**

Sets the event status enable register that is counted by the event summary bit (ESB) of the status byte.

### **\*ESR**

Queries the event status register. The event status register is cleared when read.

### **\*IDN**

Queries the model name and firmware version of the product.

### **\*OPC**

Sets the OPC bit (bit 0) of the event status register when all the commands that are in standby have been processed.

### **\*OPT**

Queries the optional interface boards that are installed in the product.

### **\*PSC**

Sets whether the event status enable register and service request enable register are cleared when the POWER switch is turned on.

### **\*RCL**

Recalls memory content.

### **\*RST**

Resets a portion of the product settings.

### **\*SAV**

Saves the panel settings to memory.

### **\*SRE**

Sets the service request enable register.

### **\*STB**

Queries the contents of the status byte register and the MSS (master summary status) message.

### **\*TRG**

Trigger command.

### **\*TST**

Executes a self-test.

### **\*WAI**

Prevents the device from executing subsequent commands until all operations that are in standby have completed.

## **ABORt Command**

---

### **ABOR**

Aborts configuration, changes, measurements, and other operations in all trigger subsystems (TRANsient, ACQuire, SIMulation, PROGram).

### **ABOR:ACQ**

Aborts measurement operations.

### **ABOR:PROG**

Stops the trigger function of the sequence operation.

### **ABOR:SIM**

Aborts the trigger function of the power line abnormality simulation.

### **ABOR:TRAN**

Aborts the trigger function for configuration changes.

## **DISPlay Command**

---

### **DISP:CONT**

Adjusts the screen brightness.

### **DISP:PHAS**

Selects the phase to display on the panel.

### **DISP:MMOD:CURRE**

Sets the current, power, or power factor to display on the screen.

### **DISP:MMOD:VOLT**

Sets the voltage (rms value, peak value, average value) to display on the screen.



**DISP:MMOD:VOLT:LTL**

Sets the voltage (phase voltage, line voltage) to display on the screen.

**HCOpy Command****HCOP:SDUM:DATA?**

Retrieves the screen capture of the present screen.

**INITiate Command****INIT:ACQ**

These commands invalidate the present measured data and start a new measurement.

**INIT:PROG**

Executes a sequence.

**INIT:SIM**

Executes a power line abnormality simulation.

**INIT:TRAN**

Starts the trigger function.

**INSTrument Command****INST/ INST:NSEL**

Selects the phase that SOURce and MEASure/FETCH commands apply to.

**INST:COUP**

Sets whether to select all phases.

**LXI Command****LXI:IDEN**

Turns the identification display on or off.

**MEASure/FETCH Command****FETC:<meas-item>/ MEAS:<meas-item>**

Queries the scalar measurement data specified with <meas-item>.

## **FETC:ARR:<harm-item>?/ MEAS:ARR:<harm-item>?**

Queries the harmonic data specified with <harm-item>.

## **MEMory Command**

### **MEM:REC**

Recalls contents saved in the ABC memory.

### **MEM:REC:CONF**

Sets whether the memory content is to be confirmed before recalling the ABC memory from the panel.

### **MEM:REC:PREV**

Displays the contents that are stored in the ABC memory.

### **MEM:SAVE**

Saves the frequency, AC voltage, DC voltage, and waveform bank number to memory.

## **OUTPut Command**

### **OUTP**

Set the output to on or off.

### **OUTP:IMP**

Enables or disables the output impedance.

### **OUTP:IMP:REAC**

Sets the reactance component of the output impedances.

### **OUTP:IMP:REAL**

Sets the resistance component of the output impedances.

### **OUTP:PHAS:OFF**

Enables or disables output-off phase control.

### **OUTP:PHAS:OFF:LEV**

Sets the output off phase angle.

### **OUTP:PHAS:ON**

Enables or disables output-on phase control.

### **OUTP:PHAS:ON:LEV**

Sets the output-on phase angle.

**OUTP:PON**

Sets the output state that the PCR-WE will be in when the power is turned on.

**OUTP:PROT:CLE**

Clears alarms.

**OUTP:PROT:WDOG**

Enables or disables the communication monitoring (WATCHDOG) timer.

**OUTP:PROT:WDOG:DEL**

Sets the delay time of the communication monitoring (WATCHDOG) timer.

**OUTP:SST**

Enables or disables soft start.

**OUTP:SST:FALL**

Enables or disables soft stop.

**OUTP:SST:TIME**

Sets the soft start rise time.

**OUTP:SST:TIME:FALL**

Sets the soft stop fall time.

**PROGram Command****PROG:CLE**

Sets all the steps in the sequence to their default values.

It takes about 3 seconds for the steps to return to their default values.

**PROG:EDIT**

Collectively sets a sequence step (frequency, voltage, signal change, step time, waveform bank, status output, trigger I/O, output).

**PROG:EDIT:FUNC:BANK**

Sets the waveform bank number of the sequence step.

**PROG:EDIT:JUMP**

Configures the jump settings of a sequence step.

**PROG:EDIT:IMP**

Sets whether to set the output impedance in the sequence step.

### **PROG:EDIT:IMP:REAC**

Sets the reactance component of the output impedance used in the sequence step.

### **PROG:EDIT:IMP:REAL**

Sets the resistance component of the output impedance used in the sequence step.

### **PROG:EDIT:PHAS:RAMP**

Sets the phase signal change of a sequence step.

### **PROG:EDIT:PHAS:STAR**

Sets the starting phase angle and whether to suddenly change the phase of the sequence step.

### **PROG:EDIT:PHAS:STOP**

Sets the ending phase angle of the sequence step.

### **PROG:EDIT:PHAS:UOFF**

Sets the U phase offset phase angle of a sequence step.

### **PROG:EDIT:PHAS:UV**

Sets the U-V phase difference of the sequence step.

### **PROG:EDIT:PHAS:UW**

Sets the U-W phase difference of the sequence step.

### **PROG:EDIT:VOLT**

Sets the unbalanced AC voltage of the sequence step

### **PROG:EDIT:VOLT:OFFS**

Sets the unbalanced DC voltage of the sequence step

### **PROG:EXEC**

Queries the sequence execution state.

### **PROG:LOOP**

Sets the sequence repetition count.

### **PROG:STAT**

Changes the execution state of the sequence.

### **PROG:STEP:END**

Sets the sequence ending step number

### **PROG:STEP:STAR**

Sets the sequence starting step number

## SENSe Command

### SENS:AVER:COUN

Sets the moving average count for current measurement.

### SENS:CURR:HOLD:CLE

Clears the hold time of the peak current..

### SENS:CURR:HOLD:TIME

Sets the hold time of the peak current.

### SENS:VOLT:AVER:COUN

Sets the moving average count for voltage measurement.

### SENS:VOLT:HOLD:CLE

Clears the hold time of the peak voltage.

### SENS:VOLT:HOLD:TIME

Sets the hold time of the peak voltage.

## SIMulation Command

### SIM:EXEC

Queries the execution status of power line abnormality simulations.

### SIM:POL

Sets the voltage regulation polarity of power line abnormality simulations.

### SIM:REP:COUN

Sets the number of repetitions of power line abnormality simulations.

### SIM:STAT

Executes or stops a power line abnormality simulation.

### SIM:T1:PHAS

Sets the voltage regulation starting phase of power line abnormality simulations.

### SIM:T1:PHAS:STAT

Selects whether the voltage regulation start of power line abnormality simulations will be set in terms of time or in terms of phase.

### SIM:T1:TIME

Sets the voltage regulation starting time of power line abnormality simulations.

### **SIM:T2:TIME**

Sets slope time 1 of power line abnormality simulations.

### **SIM:T3:TIME**

Sets the voltage regulation time of power line abnormality simulations.

### **SIM:T3:VOLT**

Sets the regulated voltage of power line abnormality simulations.

### **SIM:T4:TIME**

Sets slope time 2 of power line abnormality simulations.

### **SIM:T5:CYCL**

Sets the number of return cycles of power line abnormality simulations.

### **SIM:T5:CYCL:STAT**

Selects whether the period that the PCR-WE remains in the returned state is configured in terms of time or in terms of cycles for power line abnormality simulations.

### **SIM:T5:TIME**

Sets the return time of power line abnormality simulations.

## **[SOURce:]CURRent Command**

### **CURR**

Sets the upper limit of the output current.

### **CURR:PEAK**

Sets the output current peak limit of the positive electric potential

### **CURR:PEAK:LOW**

Sets the output current peak limit of the negative electric potential

### **CURR:PROT:STAT**

Sets how the PCR-WE acts when the current limit is exceeded.

### **CURR:PROT:TRIP:DEL**

Sets the time that must elapse before the output is turned off when the current limit is exceeded

## [SOURce:]FREQuency Command

### FREQ

Sets the frequency.

### FREQ:LIM:LOW

Sets the lower frequency limit.

### FREQ:LIM:UPP

Sets the upper frequency limit.

### FREQ:SYNC

Sets the sync function.

### FREQ:SYNC:MODE

Sets the input sync signal of the sync function.

### FREQ:SYNC:PHAS:DEL

Sets the synchronization delay phase angle of the synchronization function.

### FREQ:TRIG

Sets the frequency to change to when INIT/INIT:TRAN or a software trigger is sent.

## [SOURce:]FUNCtion Command

### FUNC:BANK

Specifies the number of the waveform bank that you want to execute.

## [SOURce:]VOLTage Command

### VOLT

Set the AC voltage.

### VOLT:COMP:MODE

Sets the compensation (voltage compensation).

### VOLT:COMP:RADJ

Sets the voltage to correct with regulation adjustment.

### VOLT:COMP:SOFT:CONT

Set the soft sensing control target.

### **VOLT:COMP:SOFT:TERM**

Set the sensing point.

### **VOLT:EXT:INP:EXTDC:ADJ:GAIN**

Sets the gain for when the input waveform is amplified using an external analog signal.

### **VOLT:EXT:INP:EXTDC:ADJ:OFFS**

Sets the offset for when the input waveform is amplified using an external analog signal.

### **VOLT:EXT:INP:EXTDC:APER**

Sets the measurement time for when the input waveform is amplified using an external analog signal.

### **VOLT:EXT:INP:EXTDC:SIGN:POL**

Sets the signal polarity of each channel for when the input waveform is amplified using an external analog signal.

### **VOLT:EXT:INP:EXTDC:SIGN:SOUR**

Sets the signal source for when the input waveform is amplified using an external analog signal.

### **VOLT:EXT:INP:FUNC:MODE**

Selects the parameter to control with the external analog signal.

### **VOLT:EXT:INP:VPR:ADJ:GAIN**

Sets the gain for when varying the voltage or frequency with the external analog signal.

### **VOLT:EXT:INP:VPR:ADJ:OFFS**

Sets the offset for when varying the voltage or frequency with the external analog signal.

### **VOLT:EXT:INP:VPR:MAP**

Sets the channel configuration for when varying the voltage or frequency with the external analog signal.

### **VOLT:EXT:INP:VPR:STAT**

Turns on or off the output of each channel for when varying the voltage or frequency with the external analog signal.

### **VOLT:LIM:LOW**

Sets the lower AC voltage limit.

### **VOLT:LIM:UPP**

Sets the upper AC voltage limit.



**VOLT:LTL**

Sets the line AC voltage.

**VOLT:OFFS**

Sets the DC voltage.

**VOLT:OFFS:LIM:LOW**

Sets the lower DC voltage limit.

**VOLT:OFFS:LIM:UPP**

Sets the upper DC voltage limit.

**VOLT:OFFS:LTL**

Sets the line DC voltage.

**VOLT:OFFS:TRIG**

Sets the DC voltage to change to when INIT:TRAN or a software trigger is sent.

**VOLT:PROT:LOW**

Sets the UVP value.

**VOLT:PROT:PEAK:LOW**

Sets the negative peak OVP value.

**VOLT:PROT:PEAK:UPP**

Sets the positive peak OVP value.

**VOLT:PROT:LOW:STAT**

Enables/disables UVP.

**VOLT:PROT:UPP**

Sets the OVP (rms) value.

**VOLT:RANG**

Sets the voltage range.

**VOLT:RESP**

Sets the response speed.

**VOLT:TRIG**

Sets the AC voltage to change to when INIT:TRAN or a software trigger is sent.

## STATus Command

### STAT:OPER

Queries the event of the OPERation status register.

### STAT:OPER:COND

Queries the condition of the OPERation status register.

### STAT:OPER:ENAB

Sets the enable register of the OPERation status register.

### STAT:OPER:NTR

Sets the negative transition filter of the OPERation status register.

### STAT:OPER:PTR

Sets the positive transition filter of the OPERation status register.

### STAT:OPER:INST

Queries the event of the OPERATION:INSTrument subregister.

### STAT:OPER:INST:COND

Queries the condition of the OPERATION:INSTrument subregister.

### STAT:OPER:INST:ENAB

Sets the enable register of the OPERATION:INSTrument subregister.

### STAT:OPER:INST:NTR

Sets the negative transition filter of the OPERATION:INSTrument subregister.

### STAT:OPER:INST:PTR

Sets the positive transition filter of the OPERATION:INSTrument subregister.

### STAT:OPER:INST:ISUM{1|2|3}

Queries the event of the OPERATION:INSTrument:ISUMmary{1|2|3} subregister.

### STAT:OPER:INST:ISUM{1|2|3}:COND

Queries the condition of the OPERATION:INSTrument:ISUMmary{1|2|3} subregister.

### STAT:OPER:INST:ISUM{1|2|3}:ENAB

Sets the enable register of the OPERATION:INSTrument:ISUMmary{1|2|3} subregister.

### STAT:OPER:INST:ISUM{1|2|3}:NTR

Sets the negative transition filter of the OPERATION:INSTrument:ISUMmary{1|2|3} subregister.

**STAT:OPER:INST:ISUM{1|2|3}:PTR**

Sets the positive transition filter of the OPERation:INSTrument:ISUMmary{1|2|3} subregister.

**STAT:QUES**

Queries the event of the QUEStionable status register.

**STAT:QUES:COND**

Queries the condition of the QUEStionable status register.

**STAT:QUES:ENAB**

Sets the enable register of the QUEStionable status register.

**STAT:QUES:NTR**

Sets the negative transition filter of the QUEStionable status register.

**STAT:QUES:PTR**

Sets the positive transition filter of the QUEStionable status register.

**STAT:QUES:INST**

Queries the event of the QUEStionable:INSTrument subregister.

**STAT:QUES:INST:COND**

Queries the condition of the QUEStionable:INSTrument subregister.

**STAT:QUES:INST:ENAB**

Sets the enable register of the QUEStionable:INSTrument subregister.

**STAT:QUES:INST:NTR**

Sets the negative transition filter of the QUEStionable:INSTrument subregister.

**STAT:QUES:INST:PTR**

Sets the positive transition filter of the QUEStionable:INSTrument subregister.

**STAT:QUES:INST:ISUM{1|2|3}**

Queries the event of the QUEStionable:INSTrument:ISUMmary{1|2|3} subregister.

**STAT:QUES:INST:ISUM{1|2|3}:COND**

Queries the condition of the QUEStionable:INSTrument:ISUMmary{1|2|3} subregister.

**STAT:QUES:INST:ISUM{1|2|3}:ENAB**

Sets the enable register of the QUEStionable:INSTrument:ISUMmary{1|2|3} subregister.

### **STAT:QUES:INST:ISUM{1|2|3}:NTR**

Sets the negative transition filter of the QUESTIONable:INSTrument:ISUMmary{1|2|3} subregister.

### **STAT:QUES:INST:ISUM{1|2|3}:PTR**

Sets the positive transition filter of the QUESTIONable:INSTrument:ISUMmary{1|2|3} subregister.

### **STAT:PRES**

Resets the ENABLE, PTRansition, and NTRansition filter registers of all status registers (including sub registers) to their default values.

## **SYSTem Command**

### **SYST:COMM:RLST**

Sets the product to remote or local mode.

### **SYST:CONF:ACC**

Enables/disables AC coupling.

### **SYST:CONF:ADJ:VOLT:FINE**

Sets the output voltage offset.

### **SYST:CONF:ADJ:VOLT:TERM:MODE**

Set whether the sensing function is enabled or disabled for the voltage offset setting.

### **SYST:CONF:FORM:FRAM**

Queries the number of units operating in parallel.

### **SYST:CONF:FORM:FRAM:INFO**

Queries the information about the specified PCR-WE.

### **SYST:CONF:FORM:PMOD**

Queries the number of power modules in the specified PCR-WE.

### **SYST:CONF:FORM:PMOD:INFO**

Queries the information about the specified power module.

### **SYST:CONF:FORM:PSAV:MAX**

Sets the maximum expected power of the power-saving function.

### **SYST:CONF:FORM:PSAV:MOD**

Sets all power modules to run using the power-saving function.

**SYST:CONF:FORM:PSAV:RES**

Resets the maximum expected power setting of the power-saving function.

**SYST:CONF:PHAS:UOFF**

Sets the absolute phase angle of the U phase relative to the reference phase.

**SYST:CONF:PHAS:UV**

Sets the U-V phase difference.

**SYST:CONF:PHAS:UW**

Sets the U-W phase difference.

**SYST:CONF:PON:STAT**

Sets the condition panel setting state when the POWER switch is turned on.

**SYST:CONF:SSUP**

Enables or disables the voltage surge suppression function.

**SYST:CONF:WIR/ SYST:CONF:NOUT**

Sets the output method.

**SYST:DATE**

Sets the date (UTC).

**SYST:ERR**

Reads the oldest error information or event information from the error queue.

**SYST:ERR:COUN**

Queries the number of errors occurring currently.

**SYST:EXT:DIG:READ**

Queries all the signal input states of SIGNAL IN channels (CTRL.1 to CTRL.4) and SIGNAL IO channels (DIO.1, DIO.2).

**SYST:EXT:DIG:WRIT**

Outputs the SIGNAL OUT channels (STAT.1 to STAT.4) and SIGNAL IO channels (DIO.1, DIO.2) collectively.

**SYST:EXT:SIGIN:MAP**

Maps SIGNAL IN channels (CTRL.1 to CTRL.3).

**SYST:EXT:SIGIN:POL**

Sets the polarity of the parameter to map to SIGNAL IN.

### **SYST:EXT:SIGIN:STAT**

Queries the signal level of the SIGNAL IN channel.

### **SYST:EXT:SIGOUT:MAP**

Maps SIGNAL OUT channels (STAT.1 to STAT.3).

### **SYST:EXT:SIGOUT:POL**

Sets the polarity of the parameter to map to SIGNAL OUT.

### **SYST:EXT:SIGOUT:STAT**

Sets the signal level of the SIGNAL OUT channel.

### **SYST:EXT:SSIGIO:MAP**

Maps SIGNAL IO channels (DIO.1, DIO.2).

### **SYST:EXT:SSIGIO:POL**

Sets the polarity of the parameter to map to SIGNAL IO.

### **SYST:EXT:SSIGIO:STAT**

Sets the signal level of the SIGNAL IO channel.

### **SYST:KLOC**

Sets or releases panel control lock.

### **SYST:LOC/ SYST:REM/ SYST:RWL**

This is an old style command.

### **SYST:OPT**

Queries the optional interface boards that are installed in the product.

### **SYST:PASS**

Enables a password-protected command.

### **SYST:PASS:CDIS**

Disable the password-protected command.

### **SYST:PASS:NEW**

Set the password.

### **SYST:PASS:STAT**

Queries whether a password-protected command is valid or invalid.

### **SYST:SEC:IMM**

Sanitizes the product to its factory default settings.

**SYST:SLE**

Turns the sleep function on and off.

**SYST:SLE:EXEC**

Activates sleep mode immediately.

**SYST:SLE:TIME**

Sets the time that must elapse before the product enters sleep mode.

**SYST:TIME**

Sets the time.

**SYST:TIME:ADJ**

Automatically synchronizes the system clock using the NTP server on the network.

**SYST:TZON**

Sets the time zone of the system clock.

**SYST:TZON:CAT**

Queries the time zone IDs that can be used.

**SYST:VERS**

Queries the version of the SCPI specifications that the product complies with.

**TRIGger Command**

---

**TRIG:ACQ**

Executes a software trigger on the ACQuire trigger subsystem.

**TRIG:ACQ:SOUR**

Sets the condition (trigger source) for actually starting the measurement after the ACQuire trigger subsystem receives an INIT:ACQ.

**TRIG:PROG**

Executes a software trigger on the PROGram trigger subsystem.

**TRIG:PROG:SOUR**

Sets the condition (trigger source) for actually starting the sequence operation after the PROGram trigger subsystem receives an INIT:PROG.

**TRIG:SIM**

Executes a software trigger on the SIMulation trigger subsystem.

## **TRIG:SIM:SOUR**

Sets the condition (trigger source) for actually executing the simulation after the SIMulation trigger subsystem receives an INIT:SIM.

## **TRIG:TRAN**

Executes a software trigger on the TRANsient trigger subsystem.

## **TRIG:TRAN:SOUR**

Sets the condition (trigger source) for actually changing the settings after the TRANsient trigger subsystem receives an INIT:TRAN.

## **WAVE Command**

---

### **WAVE:DATA:ARB**

Sets a user-defined waveform with block data at the waveform bank that you specify by its number.

### **WAVE:DATA:CLIP**

Sets the crest factor of the peak clipped waveform at the waveform bank that you specify by its number.

### **WAVE:DATA:IECP**

Sets the clip factor of the flat curve waveform at the waveform bank that you specify by its number.

### **WAVE:DATA:POIN**

Sets a user-defined waveform by specifying the waveform bank number and the point.

### **WAVE:DATA:SIN**

Sets the waveform bank that you specify by its number to sine wave.

### **WAVE:DATA:TYPE**

Queries the waveform type at the waveform bank that you specify by its number.



# Introduction

---

The PCR-WE Series Communication Interface Manual explains the settings that are used to control the PCR-WE series remotely through the following interfaces and the available commands.

- RS232C interface (standard)
- USB interface (standard)
- LAN interface (standard)
- GPIB interface (option)

When the product is operating under remote control, REMOTE appears on the front panel display. To switch the product back to local mode from the front panel, press LOCAL.

## Reading environment

---

We recommend the following environment for viewing this manual.

PDF Reader: Adobe Reader

## Intended readers

---

This manual is written for readers with sufficient basic knowledge of how to control measuring instruments using a PC.

Familiarize yourself with the syntax of the SCPI commands that are used with the product before you use them.

## Structure of the manual

---

This manual consists of the following sections.

- Overview
- Setup
- Message Overview
- Command
- Appendix
- Tutorial

## Trademarks

---

Microsoft Windows is a trademark of Microsoft Corporation in the United States and/or other countries.

All other company and product names used in this guide are trademarks or registered trademarks of their respective owners.

## Firmware version of the product to which this manual applies

---

This manual applies to products with firmware versions 1.4x.

## Measuring instrument interface standards

---

This product complies with the following standards.

- IEEE Std 488.2-1992 IEEE Standard Codes, Formats, Protocols, and Common Commands For Use With IEEE Std 488.1-1987
- IEEE Std 488.1-1987 IEEE Standard Digital Interface for Programmable Instrumentation
- Standard Commands for Programmable Instruments (SCPI) version 1999.0
- Universal Serial Bus Specification Rev 2.0
- Universal Serial Bus Test and Measurement Class Specification (USBTMC) Rev 1.0
- Universal Serial Bus Test and Measurement Class, Subclass USB488 Specification (USBTMC-USB488) Rev 1.0
- TCP/IP Instrument Protocol Specification VXI-11 Rev 1.0 1995
- TCP/IP-IEEE488.2 Interface Specification VXI-11.3 Draft 0.3 1995
- LXI Device Specification 2016 Rev 1.5
- LXI HiSLIP Extended Function Rev 1.02
- LXI Extended Function IPv6 Rev 1.1
- IVI-6.1 IVI High-Speed LAN Instrument Protocol (HiSLIP) Rev 1.1
- VPP-4.3 The VISA Library 2015 Rev 5.5

## Copyright

---

The contents of this manual may not be reproduced, in whole or in part, without the prior consent of the copyright holder.

The specifications of this product and the contents of this manual are subject to change without prior notice.

Copyright©2018

## Installing the VISA Library

VISA (Virtual Instrument Software Architecture) was developed by the IVI Foundation. It is the standard specification for measurement instrument connection software.

To use the VISA library (VISA COM) with the I/O library, the VISA library must be installed on the controller (Windows).

To use the LAN interface to control the product, middleware that supports the SC-PI-Telnet, VXI-11, HiSLIP, or SCPI-RAW protocol is required. The middleware is installed automatically by the VISA library.

You have to install one of the following VISA libraries (driver software that is implemented according to the VISA specifications).

- NI-VISA by National Instruments Corporation (Ver. 5.1.1 or later)
- Keysight VISA (Keysight IO Libraries Suite 16.0 or later) by Keysight Technologies
- KI-VISA Ver. 5.0.4 or later

—Note—

Depending on the interface, you may not be able to use your VISA library if it is an older version than that specified.

## Setting Up the Interface

The product is standard equipped with RS232C, USB, and LAN interfaces.

There is no need to switch interfaces. All interfaces can be used simultaneously. Each interface can be turned off using CONFIG settings.

[RS232C \(standard\)](#)

[USB \(standard\)](#)

[LAN \(standard\)](#)

[Accessing and Operating the Product from a Web Browser \(LAN\)](#)

[GPIB \(option\)](#)



### **WARNING**

**If the remote control via digital interface fails to work properly, an unexpected operation may occur that may cause electric shock, fire, physical damage to the load, and so on. If you are going to remotely control the PCR-WE from a distance, take safety measures such as using a watchdog timer.**

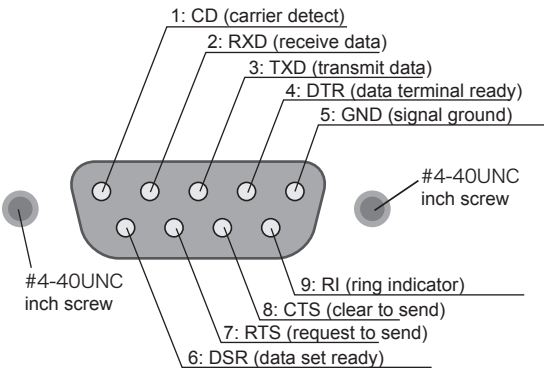
# RS232C (standard)

## ■ RS232C connection

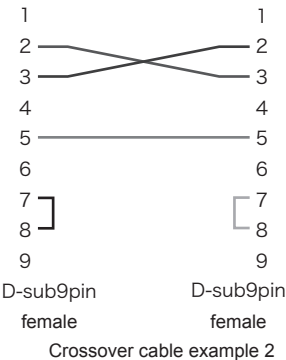
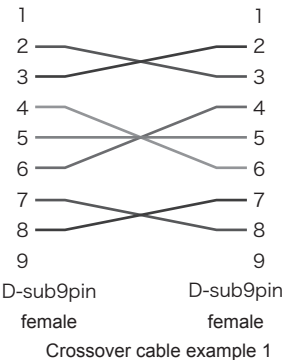
Turn off the product and the PC.

For the RS232C cable, use a D-sub, 9-pin, female-to-female crossover cable. The port pinout is shown below.

If you are not using flow control, you do not have to connect all the pins. (See the second crossover cable wiring example below.)



Viewing the front panel



9-pin AT connector (Flow control cannot be performed using the cable described in the second crossover cable wiring example.)

## ■ RS232C settings

The RS232C protocol is shown in the following table.

The factory default RS232C settings are RS232C enabled and 19200 bps data rate.

| Parameter    | Value  |
|--------------|--|
| Baudrate:    | 9 600 bps, <u>19 200 bps</u> , 38 400 bps, 57 600 bps, 115 200 bps |
| Data length: | 8 bits   |
| Stop bits:   | 1 bit  |
| Parity:      | None   |
| Flow Ctrl:   | NONE, Xon/Xoff   |

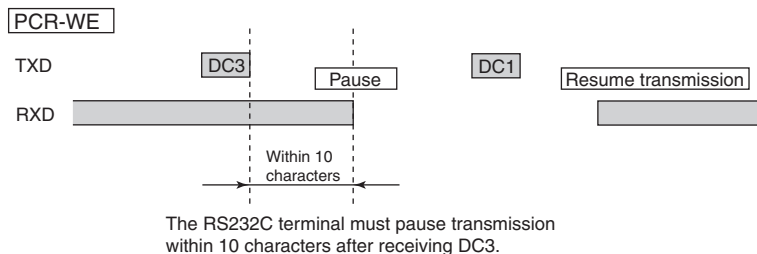
- 1 Press CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > RS232C (F3).**  
The present RS232C settings are displayed.
- 2 To enable RS232C, press ENABLE (F1). To disable it, press DISABLE (F2).**
- 3 Press BITRATE (F3) to set the data rate.**  
For the settings, see the table under Protocol below.
- 4 Press FLOW CTRL (F4) to set flow control.**  
To disable flow control, press NONE (F1). To enable it, press XON/XOFF (F2).
- 5 Press ESC > APPLY (F5).**  
To cancel, press CANCEL (F6).
- 6 Turn the PCR-WE's POWER switch off and then back on.**  
The settings take effect.

## ■ X-Flow control

The transmission/reception on the PCR-WE/WE2 can be controlled using Xon/Xoff. DC (device control) codes are used as control codes.

Data may not be received properly if flow control is not used.

| Code       | Function                  | ASCII Code |
|------------|---------------------------|------------|
| DC1 (Xon)  | Transmission request      | 11H        |
| DC3 (Xoff) | Transmission stop request | 13H        |



## ■ Break signal

The break signal is used as a substitute for the IEEE488.1 dcl/sdc (Device Clear, Selected Device Clear) message.

### —Note—

To use the RS232C interface, a "SYSTem:COMM:RLST REM" command must be sent to set the product to remote mode. To use remote programming, send "SYST:COMM:RLST REM" at the beginning of the program.



## USB (standard)

---

To use the USB interface to control the product, a device driver that supports the USB Test & Measurement class (USBTMC) must be installed on the controller. The USBTMC driver is installed automatically by the VISA library.

### ■ USB connection

Connect the product to a PC using a USB cable.

### ■ USB settings

The factory default USB setting is “USB enabled.”

- 1** Press **CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > USB (F2)**.
- 2** To enable USB, press **ENABLE (F1)**. To disable it, press **DISABLE (F2)**.
- 3** Press **APPLY (F5)**.  
To cancel, press **CANCEL (F6)**.
- 4** Turn the **PCR-WE's POWER switch off and then back on**.  
The settings take effect.

### ■ Service request

The product is equipped with service request and serial polling functions.

### ■ USB function

Complies with USB Specification 2.0

Complies with USBTMC Specification 1.0 and USBTMC-USB488 Specification 1.0

Baud rate: 480 Mbps maximum (high speed)

VID (vendor ID)

0x0B3E

PID (product ID)

PCR-WE: 0x104D

PCR-WE2: 0x104E

PCR-WE2R: 0x1054

## LAN (standard)

---

To use the LAN interface to control the product, middleware that supports the SCPI-Telnet, VXI-11, HiSLIP, or SCPI-RAW protocol is required. The middleware is installed automatically by the VISA library.

The LAN interface board has a Web browser interface (Web Browser Interface). You can configure the LAN interface settings from your PC's Web browser.

For information on topics such as connecting to your corporate LAN, your IP address, your host name, and security, contact your network administrator.

If you are using a host name (a Bonjour host name), you have to install Apple Bonjour.

### ■ LAN connection

Use a standard LAN cable (category 5 and straight) to connect the product to a network hub or router.

## ■ LAN settings

### Checking the settings

For IPv4, press CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > LAN (F1). For IPv6, press CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > LAN (F1) > IPv6 (F6).

| Parameter          | IPv4 value            | IPv6 value                   |
|--------------------|-----------------------|------------------------------|
| Status             | Present LAN status    |                              |
| IP Address Source  | IP address assignment |                              |
| Grobal Address     | —                     | Global address               |
| IP Address         | IP address            | —                            |
| LinkLocal Address  | —                     | IP address                   |
| Subnet Mask        | Subnet mask           | —                            |
| Default Gateway    | Default gateway       |                              |
| DNS Servers        | DNS server address    | —                            |
| WINS Servers       | WINS server address   | —                            |
| Primary DNS        | —                     | Primary DNS server address   |
| Secondary DNS      | —                     | Secondary DNS server address |
| mDNS Hostname      | mDNS host name        |                              |
| mDNS Serice Name   | mDNS service name     |                              |
| DDNS Hostname      | DDNS host name        |                              |
| Domain             | domain                |                              |
| NetBIOS Name       | NetBIOS name          | —                            |
| SCPI-RAW Port SCPI | TCP/IP socket port    |                              |
| SCPI-Telnet Port   | SCPI Telnet port      |                              |
| HiSLIP Port        | HiSLIP port           |                              |
| MAC Address        | MAC address           |                              |

## Change the parameter value.

Normally, set “IP Address Assign” to “AUTO” (factory default setting).

| Parameter                            | Value <sup>*1</sup>  |
|--------------------------------------|--|
| IP Address Assign                    | <u>AUTO</u> (auto), STATIC (fixed), DISABLE (disable IPv4/IPv6)  |
| IP Address <sup>*2</sup>             | 0.0.0.0 to 254.254.254.254 (IPv4)<br>0 to 9 and a to f allowed (IPv6)  |
| Subnet Mask <sup>*2*3</sup>          | 0.0.0.0 to 255.255.255.255   |
| Subnet Prefix Length <sup>*2*4</sup> | 16 to 112  |
| Default Gateway <sup>*2</sup>        | 0.0.0.0 to 254.254.254.254 (IPv4)<br>0 to 9 and a to f allowed (IPv6)  |
| Primary DNS <sup>*2</sup>            | 0.0.0.0 to 254.254.254.254 (IPv4)<br>0 to 9 and a to f allowed (IPv6)  |
| Secondary DNS <sup>*2</sup>          | 0.0.0.0 to 254.254.254.254 (IPv4)<br>0 to 9 and a to f allowed (IPv6)  |
| Primary WINS <sup>*2*3</sup>         | 0.0.0.0 to 254.254.254.254   |
| Secondary WINS <sup>*2*3</sup>       | 0.0.0.0 to 254.254.254.254   |
| Hostname                             | Enter the host name (up to 15 characters). Factory default is model name and serial number.  |
| Description                          | Enter the service name (up to 63 characters). The factory default setting is KIKUSUI XXXX AC Power Supply (where XXXX is the model name) and serial number |
| mDNS                                 | <u>Enable/Disable</u>  |
| Dynamic DNS                          | <u>Enable/Disable</u>  |
| NetBbIOS Over TCP/IP <sup>*3</sup>   | <u>Enable/Disable</u>  |

<sup>\*1</sup>. Factory default settings are underlined.

<sup>\*2</sup>. Can be set when the IP address assignment method is Static.

<sup>\*3</sup>. IPv4 only.

<sup>\*4</sup>. IPv6 only.

- 1 For IPv4, press **CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > LAN (F1)**. For IPv6, press **CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > LAN (F1) > IPv6 (F6)**.

The present settings are displayed.

- 2 Press **MODIFY (F3)**.

- 3 Select the parameter you want to change, press **EDIT (F3)**, and change the value.

For parameters other than the IP address assignment method, multicast DNS, dynamic DNS, and NetBIOS Over TCP/IP, press **OK (F4)** after you set the value. You can cancel a setting by pressing **CANCEL (F6)**.

- 4 When you are finished with the settings, press **APPLY (F5)**.

To cancel, press **CANCEL (F6)**.

- 5 Turn the PCR-WE's **POWER** switch off and then back on.

The settings take effect.



#### **WARNING**

Possible damage to the equipment and electric shock. The LAN interface can be accessed from anywhere on the network that the product is connected to. Change the security settings if necessary. The security settings that you can apply are: password protection and IP address access control.

#### **—Note—**

To use the LAN interface, a "SYST:COMM:RLST REM" command must be sent to set the product to remote mode. To use remote programming, send "SYST:COMM:RLST REM" at the beginning of the program.

### **■ Service request**

The product is equipped with service request and serial polling functions.

## ■ LAN function

Depending on the operations that you perform through your Web browser, the product may need to connect to the Internet.

Complies with LXI 1.5 Device Specification 2016

Complies with the SCPI-Telnet, VXI-11, HiSLIP, and SCPI-RAW protocols

Baud rate: 100 Mbps maximum (auto negotiation)

AUTO MDIX

Web browser interface features

Displays measuring instrument information, network information, VISA resource information, system information, license information

Change network settings, configure security settings, set passwords

Use the simple power supply control application

## ■ Resetting the LAN settings

You can return all LAN settings except the host name and service name to their factory default settings.

1

**Press CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > LAN (F1) > RESET (F4).**

Pressing CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > LAN (F1) > IPv6 (F6) > RESET (F4) performs the same operation.

A confirmation screen appears.

2

**Press OK (F4).**

The interface setting is reset.

To cancel, press CANCEL (F5).

## ■ Returning the LAN settings to their factory default settings

You can return the LAN settings to their factory default settings.

### 1 Press **CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > LAN (F1) > DEFAULT (F4)**.

Pressing **CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > LAN (F1) > IPv6 (F6) > DEFAULT (F4)** performs the same operation.

A confirmation screen appears.

### 2 Press **OK (F4)**.

The interface settings are returned to the factory default values.

To cancel, press **CANCEL (F5)**.

## ■ When using SCPI-Telnet or SCPI-RAW

If you send consecutive setting commands at a high speed, the product reception buffer may overflow. Either do not send consecutive commands—send query commands and read the responses at fixed intervals—or reduce the command transmission frequency. If an error (-363 Input buffer overrun) occurs, close VISA once, and then open it again to reconnect.

## **Accessing and Operating the Product from a Web Browser (LAN)**

You can configure the LAN interface settings from your PC's Web browser. Use the latest browser version.

The website URL is the product's IP address with http:// added in front of it.

If a VISA library is in use, a function for searching the VXI-11 measurement instrument with the application supplied by the vendor (National Instruments NI-MAX, Keysight Connection Expert, Kikusui KI-VISA Instrument Explorer, or the like) is available. You can open the Web browser interface by simply searching for the instrument and clicking the Web link that appears in the search result.

Example: When the IP address is 169.254.7.8

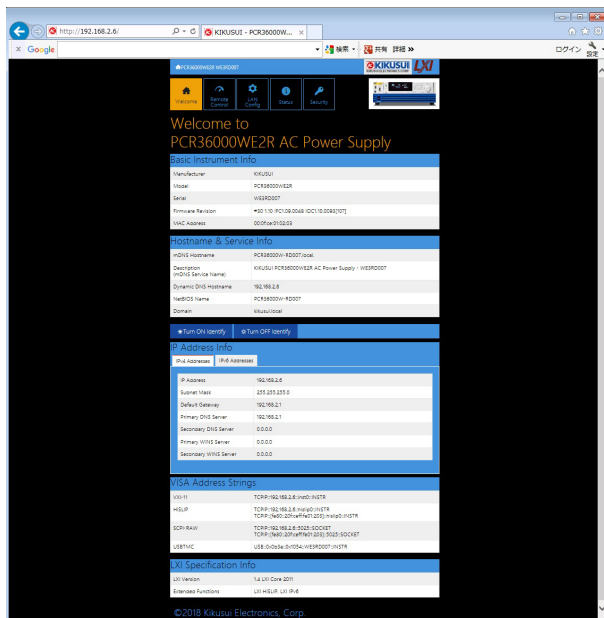
http://169.254.7.8



## Welcome page

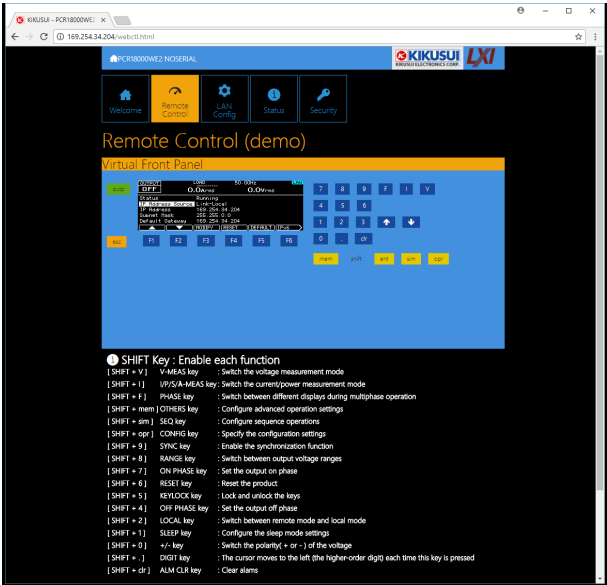
When you connect to the Web browser interface, a WELCOME page appears first.

This page shows the measuring instrument information, network information, and VISA resource (I/O resource) information. Click the navigation menu to go to another page.



## ■ Remote Control page

You can remotely control the PCR-WE from a browser. The various buttons have the same functions as those on the front panel of the PCR-WE.

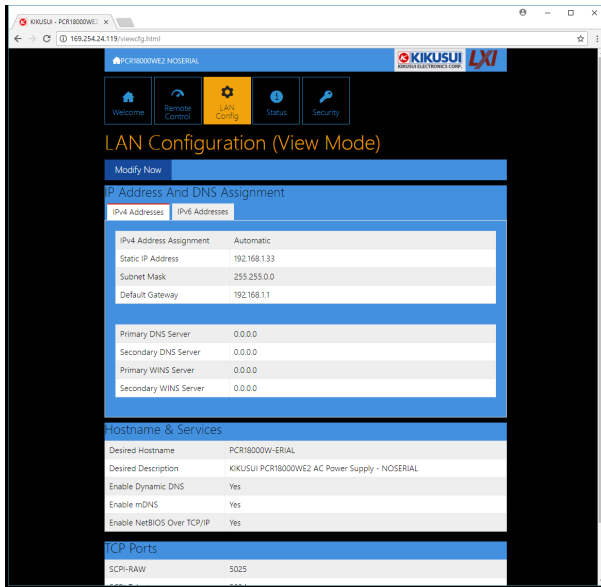


### SHIFT Key

Clicking SHIFT and then a button enables the displayed function.

## ■ LAN Configuration page

You can display (View Mode) and change (Modify Mode) the network settings.



### Navigation (View Mode)

Modify Now: Changes to the network setting edit screen (Modify Mode).

### Navigation (Modify Mode)

Undo: Returns the edited contents to the state before editing.

Apply: Applies the edited contents.

Reset: Resets the network settings.

Default: Returns the network settings to the factory default settings.

Back to View Mode: Changes to the network setting view screen (View Mode).

## **IP Address Assignment**

You can set the IP address. You can choose between automatic assignment and assignment of a fixed address.

In the case of automatic assignment of IP address, we recommend using the DHCP server function using a router as far as possible.

If the DHCP server function is not used, it takes about 60 seconds until determination that address assignment with DHCP has failed. Then, an address between 169.254.0.0 to 169.254.255.255 is assigned by link local address (Auto-IP).

## **DNS Server Assignment**

Sets the address of the DNS server.

## **WINS Server Assignment**

Sets the address of the WINS server.

## **Hostname & Services**

You can set the host name and so on. If you set the host name, you can use it in place of the IP address to access the LAN interface. Normally, we recommend that you select "Enable Dynamic DNS", "Enable mDNS", and "Enable NetBIOS Over TCP/IP".

If you leave the Hostname and Description boxes empty and click "Apply," the host name will be created from the model name and serial number.

## **TCP Ports (View Mode)**

The number of the TCP port in use is displayed. You cannot change the port number.

## Reset and factory default settings

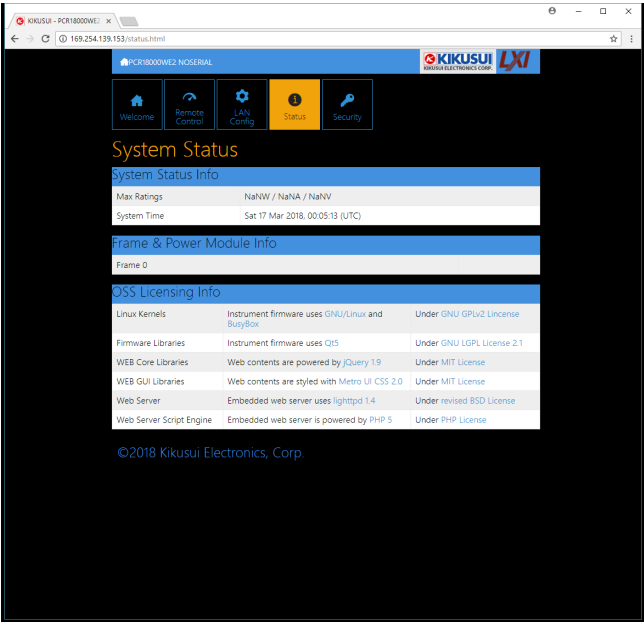
If you click Reset or Default, network settings are changed as follows.

The items with an X mark are returned to their default values.

| Reset | Default | Parameter                  | Default value   |
|-------|---------|----------------------------|---|
| X     | X       | Assignment Method          | DHCP:ON, Auto-IP:ON, Static:OFF                           |
| X     | X       | DNS Server Assignment      | 0.0.0.0   |
| X     | X       | WINS Server Assignment     | 0.0.0.0   |
| —     | X       | Desired Hostname           | <Model name> - <Last 5 digits of serial number>           |
| —     | X       | Desired Description        | KIKUSUI <model name> AC Power Supply<br>- <serial number> |
| X     | X       | Enable Dynamic DNS         | Enable  |
| X     | X       | Enable mDNS                | Enable  |
| X     | X       | Enable NetBIOS Over TCP/IP | Enable  |

## ■ System Status page

This page shows the system information and the license information of the open-source software.



The screenshot shows a web browser window displaying the 'System Status' page of a Kikusui LXI device. The page has a dark blue header with the Kikusui logo and 'LXI' branding. Below the header is a navigation bar with icons for Welcome, Remote Control, LAN Config, Status (highlighted), and Security. The main content area is titled 'System Status' and contains several sections:

- System Status Info**: A table showing Max Ratings (NaNW / NaNV / NaNV) and System Time (Sat 17 Mar 2018, 00:05:13 (UTC)).
- Frame & Power Module Info**: A table showing Frame 0.
- OS/Licensing Info**: A table listing various software components and their licenses.

At the bottom of the page, it says '©2018 Kikusui Electronics, Corp.'

| System Status Info |                                 |  |
|--------------------|---------------------------------|--|
| Max Ratings        | NaNW / NaNV / NaNV              |  |
| System Time        | Sat 17 Mar 2018, 00:05:13 (UTC) |  |

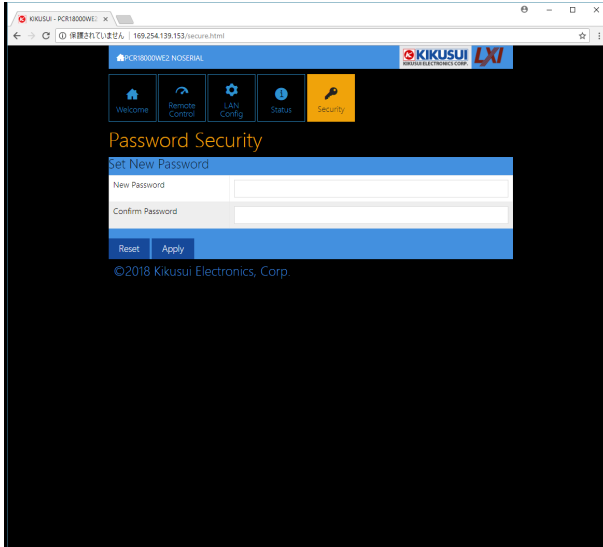
| Frame & Power Module Info |  |  |
|---------------------------|--|--|
| Frame 0                   |  |  |

| OS/Licensing Info        |  |                            |
|--------------------------|--|----------------------------|
| Linux Kernels            | Instrument firmware uses GNU/Linux and Busybox | Under GNU GPLv2 License    |
| Firmware Libraries       | Instrument firmware uses Qt5                   | Under GNU LGPL License 2.1 |
| WEB Core Libraries       | Web contents are powered by JQuery 1.9         | Under MIT License          |
| WEB GUI Libraries        | Web contents are styled with Metro UI CSS 2.0  | Under MIT License          |
| Web Server               | Embedded web server uses lighttpd 1.4          | Under revised BSD License  |
| Web Server Script Engine | Embedded web server is powered by PHP 5        | Under PHP License          |

## ■ Password Security page

You can set and change the password for the Web browser interface here.



When a password has been set, that password is required in order to use the following functions.

- Remote control from Remote Control page

- Editing of LAN Configuration page

- Changing/deleting the password

### Set New Password

Enter the password.

You can use alphanumeric characters, hyphens, and underscores for the password. The first character must be an alphabet. You can enter up to 15 characters.

### Changing or deleting the password

After the password has been set, the screen for changing the password appears when you enter the password.

To change the password, enter the present password in “Current Password”, enter the new password in “New Password” and “Confirm Password”, and then click “Apply”.

To disable password protection, enter the present password in “Current Password”, leave “New Password” and “Confirm Password” blank, and click “Apply”.

**If you forget the password**

If you forget the password, reset the LAN interface setting in the CONFIG settings or initialize the product to its factory default settings.



## **GPIB (option)**

The information in this section is valid only when the optional GPIB interface board is installed.

### **■ GPIB connection**

Connect the product to a PC using a standard IEEE488 cable.

### **■ GPIB settings**

- 1 Press CONFIG (SHIFT+OPR MODE) > COM-I/F (F1) > GPIB (F4).**  
 The present GPIB settings are displayed.
- 2 To enable GPIB, press ENABLE (F1). To disable it, press DISABLE (F2).**
- 3 Press ADDRESS (F2) to set the GPIB address.**  
 The factory default GPIB address is 5.
- 4 Press ESC > APPLY (F5).**  
 To cancel, press CANCEL (F6).
- 5 Turn the PCR-WE's POWER switch off and then back on.**  
 The settings take effect.

### **■ GPIB function**

| Function             | Subset | Description           |
|----------------------|--------|-----------------------|
| Source handshaking   | SH1    | Full capability       |
| Acceptor handshaking | AH1    | Full capability       |
| Talker               | T8     | Function available    |
| Listener             | L4     | Function available    |
| Service request      | SR0    | No capability         |
| Remote local         | RL0    | No capability         |
| Parallel polling     | PP0    | No capability         |
| Device clear         | DC0    | No capability         |
| Device trigger       | DT0    | No capability         |
| Controller           | C0     | No capability         |
| Electrical interface | E1     | Open-collector driver |

## ■ Functional Limitations

The following GPIB-specific functions cannot be used on the PCR-WE/WE2.

Serial polling, parallel polling

Remote local control

Device clear

However, for serial polling and remote local control, equivalent functions can be used through similar commands.

| GPIB function               | Substitute command |
|-----------------------------|--------------------|
| Serial Polling              | *STB?              |
| GET (Group Execute Trigger) | *TRG               |
| GTL (Go To Local)           | SYST:COMM:RLST LOC |
| REN (Remort Enable)         | SYST:COMM:RLST REM |
| LLO (Local Lock Out)        | SYST:COMM:RLST RWL |

# About Commands

The information that is transferred between the controller (PC) and the device (PCR-WE/WE2 series) is referred to as messages.

This product uses the SCPI language for these messages.

The messages that the PC sends to the product are commands. The messages that the product sends to the PC are responses.

## Command Hierarchy

SCPI is an ASCII-based command language that was designed for test and measuring equipment. The command structure is composed of the common roots and nodes that are the building blocks of the SCPI subsystem. A command consists of a program header, parameters, and punctuation marks.

The following table uses the SOURce subsystem as an example to explain the hierarchy.

| Program header | Parameter | Node level |
|----------------|-----------|------------|
| SOUR:          |           | Root node  |
| FREQ           |           | 2nd level  |
| :LIM           |           | 3rd level  |
| :UPP           | <numeric> | 4th level  |
| :LOW           | <boolean> | 4th level  |
| VOLT           |           | 2nd level  |
| :RANGE         |           | 3rd level  |
| :UPP           | <numeric> | 4th level  |
| :AUTO          | <boolean> | 4th level  |

A colon (:) separates a higher node from a lower node.

## Command Syntax

---

In this manual, SCPI commands are expressed in the following format.

```
MEASure[:SCALar]:CURRent:DC? {<numeric>|MINimum|MAXimum}
```

SCPI commands are also available in the short form. In the short form, the lowercase characters in SCPI commands are omitted.

SCPI commands can be sent either in the long form or short form. Because SCPI commands are not case-sensitive, CURR, CURRent, and curr are all acceptable as short form notations. In the long form, CURRENT, Current, and current are all acceptable.

- A space separates a program header and its parameters.
- Multiple parameters are separated by commas.
- Multiple commands are separated by semicolons (compound command).

```
OUTPut:PHASe:ON:STATe ON;LEVel 90
```

In the second command, OUTPut:PHASe:ON is omitted. This is possible because that path is set to OUTPut:PHASe:ON by the first command (OUTPut:PHASe:ON:STATe ON).

This compound command is equivalent to entering the following commands.

```
OUTPut:PHASe:ON:STATe ON
```

```
OUTPut:PHASe:ON:LEVel 90
```

If you specify a node that is not defined in the current path, an error will occur.

By using colons and semicolons, you can concatenate commands of different sub-systems.

```
SOURce:CURRent MINimum;:MEASure:CURRent:AC?
```

There are two root nodes in this compound command: SOURce and MEASure.

When the second command or later begins with a colon, the path that was specified by the previous command is cleared.

- The maximum length of a command that you can transmit on a single line is 128 bytes.

## ■ Special symbols

The special symbols that are used in this manual for the SCPI command syntax are explained below.

- Characters and numbers enclosed by { and } and delimited by "|" indicate that one of the delimited items is to be selected.

Do not include the { and } symbols in the actual program.

- <> denotes program data.

Do not include the < and > symbols in the actual program.

- [ ] denotes optional data.

When optional data is not sent with the program, the default value is applied.

Do not include the [ and ] symbols in the actual program.

## ■ Query

You can query the device settings and status.

To make a query, append a question mark to the end of the program header section.

If the query has parameters, insert a space after the question mark, and then write the parameters.

```
CURRent? MIN
```

## Response

This is the response to a query. It is a message always sent from the device to the host PC. It conveys device status or measured value to the PC.

—Note—

If you want to send two queries on separate lines, send the second query after you have received the response to the first one.

## ■ Program terminator

All commands must be terminated with a valid terminator.

|           | RS232C | USB       | LAN            |          |
|-----------|--------|-----------|----------------|----------|
|           |        |           | VXI-11, HiSLIP | SCPI-RAW |
| Receiving | LF     | LF or EOM | LF or END      | LF       |
| Sending   | LF     | LF+EOM    | LF+END         | LF       |

When you terminate a command string, the path is reset to the root level.

—Note—

CR (ASCII 0x0D) is not a terminator.

## ■ Common Commands

There are commands that are common to the IEEE-488.2 and SCPI standards for functions such as resetting devices and performing self-diagnoses. These common commands start with an asterisk ("\*"). These commands may have one or multiple parameters.

# Parameters

The SCPI parameter format is derived from the program parameter format that is defined in IEEE 488.2.

The program data expression format that the this product uses is shown below.

## Non-numeric parameters

---

### String data (String)

String data is used when a series of ASCII characters (20H to 7EH) are requested.

Enclose strings in single ( ' ') or double quotation ( " ") marks. The opening and closing quotation marks must match (you cannot mix single and double quotation marks).

```
SYSTem:TZONe "Asia/Tokyo"
```

If you want to include a quotation mark as part of the string, enter consecutive quotation marks (with no characters between them).

### Character data (Character)

Character data is used when only a limited number of values are available for a program setting. Responses are returned in short form.

```
TRIGger:SOURce {BUS|IMMediate}
```

### Boolean data (Boolean)

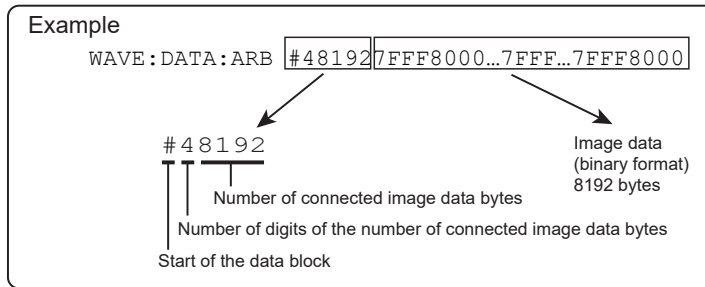
Boolean data is used to express a condition of 1 or 0, or ON or OFF. Responses are returned as 1 or 0.

```
OUTPut {ON|OFF|1|0}
```

## Block data (block)

Arbitrary block data that starts with #.

```
WAVE:DATA:ARB 5, #481927FFF8000.....7FFF8000
```



## Numeric parameters

### NR1

Represents an integer value.

Details are given in the “IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.”

### NR2

Represents a real number in floating-point format.

Details are given in the “IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.”

### NR3

Represents a real number in scientific notation.

Details are given in the “IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.”

If 380 is returned in the response data, it is returned as +3.80000E+02. Five decimal places are used.

### NRf

NRf is a generic term that includes NR1, NR2, and NR3.



## Numeric

Numeric parameter for values such as the decimal point, optional prefixes, and measurement units.

Numbers are expressed the same as NRf.

MINimum, MAXimum, and the like are available as substitutes for declaring certain values.

You can also use units such as V, A, and W in numeric parameters.

## Special form numeric parameters

---

The special form numeric parameters MINimum and MAXimum can be used as substitutes for the actual maximum and minimum values when the parameter is numeric.

The following example sets the overcurrent protection value to the minimum value.

```
SOURce:CURRent MINimum
```

You can query the minimum and maximum values for most parameters.

```
SOURce:CURRent? MAX
```

```
SOURce:CURRent? MIN
```

## Phase designation (channel list)

---

When the parameter is numeric, this product allows you to designate the phases you want to set using the syntax (@chanlist). The INSTRUMENT command settings do not apply to phase-designated commands.

U phase is (@1), V phase (@2), and W phase (@3).

In the following example, the V phase is set to 130 V.

```
VOLT 130, (@2)
```

Multiple phases can be designated at once. To designate U phase (@1) and W phase (@3)

```
VOLT 130, (@1, 3)
```

To designate all phases (@1 to @3)

```
VOLT 130, (@1:3)
```

## Measurement units

---

The default measurement units are listed below. Commands are accepted even if measurement units are not specified.

- |                        |                 |                  |                       |
|------------------------|-----------------|------------------|-----------------------|
| • V (voltage)          | • A (current)   | • W (power)      | • VA (apparent power) |
| • VAR (reactive power) | • DEG (degrees) | • HZ (frequency) | • H (reactance)       |
| • HR (hours)           | • MIN (minutes) | • S (seconds)    | • PCT (%)             |
| • OHM (resistance)     |                 |                  |                       |

The following optional prefixes are supported. If you use optional prefixes, specify the measurement unit.

- |             |            |             |
|-------------|------------|-------------|
| • M (milli) | • K (kilo) | • U (micro) |
|-------------|------------|-------------|

### —Note—

- The unit symbols in the International System of Units contain lowercase characters. The IEEE standard uses uppercase characters. SCPI commands are not case sensitive.
- Commands are accepted whether or not measurement units are specified.
- To enter “μ” in the data, use “U” instead.

## IEEE 488.2 common commands

### \*CLS

Clears all event registers including the status byte, event status, and error queue.

Clears the operation complete standby that was created by the \*OPC or \*OPC? command.

### **Command**

\*CLS

## \*ESE

Sets the event status enable register that is counted by the event summary bit (ESB) of the status byte.

### **Command**

\*ESE <NRf>

\*ESE?

### **Parameter**

Value: 0 to 255

Example: When \*ESE 16 is transmitted, bit 4 of the event status enable register is set. Each time the execution error bit (bit 4) of the event status register is set, the summary bit (ESB) of the status byte is set.

Response: NR1

**\*ESR**

Queries the event status register. The event status register is cleared when read.

**Command**

\*ESR?

Response: NR1

## \*IDN

Queries the model name and firmware version of the product.

### **Command**

\*IDN?

### **Response**

The response to \*IDN? is indicated below.

Response example for model PCR3000WE2, serial number WE3RD008, firmware version 1.00 IFC0.03.0035 IOC0.05.0079

```
KIKUSUI,PCR3000WE2,WE3RD008,1.00 IFC0.03.0035 IOC0.05.0079
```

is returned.

**\*OPC**

Sets the OPC bit (bit 0) of the event status register when all the commands that are in standby have been processed.

See IEEE 488.2-1992 section 12.5.3.

**Command**

\*OPC

\*OPC?

**Response**

Returns “1” when all the commands that are in standby have been processed.

## \*OPT

Queries the optional interface boards that are installed in the product.

During parallel operation, queries the optional interface boards that are installed in the master unit.

### **Command**

\*OPT?

### **Response**

Returns the installed options in comma-separated string format. Returns "0" if no options are installed.

"GPIB"            IB07-PCR-WE GPIB interface board

"500HZLMT"    500 Hz limit model



**\*PSC**

Sets whether the event status enable register and service request enable register are cleared when the POWER switch is turned on.

**Command**

\*PSC <boolean>

\*PSC?

Parameter <boolean>

|              |   |
|--------------|---|
| Value: ON(1) | When the POWER switch is turned on, the *ESE and *SRE settings are cleared.     |
| OFF(0)       | When the POWER switch is turned on, the *ESE and *SRE settings are not cleared. |

**Example**

\*PSC 0

Response: NR1

## **\*RCL**

Recalls memory content.

Clears alarms.

Aborts the trigger subsystem operation.

### **Command**

```
*RCL <NRf>
```

Parameter

Value: 0 to 9 memory number

Example

```
*RCL 1
```

**\*RST**

Resets a portion of the product settings.

Clears alarms. Aborts the trigger subsystem operation.

Clears \*OPC and \*OPC?. Clears the contents of the preset memory.

| Command              | Value when *RST is transmitted     |
|----------------------|------------------------------------|
| DISP:PHAS            | 0                                  |
| DISP:MMOD:CURRE      | RMS                                |
| DISP:MMOD:VOLT       | RMS                                |
| DISP:MMOD:VOLT:LTL   | OFF                                |
| INST                 | OUTP1                              |
| INST:NSEL            | 1                                  |
| INST:COUP            | ALL                                |
| OUTP                 | OFF                                |
| OUTP:IMP             | OFF                                |
| OUTP:IMP:REAC        | 0                                  |
| OUTP:IMP:REAL        | 0                                  |
| OUTP:PHAS:OFF        | OFF                                |
| OUTP:PHAS:OFF:LEV    | 0                                  |
| OUTP:PHAS:ON         | OFF                                |
| OUTP:PHAS:ON:LEV     | 0                                  |
| OUTP:PON             | SAFE                               |
| OUTP:SST             | OFF                                |
| OUTP:SST:FALL        | OFF                                |
| OUTP:SST:TIME        | 0.1                                |
| OUTP:SST:TIME:FALL   | 0.1                                |
| SENS:AVER:COUN       | 1                                  |
| SENS:CURRE:HOLD:TIME | 1                                  |
| SENS:VOLT:AVER:COUN  | 1                                  |
| SENS:VOLT:HOLD:TIME  | 1                                  |
| CURRE                | MAXimum                            |
| CURRE:PEAK           | MAXimum                            |
| CURRE:PEAK:LOW       | MAXimum                            |
| CURRE:PROT:STAT      | ON                                 |
| CURRE:PROT:TRIP:DEL  | 10                                 |
| FREQ                 | 50                                 |
| FREQ:LIM:LOW         | 1                                  |
| FREQ:LIM:UPP         | 5000 (500 on the 500 Hz LMT model) |
| REQ:SYNC             | OFF                                |
| FREQ:SYNC:MODE       | LINE                               |
| FREQ:SYNC:PHAS:DEL   | 0                                  |
| FREQ:TRIG            | 50                                 |
| FUNC:BANK            | 0                                  |
| ROSC                 | INTernal                           |
| VOLT                 | 0                                  |
| VOLT:COMP:MODE       | DISabled                           |

| Command                      | Value when *RST is transmitted |
|------------------------------|--------------------------------|
| VOLT:COMP:RADJ               | 0                              |
| VOLT:COMP:SOFT:CONT          | AC                             |
| VOLT:COMP:SOFT:TERM          | SENS                           |
| VOLT:EXT:INP:EXTDC:ADJ:GAIN  | 100 (all channels)             |
| VOLT:EXT:INP:EXTDC:ADJ:OFFS  | 0 (all channels)               |
| VOLT:EXT:INP:EXTDC:APER      | 0.1                            |
| VOLT:EXT:INP:EXTDC:SIGN:POL  | NORMal                         |
| VOLT:EXT:INP:EXTDC:SIGN:SOUR | EXternal                       |
| VOLT:EXT:INP:FUNC:MODE       | OFF                            |
| VOLT:EXT:INP:VPR:ADJ:GAIN    | 10 (all channels)              |
| VOLT:EXT:INP:VPR:ADJ:OFFS    | 0 (all channels)               |
| VOLT:EXT:INP:VPR:MAP         | ALL                            |
| VOLT:EXT:INP:VPR:STAT        | OFF                            |
| VOLT:LIM:LOW                 | 0                              |
| VOLT:LIM:UPP                 | 315                            |
| VOLT:LTL                     | 0                              |
| VOLT:OFFS                    | 0                              |
| VOLT:OFFS:LIM:LOW            | -445                           |
| VOLT:OFFS:LIM:UPP            | 445                            |
| VOLT:OFFS:LTL                | 0                              |
| VOLT:OFFS:TRG                | 0                              |
| VOLT:PROT:LOW                | 0                              |
| VOLT:PROT:PEAK:LOW           | -489.5                         |
| VOLT:PROT:PEAK:UPP           | 489.5                          |
| VOLT:PROT:LOW:STAT           | OFF                            |
| VOLT:PROT:UPP                | 489.5                          |
| VOLT:RANG                    | 157.5                          |
| VOLT:RESP                    | MEDium                         |
| VOLT:TRIG                    | 0                              |
| SYST:CONF:ACC                | OFF                            |
| SYST:CONF:FORM:PSAV:MAX      | Rated power                    |
| SYST:CONF:PHAS:UOFF          | 0                              |
| SYST:CONF:PHAS:UV            | 120                            |
| SYST:CONF:PHAS:UW            | 240                            |
| SYST:SLE                     | OFF                            |
| SYST:SLE:TIME                | 3600                           |
| TRIG:ACQ:SOUR                | IMMediate                      |
| TRIG:PROG:SOUR               | IMMediate                      |
| TRIG:SIM:SOUR                | IMMediate                      |
| TRIG:TRAN:SOUR               | IMMediate                      |

## Command

\*RST

**\*SAV**

Saves the panel settings to memory.

**Command**

```
*SAV <NRf>
```

**Parameter**

Value: 0 to 9 memory number

**Example**

```
*SAV 1
```

## \*SRE

Sets the service request enable register.

The service request enable register can be used to select which summary messages in the status byte register will perform service requests.

To clear the service request enable register, send \*SRE 0. If the register is cleared, service requests cannot be generated using status information.

### **Command**

\*SRE <NRf>

\*SRE?

### **Parameter**

Value: 0 to 255

Example: Sending \*SRE8 sets bit 3 of the service request enable register. Each time the summary bit (bit 3) of the QUEStionable status register in the status byte is set, a service request message is generated.

Response: NR1

**\*STB**

Queries the contents of the status byte register and the MSS (master summary status) message.

The response is the same as serial polling only with the exception that the MSS message appears in place of the RQS message in bit 6.

**Command**

\*STB?

Response: NR1

## \*TRG

Trigger command.

Executes triggers on the TRANsient trigger group and ACQuire trigger group.

This is a substitute command for IEEE 488.1 get (Group Execute Trigger).

If the device is in a state in which it does not accept triggers, an SCPI error (-211, "Trigger ignored") occurs.

See IEEE 488.2-1992 section 10.37.

### **Command**

\*TRG



**\*TST**

Executes a self-test.

You can check which error occurred with SYST:ERR? command. See IEEE 488.2-1992 section 10.38.

**Command**

\*TST?

**Response**

Returns 0 if there are no errors. Returns an error code if there is a problem.

## \*WAI

Prevents the device from executing subsequent commands until all operations that are in standby have completed.

### **Command**

\*WAI

# ABORt Command

This product has four different trigger subsystems (TRANsient, ACQuire, SIMulation, PROGram).

TRANsient is a subsystem for changing settings.

ACQuire is a measurement trigger subsystem. This subsystem is used to measure voltage, current, and power.

SIMulation executes power line abnormality simulations.

PROGram runs sequences.

## ABOR

Aborts configuration, changes, measurements, and other operations in all trigger subsystems (TRANsient, ACQuire, SIMulation, PROGram).

The product's trigger state immediately after it turns on is the same as its trigger state after it receives an ABOR command.

If you send an ABOR command while the PCR is executing measurements, the measured data is discarded.

You cannot specify a trigger subsystem with the ABOR command. It is always interpreted as ALL.

### Command

```
ABORt [:ALL]
```

## ABOR:ACQ

Aborts measurement operations.

If you send an ABOR command without initiating, the measured data is not discarded.

### **Command**

ABORt:ACQuire

## **ABOR:PROG**

Stops the trigger function of the sequence operation.

### **Command**

ABORt:PROGram

## **ABOR:SIM**

Aborts the trigger function of the power line abnormality simulation.

### **Command**

ABORt:SIMulation

**ABOR:TRAN**

Aborts the trigger function for configuration changes.

**Command**

```
ABORt:TRANsient
```

# DISPlay Command

## DISP:CONT

Adjusts the screen brightness.

### Command

```
DISPlay:CONTRast <NRf>
```

```
DISPlay:CONTRast?
```

### Parameter

Value: 1 to 3 (The default value is 3)

### Example

```
DISP:CONT 2
```

Response: NR1



## DISP:PHAS

Selects the phase to display on the panel.

This is invalid for single-phase operation.

### Command

```
DISPlay:PHASe <NR1>
```

```
DISPlay:PHASe?
```

### Parameter

|        |   |                      |
|--------|---|----------------------|
| Value: | 0 | All phases (default) |
|        | 1 | U phase              |
|        | 2 | V phase              |
|        | 3 | W phase              |

Settings are reset to default values when an \*RST command is sent.

### Example

```
DISP:PHAS 1
```

Response: NR1

## DISP:MMOD:CURR

Sets the current, power, or power factor to display on the screen.

### Command

```
DISPlay:MMODe:CURRent <character>
```

```
DISPlay:MMODe:CURRent?
```

### Parameter

|        |          |   |
|--------|----------|---|
| Value: | RMS      | Rms current (default)   |
|        | PEAK     | Peak current  |
|        | DC       | Average current   |
|        | WATTage  | Power   |
|        | VA       | Apparent power  |
|        | PF       | Power factor  |
|        | TWATtage | Total power (single-phase three-wire output and three-phase output only)          |
|        | TVA      | Total apparent power (single-phase three-wire output and three-phase output only) |
|        | TPF      | Total power factor (single-phase three-wire output and three-phase output only)   |

Settings are reset to default values when an \*RST command is sent.

### Example

```
DISP:MMOD:CURR RMS
```

Response: Characters

## DISP:MMOD:VOLT

Sets the voltage (rms value, peak value, average value) to display on the screen.

### Command

```
DISPlay:MMODE:VOLTage <character>
```

```
DISPlay:MMODE:VOLTage?
```

### Parameter

|        |      |                       |
|--------|------|-----------------------|
| Value: | RMS  | Rms voltage (default) |
|        | PEAK | Peak voltage          |
|        | DC   | Average voltage       |

Settings are reset to default values when an \*RST command is sent.

### Example

```
DISP:MMOD:VOLT PEAK
```

Response: Characters

## DISP:MMOD:VOLT:LTL

Sets the voltage (phase voltage, line voltage) to display on the screen.

This is invalid for single-phase operation.

### Command

```
DISPlay:MMODE:VOLTage:LTLLine[:STATe] <boolean>
```

```
DISPlay:MMODE:VOLTage:LTLLine[:STATe]?
```

### Parameter

|        |        |                         |
|--------|--------|-------------------------|
| Value: | ON(1)  | Line voltage            |
|        | OFF(0) | Phase voltage (default) |

Settings are reset to default values when an \*RST command is sent.

### Example

```
DISP:MMOD:VOLT:LTL ON
```

Response: NR1

# HCOPy Command

## HCOP:SDUM:DATA?

Retrieves the screen capture of the present screen.

### **Command**

HCOPy:SDUMp:DATA?

Response: Block (length: 3 KBytes to 3.5 KBytes)

# INITiate Command

This command cannot be executed simultaneously with the TRANSient, SIMulation, or PROGRAM subsystem.

## INIT:ACQ

These commands invalidate the present measured data and start a new measurement.

When the trigger source is set to IMM, measurement is started immediately. When the trigger source is set to BUS, the device waits for a software trigger and then starts a measurement.

### Command

```
INITiate[:IMMEDIATE]:ACquire
```

### Related command

```
TRIG:ACQ
```

```
TRIG:ACQ:SOUR
```

## INIT:PROG

Executes a sequence.

When the trigger source is set to IMM, the sequence is executed immediately. When the trigger source is set to BUS, the device waits for a software trigger and then executes the sequence.

### **Command**

```
INITiate[:IMMediate]:PROGram
```

### **Related command**

```
TRIG:PROG
```

```
TRIG:PROG:SOUR
```

## INIT:SIM

Executes a power line abnormality simulation.

When the trigger source is set to IMM, the sequence is executed immediately. When the trigger source is set to BUS, the device waits for a software trigger and then executes the sequence.

This is invalid when the output is off.

This is invalid if the DC voltage is not zero.

This is invalid when the T3 voltage is set in the H range area or when the present voltage range is L.

### **Command**

```
INITiate[:IMMEDIATE]:SIMulation
```

### **Related command**

```
TRIG:SIM
```

```
TRIG:SIM:SOUR
```



## INIT:TRAN

Starts the trigger function.

If the trigger source is set to IMM, change is immediately started. When the trigger source is set to BUS, the device waits for a software trigger and then starts the change.

### **Command**

```
INITiate[:IMMediate]:TRANsient
```

### **Related command**

```
TRIG:TRAN
```

```
TRIG:TRAN:SOUR
```

# INSTrument Command

This command selects the phase that SOURce and MEASure/FETCh commands apply to for single-phase three-wire output or three-phase output. The INSTrument command designation does not apply to phase-designated commands.

In the factory default condition, at power-on, and after a \*RST or \*RCL is sent, all phases are selected (INST:COUP ALL).

## INST/ INST:NSEL

Selects the phase that SOURce and MEASure/FETCh commands apply to.

This does not apply to phase-designated commands.

This is invalid when INST:COUP is set to ALL.

### Command

```
INSTrument[:SElect] <character>
```

```
INSTrument[:SElect]?
```

```
INSTrument:NSElect <NRf>
```

```
INSTrument:NSElect?
```

#### Parameter INST command

|        |         |                   |
|--------|---------|-------------------|
| Value: | OUTPut1 | U phase (default) |
|        | OUTPut2 | V phase           |
|        | OUTPut3 | W phase           |

#### Parameter INST:NSEL command

|        |   |                   |
|--------|---|-------------------|
| Value: | 1 | U phase (default) |
|        | 2 | V phase           |
|        | 3 | W phase           |

Settings are reset to default values when an \*RST command is sent.

### Example

```
INST OUTP1
```

### Response

Returns the designated phase in character format in response to INST?.

Returns the designated phase in NR1 format in response to INST:NSEL?.

## INST:COUP

Sets whether to select all phases.

This is invalid for single-phase output.

### Command

```
INSTRument:COUPle <character>
```

```
INSTRument:COUPle?
```

### Parameter

|        |      |                                    |
|--------|------|------------------------------------|
| Value: | NONE | None of the phases are selected.   |
|        | ALL  | All phases are selected (default). |

Settings are reset to default values when an \*RST command is sent.

### Example

```
INST:COUP ALL
```

Response: Characters

# LXI Command

## LXI:IDEN

Turns the identification display on or off.

When turned on, the LAN indicator blinks in the upper right of the front panel display of the PCR-WE that is being controlled through the LAN interface.

### Command

```
LXI:IDENtify[:STATe] <boolean>
```

```
LXI:IDENtify[:STATe]?
```

### Parameter

|        |        |                   |
|--------|--------|-------------------|
| Value: | ON(1)  | LAN blinking      |
|        | OFF(0) | LAN lit (Default) |

### Example

```
LXI:IDEN ON
```

Response: NR1

## MEASure/FETCh Command

The measurement function is mapped to the ACQuire trigger subsystem.

If you use the root node MEASure, the measurement is performed, and then the measured value is queried. If you use FETCh, the measured value is queried without a measurement being performed.

<meas-item>/<harm-item> is the last node of the command header.

->[Tutorial "Measurement \(ACQuire\)"](#)(p. 321)(p. 321)

## FETC:<meas-item>/ MEAS:<meas-item>

Queries the scalar measurement data specified with <meas-item>.

### Command

```
FETCh[:SCALar]:<meas-item>? [(@chanlist)]
```

```
MEASure[:SCALar]:<meas-item>? [(@chanlist)]
```

### Measurement <meas-item> list of each phase

Phase designation (@chanlist) possible

| <meas-item>                    | Description                             | Unit |
|--------------------------------|---|------|
| ALL                            | All (the 25 items below)                |      |
| CURRent:AC                     | Rms current without a DC component      | Arms |
| CURRent:ACDC                   | Rms current                             | Arms |
| CURRent:AMPLitude:MAXimum      | Current peak value                      | A    |
| CURRent:AMPLitude:MAXimum:HOLD | Hold value of peak current              | A    |
| CURRent:CREStfactor            | Current crest factor                    | —    |
| CURRent[:DC]                   | Average current                         | A    |
| FREQ                           | Frequency                               | Hz   |
| POWer[:DC]                     | Average power                           | W    |
| POWer:AC[:REAL]                | Power without a DC component            | W    |
| POWer:AC:APParent              | Apparent power without a DC component   | VA   |
| POWer:AC:REACtive              | Reactive power without a DC component   | VAR  |
| POWer:AC:PFACTOR               | Power factor without a DC component     | —    |
| POWer:ACDC[:REAL]              | Power                                   | W    |
| POWer:ACDC:APParent            | Apparent power                          | VA   |
| POWer:ACDC:REACtive            | Reactive power                          | VAR  |
| POWer:ACDC:PFACTOR             | Power factor                            | —    |
| VOLTage[:DC]                   | Average voltage                         | V    |
| VOLTage:AC                     | Rms voltage without a DC component      | Vrms |
| VOLTage:ACDC                   | Rms voltage                             | Vrms |
| VOLTage:AMPLitude:MAXimum      | Voltage peak value                      | V    |
| VOLTage:AMPLitude:MAXimum:HOLD | Hold value of peak voltage              | V    |
| LTLVoltage:DC                  | Average line voltage                    | V    |
| LTLVoltage:AC                  | Rms line voltage without a DC component | Vrms |
| LTLVoltage:ACDC                | Rms line voltage                        | Vrms |
| LTLVoltage:AMPLitude:MAXimum   | Line voltage peak value                 | V    |

## Measurement <meas-item> list of the total values of all phases

Phase designation (@chanlist) not possible

| <meas-item>               | Description                                 | Unit |
|---------------------------|---|------|
| POWER[:DC]:TOTal          | Total average power                         | W    |
| POWER:AC[:REAL]:TOTal     | Total power without a DC component          | W    |
| POWER:AC:APParent:TOTal   | Total apparent power without a DC component | VA   |
| POWER:AC:REActive:TOTal   | Total reactive power without a DC component | VAR  |
| POWER:AC:PFACtor:TOTal    | Total power factor without a DC component   | —    |
| POWER:ACDC[:REAL]:TOTal   | Total power                                 | W    |
| POWER:ACDC:APParent:TOTal | Total apparent power                        | VA   |
| POWER:ACDC:REActive:TOTal | Total reactive power                        | VAR  |
| POWER:ACDC:PFACtor:TOTal  | Total power factor                          | —    |

### Response

Returns the measurement of the designated phase in NR3[,<NR3>...] format in response to FETC:<meas-item>? [(@chanlist)]/ MEAS:<meas-item>? [(@chanlist)].

Returns the measurement of the phase designated by the INST command in NR3[,<NR3>...] format in response to FETC:<meas-item>?/ MEAS:<meas-item>?.

Returns 0 in response to a line voltage query during single-phase output.

Returns the measured values in the <meas-item> list order in comma-separated NR3 format in response to FETC:ALL?/MEAS:ALL?. If you specify several phases, the measured values of the first specified phase are returned in the <meas-item> list order. Then, the measured values of the next specified phase are returned in the <meas-item> list order.

### Example

```
INST:NSEL 1;COUP NONE
```

```
FETC:CURREN?           Returns the average current of the U phase.
```

```
INST:COUP ALL
```

```
FETC:CURREN?           Returns the average currents of all phases.
```

```
FETC:CURREN? (@2)       Returns the average current of the V phase.
```

```
FETC:CURREN? (@1:3)     Returns the average currents of all phases.
```

```
FETC:ALL? (@1)          Returns all scalar measurements (25 items, excluding the total value) of the U phase.
```

```
FETC:ALL?               Returns all scalar measurements (25 items × 3, excluding the total value) of all phases.
```

## FETC:ARR:<harm-item>?/ MEAS:ARR:<harm-item>?

Queries the harmonic data specified with <harm-item>.

### Command

FETCh:ARRay:<harm-item>? [(@chanlist)]

MEASure:ARRay:<harm-item>? [(@chanlist)]

### Measurement <harm-item> list of each phase

Phase designation (@chanlist) possible

0th to 50th

| <harm-item>                 | Description                       | Unit |
|-----------------------------|-----------------------------------|------|
| CURRent:HARMonic:AMPLitude] | Harmonic current                  | Arms |
| CURRent:HARMonic:PHASe      | Harmonic current phase            | Deg  |
| CURRent:HARMonic:THD        | Total current harmonic distortion | PCT  |
| VOLTage:HARMonic:AMPLitude] | Harmonic voltage                  | Vrms |
| VOLTage:HARMonic:PHASe      | Harmonic voltage phase            | Deg  |
| VOLTage:HARMonic:THD        | Total voltage harmonic distortion | PCT  |

### Response

Returns the measurements (0th to 50th) of the designated phase in a comma-separated NR3 format in response to FETC:ARR:<harm-item>? [(@chanlist)]/ MEAS:ARR:<harm-item>? [(@chanlist)].

Returns the measurements (0th to 50th) of the phase designated by the INST command in a comma-separated NR3 format in response to FETC:<meas-item>?/ MEAS:<meas-item>?.

Returns the measurements (0th to 50th) of the U phase in a comma-separated NR3 format when multiple phases are designated. Then, the V-phase measurements and W-phase measurements are returned in the same manner.

### Example

INST:NSEL 1;COUP NONE

FETC:ARR:CURR:HARM?

Returns the harmonic current of the U phase.

FETC:ARR:CURR:HARM? (@2)

Returns the harmonic current of the V phase.

FETC:ARR:CURR:HARM? (@1:3)

Returns the harmonic currents of all phases.



# MEMory Command

## MEM:REC

Recalls contents saved in the ABC memory.

You can view the contents that are stored in memory by using the MEM:REC:PREV command.

### Command

```
MEMory:RECall[:IMMediate] <NRf>
```

### Parameter

|        |   |          |
|--------|---|----------|
| Value: | 1 | Memory A |
|        | 2 | Memory B |
|        | 3 | Memory C |

## MEM:REC:CONF

Sets whether the memory content is to be confirmed before recalling the ABC memory from the panel.

### Command

```
MEMory:RECall:CONFirmation[:STATe] <boolean>
```

```
MEMory:RECall:CONFirmation[:STATe]?
```

### Parameter

|        |        |  |
|--------|--------|--|
| Value: | ON(1)  | The memory content is to be confirmed (default). |
|        | OFF(0) | The memory content is not to be confirmed.       |

### Example

```
MEM:REC:CONF ON
```

Response: NR1

**MEM:REC:PREV**

Displays the contents that are stored in the ABC memory.

**Command**

```
MEMory:RECall:PREView? <NRf>
```

**Parameter**

|        |   |          |
|--------|---|----------|
| Value: | 1 | Memory A |
|        | 2 | Memory B |
|        | 3 | Memory C |

**Response**

Returns the following items that are stored in the specified ABC memory. The items are returned in order in a comma-separated format.

For single-phase output, AC voltage <NR3>, DC voltage <NR3>, frequency <NR3>, waveform bank number <NR1>

For single-phase three-wire output, U phase AC voltage<NR3>, V phase AC voltage<NR3>, U phase DC voltage<NR3>, V phase DC voltage<NR3>, frequency<NR3>, U phase waveform bank number<NR1>, V phase waveform bank number<NR1>

For three-phase output, U phase AC voltage<NR3>, V phase AC voltage<NR3>, W phase AC voltage<NR3>, U phase DC voltage<NR3>, V phase DC voltage<NR3>, W phase DC voltage<NR3>, frequency<NR3>, U phase waveform bank number<NR1>, V phase waveform bank number<NR1>, W phase waveform bank number<NR1>

## MEM:SAVE

Saves the frequency, AC voltage, DC voltage, and waveform bank number to memory.

### Command

```
MEMory:SAVE[:IMMEDIATE] <NRf>
```

### Parameter

|        |   |          |
|--------|---|----------|
| Value: | 1 | Memory A |
|        | 2 | Memory B |
|        | 3 | Memory C |

# OUTPut Command

## OUTP

Set the output to on or off.

This command is invalid when a protection is activated, a simulation is running, or a sequence is running.

### Command

```
OUTPut[:STATe] <boolean>
```

```
OUTPut[:STATe] ?
```

### Parameter

|        |        |                      |
|--------|--------|----------------------|
| Value: | ON(1)  | Output on            |
|        | OFF(0) | Output off (default) |

Settings are reset to default values when an \*RST command is sent.

### Example

```
OUTP 1
```

Response: NR1

## OUTP:IMP

Enables or disables the output impedance.

Use OUTP:IMP:REAL or OUTP:IMP:REAC to set the output impedance.

### **Command**

```
OUTPut:IMPedance[:STATe] <boolean>
```

```
OUTPut:IMPedance[:STATe]?
```

### **Parameter**

Value:    ON(1)    Enables the output impedance  
         OFF(0)   Disables the output impedance (default)

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
OUTP:IMP ON
```

Response: NR1

## OUTP:IMP:REAC

Sets the reactance component of the output impedances.

This command is valid when the output impedance is on (OUTP:IMP ON).

### Command

```
OUTPut:IMPedance:REActive <numeric>[, (@chanlist)]
```

```
OUTPut:IMPedance:REActive? [ (@chanlist)]
```

### Parameter

Value: Reactance component (The default value is 0.)

|             | L range                    |  | H range                     |  |
|-------------|----------------------------|--|-----------------------------|--|
|             | Single-phase               | Single-phase<br>three-wire,<br>Three phase | Single-phase                | Single-phase<br>three-wire,<br>Three phase |
| PCR1000WE   | 80 $\mu$ H to 2000 $\mu$ H | --   | 320 $\mu$ H to 8000 $\mu$ H | --   |
| PCR2000WE   | 40 $\mu$ H to 1000 $\mu$ H | --   | 160 $\mu$ H to 4000 $\mu$ H | --   |
| PCR3000WE2  | 27 $\mu$ H to 667 $\mu$ H  | 80 $\mu$ H to 2000 $\mu$ H                 | 107 $\mu$ H to 2667 $\mu$ H | 320 $\mu$ H to 8000 $\mu$ H                |
| PCR6000WE2  | 13 $\mu$ H to 333 $\mu$ H  | 40 $\mu$ H to 1000 $\mu$ H                 | 53 $\mu$ H to 1333 $\mu$ H  | 160 $\mu$ H to 4000 $\mu$ H                |
| PCR12000WE2 | 7 $\mu$ H to 167 $\mu$ H   | 20 $\mu$ H to 500 $\mu$ H                  | 27 $\mu$ H to 667 $\mu$ H   | 80 $\mu$ H to 2000 $\mu$ H                 |
| PCR18000WE2 | 4 $\mu$ H to 111 $\mu$ H   | 13 $\mu$ H to 333 $\mu$ H                  | 18 $\mu$ H to 444 $\mu$ H   | 53 $\mu$ H to 1333 $\mu$ H                 |
| PCR24000WE2 | 3 $\mu$ H to 83 $\mu$ H    | 10 $\mu$ H to 250 $\mu$ H                  | 13 $\mu$ H to 333 $\mu$ H   | 40 $\mu$ H to 1000 $\mu$ H                 |
| PCR30000WE2 | 3 $\mu$ H to 67 $\mu$ H    | 8 $\mu$ H to 200 $\mu$ H                   | 11 $\mu$ H to 267 $\mu$ H   | 32 $\mu$ H to 800 $\mu$ H                  |
| PCR36000WE2 | 2 $\mu$ H to 56 $\mu$ H    | 7 $\mu$ H to 167 $\mu$ H                   | 9 $\mu$ H to 222 $\mu$ H    | 27 $\mu$ H to 667 $\mu$ H                  |

Unit: H

Settings are reset to default values when an \*RST command is sent.

### Example

```
OUTP:IMP:REAC 5UH
```

Response: NR3

## OUTP:IMP:REAL

Sets the resistance component of the output impedances.

This command is valid when the output impedance is on (OUTP:IMP ON).

### Command

```
OUTPut:IMPedance:REAL <numeric>[, (@chanlist)]
```

```
OUTPut:IMPedance:REAL? [ (@chanlist)]
```

### Parameter

Value: Resistance component (The default value is 0.)

|             | L range                       |  | H range                        |  |
|-------------|-------------------------------|--|--------------------------------|--|
|             | Single-phase                  | Single-phase<br>three-wire,<br>Three phase | Single-phase                   | Single-phase<br>three-wire,<br>Three phase |
| PCR1000WE   | 0 $\Omega$ to 2000 m $\Omega$ | --   | 0 $\Omega$ to 8000 m $\Omega$  | --   |
| PCR2000WE   | 0 $\Omega$ to 1000 m $\Omega$ | --   | 0 $\Omega$ to 4000 m $\Omega$  | --   |
| PCR3000WE2  | 0 $\Omega$ to 667 m $\Omega$  | 0 $\Omega$ to 2000 m $\Omega$              | 0 $\Omega$ to 2 667 m $\Omega$ | 0 $\Omega$ to 8000 m $\Omega$              |
| PCR6000WE2  | 0 $\Omega$ to 333 m $\Omega$  | 0 $\Omega$ to 1000 m $\Omega$              | 0 $\Omega$ to 1333 m $\Omega$  | 0 $\Omega$ to 4000 m $\Omega$              |
| PCR12000WE2 | 0 $\Omega$ to 167 m $\Omega$  | 0 $\Omega$ to 500 m $\Omega$               | 0 $\Omega$ to 667 m $\Omega$   | 0 $\Omega$ to 2000 m $\Omega$              |
| PCR18000WE2 | 0 $\Omega$ to 111 m $\Omega$  | 0 $\Omega$ to 333 m $\Omega$               | 0 $\Omega$ to 444 m $\Omega$   | 0 $\Omega$ to 1333 m $\Omega$              |
| PCR24000WE2 | 0 $\Omega$ to 83 m $\Omega$   | 0 $\Omega$ to 250 m $\Omega$               | 0 $\Omega$ to 333 m $\Omega$   | 0 $\Omega$ to 1000 m $\Omega$              |
| PCR30000WE2 | 0 $\Omega$ to 67 m $\Omega$   | 0 $\Omega$ to 200 m $\Omega$               | 0 $\Omega$ to 267 m $\Omega$   | 0 $\Omega$ to 800 m $\Omega$               |
| PCR36000WE2 | 0 $\Omega$ to 56 m $\Omega$   | 0 $\Omega$ to 167 m $\Omega$               | 0 $\Omega$ to 222 m $\Omega$   | 0 $\Omega$ to 667 m $\Omega$               |

Unit: OHM

Settings are reset to default values when an \*RST command is sent.

### Example

```
OUTP:IMP:REAL 50MOHM, (@1)
```

Response: NR3



## OUTP:PHAS:OFF

Enables or disables output-off phase control.

If the signal source is set to external signal source VOLT:EXT:INP:EXTDC:SIGN:-SOUR EXT), this command is invalid.

Use OUTP:PHAS:OFF:LEV to set the output off phase angle.

### Command

```
OUTPut:PHAS:OFF[:STATe] <boolean>
```

```
OUTPut:PHAS:OFF[:STATe]?
```

### Parameter

Value:    ON(1)    Enables output-off phase control  
         OFF(0)    Disables output-off phase control (default)

Settings are reset to default values when an \*RST command is sent.

### Example

```
OUTP:PHAS:OFF ON
```

Response: NR1

## OUTP:PHAS:OFF:LEV

Sets the output off phase angle.

This command is valid for AC output when the output-off phase control is on (OUTP:PHAS:OFF ON).

If the signal source is set to external signal source VOLT:EXT:INP:EXTDC:SIGN:-SOUR EXT), this command is invalid.

### **Command**

```
OUTPut:PHASe:OFF:LEVel <numeric>
```

```
OUTPut:PHASe:OFF:LEVel?
```

### **Parameter**

Value: 0.0 to 360.0 (The default value is 0)

Unit: DEG

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
OUTP:PHAS:OFF:LEV 90
```

Response: NR3

**OUTP:PHAS:ON**

Enables or disables output-on phase control.

If the signal source is set to external signal source VOLT:EXT:INP:EXTDC:SIGN:-SOUR EXT), this command is invalid.

Use OUTP:PHAS:ON:LEV to set the output-on phase angle.

**Command**

```
OUTPut:PHASe:ON[:STATe] <boolean>
```

```
OUTPut:PHASe:ON[:STATe] ?
```

**Parameter**

Value:    ON(1)    Enables output-on phase control  
          OFF(0)    Disables output-on phase control (default)

Settings are reset to default values when an \*RST command is sent.

**Example**

```
OUTP:PHAS:ON ON
```

Response: NR1

## OUTP:PHAS:ON:LEV

Sets the output-on phase angle.

This command is valid for AC output when the output-on phase control is on (OUTP:PHAS:ON ON).

If the signal source is set to external signal source VOLT:EXT:INP:EXTDC:SIGN:-SOUR EXT), this command is invalid.

### **Command**

```
OUTPut:PHASe:ON:LEVel <numeric>
```

```
OUTPut:PHASe:ON:LEVel?
```

### **Parameter**

Value: 0.0 to 360.0 (The default value is 0)

Unit: DEG

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
OUTP:PHAS:ON:LEV 90
```

Response: NR3

## OUTP:PON

Sets the output state that the PCR-WE will be in when the power is turned on.

This command is valid when the power-on state setting is set to AUTO (SYST:CONF:OUTP:PON:STAT AUTO).

### Command

```
OUTPut:PON[:STATe] <character>
```

```
OUTPut:PON[:STATe]?
```

### Parameter

Value:    **SAFE**    Starts with the output turned off (default)  
         **FORCE**   Starts with the output turned on.

Settings are reset to default values when an \*RST command is sent.

### Example

```
OUTP:PON FORC
```

Response: Characters

## OUTP:PROT:CLE

Clears alarms.

### **Command**

OUTPut:PROTection:CLEar

## OUTP:PROT:WDOG

Enables or disables the communication monitoring (WATCHDOG) timer.

Use OUTP:PROT:WDOG:DEL to set the delay time of the communication monitoring (WATCHDOG) timer.

### Command

```
OUTPut:PROTection:WDOG[:STATe] <boolean>
```

```
OUTPut:PROTection:WDOG[:STATe]?
```

### Parameter

|        |        |   |
|--------|--------|---|
| Value: | ON(1)  | Enables the communication monitoring timer            |
|        | OFF(0) | Disables the communication monitoring timer (default) |

### Example

```
OUTP:PROT:WDOG ON
```

Response: NR1

## OUTP:PROT:WDOG:DEL

Sets the delay time of the communication monitoring (WATCHDOG) timer.

When an alarm occurs, disable the communication monitoring timer (OUTP:PROT:WDOG OFF) first and then clear the alarms (OUTP:PROT:CLE).

This command is valid when the communication monitoring timer is enabled (OUTP:PROT:WDOG ON).

### **Command**

```
OUTPut:PROTection:WDOG:DElay <numeric>
```

```
OUTPut:PROTection:WDOG:DElay?
```

### **Parameter**

Value: 1 to 3600 (The default value is 60)

Unit: S

### **Example**

```
OUTP:PROT:WDOG:DEL 60
```

Response: NR3



**OUTP:SST**

Enables or disables soft start.

Use OUTP:SST:TIME to set the rise time.

You cannot enable soft start if the compensation is set to soft sensing function or regulation adjustment function (VOLT:COMP:MODE SOFT|RADJ).

**Command**

```
OUTPut:SStart[:STATe][:RISE] <boolean>
```

```
OUTPut:SStart[:STATe][:RISE]?
```

**Parameter**

Value:    ON(1)    Enables soft start  
         OFF(0)    Disables soft start (default)

Settings are reset to default values when an \*RST command is sent.

**Example**

```
OUTP:SST ON
```

Response: NR1

## OUTP:SST:FALL

Enables or disables soft stop.

Use OUTP:SST:TIME:FALL to set the fall time.

You cannot enable soft stop if the compensation is set to soft sensing function or regulation adjustment function (VOLT:COMP:MODE SOFT|RADJ).

### **Command**

```
OUTPut:SStart[:StAtE]:FALL <boolean>
```

```
OUTPut:SStart[:StAtE]:FALL?
```

### **Parameter**

Value:    ON(1)    Enables soft stop  
         OFF(0)   Disables soft stop (default)

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
OUTP:SST:FALL ON
```

Response: NR1

## OUTP:SST:TIME

Sets the soft start rise time.

This command is valid when soft start is on (OUTP:SST ON).

### **Command**

```
OUTPut:SStart:TIME[:RISE] <numeric>
```

```
OUTPut:SStart:TIME[:RISE] ?
```

### **Parameter**

Value: 0.1 to 30.0 (The default value is 0.1)

Unit: S

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
OUTP:SST:TIME 1.2
```

Response: NR3

## OUTP:SST:TIME:FALL

Sets the soft stop fall time.

This command is valid when soft stops are on (OUTP:SST:FALL ON).

### **Command**

```
OUTPut:SStart:TIME:FALL <numeric>
```

```
OUTPut:SStart:TIME:FALL?
```

### **Parameter**

Value: 0.1 to 3.0 (The default value is 0.1)

Unit: S

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
OUTP:SST:TIME:FALL 1.2
```

Response: NR3

# PROGram Command

The sequence function is mapped to the PROGram trigger subsystem.

->Tutorial "[Sequence Operation \(PROGram\)](#)"(p. 329)(p. 329)

## PROG:CLE

Sets all the steps in the sequence to their default values.

It takes about 3 seconds for the steps to return to their default values.

### **Command**

PROGram:CLEar

## PROG:EDIT

Collectively sets a sequence step (frequency, voltage, signal change, step time, waveform bank, status output, trigger I/O, output).

### Command

```
PROG:EDIT <step_NRf>,<freq-ramp_boolean>,<freq_numeric>,<acv-ramp_boolean>,<acv_numeric>,<dcv-ramp_boolean>,<dcv_numeric>,<time_numeric>,<bank_NRf>,<status-out_boolean>,<trig-out_boolean>,<trig-in_boolean>,<output_boolean>
```

```
PROG:EDIT? <step_NRf>
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <freq-ramp\_boolean>

Value: ON(1) Enables the ramped frequency signal change

OFF(0) Disables the ramped frequency signal change (default)

Parameter <freq\_numeric>

Value: 1 to 5000 Frequency (The default value is 50)

Unit: HZ

Parameter <acv-ramp\_boolean>

Value: ON(1) Enables the ramped AC voltage signal change

OFF(0) Disables the ramped AC voltage signal change (default)

Parameter <acv\_numeric>

Value: 0 to 315.0 AC voltage (The default value is 0)

Unit: V

Parameter <dcv-ramp\_boolean>

Value: ON(1) Enables the ramped DC voltage signal change

OFF(0) Disables the ramped DC voltage signal change (default)

Parameter <dcv\_numeric>

Value: 0 to  $\pm 445.0$  DC voltage (The default value is 0)

Unit: V

Parameter <time\_numeric>

Value: 0.0001 s to 1000h Step time (The default value is 0.01 s)

Unit: S

Parameter <bank\_NRf>

Value: 0 to 256 Number of the waveform bank to use (The default value is 0)

Parameter <status-out\_boolean>

Value: ON(1) Enables status output  
OFF(0) Disables status output (default)

Parameter <trig-out\_boolean>

Value: ON(1) Enables trigger output  
OFF(0) Disables trigger output (default)

Parameter <trig-in\_boolean>

Value: ON(1) Enables trigger input  
OFF(0) Disables trigger input (default)

Parameter <output\_boolean>

Value: ON(1) Output on (default)  
OFF(0) Output off

#### Example

```
PROG:EDIT 1,OFF,60HZ,OFF,100V,OFF,0V,10S,0,OFF,OFF,OFF,ON
```

#### Response

Returns the settings of the specified step in the following order in response to PROG:EDIT? <step\_NRf>.

Frequency signal change <NR1>, frequency <NR3>, AC voltage signal change <NR1>, AC voltage<sup>\*2</sup> <NR3>, DC voltage signal change <NR1>, DC voltage<sup>\*3</sup> <NR3>, step execution time <NR3>, waveform bank number <NR1>, status output <NR1>, trigger output <NR1>, trigger input <NR1>, output on/off <NR1>

\*2. For single-phase three-wire output and three-phase output, the U phase AC voltage is returned when the unbalanced voltage is set.

\*3. For single-phase three-wire output and three-phase output, the U phase DC voltage is returned when the unbalanced voltage is set.

## PROG:EDIT:FUNC:BANK

Sets the waveform bank number of the sequence step.

### Command

```
PROG:EDIT:FUNCTION[:SHAPE]:BANK[:INDEX] <step_NRf>,<bank_NRf>[,@  
chanlist]
```

```
PROG:EDIT:FUNCTION[:SHAPE]:BANK[:INDEX]? <step_NRf>[,@chanlist]
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <bank\_NRf>

Value: 0 to 256 Number of the waveform bank to use (The default value is 0)

### Example

```
PROG:EDIT:FUNC:BANK 1,256
```

### Response

Returns the waveform bank number of the designated step in NR1 format in response to PROG:EDIT:FUNC:BANK? <step\_NRf>.



## PROG:EDIT:JUMP

Configures the jump settings of a sequence step.

### Command

```
PROGrama:EDIT:JUMP <step_NRf>,<jump-enable_boolean>,<jump-step_
NRf>,<jump-count_NRf>
```

```
PROGrama:EDIT:JUMP? <step_NRf>
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <jump-enable\_boolean>

Value: ON(1) Execution will jump to the specified step.

OFF(0) Execution will proceed to the subsequent step. (default)

Parameter <jump-step\_NRf>

Value: 0 to 599 Jump destination step number (The default value is 0)

Parameter <jump-count\_NRf>

Value: 1 to 99998 Jump repetition count (The default value is 1.)

99999 Repeated indefinitely

### Example

```
PROG:EDIT:JUMP 5,ON,1,10
```

### Response

Returns the jump settings of the specified step in the following order.

Whether a jump will be performed <NR1>,,jump destination step number  
<NR1>,,number of jump repetitions <NR1>

## PROG:EDIT:IMP

Sets whether to set the output impedance in the sequence step.

Use PROG:EDIT:IMP:REAC/PROG:EDIT:IMP:REAL to set the output impedance.

### **Command**

```
PROG:EDIT:IMPerdance[:STATe] <step_NRf>,<imp_boolean>
```

```
PROG:EDIT:IMPerdance[:STATe]? <step_NRf>
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <imp\_boolean>

Value: ON(1) The output impedance will be set.

OFF(0) The output impedance will not be set (default).

### **Example**

```
PROG:EDIT:IMP 1,ON
```

Response: NR1

## PROG:EDIT:IMP:REAC

Sets the reactance component of the output impedance used in the sequence step.

This command is valid when the output impedance (PROG:EDIT:IMP) of the sequence is to be set.

### Command

```
PROG:EDIT:IMP:REAC:REACTIVE <step_NRf>,<reac_NRf>[,@chanlist]
```

```
PROG:EDIT:IMP:REAC:REACTIVE? <step_NRf>[, (@chanlist)]
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <reac\_NRf>

Value: Reactance component (The default value is 0.)

|             | L range                    |  | H range                     |  |
|-------------|----------------------------|--|-----------------------------|--|
|             | Single-phase               | Single-phase<br>three-wire,<br>Three phase | Single-phase                | Single-phase<br>three-wire,<br>Three phase |
| PCR1000WE   | 80 $\mu$ H to 2000 $\mu$ H | --   | 320 $\mu$ H to 8000 $\mu$ H | --   |
| PCR2000WE   | 40 $\mu$ H to 1000 $\mu$ H | --   | 160 $\mu$ H to 4000 $\mu$ H | --   |
| PCR3000WE2  | 27 $\mu$ H to 667 $\mu$ H  | 80 $\mu$ H to 2000 $\mu$ H                 | 107 $\mu$ H to 2667 $\mu$ H | 320 $\mu$ H to 8000 $\mu$ H                |
| PCR6000WE2  | 13 $\mu$ H to 333 $\mu$ H  | 40 $\mu$ H to 1000 $\mu$ H                 | 53 $\mu$ H to 1333 $\mu$ H  | 160 $\mu$ H to 4000 $\mu$ H                |
| PCR12000WE2 | 7 $\mu$ H to 167 $\mu$ H   | 20 $\mu$ H to 500 $\mu$ H                  | 27 $\mu$ H to 667 $\mu$ H   | 80 $\mu$ H to 2000 $\mu$ H                 |
| PCR18000WE2 | 4 $\mu$ H to 111 $\mu$ H   | 13 $\mu$ H to 333 $\mu$ H                  | 18 $\mu$ H to 444 $\mu$ H   | 53 $\mu$ H to 1333 $\mu$ H                 |
| PCR24000WE2 | 3 $\mu$ H to 83 $\mu$ H    | 10 $\mu$ H to 250 $\mu$ H                  | 13 $\mu$ H to 333 $\mu$ H   | 40 $\mu$ H to 1000 $\mu$ H                 |
| PCR30000WE2 | 3 $\mu$ H to 67 $\mu$ H    | 8 $\mu$ H to 200 $\mu$ H                   | 11 $\mu$ H to 267 $\mu$ H   | 32 $\mu$ H to 800 $\mu$ H                  |
| PCR36000WE2 | 2 $\mu$ H to 56 $\mu$ H    | 7 $\mu$ H to 167 $\mu$ H                   | 9 $\mu$ H to 222 $\mu$ H    | 27 $\mu$ H to 667 $\mu$ H                  |

Unit: H

### Example

```
PROG:EDIT:IMP:REAC 1,1UH
```

Response: NR3

## PROG:EDIT:IMP:REAL

Sets the resistance component of the output impedance used in the sequence step.

This command is valid when the output impedance (PROG:EDIT:IMP) of the sequence is to be set.

### Command

```
PROG:EDIT:IMP:REAL <step_NRf>,<real_NRf>[,@chanlist]
```

```
PROG:EDIT:IMP:REAL? <step_NRf>[,@chanlist]
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <real\_NRf>

Value: Resistance component (The default value is 0.)

|             | L range                       |  | H range                        |  |
|-------------|-------------------------------|--|--------------------------------|--|
|             | Single-phase                  | Single-phase<br>three-wire,<br>Three phase | Single-phase                   | Single-phase<br>three-wire,<br>Three phase |
| PCR1000WE   | 0 $\Omega$ to 2000 m $\Omega$ | --   | 0 $\Omega$ to 8000 m $\Omega$  | --   |
| PCR2000WE   | 0 $\Omega$ to 1000 m $\Omega$ | --   | 0 $\Omega$ to 4000 m $\Omega$  | --   |
| PCR3000WE2  | 0 $\Omega$ to 667 m $\Omega$  | 0 $\Omega$ to 2000 m $\Omega$              | 0 $\Omega$ to 2 667 m $\Omega$ | 0 $\Omega$ to 8000 m $\Omega$              |
| PCR6000WE2  | 0 $\Omega$ to 333 m $\Omega$  | 0 $\Omega$ to 1000 m $\Omega$              | 0 $\Omega$ to 1333 m $\Omega$  | 0 $\Omega$ to 4000 m $\Omega$              |
| PCR12000WE2 | 0 $\Omega$ to 167 m $\Omega$  | 0 $\Omega$ to 500 m $\Omega$               | 0 $\Omega$ to 667 m $\Omega$   | 0 $\Omega$ to 2000 m $\Omega$              |
| PCR18000WE2 | 0 $\Omega$ to 111 m $\Omega$  | 0 $\Omega$ to 333 m $\Omega$               | 0 $\Omega$ to 444 m $\Omega$   | 0 $\Omega$ to 1333 m $\Omega$              |
| PCR24000WE2 | 0 $\Omega$ to 83 m $\Omega$   | 0 $\Omega$ to 250 m $\Omega$               | 0 $\Omega$ to 333 m $\Omega$   | 0 $\Omega$ to 1000 m $\Omega$              |
| PCR30000WE2 | 0 $\Omega$ to 67 m $\Omega$   | 0 $\Omega$ to 200 m $\Omega$               | 0 $\Omega$ to 267 m $\Omega$   | 0 $\Omega$ to 800 m $\Omega$               |
| PCR36000WE2 | 0 $\Omega$ to 56 m $\Omega$   | 0 $\Omega$ to 167 m $\Omega$               | 0 $\Omega$ to 222 m $\Omega$   | 0 $\Omega$ to 667 m $\Omega$               |

Unit: OHM

### Example

```
PROG:EDIT:IMP:REAL 1,50MOHM
```

Response: NR3

**PROG:EDIT:PHAS:RAMP**

Sets the phase signal change of a sequence step.

Set the U phase angle with the U phase offset (PROG:EDIT:PHAS:UOFF).

Set the V phase angle with the U-V phase difference (PROG:EDIT:PHAS:UV).

Set the W phase angle with the U-W phase difference (PROG:EDIT:PHAS:UW).

This command is valid for three-phase output.

**Command**

```
PROG:EDIT:PHAS:RAMP <step_NRf>,<u-phase_character>[,<v-phase_
character>[,<w-phase_character>]]
```

```
PROG:EDIT:PHAS:RAMP? <step_NRf>
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <u-phase\_character> U phase signal change, <v-phase\_character> V phase signal change, <w-phase\_character> W phase signal change

|        |      |                    |
|--------|------|--------------------|
| Value: | OFF  | Ramp off (default) |
|        | LEAD | Ramp on, leading   |
|        | LAG  | Ramp on, lagging   |

**Example**

```
PROG:EDIT:PHAS:RAMP 1,LAG,OFF,OFF
```

```
PROG:EDIT:PHAS:RAMP? 1
```

**Response**

Returns the phase signal change settings of the specified step in the following order.

U phase signal change<characters>, V phase signal change<characters>, W phase signal change<characters>

## PROG:EDIT:PHAS:STAR

Sets the starting phase angle and whether to suddenly change the phase of the sequence step.

### Command

```
PROG:EDIT:PHAS:STARt <step_NRf>,<enable_boolean>[,<phase_
NR3>[,<phase_change_boolean>]]
```

```
PROG:EDIT:PHAS:STARt? <step_NRf>
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <enable\_boolean>

Value: ON(1) Enables phase control  
OFF(0) Disables phase control (default)

Parameter <phase\_NR3>

Value: 0.0 to 360.0 Starting phase angle (The default value is 0)  
0 and 360 are the same.

Unit: DEG

Parameter <phase\_change\_boolean>

Value: ON(1) Enables sudden phase change  
OFF(0) Disables sudden phase change (default)

### Example

```
PROG:EDIT:PHAS:STAR 1,ON,90,OFF
```

```
PROG:EDIT:PHAS:STAR? 1
```

### Response

Returns the starting phase angle and the sudden phase change setting of the specified step in the following order.

Phase control enabled/disabled<NR1>, starting phase angle<NR3>, sudden phase change setting<NR1>

**PROG:EDIT:PHAS:STOP**

Sets the ending phase angle of the sequence step.

**Command**

```
PROG:EDIT:PHAS:STOP <step_Nrf>,<enable_boolean>,[<phase_NR3>]
```

```
PROG:EDIT:PHAS:STOP? <step_Nrf>
```

Parameter <step\_Nrf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <enable\_boolean>

Value: ON(1) Enables phase control  
OFF(0) Disables phase control (default)

Parameter <phase\_NR3>

Value: 0.0 to 360.0 Ending phase angle (The default value is 0)  
0 and 360 are the same.

Unit: DEG

**Example**

```
PROG:EDIT:PHAS:STOP 1,ON,90
```

```
PROG:EDIT:PHAS:STOP? 1
```

**Response**

Returns the ending phase angle of the specified step in the following order.

Phase control enabled/disabled<NR1>, ending phase angle<NR3>

## PROG:EDIT:PHAS:UOFF

Sets the U phase offset phase angle of a sequence step.

Use PROG:EDIT:PHAS:RAMP to set the phase signal change.

This command is valid for three-phase output.

### **Command**

```
PROG:EDIT:PHAS:UOFFset <step_NRf>,<enable_boolean>,<phase_NR3>
```

```
PROG:EDIT:PHAS:UOFFset? <step_NRf>
```

Parameter <step\_NRf>

Value: 1 to 599 Number of the step that you want to configure

Parameter <enable\_boolean>

Value: ON(1) Enables U phase offset  
OFF(0) Disables the U phase offset (default)

Parameter <phase\_NR3>

Value: -360.00 to 360.00 U phase offset value (The default value is 0)

Unit: DEG

### **Example**

```
PROG:EDIT:PHAS:UOFF 1,ON,90
```

### **Response**

Returns the U phase offset value of the specified step in the following order.

U phase offset enabled/disabled<NR1>, U phase offset value<NR3>



**PROG:EDIT:PHAS:UV**

Sets the U-V phase difference of the sequence step.

Use PROG:EDIT:PHAS:RAMP to set the phase signal change.

This command is valid for three-phase output.

**Command**

```
PROG:EDIT:PHAS:UV <step_NRf>,<enable_boolean>,<phase_NR3>
```

```
PROG:EDIT:PHAS:UV? <step_NRf>
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <enable\_boolean>

Value: ON(1) Enables U-V phase difference control

OFF(0) Disables U-V phase difference control (default)

Parameter <phase\_NR3>

Value: 0.00 to 360.00 U-V phase difference (The default value is 0)  
0 and 360 are the same.

Unit: DEG

**Example**

```
PROG:EDIT:PHAS:UV 1,ON,90
```

```
PROG:EDIT:PHAS:UV? 1
```

**Response**

Returns the U-V phase difference of the specified step in the following order.

U-V phase difference control enabled/disabled<NR1>, U-V phase difference<NR3>

## PROG:EDIT:PHAS:UW

Sets the U-W phase difference of the sequence step.

Use PROG:EDIT:PHAS:RAMP to set the phase signal change.

This command is valid for three-phase output.

### Command

```
PROG:EDIT:PHAS:UW <step_NRf>,<enable_boolean>,<phase_NR3>
```

```
PROG:EDIT:PHAS:UW? <step_NRf>
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter <enable\_boolean>

Value: ON(1) Enables U-W phase difference control

OFF(0) Disables U-W phase difference control (default)

Parameter <phase\_NR3>

Value: 0.00 to 360.00 U-W phase difference (The default value is 0)

Unit: DEG

### Example

```
PROG:EDIT:PHAS:UW 1,ON,90
```

```
PROG:EDIT:PHAS:UW? 1
```

### Response

Returns the U-W phase difference of the specified step in the following order.

U-W phase difference control enabled/disabled<NR1>, U-W phase difference<NR3>

## **PROG:EDIT:VOLT**

Sets the unbalanced AC voltage of the sequence step

### **Command**

```
PROGrama:EDIT:VOLTage <step_NRf>,<volt_NRf>[,@chanlist]
```

```
PROGrama:EDIT:VOLTage? <step_NRf>
```

Parameter <step\_NRf>

Value: 0 to 599 Number of the step that you want to configure

Parameter

Value: 0.0 to 315.0 Unbalanced AC voltage (The default value is 0)

Unit: V

Example

```
PROG:EDIT:VOLT 2,50,(@2)
```

```
PROG:EDIT:VOLT? 2
```

Response: NR3

## PROG:EDIT:VOLT:OFFS

Sets the unbalanced DC voltage of the sequence step

### Command

```
PROG:EDIT:VOLTage:OFFSet <step_Nrf>,<volt_Nrf>[,@chanlist]
```

```
PROG:EDIT:VOLTage:OFFSet? <step_Nrf>
```

Parameter <step\_Nrf>

Value: 0 to 599 Number of the step that you want to configure

Parameter

Value: -445.0 to 445.0 Unbalanced DC voltage (The default value is 0)

Unit: V

### Example

```
PROG:EDIT:VOLT:OFFS 1,-50
```

```
PROG:EDIT:VOLT:OFFS? 1
```

Response: NR3

## **PROG:EXEC**

Queries the sequence execution state.

### **Command**

PROGram:EXECuting?

### **Response**

Returns the execution state (STOP, RUN, or PAUSE) <characters>, elapsed step time (the unit is seconds) <NR3>, present repetition number <NR1>, and step number ("-1" if the sequence is not being executed) <NR1> as a comma-separated list.

## PROG:LOOP

Sets the sequence repetition count.

This command is invalid while a sequence is running.

### **Command**

```
PROG:LOOP <NR1>
```

```
PROG:LOOP?
```

### **Parameter**

|        |            |  |
|--------|------------|--|
| Value: | 1 to 99998 | Repeat count (The default value is 1.) |
|        | 99999      | Repeated indefinitely                  |

### **Example**

```
PROG:LOOP 100
```

Response: NR1

## PROG:STAT

Changes the execution state of the sequence.

If the signal source is set to external signal source VOLT:EXT:INP:EXTDC:SIGN:-SOUR EXT), this command is invalid.

### Command

```
PROGrama:STATe <character>
```

### Parameter

|        |          |   |
|--------|----------|---|
| Value: | STOP     | Stops the sequence                          |
|        | RUN      | Executing Sequences                         |
|        | PAUSE    | Pausing a sequence                          |
|        | CONTINUE | Continues the sequence that has been paused |

### Example

```
PROG:STAT PAUS
```

## PROG:STEP:END

Sets the sequence ending step number

### **Command**

```
PROG:STEP:END <NR1>
```

```
PROG:STEP:END?
```

### **Parameter**

Value: 0 to 599

### **Example**

```
PROG:STEP:END 20
```

Response: NR1



**PROG:STEP:STAR**

Sets the sequence starting step number

**Command**

```
PROGram:STEP:STARt <NR1>
```

```
PROGram:STEP:STARt?
```

**Parameter**

Value: 0 to 599

**Example**

```
PROG:STEP:STAR 10
```

Response: NR1

# SENSe Command

## SENS:AVER:COUN

Sets the moving average count for current measurement.

Averaging is not performed when this is set to 1.

### Command

```
SENSe:AVERage:COUNT <NR1>
```

```
SENSe:AVERage:COUNT?
```

### Parameter

Value: 1 to 32 (The default value is 1)

Settings are reset to default values when an \*RST command is sent.

### Example

```
SENS:AVER:COUN 16
```

Response: NR1

**SENS:CURR:HOLD:CLE**

Clears the hold time of the peak current..

**Command**

```
SENSe:CURRent [:PEAK]:HOLD:CLEar
```

## SENS:CURR:HOLD:TIME

Sets the hold time of the peak current.

### Command

```
SENSe:CURRent[:PEAK]:HOLD:TIME {<numeric>|INFinity}
```

```
SENSe:CURRent[:PEAK]:HOLD:TIME?
```

### Parameter

|        |                            |                                    |
|--------|----------------------------|------------------------------------|
| Value: | 1 to 10                    | Hold time (The default value is 1) |
|        | 11 s or higher or INFinity | Infinite                           |

Unit: S

Settings are reset to default values when an \*RST command is sent.

### Example

```
SENS:CURR:HOLD:TIME 2
```

Response: NR3

Returns +9.90000E+37 when the hold time is set to infinity.

## SENS:VOLT:AVER:COUN

Sets the moving average count for voltage measurement.

Averaging is not performed when this is set to 1.

### Command

```
SENSe:VOLTage:AVERage:COUNT <NRf>
```

```
SENSe:VOLTage:AVERage:COUNT?
```

### Parameter

Value: 1 to 32 (The default value is 1)

Settings are reset to default values when an \*RST command is sent.

### Example

```
SENS:VOLTage:AVER:COUN 16
```

Response: NR1

## SENS:VOLT:HOLD:CLE

Clears the hold time of the peak voltage.

### **Command**

```
SENSe:VOLTage[:PEAK]:HOLD:CLEar
```

**SENS:VOLT:HOLD:TIME**

Sets the hold time of the peak voltage.

**Command**

```
SENSe:VOLTage[:PEAK]:HOLD:TIME {<numeric>|INFINITY}
```

```
SENSe:VOLTage[:PEAK]:HOLD:TIME?
```

**Parameter**

|        |                            |                                    |
|--------|----------------------------|------------------------------------|
| Value: | 1 to 10                    | Hold time (The default value is 1) |
|        | 11 s or higher or INFINITY | Infinite                           |

|       |   |
|-------|---|
| Unit: | S |
|-------|---|

Settings are reset to default values when an \*RST command is sent.

**Example**

```
SENS:VOLT:HOLD:TIME 2
```

**Response: NR3**

Returns +9.90000E+37 when the hold time is set to infinity.

# SIMulation Command

Power line abnormality simulation is mapped to the SIMulation trigger subsystem.

Power line abnormality simulation is valid for AC output.

->Tutorial "[Power line abnormality simulations \(SIMulation\)](#)"(p. 326)(p. 326)

## SIM:EXEC

Queries the execution status of power line abnormality simulations.

### Command

```
SIMulation:EXECuting?
```

### Response

Returns the execution state (STOP or RUN) <characters> and the number of repetitions <NR1> as a comma-separated list.



**SIM:POL**

Sets the voltage regulation polarity of power line abnormality simulations.

**Command**

```
SIMulation:POLarity <character>
```

```
SIMulation:POLarity?
```

**Parameter**

|        |          |                                      |
|--------|----------|--------------------------------------|
| Value: | NORMAL   | Positive polarity is used (default). |
|        | INVERTed | Negative polarity is used.           |

**Example**

```
SIM:POL INV
```

Response: Characters

## SIM:REP:COUN

Sets the number of repetitions of power line abnormality simulations.

### Command

```
SIMulation:REPeat:COUNT <NRf>
```

```
SIMulation:REPeat:COUNT?
```

### Parameter

|        |           |   |
|--------|-----------|---|
| Value: | 1 to 9998 | Repeat count (The default value is 9999.) |
|        | 9999      | Repeated indefinitely                     |

### Example

```
SIM:REP:COUN 100
```

Response: NR1

## **SIM:STAT**

Executes or stops a power line abnormality simulation.

This command is valid for AC output.

This command is invalid when the output is off (OUTP OFF).

If the signal source is set to external signal source (FUNC:SOUR EXT), this command is invalid.

### **Command**

```
SIMulation:STATe <character>
```

#### **Parameter**

|        |      |  |
|--------|------|--|
| Value: | STOP | Stops the power line abnormality simulation    |
|        | RUN  | Executes the power line abnormality simulation |

#### **Example**

```
SIM:STAT STOP
```

## SIM:T1:PHAS

Sets the voltage regulation starting phase of power line abnormality simulations.

This command is valid when voltage regulations are configured to be set by phase (SIM:T1:PHAS:STAT ON).

### **Command**

```
SIMulation:T1:PHASe[:LEVel] <numeric>
```

```
SIMulation:T1:PHASe[:LEVel]?
```

### **Parameter**

Value: 0.0 to 359.9 Voltage regulation starting phase (The default value is 0)

Unit: DEG

### **Example**

```
SIM:T1:PHAS 89.5
```

Response: NR3

## SIM:T1:PHAS:STAT

Selects whether the voltage regulation start of power line abnormality simulations will be set in terms of time or in terms of phase.

When you have selected time, use SIM:T1:TIME to set the voltage regulation start time.

When you have selected phase, use SIM:T1:PHAS to set the voltage regulation starting phase.

### Command

```
SIMulation:T1:PHASe:STATe <boolean>
```

```
SIMulation:T1:PHASe:STATe?
```

### Parameter

Value:    ON(1)    The value is set in terms of phase.  
         OFF(0)    The value is set in terms of time (default).

### Example

```
SIM:T1:PHAS:STAT 0
```

Response: NR1

## SIM:T1:TIME

Sets the voltage regulation starting time of power line abnormality simulations.

This command is valid when voltage regulations are configured to be set by time (SIM:T1:PHAS:STAT OFF).

### **Command**

```
SIMulation:T1:TIME[:LEVel] <numeric>
```

```
SIMulation:T1:TIME[:LEVel]?
```

### **Parameter**

Value: 0.0000 to 0.9999 Voltage regulation starting time (The default value is 0.1)

Unit: S

### **Example**

```
SIM:T1:TIME 4.5MS
```

Response: NR3

## **SIM:T2:TIME**

Sets slope time 1 of power line abnormality simulations.

### **Command**

```
SIMulation:T2:TIME[:LEVel] <numeric>
```

```
SIMulation:T2:TIME[:LEVel]?
```

### **Parameter**

Value: 0.000 to 99.990 Slope time 1 (The default value is 0)

Unit: S

### **Example**

```
SIM:T2:TIME 45MS
```

Response: NR3

## SIM:T3:TIME

Sets the voltage regulation time of power line abnormality simulations.

### Command

```
SIMulation:T3:TIME[:LEVel] <numeric>
```

```
SIMulation:T3:TIME[:LEVel]?
```

### Parameter

Value: 0.0000 to 9.9990 Voltage regulation time (The default value is 0.1)

Unit: S

### Example

```
SIM:T3:TIME 4.5MS
```

Response: NR3



## **SIM:T3:VOLT**

Sets the regulated voltage of power line abnormality simulations.

### **Command**

```
SIMulation:T3:VOLTage[:LEVel] <numeric>
```

```
SIMulation:T3:VOLTage[:LEVel]?
```

### **Parameter**

Value: 0.0 to 315.0 Regulated voltage (The default value is 0)

Unit: V

### **Example**

```
SIM:T3:VOLT 120V
```

Response: NR3

## SIM:T4:TIME

Sets slope time 2 of power line abnormality simulations.

### Command

```
SIMulation:T4:TIME[:LEVel] <numeric>
```

```
SIMulation:T4:TIME[:LEVel]?
```

### Parameter

Value: 0.000 to 99.990 Slope time 2 (The default value is 0)

Unit: S

### Example

```
SIM:T4:TIME 45MS
```

Response: NR3

## **SIM:T5:CYCL**

Sets the number of return cycles of power line abnormality simulations.

This command is valid when the period that the PCR-WE remains in the returned state is configured to be set in terms of cycles (SIM:T5:CYCL:STAT ON).

### **Command**

```
SIMulation:T5:CYCLe[:LEVel] <numeric>
```

```
SIMulation:T5:CYCLe[:LEVel]?
```

### **Parameter**

Value: 0 to 999900 (The default value is 0)

### **Example**

```
SIM:T5:CYCL 100
```

Response: NR1

## SIM:T5:CYCL:STAT

Selects whether the period that the PCR-WE remains in the returned state is configured in terms of time or in terms of cycles for power line abnormality simulations.

When you have selected time, use SIM:T5:TIME to set the return time.

When you have selected cycles, use SIM:T5:CYCL to set the number of return cycles.

### **Command**

```
SIMulation:T5:CYCLe:STATe <boolean>
```

```
SIMulation:T5:CYCLe:STATe?
```

### **Parameter**

Value:    ON(1)    The value is set in terms of cycles.  
         OFF(0)   The value is set in terms of time (default).

### **Example**

```
SIM:T5:CYCL:STAT 1
```

Response: NR1

## **SIM:T5:TIME**

Sets the return time of power line abnormality simulations.

This command is valid when the period that the PCR-WE remains in the returned state is configured to be set in terms of time (SIM:T5:CYCL:STAT OFF).

### **Command**

```
SIMulation:T5:TIME[:LEVel] <numeric>
```

```
SIMulation:T5:TIME[:LEVel]?
```

### **Parameter**

Value: 0.000 to 99.990 Return time (The default value is 0.1)

Unit: S

### **Example**

```
SIM:T5:TIME 45MS
```

Response: NR3

# [SOURce:]CURRent Command

## CURR

Sets the upper limit of the output current.

### Command

```
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <numeric>[, (@chan-  
list)]
```

```
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? [(@chanlist)]
```

### Parameter

Value: 10 % of the maximum current [A] to 110 % of the maximum current [A]  
(The default value is MAXimum)

Unit: A

Settings are reset to default values when an \*RST command is sent.

### Example

```
CURR 25
```

Response: NR3

## **CURR:PEAK**

Sets the output current peak limit of the positive electric potential

### **Command**

```
[SOURce:]CURRent:PEAK[:UPPer][:IMMediate] <numeric>[, (@chanlist)]
```

```
[SOURce:]CURRent:PEAK[:UPPer][:IMMediate]? [(@chanlist)]
```

### **Parameter**

Value: 10 % of the maximum current [A] to 420 % of the maximum current [A]  
(The default value is MAXimum)

Unit: A

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
CURR:PEAK 120
```

Response: NR3

## CURR:PEAK:LOW

Sets the output current peak limit of the negative electric potential

### Command

```
[SOURce:]CURRent:PEAK:LOWer[:IMMediate] <numeric>[, (@chanlist)]
```

```
[SOURce:]CURRent:PEAK:LOWer[:IMMediate]? [ (@chanlist)]
```

### Parameter

Value: 10 % of the maximum current [A] to 420 % of the maximum current [A]  
(The default value is MAXimum)

Unit: A

Settings are reset to default values when an \*RST command is sent.

### Example

```
CURR:PEAK:LOW -120
```

Response: NR3



## CURR:PROT:STAT

Sets how the PCR-WE acts when the current limit is exceeded.

When you have specified the ON parameter (to select TRIP), use the CURR:PROT:TRIP:DEL command to set the time that must elapse after a limit has been exceeded before the protection functions are tripped.

### Command

```
[SOURce:]CURRent:PROTection:STATe <boolean>
```

```
[SOURce:]CURRent:PROTection:STATe?
```

### Parameter

Value:    ON(1)    TRIP (when an overload occurs for longer than the specified time, the output is turned off, and an alarm is generated) (default)

         OFF(0)    OFF(0) CC (decreases the output so that the current limit is not exceeded when an overloading occurs)

Settings are reset to default values when an \*RST command is sent.

### Example

```
CURR:PROT:STAT 1
```

Response: NR1

## CURR:PROT:TRIP:DEL

Sets the time that must elapse before the output is turned off when the current limit is exceeded

The action that is performed when the limit is exceeded is valid when TRIP (CURR:PROT:STAT ON) has been selected.

### **Command**

```
[SOURce:]CURRent:PROTection:TRIP:DElay <numeric>
```

```
[SOURce:]CURRent:PROTection:TRIP:DElay?
```

### **Parameter**

Value: 0.0 to 10.0 Time until the output is turned off (The default value is 10)

Unit: S

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
CURR:PROT:TRIP:DEL 3.5
```

Response: NR3

# [SOURce:]FREQuency Command

## FREQ

Sets the frequency.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

### Command

```
[SOURce:]FREQuency[:IMMediate] <numeric>
```

```
[SOURce:]FREQuency[:IMMediate]?
```

### Parameter

Value: 1.00 to 5000      Frequency (The default value is 50.0)  
1.00 to 500.0 on the 500 Hz limit model

Unit:      HZ

Settings are reset to default values when an \*RST command is sent.

### Example

```
FREQ 400
```

Response: NR3

## FREQ:LIM:LOW

Sets the lower frequency limit.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

The following relationship must be met:  $\text{FREQ:LIM:LOW} \leq \text{FREQ:FREQ:TRIG} \leq \text{FREQ:LIM:UPP}$

### Command

[SOURce:]FREQuency:LIMit:LOWer <numeric>

[SOURce:]FREQuency:LIMit:LOWer?

### Parameter

Value: 1.00 to 5000      Frequency (The default value is 1.00)  
1.00 to 500.0 on the 500 Hz limit model

Unit:      HZ

Settings are reset to default values when an \*RST command is sent.

### Example

FREQ:LIM:LOW 40

Response: NR3

**FREQ:LIM:UPP**

Sets the upper frequency limit.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

The following relationship must be met:  $\text{FREQ:LIM:LOW} \leq \text{FREQ}| \text{FREQ:TRIG} \leq \text{FREQ:LIM:UPP}$

**Command**

```
[SOURce:]FREQuency:LIMit:UPPer <numeric>
```

```
[SOURce:]FREQuency:LIMit:UPPer?
```

**Parameter**

Value: 1.00 to 5000      Frequency (The default value is 5000)  
    1.00 to 500.0 on the 500 Hz limit model (The default value is 500)

Unit:      HZ

Settings are reset to default values when an \*RST command is sent.

**Example**

```
FREQ:LIM:UPP 70
```

Response: NR3

## FREQ:SYNC

Sets the sync function.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

Use FREQ:SYNC:PHASE:DEL to set the synchronization delay phase angle.

### Command

```
[SOURce:]FREQuency:SYNChronize[:STATe] <boolean>
```

```
[SOURce:]FREQuency:SYNChronous[:STATe]?
```

### Parameter

Value:    ON(1)    Using the Synchronization Function  
         OFF(0)   The synchronization function is disabled (default).

Settings are reset to default values when an \*RST command is sent.

### Example

```
FREQ:SYNC ON
```

Response: NR1

## FREQ:SYNC:MODE

Sets the input sync signal of the sync function.

### Command

```
[SOURce:]FREQuency:SYNChronous:MODE <character>
```

```
[SOURce:]FREQuency:SYNChronous:MODE?
```

### Parameter

|        |          |  |
|--------|----------|--|
| Value: | LINE     | Synchronizes to the input power supply (default) |
|        | EXternal | Synchronizes to an external sync input signal    |

Settings are reset to default values when an \*RST command is sent.

### Example

```
FREQ:SYNC:MODE LINE
```

Response: Characters

## FREQ:SYNC:PHAS:DEL

Sets the synchronization delay phase angle of the synchronization function.

This command is valid when the sync function is on (FREQ:SYNC ON).

### Command

```
[SOURce:]FREQuency:SYNChronous:PHASe:DELay <numeric>
```

```
[SOURce:]FREQuency:SYNChronous:PHASe:DELay?
```

### Parameter

Value: 0 to 360.0 Synchronization delay phase angle (The default value is 0)  
0 and 360 are the same.

Unit: DEG

Settings are reset to default values when an \*RST command is sent.

### Example

```
FREQ:SYNC:PHAS:DEL 70.5
```

Response: NR3



## **FREQ:TRIG**

Sets the frequency to change to when INIT/INIT:TRAN or a software trigger is sent.

### **Command**

```
[SOURce:]FREQuency:TRIGgered <numeric>
```

```
[SOURce:]FREQuency:TRIGgered?
```

### **Parameter**

Value: 1.00 to 5000      Frequency (The default value is 50.0)  
1.00 to 500.0 on the 500 Hz limit model

Unit:      HZ

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
FREQ:TRIG 400
```

Response: NR3

# [SOURce:]FUNCTION Command

## FUNC:BANK

Specifies the number of the waveform bank that you want to execute.

This command is invalid when the sync function is enabled (FREQ:SYNC ON).

This command is invalid if the compensation function is set to the soft sensing function or the regulation adjustment function (VOLT:COMP:MODE SOFT|RADJ).

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

### Command

```
[SOURce:]FUNCTION[:SHAPE]:BANK[:INDEX] <NR1>[, (@chanlist)]
```

```
[SOURce:]FUNCTION[:SHAPE]:BANK[:INDEX]? [ (@chanlist)]
```

### Parameter

Value 0 to 256 (The default value is 0)

Settings are reset to default values when an \*RST command is sent.

### Example

```
FUNC:BANK 5
```

Response: NR1

# [SOURce:]VOLTage Command

## VOLT

Set the AC voltage.

This command is invalid when the combined value with the DC voltage is outside the allowable range (L range: -222.5 V to 222.5 V, H range: -445.0 V to 445.0 V).

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

This command is invalid when an external analog signal is used to control the voltage or frequency (VOLT:EXT:FUNC:MODE VPR).

### Command

```
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <numeric>[, (@chan-  
list)]
```

```
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [(@chanlist)]
```

### Parameter

Value:    0 to 157.5    AC voltage in the L range  
          0 to 315.0    AC voltage in the H range  
                          (The default value is 0.)

Unit:     V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT 120
```

Response: NR3

## VOLT:COMP:MODE

Sets the compensation (voltage compensation).

### Command

```
[SOURce:]VOLTage:COMPensate:MODE <character>
```

```
[SOURce:]VOLTage:COMPensate:MODE?
```

### Parameter

|        |          |                                     |
|--------|----------|-------------------------------------|
| Value: | DISabled | Disables the compensation (default) |
|        | HARD     | Enables hard sensing                |
|        | SOFT     | Enables soft sensing                |
|        | RADJust  | Enables regulation adjustment       |

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:COMP:MODE HARD
```

Response: Characters

## VOLT:COMP:RADJ

Sets the voltage to correct with regulation adjustment.

This is valid when compensation is set to regulation adjustment (VOLT:COMP:MODE RADJ) and the output is on (OUTP ON).

### Command

```
[SOURce:]VOLTage:COMPensate:RADJust[:RATio] <numeric>
```

```
[SOURce:]VOLTage:COMPensate:RADJust[:RATio]?
```

### Parameter

Value: 0 to 10      Regulation adjustment ratio (The default value is 0)

Unit: PCT

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:COMP:RADJ 4.0
```

Response: NR3

## VOLT:COMP:SOFT:CONT

Set the soft sensing control target.

This command is invalid when the output is on (OUTP ON).

### Command

```
[SOURce:]VOLTage:COMPensate:SOFT:CONTrol[:STATus] <character>
```

```
[SOURce:]VOLTage:COMPensate:SOFT:CONTrol[:STATus]?
```

### Parameter

|        |    |   |
|--------|----|---|
| Value: | AC | Corrects the AC voltage (default)<br>This is invalid if the DC voltage is not set to 0 V. |
|        | DC | Corrects the DC voltage<br>This is invalid if the AC voltage is not set to 0 V.           |

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:COMP:SOFT:CONT AC
```

Response: Characters

**VOLT:COMP:SOFT:TERM**

Set the sensing point.

This command is invalid when the output is on (OUTP ON).

**Command**

```
[SOURce:]VOLTage:COMPensate:SOFT:TERMinal <character>
```

```
[SOURce:]VOLTage:COMPensate:SOFT:TERMinal?
```

**Parameter**

|        |         |  |
|--------|---------|--|
| Value: | OUTPut  | Output terminal (output voltage correction function)                       |
|        | SENSing | Sensing terminal (load wire voltage drop correction function)<br>(default) |

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:COMP:SOFT:TERM OUTP
```

Response: Characters

## VOLT:EXT:INP:EXTDC:ADJ:GAIN

Sets the gain for when the input waveform is amplified using an external analog signal.

This command is valid when the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT).

### Command

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:ADJust:GAIN <NRf_ch>,<NRf_gain>
```

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:ADJust:GAIN? <NRf_ch>
```

Parameter <NRf\_ch> Channel to be configured

|        |   |      |
|--------|---|------|
| Value: | 0 | Ch.A |
|        | 1 | Ch.B |
|        | 2 | Ch.C |

Parameter <NRf\_gain> Gain

Value: 5 to 220 (The default value is 100)

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:EXT:INP:EXTDC:ADJ:GAIN 0,10
```

Response: NR3



**VOLT:EXT:INP:EXTDC:ADJ:OFFS**

Sets the offset for when the input waveform is amplified using an external analog signal.

This command is valid when the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT).

**Command**

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:ADJust:OFFSet <NRf_ch>,<NRf_offset>
```

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:ADJust:OFFSet? <NRf_ch>
```

Parameter <NRf\_ch> Channel to be configured

|       |   |      |
|-------|---|------|
| 設定値 : | 0 | Ch.A |
|       | 1 | Ch.B |
|       | 2 | Ch.C |

Parameter <NRf\_offset> Offset

Value: -200 to 200(The default value is 0)

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:EXT:INP:EXTDC:ADJ:OFFS 0,10
```

Response: NR3

## VOLT:EXT:INP:EXTDC:APER

Sets the measurement time for when the input waveform is amplified using an external analog signal.

This command is valid when the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT).

### Command

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:APERture <numeric>
```

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:APERture?
```

### Parameter

Value: 0.1 to 1.0 Measurement time (The default value is 0.1)

Unit: S

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:EXT:INP:EXTDC:APER 0.5
```

Response: NR3

**VOLT:EXT:INP:EXTDC:SIGN:POL**

Sets the signal polarity of each channel for when the input waveform is amplified using an external analog signal.

**Command**

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:SIGNal:POLarity <NR1>,<character>
```

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDc:SIGNal:POLarity? <NR1>
```

Parameter <NR1>

|        |   |           |
|--------|---|-----------|
| Value: | 0 | Channel A |
|        | 1 | Channel B |
|        | 2 | Channel C |

Parameter <character>

|        |          |   |
|--------|----------|---|
| Value: | NORMAL   | Outputs a signal whose polarity is the same as the input signal (default) |
|        | INVERTed | Outputs a signal whose polarity is opposite to that of the input signal.  |

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:EXT:INP:EXTDC:SIGN:POL 0,INV
```

```
VOLT:EXT:INP:EXTDC:SIGN:POL? 0
```

Response: Characters

## VOLT:EXT:INP:EXTDC:SIGN:SOUR

Sets the signal source for when the input waveform is amplified using an external analog signal.

### Command

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:SIGNal:SOURce <character>
```

```
[SOURce:]VOLTage:EXTernal:INPut:EXTDC:SIGNal:SOURce?
```

### Parameter

|        |          |                                     |
|--------|----------|-------------------------------------|
| Value: | EXTernal | External signal (default)           |
|        | INT_EXT  | Internal signal and external signal |

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:EXT:INP:EXTDC:SIGN:SOUR INT_EXT
```

Response: Characters

**VOLT:EXT:INP:FUNC:MODE**

Selects the parameter to control with the external analog signal.

This command is invalid when the output is on (OUTP ON).

**Command**

```
[SOURce:]VOLTage:EXTernal:INPut:FUNCtion:MODE <character>
```

```
[SOURce:]VOLTage:EXTernal:INPut:FUNCtion:MODE?
```

**Parameter**

|        |          |  |
|--------|----------|--|
| Value: | OFF      | Disables the use of the external analog signal (default) |
|        | EXTDC    | Amplifies the input waveform                             |
|        | VPRogram | Varies the output voltage or frequency                   |

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:EXT:INP:FUNC:MODE EXTDC
```

Response: Characters

## VOLT:EXT:INP:VPR:ADJ:GAIN

Sets the gain for when varying the voltage or frequency with the external analog signal.

### Command

```
[SOURce:]VOLTage:EXTernal:INPut:VPRogram:ADJust:GAIN <NRf_ch>,<NRf_gain>
```

```
[SOURce:]VOLTage:EXTernal:INPut:VPRogram:ADJust:GAIN? <NRf_ch>
```

Parameter <NRf\_ch> Channel to be configured

|       |   |      |
|-------|---|------|
| 設定値 : | 0 | Ch.A |
|       | 1 | Ch.B |
|       | 2 | Ch.C |

Parameter <NRf\_gain> Gain

Value: 5 to 50 (The default value is 10)

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:EXT:INP:VPR:ADJ:GAIN 0,10
```

Response: NR3

**VOLT:EXT:INP:VPR:ADJ:OFFS**

Sets the offset for when varying the voltage or frequency with the external analog signal.

**Command**

```
[SOURce:]VOLTage:EXTernal:INPut:VPRogram:ADJust:OFFSet <NRf_
ch>,<NRf_offset>
```

```
[SOURce:]VOLTage:EXTernal:INPut:VPRogram:ADJust:OFFSet? <NRf_ch>
```

Parameter <NRf\_ch> Channel to be configured

|       |   |      |
|-------|---|------|
| 設定値 : | 0 | Ch.A |
|       | 1 | Ch.B |
|       | 2 | Ch.C |

Parameter <NRf\_offset> Offset

Value: -200 to 200 (The default value is 0)

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:EXT:INP:VPR:ADJ:OFFS 0,10
```

Response: NR3

## VOLT:EXT:INP:VPR:MAP

Sets the channel configuration for when varying the voltage or frequency with the external analog signal.

This command is invalid for single-phase output and single-phase three-wire output.

### Command

```
[SOURce:]VOLTage:EXTernal:INPut:VPRogram:MAP <character>
```

```
[SOURce:]VOLTage:EXTernal:INPut:VPRogram:MAP?
```

### Parameter

Value:    ALL            ChA: AC voltage, ChB: DC voltage, ChC: Frequency (default)  
         ACVoltage    ChA: U phase AC voltage, ChB: V phase AC voltage,  
                        ChC: W phase AC voltage  
         DCVoltage    ChA: U phase DC voltage, ChB: V phase DC voltage,  
                        ChC: W phase DC voltage

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:EXT:INP:VPR:MAP DCV
```

Response: Characters



**VOLT:EXT:INP:VPR:STAT**

Turns on or off the output of each channel for when varying the voltage or frequency with the external analog signal.

**Command**

```
[SOURce:]VOLTage:EXTernal:INPut:VPRogram:STATe <NRf>,<boolean>
```

```
[SOURce:]VOLTage:EXTernal:INPut:VPRogram:STATe? <NRf>
```

Parameter <NRf>

|        |   |           |
|--------|---|-----------|
| Value: | 0 | Channel A |
|        | 1 | Channel B |
|        | 2 | Channel C |

Parameter <boolean>

|        |        |                      |
|--------|--------|----------------------|
| Value: | ON(1)  | Output on            |
|        | OFF(0) | Output off (default) |

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:EXT:INP:VPR:STAT 0,ON
```

Response: NR1

## VOLT:LIM:LOW

Sets the lower AC voltage limit.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

The following relationship must be met:  $VOLT:LIM:LOW \leq VOLT|VOLT:TRIG \leq VOLT:LIM:UPP$

### Command

```
[SOURce:]VOLTage[:LEVel]LIMit:LOWer <numeric>
```

```
[SOURce:]VOLTage[:LEVel]LIMit:LOWer?
```

### Parameter

Value: 0 to 315.0 Lower AC voltage limit (The default value is 0)

Unit: V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:LIM:LOW 119
```

Response: NR3

**VOLT:LIM:UPP**

Sets the upper AC voltage limit.

This command is valid for AC output.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

The following relationship must be met:  $VOLT:LIM:LOW \leq VOLT|VOLT:TRIG \leq VOLT:LIM:UPP$

**Command**

```
[SOURce:]VOLTage[:LEVel]:LIMit:UPPer <numeric>
```

```
[SOURce:]VOLTage[:LEVel]:LIMit:UPPer?
```

**Parameter**

Value: 0 to 315.0 Upper AC voltage limit (The default value is 315.0)

Unit: V

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:LIM:UPP 121
```

Response: NR3

## VOLT:LTL

Sets the line AC voltage.

This command is valid for single-phase three-wire output.

This command is valid during three-phase output when the U-V phase difference is 120 (SYST:CONF:PHAS:UV 120) and the U-W phase difference is 240 (SYST:CONF:PHAS:UW 240).

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

This command is invalid when an external analog signal is used to control the voltage or frequency (VOLT:EXT:FUNC:MODE VPR).

### Command

```
[SOURce:]VOLTage:LTLLine <numeric>
```

```
[SOURce:]VOLTage:LTLLine?
```

### Parameter

|        |            |  |
|--------|------------|--|
| Value: | 0 to 315.0 | Line AC voltage in the single-phase three-wire output L range                  |
|        | 0 to 630.0 | Line AC voltage in the single-phase three-wire output H range                  |
|        | 0 to 272.7 | Line AC voltage in the three-phase output L range                              |
|        | 0 to 545.5 | Line AC voltage in the three-phase output H range<br>(The default value is 0.) |

Unit: V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:LTL 173.0
```

Response: NR3

Returns +9.91E+37 in the case of unbalanced phase or unbalanced voltage.

**VOLT:OFFS**

Sets the DC voltage.

This command is invalid when the combined value with the AC voltage is outside the allowable range (L range: -222.5 V to 222.5 V, H range: -445.0 V to 445.0 V).

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

For single-phase three-wire output, set the U phase voltage. The V phase is automatically outputs to the same value as the U phase but with reverse polarity.

**Command**

```
[SOURce:]VOLTage:OFFSet[:IMMediate] <numeric>[, (@chanlist)]
[SOURce:]VOLTage:OFFSet[:IMMediate]? [(@chanlist)]
```

**Parameter**

Value:    -222.5 to +222.5 DC voltage in the L range  
          -445.0 to +445.0 DC voltage in the H range  
          (The default value is 0.)

Unit:     V

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:OFFS -10.5
```

Response: NR3

Response

## VOLT:OFFS:LIM:LOW

Sets the lower DC voltage limit.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

For single-phase three-wire, set this value with the phase voltage of phase U.

The following relationship must be met:  $VOLT:OFFS:LIM:LOW \leq VOLT:OFFS:-VOLT:OFFS:TRIG \leq VOLT:OFFS:LIM:UPP$

### Command

```
[SOURce:]VOLTage:OFFSet:LIMit:LOWer <numeric>
```

```
[SOURce:]VOLTage:OFFSet:LIMit:LOWer?
```

### Parameter

Value: -445.0 to +445.0 Lower DC voltage limit (The default value is -445.0)

Unit: V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:OFFS:LIM:LOW -12
```

Response: NR3

**VOLT:OFFS:LIM:UPP**

Sets the upper DC voltage limit.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

For single-phase three-wire, set this value with the phase voltage of phase U.

The following relationship must be met:  $\text{VOLT:OFFS:LIM:LOW} \leq \text{VOLT:OFFS:LIM:UPP}$

**Command**

```
[SOURce:]VOLTage:OFFSet:LIMit:UPPer <numeric>
```

```
[SOURce:]VOLTage:OFFSet:LIMit:UPPer?
```

**Parameter**

Value: -445.0 to +445.0 Upper DC voltage limit (The default value is 445.0)

Unit: V

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:OFFS:LIM:UPP -9
```

Response: NR3

## VOLT:OFFS:LTL

Sets the line DC voltage.

This command is valid for single-phase three-wire output.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

### Command

```
[SOURce:]VOLTage:OFFSet:LTLLine <numeric>
```

```
[SOURce:]VOLTage:OFFSet:LTLLine?
```

### Parameter

Value:    -445.0 to +445.0 Line DC voltage in the L range  
         -890.0 to +890.0 Line DC voltage in the H range  
                          (The default value is 0.)

Unit:     V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:OFFS:LTL +50
```

Response: NR3

Returns +9.91E+37 in the case of unbalanced phase or unbalanced voltage.



**VOLT:OFFS:TRIG**

Sets the DC voltage to change to when INIT:TRAN or a software trigger is sent.

This command is invalid when the combined value with the AC voltage is outside the allowable range (L range: -222.5 V to 222.5 V, H range: -445.0 V to 445.0 V).

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

**Command**

```
[SOURce:]VOLTage:OFFSet:TRIGgered <numeric>[, (@chanlist)]
```

```
[SOURce:]VOLTage:OFFSet:TRIGgered?
```

**Parameter**

Value:    -222.5 to +222.5 DC voltage in the L range  
          -445.0 to +445.0 DC voltage in the H range  
          (The default value is 0.)

Unit:     V

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:OFFS:TRIG -10.5
```

Response: NR3

## VOLT:PROT:LOW

Sets the UVP value.

For single-phase three-wire output and three-phase output, set the limits using phase voltages.

This command is valid when using the UVP function (VOLT:PROT:LOW:STAT ON).

### Command

```
[SOURce:]VOLTage:PROTection:LOWer <numeric>
```

```
[SOURce:]VOLTage:PROTection:LOWer?
```

### Parameter

Value: 0 to +489.5      UVP value (The default value is 0)

Unit: V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:PROT:LOW 120.0
```

Response: NR3

## VOLT:PROT:PEAK:LOW

Sets the negative peak OVP value.

### Command

```
[SOURce:]VOLTage:PROTection:PEAK:LOWer <numeric>
```

```
[SOURce:]VOLTage:PROTection:PEAK:LOWer?
```

### Parameter

Value: -489.5 to 0 (The default value is -489.5.)

Unit: V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:PROT:PEAK:LOW -120.0
```

Response: NR3

## VOLT:PROT:PEAK:UPP

Sets the positive peak OVP value.

### Command

```
[SOURce:]VOLTage:PROTection:PEAK:UPPer <numeric>
```

```
[SOURce:]VOLTage:PROTection:PEAK:UPPer?
```

### Parameter

Value: 0 to 489.5 (The default value is 489.5.)

Unit: V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:PROT:PEAK:UPP 120.0
```

Response: NR3

## VOLT:PROT:LOW:STAT

Enables/disables UVP.

Use VOLT:PROT:LOW to set the UVP value.

### Command

```
[SOURce:]VOLTage:PROTection:LOWer:STATe <boolean>
```

```
[SOURce:]VOLTage:PROTection:LOWer:STATe?
```

### Parameter

Value:    ON(1)    Enables the UVP  
         OFF(0)    Disables the UVP (default)

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:PROT:LOW:STAT ON
```

Response: NR1

## VOLT:PROT:UPP

Sets the OVP (rms) value.

For single-phase three-wire output and three-phase output, set the limits using phase voltages.

### Command

```
[SOURce:]VOLTage:PROTection:UPPer <numeric>
```

```
[SOURce:]VOLTage:PROTection:UPPer?
```

### Parameter

Value: 14.0 to 489.5      OVP value (The default value is 489.5)

Unit: V

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:PROT:UPP 120.0
```

Response: NR3

## VOLT:RANG

Sets the voltage range.

If the AC voltage and DC voltage (including the value changed by a trigger) are set within the H range, the voltage range cannot be set to L range.

This command is invalid when the output is on (OUTP ON).

### Command

```
[SOURce:]VOLTage:RANGe[:UPPer] <numeric>
```

```
[SOURce:]VOLTage:RANGe[:UPPer]?
```

### Parameter

|        |       |                   |
|--------|-------|-------------------|
| Value: | 157.5 | L range (default) |
|        | 315   | H range           |

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:RANG 315
```

Response: NR3

## VOLT:RESP

Sets the response speed.

This command is invalid when the output is on (OUTP ON).

This command is invalid when a voltage abnormality simulation or a sequence is running.

### Command

```
[SOURce:]VOLTage:RESPonse <character>
```

```
[SOURce:]VOLTage:RESPonse?
```

### Parameter

|        |        |   |
|--------|--------|---|
| Value: | SLOW   | High stability  |
|        | MEDium | Normal speed (default)  |
|        | FAST   | High-speed response. This is invalid for parallel connection. |

Settings are reset to default values when an \*RST command is sent.

### Example

```
VOLT:RESP MED
```

Response: Characters



**VOLT:TRIG**

Sets the AC voltage to change to when INIT:TRAN or a software trigger is sent.

This command is valid for AC output.

This command is invalid when the combined value with the DC voltage is outside the allowable range (L range: -222.5 V to 222.5 V, H range: -445.0 V to 445.0 V).

**Command**

```
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <numeric>[, (@chan-  
list)]
```

```
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]?
```

**Parameter**

Value:    0 to 157.5    AC voltage in the L range  
           0 to 315.0    AC voltage in the H range  
                           (The default value is 0.)

Unit:      V

Settings are reset to default values when an \*RST command is sent.

**Example**

```
VOLT:TRIG 120
```

Response: NR3

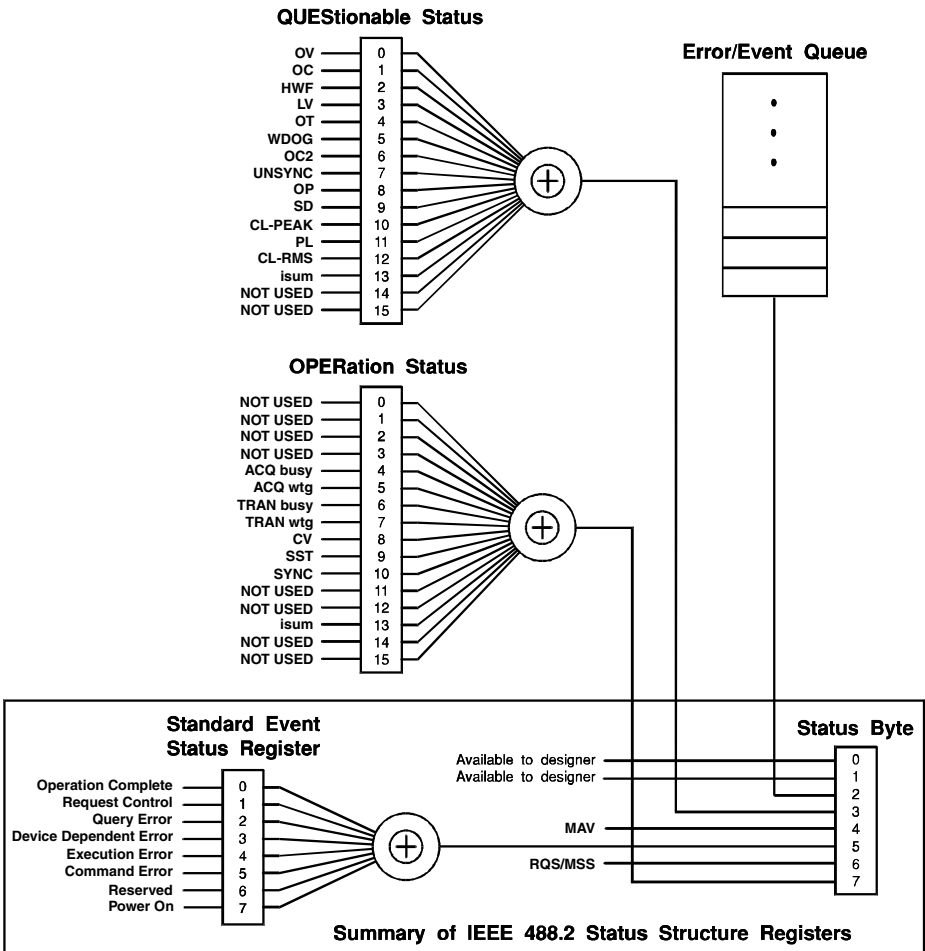
# STATus Command

## Status Report Structure

A "+" represents the logical OR of the register bits.

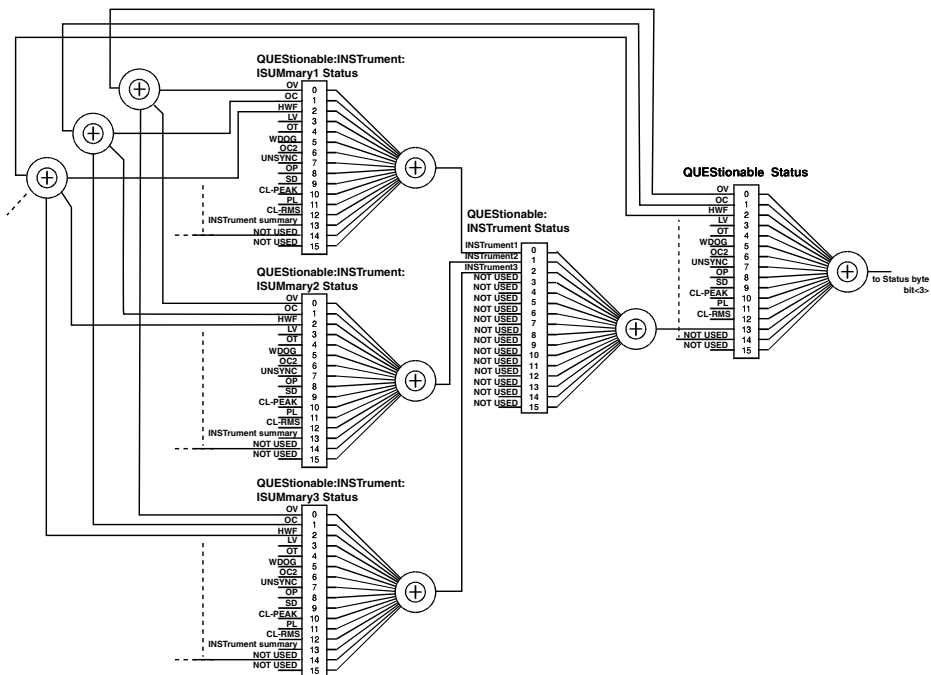
### Single-phase output

#### 1999 SCPI Syntax & Style



## Single-phase three-wire output and three-phase output

The OPERation Status, OPERation:INSTRument Status, and OPERation:INSTRument:ISUMmary Status registers all operate the same way.



## Architecture

---

IEEE 488.2 and SCPI registers are used for status reports.

In each SCPI status register, there are the following sub registers: the CONDition register, the EVENT register, the ENABLE register, the PTRansition filter, and the NTRansition filter.

### CONDition register

Transitions of the CONDition register are automatic and reflect the condition of the product in real time. Reading this register does not affect its contents.

### EVENT register

The EVENT register bits are automatically set according to the changes in the CONDition register. The rule for setting the bits varies depending on the positive and negative transition registers (PTRansition and NTRansition). The EVENT register is reset when it is read.

### ENABLE register

The ENABLE register enables reports to the summary bit or status bit of the event bits.

### Transition filters

Use the PTRansition (positive transition) filter to report events when the condition changes from false to true.

Use the NTRansition (negative transition) filter to report events when the condition changes from true to false.

If both the positive filter and negative filter are set to true, events can be reported each time the status changes.

If both filters are cleared, event reporting is disabled.

## Status byte register

The status byte register stores STB and RQS (MSS) messages as defined by the IEEE 488.1 standard. The status byte register can be read by using IEEE 488.1 serial polling or the IEEE 488.2 common command \*STB?.

When the controller executes serial polling, bit 6 responds with request service (RQS). The status byte value is not changed by serial polling.

\*STB? makes the device transmit the contents of the status byte register and the master status summary (MSS) message.

\*STB? does not change the status byte, MSS, and RQS.

| Bit  | Bit weight | Bit name                                | Description  |
|------|------------|---|--|
| 0    | 1          | Reserved                                | Reserved for future use by IEEE 488. The bit value is notified as zero.  |
| 1    | 2          | Reserved                                |  |
| 2    | 4          | Error/Event Queue                       | If data exists in the error or event queue, this bit is set to true.   |
| 3    | 8          | Questionable Status Register (QUES)     | This bit is set to true when a bit is set in the QUESTIONable event status register and the corresponding bit in the QUESTIONable status enable register is true.      |
| 4    | 16         | Message Available (MAV)                 | This bit is set to true when a request is received from the digital programming interface and the product is ready to generate the data byte.                          |
| 5    | 32         | Standard Event Status Bit Summary (ESB) | This bit is set to true when a bit is set in the event status register.  |
| 6    | 64         | Request Service (RQS)                   | This bit is set to true when a bit is set in the service request enable register and the corresponding bit exists in the status byte. The SRQ line of the GPIB is set. |
|      |            | Master Status Summary (MSS)             | This bit is set to true when any bit in the status byte register is set to 1 and the corresponding bit in the service request enable register is set to 1.             |
| 7    | 128        | Operation Status Register (OPER)        | This bit is set to true when a bit is set in the OPERATION event status register and the corresponding bit in the OPERATION status enable register is set.             |
| 8-15 |            | Not Used                                | Not used   |

## Event status register

The event status register bits are set when certain events occur during product operation. All the event status register bits are set by the error event queue.

This register is defined by the IEEE 488.2 standard and is controlled using the IEEE 488.2 common commands \*ESE, \*ESE?, and \*ESR?.

You can check the error content with SYST:ERR?.

| Bit  | Bit weight | Bit name                    | Description  | Error number               |
|------|------------|-----------------------------|--|----------------------------|
| 0    | 1          | Operation Complete(OPC)     | Set when an *OPC command is received and all operations in standby have been completed.  | -800 to -899               |
| 1    | 2          | Request Control (RQC)       | Not used   | --                         |
| 2    | 4          | Query Error(QYE)            | Set when an attempt is made to read data from the output queue when there is no data or when the output queue is not in the wait state. This indicates that there is no data in the output queue.  | -400 to -499               |
| 3    | 8          | Device Dependent Error(DDE) | Set when there is a device-specific error.   | -300 to -399<br>100 to 999 |
| 4    | 16         | Execution Error(EXE)        | Set when the product evaluates that the program data after the header is outside the formal input range or does not match the specifications of the product. This indicates that a valid SCPI command may not be executed correctly depending on the state of the product. | -200 to -299               |
| 5    | 32         | Command Error(C-ME)         | Set when an IEEE 488.2 syntax error is detected by the parser, when an unidentifiable header is received, or when a group execution trigger enters the internal IEEE 488.2 SCPI command input buffer.  | -100 to -199               |
| 6    | 64         | Reserved                    | Not used   | --                         |
| 7    | 128        | PON                         | Set when the power is turned on.   | --                         |
| 8-15 |            | Reserved                    | Not used   | --                         |

## OPERation status register

The OPERation status register is a 16-bit register that contains information about the normal operating conditions of the product.

| Bit | Bit weight | Bit name           | Description   |
|-----|------------|--------------------|---|
| 0   | 1          | NOT USED           | Not used  |
| 1   | 2          | NOT USED           | Not used  |
| 2   | 4          | NOT USED           | Not used  |
| 3   | 8          | NOT USED           | Not used  |
| 4   | 16         | ACQ busy           | Indicates whether measurement is in progress on the PCR.  |
| 5   | 32         | ACQ wtg            | Indicates whether the product is waiting for a measurement trigger (TRIG).  |
| 6   | 64         | TRAN busy          | Indicates whether the product is running a sequence or a power line abnormality simulation or changing the settings.                                  |
| 7   | 128        | TRAN wtg           | Indicates whether the product is waiting for a trigger (TRIG) for running a sequence or a power line abnormality simulation or changing the settings. |
| 8   | 256        | CV                 | CV output   |
| 9   | 512        | SST                | Indicates whether a soft start is being performed.  |
| 10  | 1024       | SYNC               | Indicates whether the synchronization function is in operation.   |
| 11  | 2048       | NOT USED           | Not used  |
| 12  | 4096       | NOT USED           | Not used  |
| 13  | 8192       | INSTrument Summary | Summary bit of the OPERation:INSTrument subregister   |
| 14  | 16384      | NOT USED           | Not used  |
| 15  | 32768      | NOT USED           | Always 0.   |

## STAT:OPER

Queries the event of the OPERation status register.

A query clears the contents of the register.

### **Command**

```
STATus:OPERation[:EVENT]?
```

Response: NR1



**STAT:OPER:COND**

Queries the condition of the OPERation status register.

A query does not clear the contents of the register.

**Command**

STATus:OPERation:CONDition?

Response: NR1

## STAT:OPER:ENAB

Sets the enable register of the OPERation status register.

### **Command**

```
STATus:OPERation:ENABle <NR1>
```

```
STATus:OPERation:ENABle?
```

### **Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

**STAT:OPER:NTR**

Sets the negative transition filter of the OPERation status register.

**Command**

```
STATus:OPERation:NTRansition <NR1>
```

```
STATus:OPERation:NTRansition?
```

**Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

## STAT:OPER:PTR

Sets the positive transition filter of the OPERation status register.

### **Command**

```
STATus:OPERation:PTRansition <NR1>
```

```
STATus:OPERation:PTRansition?
```

### **Parameter**

Value: 0 to 65535 (The default value is 32767)

Response: NR1

## OPERation:INSTrument subregister

This is the subregister (16 bits) of bit 13 of the OPERation status register.

| Bit | Bit weight | Bit name    | Description                           |
|-----|------------|-------------|---------------------------------------|
| 0   | 1          | INSTrument1 | U phase (OPER:INST:ISUM1) summary bit |
| 1   | 2          | INSTrument2 | V phase (OPER:INST:ISUM2) summary bit |
| 2   | 4          | INSTrument3 | W phase (OPER:INST:ISUM3) summary bit |
| 3   | 8          | NOT USED    | Not used                              |
| 4   | 16         | NOT USED    | Not used                              |
| 5   | 32         | NOT USED    | Not used                              |
| 6   | 64         | NOT USED    | Not used                              |
| 7   | 128        | NOT USED    | Not used                              |
| 8   | 256        | NOT USED    | Not used                              |
| 9   | 512        | NOT USED    | Not used                              |
| 10  | 1024       | NOT USED    | Not used                              |
| 11  | 2048       | NOT USED    | Not used                              |
| 12  | 4096       | NOT USED    | Not used                              |
| 13  | 8192       | NOT USED    | Not used                              |
| 14  | 16384      | NOT USED    | Not used                              |
| 15  | 32768      | NOT USED    | Always 0                              |

## STAT:OPER:INST

Queries the event of the OPERation:INSTrument subregister.

A query clears the contents of the register.

### **Command**

```
STATus:OPERation:INSTrument[:EVENT]?
```

Response: NR1

**STAT:OPER:INST:COND**

Queries the condition of the OPERation:INSTrument subregister.

A query does not clear the contents of the register.

**Command**

```
STATus:OPERation:INSTrument:CONDition?
```

Response: NR1

## STAT:OPER:INST:ENAB

Sets the enable register of the OPERation:INSTrument subregister.

### **Command**

STATus:OPERation:INSTrument:ENABle <NR1>

STATus:OPERation:INSTrument:ENABle?

### **Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1



**STAT:OPER:INST:NTR**

Sets the negative transition filter of the OPERation:INSTRument subregister.

**Command**

```
STATus:OPERation:INSTRument:NTRansition <NR1>
```

```
STATus:OPERation:INSTRument:NTRansition?
```

**Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

## STAT:OPER:INST:PTR

Sets the positive transition filter of the OPERation:INSTrument subregister.

### **Command**

```
STATus:OPERation:INSTrument:PTRansition <NR1>
```

```
STATus:OPERation:INSTrument:PTRansition?
```

### **Parameter**

Value: 0 to 65535 (The default value is 32767)

Response: NR1

## OPERation:INSTRument:ISUMmary{1|2|3} subregister

This is the subregister of bits 1 to 3 of the OPERation:INSTRument subregister. This is a 16-bit register that contains information about the normal operating conditions of the product for each phase.

Of the parameters {1|2|3}, 1 represents U phase, 2 represents V phase, and 3 represents W phase.

| Bit | Bit weight | Bit name           | Description   |
|-----|------------|--------------------|---|
| 0   | 1          | NOT USED           | Not used  |
| 1   | 2          | NOT USED           | Not used  |
| 2   | 4          | NOT USED           | Not used  |
| 3   | 8          | NOT USED           | Not used  |
| 4   | 16         | ACQ busy           | Indicates whether measurement is in progress on the PCR.  |
| 5   | 32         | ACQ wtg            | Indicates whether the product is waiting for a measurement trigger (TRIG).  |
| 6   | 64         | TRAN busy          | Indicates whether the product is running a sequence or a power line abnormality simulation or changing the settings.                                  |
| 7   | 128        | TRAN wtg           | Indicates whether the product is waiting for a trigger (TRIG) for running a sequence or a power line abnormality simulation or changing the settings. |
| 8   | 256        | CV                 | CV output   |
| 9   | 512        | SST                | Indicates whether a soft start is being performed.  |
| 10  | 1024       | SYNC               | Indicates whether the synchronization function is in operation.   |
| 11  | 2048       | NOT USED           | Not used  |
| 12  | 4096       | NOT USED           | Not used  |
| 13  | 8192       | INSTRument Summary | Not used  |
| 14  | 16384      | NOT USED           | Not used  |
| 15  | 32768      | NOT USED           | Always 0.   |

### STAT:OPER:INST:ISUM{1|2|3}

Queries the event of the OPERation:INSTrument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

A query clears the contents of the register.

Of the parameters {1|2|3}, 1 represents U phase, 2 represents V phase, and 3 represents W phase.

#### **Command**

U phase

```
STATus:OPERation:INSTrument:ISUMmary1 [:EVENT]?
```

V phase

```
STATus:OPERation:INSTrument:ISUMmary2 [:EVENT]?
```

W phase (three-phase output only)

```
STATus:OPERation:INSTrument:ISUMmary3 [:EVENT]?
```

Response: NR1

**STAT:OPER:INST:ISUM{1|2|3}:COND**

Queries the condition of the OPERation:INSTrument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

A query does not clear the contents of the register.

Of the parameters {1|2|3}, 1 represents U phase, 2 represents V phase, and 3 represents W phase.

**Command**

U phase

```
STATus:OPERation:INSTrument:ISUMmary1:CONDition?
```

V phase

```
STATus:OPERation:INSTrument:ISUMmary2:CONDition?
```

W phase (three-phase output only)

```
STATus:OPERation:INSTrument:ISUMmary3:CONDition?
```

Response: NR1

## STAT:OPER:INST:ISUM{1|2|3}:ENAB

Sets the enable register of the OPERation:INSTrument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

### **Command**

#### **U phase**

```
STATus:OPERation:INSTrument:ISUMmary1:ENABle <NRf>
```

```
STATus:OPERation:INSTrument:ISUMmary1:ENABle?
```

#### **V phase**

```
STATus:OPERation:INSTrument:ISUMmary2:ENABle <NRf>
```

```
STATus:OPERation:INSTrument:ISUMmary2:ENABle?
```

#### **W phase (three-phase output only)**

```
STATus:OPERation:INSTrument:ISUMmary3:ENABle <NRff>
```

```
STATus:OPERation:INSTrument:ISUMmary3:ENABle?
```

### **Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

**STAT:OPER:INST:ISUM{1|2|3}:NTR**

Sets the negative transition filter of the OPERation:INSTrument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

**Command****U phase**

```
STATus:OPERation:INSTrument:ISUMmary1:NTRansition <NRf>
```

```
STATus:OPERation:INSTrument:ISUMmary1:NTRansition?
```

**V phase**

```
STATus:OPERation:INSTrument:ISUMmary2:NTRansition <NRf>
```

```
STATus:OPERation:INSTrument:ISUMmary2:NTRansition?
```

**W phase (three-phase output only)**

```
STATus:OPERation:INSTrument:ISUMmary3:NTRansition <NRf>
```

```
STATus:OPERation:INSTrument:ISUMmary3:NTRansition?
```

**Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

## STAT:OPER:INST:ISUM{1|2|3}:PTR

Sets the positive transition filter of the OPERATION:INSTRUMENT:ISUMmary{1|2|3} sub-register.

This command is valid for single-phase three-wire output and three-phase output.

### **Command**

#### **U phase**

```
STATus:OPERation:INSTRument:ISUMmary1:PTRansition <NRf>
```

```
STATus:OPERation:INSTRument:ISUMmary1:PTRansition?
```

#### **V phase**

```
STATus:OPERation:INSTRument:ISUMmary2:PTRansition <NRf>
```

```
STATus:OPERation:INSTRument:ISUMmary2:PTRansition?
```

#### **W phase (three-phase output only)**

```
STATus:OPERation:INSTRument:ISUMmary3:PTRansition <NRf>
```

```
STATus:OPERation:INSTRument:ISUMmary3:PTRansition?
```

### **Parameter**

Value: 0 to 65535 (The default value is 32767)

Response: NR1



## QUESTionable status register

The QUESTionable status register is a 16-bit register that stores information related to the product's status and the questionable events that occur during product operation.

The QUESTionable status register bits may indicate that there are problems with the product's measured data.

| Bit | Bit weight | Bit name                                 | Description   |
|-----|------------|--|---|
| 0   | 1          | OV (Over Voltage Protection)             | Overvoltage protection activated  |
| 1   | 2          | OC (Over Current Protection)             | Overcurrent protection activated  |
| 2   | 4          | HWF (Hardware Failure)                   | Device error occurred   |
| 3   | 8          | LV (Low Voltage Protection)              | Undervoltage protection activated   |
| 4   | 16         | OT (Over Temperature Protection)         | Overheat protection activated   |
| 5   | 32         | WD OG (Watchdog protection)              | Communication monitoring activated  |
| 6   | 64         | OC2 (Over Current Protection)            | Overcurrent internal semiconductor protection activated                                       |
| 7   | 128        | UNSYNC (FREQUENCY synchronisation fault) | Sync function error occurred  |
| 8   | 256        | OP (Over Power protection)               | Overpower protection activated  |
| 9   | 512        | SD (SHUTDOWN)                            | Forced output shutdown  |
| 10  | 1024       | CL-PEAK (Current Limit on PEAK)          | Current limit control activated   |
| 11  | 2048       | PL(Power Limit)                          | Power limit activated   |
| 12  | 4096       | CL-RMS(Current Limit on RMS)             | TRIP ENABLE: Overload judgment in progress<br>TRIP DISABL: Output voltage control in progress |
| 13  | 8192       | INSTRument Summary                       | Summary bit of the QUESTionable:INSTRument subregister  |
| 14  | 16384      | Not Used                                 | Not used  |
| 15  | 32768      | Not Used                                 | Always 0.   |

## STAT:QUES

Queries the event of the QUEStionable status register.

A query clears the contents of the register.

### **Command**

```
STATus:QUEStionable[:EVENT]?
```

Response: NR1

**STAT:QUES:COND**

Queries the condition of the QUEStionable status register.

A query does not clear the contents of the register.

**Command**

```
STATus:QUEStionable:CONDition?
```

Response: NR1

## STAT:QUES:ENAB

Sets the enable register of the QUEStionable status register.

### **Command**

```
STATus:QUEStionable:ENABle <NR1>
```

```
STATus:QUEStionable:ENABle?
```

### **Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

**STAT:QUES:NTR**

Sets the negative transition filter of the QUEStionable status register.

**Command**

```
STATus:QUEStionable:NTRansition <NR1>
```

```
STATus:QUEStionable:NTRansition?
```

**Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

## STAT:QUES:PTR

Sets the positive transition filter of the QUEStionable status register.

### **Command**

```
STATus:QUEStionable:PTRansition <NR1>
```

```
STATus:QUEStionable:PTRansition?
```

### **Parameter**

Value: 0 to 65535 (The default value is 32767)

Response: NR1

## QUESTionable:INSTrument subregister

This is the subregister (16 bits) of bit 13 of the QUESTionable status register.

| Bit | Bit weight | Bit name    | Description                           |
|-----|------------|-------------|---------------------------------------|
| 0   | 1          | INSTrument1 | U phase (QUES:INST:ISUM1) summary bit |
| 1   | 2          | INSTrument2 | V phase (QUES:INST:ISUM2) summary bit |
| 2   | 4          | INSTrument3 | W phase (QUES:INST:ISUM3) summary bit |
| 3   | 8          | NOT USED    | Not used                              |
| 4   | 16         | NOT USED    | Not used                              |
| 5   | 32         | NOT USED    | Not used                              |
| 6   | 64         | NOT USED    | Not used                              |
| 7   | 128        | NOT USED    | Not used                              |
| 8   | 256        | NOT USED    | Not used                              |
| 9   | 512        | NOT USED    | Not used                              |
| 10  | 1024       | NOT USED    | Not used                              |
| 11  | 2048       | NOT USED    | Not used                              |
| 12  | 4096       | NOT USED    | Not used                              |
| 13  | 8192       | NOT USED    | Not used                              |
| 14  | 16384      | NOT USED    | Not used                              |
| 15  | 32768      | NOT USED    | Always 0                              |

## STAT:QUES:INST

Queries the event of the QUEStionable:INSTrument subregister.

A query clears the contents of the register.

### **Command**

```
STATus:QUEStionable:INSTrument[:EVENT]?
```

Response: NR1



**STAT:QUES:INST:COND**

Queries the condition of the QUEStionable:INSTRument subregister.

A query does not clear the contents of the register.

**Command**

```
STATus:QUEStionable:INSTRument:CONDition?
```

Response: NR1

## STAT:QUES:INST:ENAB

Sets the enable register of the QUEStionable:INSTrument subregister.

### **Command**

```
STATus:QUEStionable:INSTrument:ENABle <NR1>
```

```
STATus:QUEStionable:INSTrument:ENABle?
```

### **Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

**STAT:QUES:INST:NTR**

Sets the negative transition filter of the QUEStionable:INSTrument subregister.

**Command**

```
STATus:QUEStionable:INSTrument:NTRansition <NR1>
```

```
STATus:QUEStionable:INSTrument:NTRansition?
```

**Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

## STAT:QUES:INST:PTR

Sets the positive transition filter of the QUEStionable:INSTrument subregister.

### **Command**

```
STATus:QUEStionable:INSTrument:PTRansition <NR1>
```

```
STATus:QUEStionable:INSTrument:PTRansition?
```

### **Parameter**

Value: 0 to 65535 (The default value is 32767)

Response: NR1

## **QUESTionable:INSTrument:ISUMmary{1|2|3} subregister**

This is the subregister of bits 1 to 3 of the QUESTionable:INSTrument subregister. This is a 16-bit register that contains information about the product's questionable events and status that occur during operation for each phase.

Of the parameters {1|2|3}, 1 represents U phase, 2 represents V phase, and 3 represents W phase.

The QUESTionable status register bits may indicate that there are problems with the product's measured data.

| Bit | Bit weight | Bit name                                 | Description   |
|-----|------------|--|---|
| 0   | 1          | OV (Over Voltage Protection)             | Overvoltage protection activated  |
| 1   | 2          | OC (Over Current Protection)             | Overcurrent protection activated  |
| 2   | 4          | HWF (Hardware Failure)                   | Device error occurred   |
| 3   | 8          | LV (Low Voltage Protection)              | Undervoltage protection activated   |
| 4   | 16         | OT (Over Temperature Protection)         | Overheat protection activated   |
| 5   | 32         | WDOG (Watchdog protection)               | Communication monitoring activated  |
| 6   | 64         | OC2 (Over Current Protection#2)          | Overcurrent internal semiconductor protection activated                                       |
| 7   | 128        | UNSYNC (FREQUENCY synchronisation fault) | Sync function error occurred  |
| 8   | 256        | OP (Over Power protection)               | Overpower protection activated  |
| 9   | 512        | SD (SHUTDOWN)                            | Forced output shutdown  |
| 10  | 1024       | CL-PEAK (Current Limit on PEAK)          | Current limit control activated   |
| 11  | 2048       | PL(Power Limit)                          | Power limit activated   |
| 12  | 4096       | CL-RMS(Current Limit on RMS)             | TRIP ENABLE: Overload judgment in progress<br>TRIP DISABL: Output voltage control in progress |
| 13  | 8192       | INSTrument Summary                       | Not used  |
| 14  | 16384      | Not Used                                 | Not used  |
| 15  | 32768      | Not Used                                 | Always 0.   |

### STAT:QUES:INST:ISUM{1|2|3}

Queries the event of the QUESTionable:INSTrument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

A query clears the contents of the register.

Of the parameters {1|2|3}, 1 represents U phase, 2 represents V phase, and 3 represents W phase.

#### **Command**

U phase

```
STATus:QUESTionable:INSTrument:ISUMmary1[:EVENT]?
```

V phase

```
STATus:QUESTionable:INSTrument:ISUMmary2[:EVENT]?
```

W phase (three-phase output only)

```
STATus:QUESTionable:INSTrument:ISUMmary3[:EVENT]?
```

Response: NR1

**STAT:QUES:INST:ISUM{1|2|3}:COND**

Queries the condition of the QUESTIONable:INSTrument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

A query does not clear the contents of the register.

Of the parameters {1|2|3}, 1 represents U phase, 2 represents V phase, and 3 represents W phase.

**Command**

U phase

```
STATus:QUESTionable:INSTrument:ISUMmary1:CONDition?
```

V phase

```
STATus:QUESTionable:INSTrument:ISUMmary2:CONDition?
```

W phase (three-phase output only)

```
STATus:QUESTionable:INSTrument:ISUMmary3:CONDition?
```

Response: NR1

## STAT:QUES:INST:ISUM{1|2|3}:ENAB

Sets the enable register of the QUESTIONable:INSTrument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

### **Command**

#### **U phase**

```
STATus:QUESTionable:INSTrument:ISUMmary1:ENABle <NRf>
```

```
STATus:QUESTionable:INSTrument:ISUMmary1:ENABle?
```

#### **V phase**

```
STATus:QUESTionable:INSTrument:ISUMmary2:ENABle <NRf>
```

```
STATus:QUESTionable:INSTrument:ISUMmary2:ENABle?
```

#### **W phase (three-phase output only)**

```
STATus:QUESTionable:INSTrument:ISUMmary3:ENABle <NRf>
```

```
STATus:QUESTionable:INSTrument:ISUMmary3:ENABle?
```

### **Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1



**STAT:QUES:INST:ISUM{1|2|3}:NTR**

Sets the negative transition filter of the QUEStionable:INSTrument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

**Command****U phase**

```
STATus:QUEStionable:INSTrument:ISUMmary1:NTRansition <NRf>
```

```
STATus:QUEStionable:INSTrument:ISUMmary1:NTRansition?
```

**V phase**

```
STATus:QUEStionable:INSTrument:ISUMmary2:NTRansition <NRf>
```

```
STATus:QUEStionable:INSTrument:ISUMmary2:NTRansition?
```

**W phase (three-phase output only)**

```
STATus:QUEStionable:INSTrument:ISUMmary3:NTRansition <NRf>
```

```
STATus:QUEStionable:INSTrument:ISUMmary3:NTRansition?
```

**Parameter**

Value: 0 to 65535 (The default value is 0)

Response: NR1

## STAT:QUES:INST:ISUM{1|2|3}:PTR

Sets the positive transition filter of the QUESTIONable:INSTRument:ISUMmary{1|2|3} subregister.

This command is valid for single-phase three-wire output and three-phase output.

### **Command**

#### **U phase**

```
STATus:QUEStionable:INSTRument:ISUMmary1:PTRansition <NR1>
```

```
STATus:QUEStionable:INSTRument:ISUMmary1:PTRansition?
```

#### **V phase**

```
STATus:QUEStionable:INSTRument:ISUMmary2:PTRansition <NR1>
```

```
STATus:QUEStionable:INSTRument:ISUMmary2:PTRansition?
```

#### **W phase (three-phase output only)**

```
STATus:QUEStionable:INSTRument:ISUMmary3:PTRansition <NR1>
```

```
STATus:QUEStionable:INSTRument:ISUMmary3:PTRansition?
```

### **Parameter**

Value: 0 to 65535 (The default value is 32767)

Response: NR1

## Preset status

---

### STAT:PRES

Resets the ENABLE, PTRansition, and NTRansition filter registers of all status registers (including sub registers) to their default values.

Default values:

STATus:ENABle = 0x0000

STATus:PTRansition = 0x7FFF

STATus:NTRansition = 0x0000

### Command

STATus:PRESet

# SYSTem Command

## SYST:COMM:RLST

Sets the product to remote or local mode.

### Command

```
SYSTem:COMMunicate:RLState <character>
```

```
SYSTem:COMMunicate:RLState?
```

### Parameter

|              |  |
|--------------|--|
| Value: LOCAL | Sets the product to local mode (Remote Disable; the RMT turns off).<br>This enables both panel operations and commands.<br>This is a substitute command for IEEE488.1 ren FALSE (Remote Disable).                          |
| REMOte       | Sets the product to remote mode<br>All panel keys, except the LOCAL key, are locked.<br>This is a substitute command for IEEE 488.1 ren (Remote Enable).<br>This is also the substitute command for address specification. |
| RWLock       | Sets the product to remote mode<br>All panel keys (including the LOCAL key) are locked.<br>This is a substitute command for IEEE 488.1 llo (Local Lock Out).   |

### Example

```
SYST:COMM:RLST REM
```

Response: Characters

## **SYST:CONF:ACC**

Enables/disables AC coupling.

### **Command**

```
SYSTem:CONFigure:ACCoupling[:STATe] <boolean>
```

```
SYSTem:CONFigure:ACCoupling[:STATe]?
```

### **Parameter**

Value:    ON(1)    Enables AC coupling  
         OFF(0)   Disables AC coupling (default)

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
SYST:CONF:ACC ON
```

Response: NR1

## SYST:CONF:ADJ:VOLT:FINE

Sets the output voltage offset.

Use SYST:CONF:ADJ:VOLT:TERM:MODE to set whether the sensing function is enabled or disabled.

### Command

```
SYSTem:CONFigure:ADJust:VOLTage:FINE <numeric>[, (@chanlist)]
```

```
SYSTem:CONFigure:ADJust:VOLTage:FINE? [ (@chanlist)]
```

### Parameter

|             |              |   |
|-------------|--------------|---|
| Value:      | -200 to +200 | Voltage offset (The default value is 0) |
| Resolution: | 10           | The ones digit is rounded.              |

### Example

```
SYST:CONF:ADJ:VOLT:FINE 10
```

Response: NR3

## SYST:CONF:ADJ:VOLT:TERM:MODE

Set whether the sensing function is enabled or disabled for the voltage offset setting.

Use SYST:CONF:ADJ:VOLT:FINE to set the offset value.

### Command

```
SYSTem:CONFigure:ADJst:VOLTagE:TERMinal:MODE <character>
```

```
SYSTem:CONFigure:ADJst:VOLTagE:TERMinal:MODE?
```

### Parameter

Value:   OTERM   When the sensing function is disabled (default)  
          STERM   When the sensing function is enabled

### Example

```
SYST:CONF:ADJ:VOLT:TERM:MODE OTERM
```

Response: Characters

## SYST:CONF:FORM:FRAM

Queries the number of units operating in parallel.

### **Command**

```
SYSTem:CONFigure:FORMation:FRAMe[:COUNT]?
```

Response: NR1

Returns 1 if parallel operation is not being performed.



## SYST:CONF:FORM:FRAM:INFO

Queries the information about the specified PCR-WE.

### Command

```
SYSTem:CONFigure:FORMation:FRAMe:INFO? <Nrf>
```

### Parameter

|        |   |  |
|--------|---|--|
| Value: | 0 | Master unit or during standalone operation |
|        | 1 | Slave 1                                    |
|        | 2 | Slave 2                                    |
|        | 3 | Slave 3                                    |

### Example

```
SYST:CONF:FORM:FRAM:INFO? 0
```

Response example for model PCR3000WE2, serial number WE3RD008, firmware version 1.00 IFC0.03.0035 IOC0.05.0079

```
KIKUSUI,PCR3000WE2,WE3RD008,1.00 IFC0.03.0035 IOC0.05.0079
```

is returned.

## SYST:CONF:FORM:PMOD

Queries the number of power modules in the specified PCR-WE.

One power module is 6 kW.

### **Command**

```
SYSTem:CONFigure:FORMation:PMODule[:COUNT]?
```

Response: NR1

The PCR1000WE/PCR2000WE/PCR3000WE2 returns 1.

Example: If the specified model is the PCR12000WE, 2 is returned.

**SYST:CONF:FORM:PMOD:INFO**

Queries the information about the specified power module.

**Command**

```
SYSTem:CONFigure:FORMation:PMODule:INFO? <NRf_index>,<NRf_moduleIndex>
```

Parameter <NRf\_index>

|        |   |  |
|--------|---|--|
| Value: | 0 | Master unit or during standalone operation |
|        | 1 | Slave 1                                    |
|        | 2 | Slave 2                                    |
|        | 3 | Slave 3                                    |

Parameter <NRf\_moduleIndex>

Value:     Number of modules   The module to be queried

**Example**

```
SYST:CONF:FORM:PMOD:INFO? 0,1
```

**Response example for PCR-WE/WE2**

```
PUC1.07.0096[1314],PFC
```

**Response example for PCR-WE2R**

```
PUC1.07.0096[1314],INV INV1:V0.27 INV2:V0.27
```

## SYST:CONF:FORM:PSAV:MAX

Sets the maximum expected power of the power-saving function.

Which power module is to run is automatically set.

This command is invalid for the PCR1000WE/PCR2000WE/PCR3000WE2.

### **Command**

```
SYSTem:CONFigure:FORMation:PSAVer:MAXimum <NRf>
```

```
SYSTem:CONFigure:FORMation:PSAVer:MAXimum?
```

### **Parameter**

Value: 0 to total wattage of normally running power modules

Unit: VA

Settings are reset to default values when an \*RST command is sent.

Settings are changed when the SYST:CONF:FORM:PSAV:MOD or SYST:CONF:FORM:PSAV:RES command is sent.

Example: To place a limit at 6 kW

```
SYST:CONF:FORM:PSAV:MAX 6000
```

Response: NR1

**SYST:CONF:FORM:PSAV:MOD**

Sets all power modules to run using the power-saving function.

This command is invalid for the PCR1000WE/PCR2000WE/PCR3000WE2.

**Command**

```
SYSTem:CONFigure:FORMation:PSAVer:MODules <character_>[,<character>]...
```

```
SYSTem:CONFigure:FORMation:PSAVer:MODules?
```

Parameter <character>

The first parameter is the power module operating condition of the master unit.

The subsequent parameters are the power module operating conditions for slave unit 1 and later (for parallel operation only).

The number of parameters must be equal to the number of PCR-WE units, which includes the master unit and slave units.

Parameter details: {E|D}{E|D}{E|D}{E|D}{E|D}{E|D}...

(E|D) represents power module 0, power module 1, power module 2, and so on from the left. This must be set the same as the number of slots that the PCR-WE has.

|        |   |                           |
|--------|---|---------------------------|
| Value: | E | Power module operated     |
|        | D | Power module not operated |

The default value is EEEEEEE (standalone, all modules operated).

**Example**

To operate the master unit's power modules 0 and 1 and the slave unit 1's power modules 2 and 3 and set the maximum expected power to 24000 VA (6000VA×4 modules).

```
SYST:CONF:FORM:PSAV:MOD EEDDDD,DDEEDD
```

Response: Characters, [,Characters]...

## SYST:CONF:FORM:PSAV:RES

Resets the maximum expected power setting of the power-saving function.

This command is invalid when the output is on (OUTP ON).

This command is invalid for the PCR1000WE/PCR2000WE/PCR3000WE2.

### **Command**

```
SYSTem:CONFigure:FORMation:PSAVer:RESet
```

### **Response**

## **SYST:CONF:PHAS:UOFF**

Sets the absolute phase angle of the U phase relative to the reference phase.

This command is valid for three-phase output.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

### **Command**

```
SYSTem:CONFigure:PHASe:UOFFset <numeric>
```

```
SYSTem:CONFigure:PHASe:UOFFset?
```

### **Parameter**

Value: -360.00 to 360.00 Absolute phase angle (The default value is 0)

Unit: DEG

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
SYST:CONF:PHAS:UOFF 35.51
```

Response: NR3

## SYST:CONF:PHAS:UV

Sets the U-V phase difference.

This command is valid for three-phase output.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

### **Command**

```
SYSTem:CONFigure:PHASe:UV <numeric>
```

```
SYSTem:CONFigure:PHASe:UV?
```

### **Parameter**

Value: -360.00 to 360.00 U-V phase difference

The default value for three-phase output is 120.

Unit: DEG

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
SYST:CONF:PHAS:UV 121.0
```

Response: NR3



## **SYST:CONF:PHAS:UW**

Sets the U-W phase difference.

This command is valid for three-phase output.

If the signal source is set to external signal source (VOLT:EXT:EXTDC:SIGN:SOUR EXT), this command is invalid.

### **Command**

```
SYSTem:CONFigure:PHASe:UW <numeric>
```

```
SYSTem:CONFigure:PHASe:UW?
```

### **Parameter**

Value: -360.00 to 360.00 U-W phase difference (The default value is 240)

Unit: DEG

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
SYST:CONF:PHAS:UW 241.0
```

Response: NR3

## SYST:CONF:PON:STAT

Sets the condition panel setting state when the POWER switch is turned on.

### Command

```
SYSTem:CONFigure:PON:STATe <character>
```

```
SYSTem:CONFigure:PON:STATe?
```

### Parameter

|        |      |   |
|--------|------|---|
| Value: | RST  | Reset the panel settings  |
|        | RCL0 | Settings stored in memory 0   |
|        | AUTO | The previous state before the POWER switch was turned off (this is the default value)<br>Use OUTP:PON:STAT to set the output. |

### Example

```
SYST:CONF:PON:STAT RCL0
```

Response: Characters

## **SYST:CONF:SSUP**

Enables or disables the voltage surge suppression function.

### **Command**

```
SYSTem:CONFigure:SSUPpression[:STATe] <boolean>
```

```
SYSTem:CONFigure:SSUPpression[:STATe]?
```

### **Parameter**

Value:    ON(1)    Enables the voltage surge function (default)  
         OFF(0)    Disables the voltage surge function

### **Example**

```
SYST:CONF:SSUP ON
```

Response: NR1

## SYST:CONF:WIR/ SYST:CONF:NOUT

Sets the output method.

This command is invalid when the output is on (OUTP ON).

This command is invalid when a sequence or a power line abnormality simulation is running.

This command is invalid on the PCR1000WE and PCR2000WE.

### Command

```
SYSTem:CONFigire:WIRing <NR1>
```

```
SYSTem:CONFigire:WIRing?
```

```
SYSTem:CONFigire:NOUtputs <NR1>
```

```
SYSTem:CONFigire:NOUtputs?
```

### Parameter

|        |   |                                |
|--------|---|--------------------------------|
| Value: | 1 | Single-phase output            |
|        | 2 | Single-phase three-wire output |
|        | 3 | Three-phase output             |

### Example

```
SYST:CONF:NOUT 3
```

Response: NR1

## **SYST:DATE**

Sets the date (UTC).

Also set the time (SYST:TIME).

The time and date are used in the timestamps of files saved to USB memory devices.

If you specify a day that does not exist (for example, February 30), the settings are changed to the first day of the following month.

### **Command**

```
SYSTem:DATE <year_NR1>,<month_NR1>,<day_NR1>
```

```
SYSTem:DATE?
```

Parameter <year\_NR1>

Value 2016 to 2037 Year

Parameter <month\_NR1>

Value 1 to 12 Month

Parameter <day\_NR1>

Value 1 to 31 Day

### **Example**

```
SYST:DATE 2015,4,14
```

### **Response**

Returns the year, month, and day in a comma-separated NR1 format.

## SYST:ERR

Reads the oldest error information or event information from the error queue.

The error/event queue can hold up to 16 errors. -> [Tutorial "Error Checking"](#)(p. 336)

The error queue is cleared if a \*CLS command is sent.

-> ["List of Errors"](#)(p. 306)

### **Command**

`SYSTem:ERRor[:NEXT]?`

### **Response**

Returns the oldest error or event from the error/event queue in the following format, in response to SYST:ERR?.

Example: If there is no error or event

This command returns +0 "No error."

Example: If a command that cannot be executed in the present operating state is received

This command returns -221, "Settings conflict."

**SYST:ERR:COUN**

Queries the number of errors occurring currently.

**Command**

`SYSTem:ERRor:COUNT?`

Response: NR1

Returns +0 if there are no errors.

## SYST:EXT:DIG:READ

Queries all the signal input states of SIGNAL IN channels (CTRL.1 to CTRL.4) and SIGNAL IO channels (DIO.1, DIO.2).

### Command

```
SYSTem:EXTernal:DIGital:READ?
```

### Response

Returns the sum of the bit weights with high set to 1 for the signal states of the SIGNAL IN channels (CTRL.1 to CTRL.4) and SIGNAL IO channels (DIO.1, DIO.2) in NR1 format.

Ports that USERPROGIN is not mapped to is assumed to be 0 (low).

| Bit | Bit weight | channel  |
|-----|------------|----------|
| 0   | 1          | CTRL.1   |
| 1   | 2          | CTRL.2   |
| 2   | 4          | CTRL.3   |
| 3   | 8          | CTRL.4   |
| 4   | 16         | DIO.1    |
| 5   | 32         | DIO.2    |
| 6   | 64         | NOT USED |
| 7   | 128        | NOT USED |

Example: When the DIO.1 and DIO.2 signal inputs are high, 48 is returned.



**SYST:EXT:DIG:WRIT**

Outputs the SIGNAL OUT channels (STAT.1 to STAT.4) and SIGNAL IO channels (DIO.1, DIO.2) collectively.

Channels that USERPROGOUT is not mapped to are not output.

**Command**

```
SYSTem:EXTernal:DIGital:WRITe <NR1>
```

Value: 0 to 63      Sum of the bit weights with output set to 1

| Bit | Bit weight | channel  |
|-----|------------|----------|
| 0   | 1          | STAT.1   |
| 1   | 2          | STAT.2   |
| 2   | 4          | STAT.3   |
| 3   | 8          | STAT.4   |
| 4   | 16         | DIO.1    |
| 5   | 32         | DIO.2    |
| 6   | 64         | NOT USED |
| 7   | 128        | NOT USED |

Example: To output high to STAT.1 and STAT.2 and low to other channels

```
SYST:EXT:DIG:WRIT 3
```

## SYST:EXT:SIGIN:MAP

Maps SIGNAL IN channels (CTRL.1 to CTRL.3).

### Command

```
SYSTem:EXTeRnal:SIGIN:MAP <NRf_port>,<character_map>
```

```
SYSTem:EXTeRnal:SIGIN:MAP? <NRf_port>
```

Parameter <NRf\_port>

Value 1 to 3 Channel number to be mapped (CTRL.1 to CTRL.3)

Parameter <character\_map>

|        |             |  |
|--------|-------------|--|
| Value: | DISabled    | Disabled (default)                             |
|        | OUTCTL      | Output on/off                                  |
|        | SEQEXEC     | Sequence run/stop                              |
|        | ALMCLR      | Alarm clear                                    |
|        | EXTALM      | External alarm input                           |
|        | WIRCTL_1P   | Output method change (single-phase)            |
|        | WIRCTL_1P3W | Output method change (single-phase three-wire) |
|        | WIRCTL_3P   | Output method change (three-phase)             |
|        | VRANGE      | Output range change                            |
|        | RCL_A       | Recall A memory                                |
|        | RCL_B       | Recall B memory                                |
|        | RCL_C       | Recall C memory                                |
|        | OUTINH      | Output on inhibit                              |
|        | USERPROGIN  | Programmable signal in                         |

### Example

```
SYST:EXT:SIGIN:MAP 1,EXTALM
```

```
SYST:EXT:SIGIN:MAP? 1
```

Response: Characters

**SYST:EXT:SIGIN:POL**

Sets the polarity of the parameter to map to SIGNAL IN.

This command is invalid when the output is on (OUTP ON).

**Command**

```
SYSTem:EXTernal:SIGIN:POLarity <NRf_port>,<character_pol>
```

```
SYSTem:EXTernal:SIGIN:POLarity? <NRf_port>
```

Parameter <NRf\_port>

Value 1 to 3 Channel number to be mapped (CTRL.1 to CTRL.3)

Parameter <character\_pol>

|        |          |                         |
|--------|----------|-------------------------|
| Value: | POSitive | Positive edge (default) |
|        | NEGative | Negative edge           |

**Example**

```
SYST:EXT:SIGIN:POL 1,NEG
```

Response: Characters

## SYST:EXT:SIGIN:STAT

Queries the signal level of the SIGNAL IN channel.

### **Command**

```
SYSTem:EXTernal:SIGIN:STATe? <NRF>
```

### **Response**

Value    1 to 4    Channel number to be queried (CTRL.1 to CTRL.4)

### **Example**

```
SYST:EXT:SIGIN:STAT? 1
```

### **Response**

Returns 1 when the signal level is high and 0 when it is low.

Returns 0 if USERPROGIN is not mapped to the SIGNAL IN channel.

## SYST:EXT:SIGOUT:MAP

Maps SIGNAL OUT channels (STAT.1 to STAT.3).

### Command

```
SYSTem:EXTernal:SIGOUT:MAP <NRf_port>,<character_map>
```

```
SYSTem:EXTernal:SIGOUT:MAP? <NRf_port>
```

Parameter <NRf\_port>

Value 1 to 3 Channel number to be mapped (STAT.1 to STAT.3)

Parameter <character\_map>

|        |             |   |
|--------|-------------|---|
| Value: | DISabled    | Disabled (default)                                |
|        | OUTON       | Output on status                                  |
|        | IPKLIM      | Current peak limit status                         |
|        | OVERLOAD    | Overload status                                   |
|        | BUSY        | Busy status                                       |
|        | WIRING_1P   | Output method status (single-phase)               |
|        | WIRING_1P3W | Output method status (single-phase three-wire)    |
|        | WIRING_3P   | Output method status (three-phase)                |
|        | VRANGE-H    | Output voltage H range status                     |
|        | POWON       | POWER switch on status                            |
|        | SEQSTAT     | Sequence/power line abnormality simulation status |
|        | USERPROGOUT | Programmable signal status                        |

### Example

```
SYST:EXT:SIGOUT:MAP 1,BUSY
```

Response: Characters

## SYST:EXT:SIGOUT:POL

Sets the polarity of the parameter to map to SIGNAL OUT.

This command is invalid when the output is on (OUTP ON).

### **Command**

```
SYSTem:EXTernal:SIGOUT:POLarity <NRf_port>,<character_pol>
```

```
SYSTem:EXTernal:SIGOUT:POLarity? <NRf_port>
```

Parameter <NRf\_port>

Value 1 to 3 Channel number to be mapped (STAT.1 to STAT.3)

Parameter <character\_pol>

Value: HIGH (Default)  
LOW

### **Example**

```
SYST:EXT:SIGOUT:POL 1,LOW
```

Response: Characters

## SYST:EXT:SIGOUT:STAT

Sets the signal level of the SIGNAL OUT channel.

This command is valid only when USERPROGOUT is mapped to the SIGNAL OUT channel.

### Command

```
SYSTem:EXTeRnal:SIGOUT:STATe <NRf_port>,<boolean_pol>
```

Parameter <NRf\_port>

Value 1 to 3 Channel number to be mapped (STAT.1 to STAT.3)

Parameter <boolean\_pol>

Value: ON(1) HIGH (Default)  
OFF(0) LOW

An error occurs if USERPROGOUT is not mapped to the SIGNAL OUT channel.

### Example

```
SYST:EXT:SIGOUT:STAT 1,ON
```

## SYST:EXT:SSIGIO:MAP

Maps SIGNAL IO channels (DIO.1, DIO.2).

### Command

```
SYSTem:EXTeRnal:SSIGIO:MAP <NRf_port>,<character_inout>,<character_map>
```

```
SYSTem:EXTeRnal:SSIGIO:MAP? <NRf_port>
```

Parameter <NRf\_port>

Value 1 to 2 Channel number to be mapped (DIO.1, DIO.2)

Parameter <character\_inout>

|        |         |                                 |
|--------|---------|---------------------------------|
| Value: | IN      | Control using external contacts |
|        | OUT     | Monitors the operation status   |
|        | DISable | Disabled (default)              |

Parameter <character\_map> (<character\_inout> is set to IN)

|        |             |  |
|--------|-------------|--|
| Value: | SYNCCLK     | Output reference phase signal                  |
|        | SEQTRIGIN   | Sequence trigger input                         |
|        | OUTCTL      | Output on/off                                  |
|        | SEQEXEC     | Sequence run/stop                              |
|        | ALMCLR      | Alarm clear                                    |
|        | EXTALM      | External alarm input                           |
|        | WIRCTL_1P   | Output method change (single-phase)            |
|        | WIRCTL_1P3W | Output method change (single-phase three-wire) |
|        | WIRCTL_3P   | Output method change (three-phase)             |
|        | VRANGE      | Output range change                            |
|        | RCL_A       | Recall A memory                                |
|        | RCL_B       | Recall B memory                                |
|        | RCL_C       | Recall C memory                                |
|        | OUTINH      | Output on inhibit                              |
|        | USERPROGIN  | Programmable signal in                         |

Parameter <character\_map> (When <character\_inout> is set to OUT)

|        |            |                               |
|--------|------------|-------------------------------|
| Value: | STDCLK     | Output reference phase output |
|        | SEQTRIGOUT | Sequence trigger output       |
|        | OUTON      | Output on status              |
|        | IPKLIM     | Current peak limit status     |
|        | OVERLOAD   | Overload status               |
|        | BUSY       | Busy status                   |



|             |   |
|-------------|---|
| WIRING_1P   | Output method status (single-phase)               |
| WIRING_1P3W | Output method status (single-phase three-wire)    |
| WIRING_3P   | Output method status (three-phase)                |
| VRANGEH     | Output voltage H range status                     |
| POWON       | POWER switch on status                            |
| SEQSTAT     | Sequence/power line abnormality simulation status |
| USERPROGOUT | Programmable signal status                        |

### Example

```
SYST:EXT:SSIGIO:MAP 1,IN,VRANGE
```

### Response

Returns the I/O of the specified channel and the mapped parameters in order in comma-separated character format.

## SYST:EXT:SSIGIO:POL

Sets the polarity of the parameter to map to SIGNAL IO.

This command is invalid when the output is on (OUTP ON).

### Command

```
SYSTem:EXTernal:SSIGIO:POLarity <NRf_port>,<character_pol>
```

```
SYSTem:EXTernal:SSIGIO:POLarity? <NRf_port>
```

Parameter <NRf\_port>

Value 1 to 2 Channel number to be mapped (DIO.1, DIO.2)

Parameter <character\_pol> (When <character\_map> is set to input parameters)

Value: POSitive Positive edge (default)  
NEGative Negative edge

Parameter <character\_pol>

Value: HIGH (Default)  
LOW

### Example

```
SYST:EXT:SSIGIO:POL 1,LOW
```

Response: Characters

**SYST:EXT:SSIGIO:STAT**

Sets the signal level of the SIGNAL IO channel.

The set command is valid only when USERPROGOUT is mapped to the SIGNAL IO channel.

The query command is valid only when USERPROGIN is mapped to the SIGNAL IO channel.

**Command**

```
SYSTem:EXTeRnal:SSIGIO:STATe <NRf_port>,<boolean_pol>
```

```
SYSTem:EXTeRnal:SSIGIO:STATe? <NRf_port>
```

Parameter <NRf\_port>

Value 1 to 2 Channel number to be mapped (DIO.1, DIO.2)

Parameter <boolean\_pol>

Value: ON(1) HIGH (Default)  
OFF(0) LOW

An error occurs if USERPROGOUT is not mapped to the SIGNAL IO channel.

**Example**

```
SYST:EXT:SSIGIO:STAT 1,ON
```

```
SYST:EXT:SSIGIO:STAT? 1
```

**Response**

Returns 1 when the signal level is high and 0 when it is low.

Returns 0 if USERPROGIN is not mapped to the SIGNAL IO channel.

## SYST:KLOC

Sets or releases panel control lock.

### **Command**

```
SYSTem:KLOCk <boolean>
```

```
SYSTem:KLOCk?
```

### **Parameter**

Value:   ON(1)   Locks the panel control  
          OFF(0)   Unlocks the panel control

Response: NR1

**SYST:LOC/ SYST:REM/ SYST:RWL**

This is an old style command.

Use SYST:COMM:RLST(p. 244) when creating new programs.

**Command**

SYSTem:LOCal

SYSTem:REMOte

SYSTem:RWLock

## SYST:OPT

Queries the optional interface boards that are installed in the product.

This is an alias for \*OPT.

### **Command**

```
SYSTem:OPTion?
```

### **Response**

Returns the installed options in comma-separated string format. Returns "0" if no options are installed.

## **SYST:PASS**

Enables a password-protected command.

### **Command**

```
SYSTem:PASSword[:CENable] "<string>"
```

```
SYSTem:PASSword[:CENable]?
```

### **Parameter**

Value: Enter password set by SYSTem:PASSword:NEW

Response: "string"

### **Example**

```
SYST:PASS "password"
```

## SYST:PASS:CDIS

Disable the password-protected command.

### **Command**

```
SYSTem:PASSword:CDISable "<string>"
```

### **Parameter**

Value: Enter password set by SYSTem:PASSword:NEW

### **Example**

```
SYST:PASS:CDIS "password"
```



## **SYST:PASS:NEW**

Set the password.

### **Command**

```
SYSTem:PASSword:NEW "<string_exist>","<string_new>"
```

Parameter "<string\_exist>": existing password, "<string\_new>" new password

Characters that can be used: alphanumeric characters (A-Z, a-z, 0-9), underscore,  
hyphen

Number of characters: 4 to 15

The default value is "".

### **Example**

```
SYST:PASS:NEW "existing password", "new password"
```

## SYST:PASS:STAT

Queries whether a password-protected command is valid or invalid.

### **Command**

```
SYSTem:PASSword[:CENable]:STATe?
```

Response: NR1

### **Example**

```
SYST:PASS:STAT?
```

## SYST:SEC:IMM

Sanitizes the product to its factory default settings.

Communication settings are also returned to their factory default conditions.

This is valid when the password protection command is valid (SYST:PASS).

When parallel operation in use, set the rotary switches for the address and the number of slave units on the master unit and all slave units to zero, and then use this command.

— Note —

Sanitization clears all user-defined state information and user-defined I/O settings such as the IP address. Because unexpected data loss may occur, sanitization is recommended only after firmware updates.

### Command

```
SYSTem:SECurity:IMMediate
```

## SYST:SLE

Turns the sleep function on and off.

Even if the sleep function is turned off, you can activate the sleep function by sending the SYST:SLE:EXEC command.

Use SYST:SLE:TIME to set the time that must elapse before the PCR-WE enters sleep mode.

### **Command**

```
SYSTem:SLEep[:STATe] <boolean>
```

```
SYSTem:SLEep[:STATe] ?
```

### **Parameters**

#### **Parameter**

Value:    ON(1)    The sleep function turns on.  
          OFF(0)    The sleep function is turned off (default).

Settings are reset to default values when an \*RST command is sent.

Response: NR1

## SYST:SLE:EXEC

Activates sleep mode immediately.

This command is valid even when the sleep function has been turned off (SYST:SLE OFF).

This command is not valid when the output is on (OUTP ON), when an alarm has occurred, when a sequence is being executed, and when the PCR-WE is in the WTG state.

### **Command**

SYSTem:SLEep:EXECute

## SYST:SLE:TIME

Sets the time that must elapse before the product enters sleep mode.

Use SYST:SLE to turn the sleep function on and off.

### **Command**

```
SYSTem:SLEep:TIME <numeric>
```

```
SYSTem:SLEep:TIME?
```

### **Parameter**

Value: 60 to 3600 (The default value is 3600)

Unit: S

Settings are reset to default values when an \*RST command is sent.

Response: NR3

## SYST:TIME

Sets the time.

Also set the date (using SYST:CONF:DATE).

The time and date are used in the timestamps of files saved to USB memory devices.

### Command

```
SYSTem:TIME <hour_NR1>,<min_NR1>,<sec_NR1>
```

```
SYSTem:TIME?
```

Parameter <hour\_NR1>

Value 0 to 23 Hour

Parameter <min\_NR1>

Value 0 to 59 Minutes

Parameter <sec\_NR1>

Value 0 to 59 second

### Example

```
SYST:TIME 23,0,0
```

### Response

Returns the hour, minute, and second in NR1 format.

## SYST:TIME:ADJ

Automatically synchronizes the system clock using the NTP server on the network.

### **Command**

SYSTem:TIME:ADJust



## **SYST:TZON**

Sets the time zone of the system clock.

Use SYST:TZON:CAT? to check the time zone ID.

### **Command**

```
SYSTem:TZONE "<string>"
```

```
SYSTem:TZONE?
```

### **Parameter**

Value: Time zone ID or UTC (The default value is "UTC")

### **Example**

```
SYST:TZON "Asia/Tokyo"
```

Response: "string"

## SYST:TZON:CAT

Queries the time zone IDs that can be used.

### **Command**

`SYSTem:TZONe:CATalog?`

Response: Comma-separated character string

**SYST:VERS**

Queries the version of the SCPI specifications that the product complies with.

**Command**

```
SYSTem:VERSion?
```

**Response**

Returns 1999.0.

# TRIGger Command

## TRIG:ACQ

Executes a software trigger on the ACQuire trigger subsystem.

### **Command**

```
TRIGger:ACQuire[:IMMediate]
```

## **TRIG:ACQ:SOUR**

Sets the condition (trigger source) for actually starting the measurement after the AC-Quire trigger subsystem receives an INIT:ACQ.

### **Command**

```
TRIGger:ACQuire:SOURce <character>
```

```
TRIGger:ACQuire:SOURce?
```

### **Parameter**

|        |           |  |
|--------|-----------|--|
| Value: | IMMediate | Starts the measurement immediately (default)   |
|        | BUS       | Waits for a software trigger (a *TRG, TRIG:ACQ, or IEEE 488.1 get—Group Execute Trigger—command), and then begin measuring |

Settings are reset to default values when an \*RST command is sent.

### **Example**

```
TRIG:ACQ:SOUR BUS
```

Response: Characters

## TRIG:PROG

Executes a software trigger on the PROGram trigger subsystem.

### **Command**

```
TRIGger:PROGram[:IMMediate]
```

## TRIG:PROG:SOUR

Sets the condition (trigger source) for actually starting the sequence operation after the PROGRAM trigger subsystem receives an INIT:PROG.

### Command

```
TRIGger:PROGram:SOURce <character>
```

```
TRIGger:PROGram:SOURce?
```

### Parameter

|        |           |  |
|--------|-----------|--|
| Value: | IMMediate | Execute the sequence immediately (default)   |
|        | BUS       | Waits for a software trigger (a *TRG, TRIG:PROG, or IEEE 488.1 get—Group Execute Trigger—command), and then execute the sequence |

Settings are reset to default values when an \*RST command is sent.

### Example

```
TRIG:PROG:SOUR BUS
```

Response: Characters

## TRIG:SIM

Executes a software trigger on the SIMulation trigger subsystem.

### **Command**

```
TRIGger:SIMulation[:IMMediate]
```



**TRIG:SIM:SOUR**

Sets the condition (trigger source) for actually executing the simulation after the SIMulation trigger subsystem receives an INIT:SIM.

**Command**

```
TRIGger:SIMulation:SOURce <character>
```

```
TRIGger:SIMulation:SOURce?
```

**Parameter**

|        |           |   |
|--------|-----------|---|
| Value: | IMMediate | Executes the power line abnormality simulation immediately (default)  |
|        | BUS       | Waits for a software trigger (a *TRG, TRIG:TRAN, or IEEE 488.1 get—Group Execute Trigger—command), and then execute the power line abnormality simulation |

Settings are reset to default values when an \*RST command is sent.

**Example**

```
TRIG:SIM:SOUR BUS
```

Response: Characters

## TRIG:TRAN

Executes a software trigger on the TRANsient trigger subsystem.

### **Command**

```
TRIGger:TRANsient[:IMMediate]
```

## TRIG:TRAN:SOUR

Sets the condition (trigger source) for actually changing the settings after the TRANSient trigger subsystem receives an INIT:TRAN.

### Command

```
TRIGger:TRANsient:SOURce <character>
```

```
TRIGger:TRANsient:SOURce?
```

### Parameter

|        |           |   |
|--------|-----------|---|
| Value: | IMMediate | Change the settings immediately (default)   |
|        | BUS       | Change the settings when a software trigger (*TRG, TRIG:TRAN, IEEE488.1 get (Group Execute Trigger) is received |

Settings are reset to default values when an \*RST command is sent.

### Example

```
TRIG:TRAN:SOUR BUS
```

Response: Characters

# WAVE Command

## WAVE:DATA:ARB

Sets a user-defined waveform with block data at the waveform bank that you specify by its number.

If you overwrite the content of the bank selected with FUNC:BANK, the change is immediately applied.

This command is not valid when the synchronization function is in use (FREQ:SYNC ON).

### Command

WAVE:DATA:ARbitrary <NRf>,<block>

WAVE:DATA:ARbitrary

Parameter <NRf>

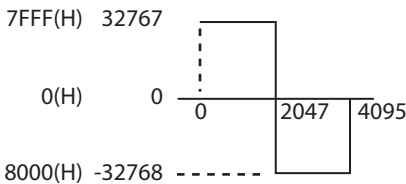
Value: 1 to 256 Waveform bank number

Parameter <block>

Value: User-defined waveform

Fixed to Big Endian. Size fixed to 4096 words (8192 octets)

16-bit integer array, each value ranging from -32768 to +32767



### Example

WAVE:DATA:ARB 5,#481927FFF7FFF....7FFF....80008000

Response: block

## WAVE:DATA:CLIP

Sets the crest factor of the peak clipped waveform at the waveform bank that you specify by its number.

This command is not valid when the synchronization function is in use (FREQ:SYNC ON).

### **Command**

```
WAVE:DATA:CLIP <BANK_NRf>,<PCLIP_numeric>
```

```
WAVE:DATA:CLIP? <BANK_NRf>
```

Parameter <BANK\_NRf>

Value: 1 to 256 Waveform bank number

Parameter <PCLIP\_numeric>

Value: 1.10 to 1.40 Crest factor of the peak clipped waveform (The default value is 1.40)

### **Example**

```
WAVE:DATA:CLIP 5,1.20
```

Response: NR3 format

## WAVE:DATA:IECP

Sets the clip factor of the flat curve waveform at the waveform bank that you specify by its number.

This command is not valid when the synchronization function is in use (FREQ:SYNC ON).

### **Command**

```
WAVE:DATA:IECPclip <BANK_NRf>,<PCLIP_numeric>
```

```
WAVE:DATA:IECPclip? <BANK_NRf>
```

Parameter <BANK\_NRf>

Value: 1 to 256 Waveform bank number

Parameter <PCLIP\_numeric>

Value: 0.4 to 1.0 Clip factor of the flat curve waveform (The default value is 1.0)

### **Example**

```
WAVE:DATA:IECP 5,0.8
```

Response: NR3

**WAVE:DATA:POIN**

Sets a user-defined waveform by specifying the waveform bank number and the point.

**Command**

```
WAVE:DATA:POINT <BANK_NRf>,<POINT_NRf>,<DATA_NRf>
```

```
WAVE:DATA:POINT? <BANK_NRf>,<POINT_NRf>
```

Parameter <BANK\_NRf>

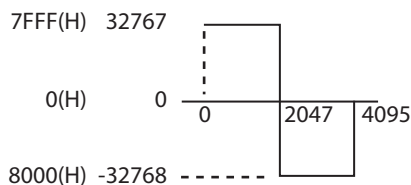
Value: 1 to 256 Waveform bank number

Parameter <POINT\_NRf>

Value: 0 to 4095 point

Parameter <DATA\_NRf>

Value: -32768 to +32767

**Example**

To set point 1023 of waveform bank number 2 to 32767

```
WAVE:DATA:POIN 2,1023,32767
```

Response: NR3

## WAVE:DATA:SIN

Sets the waveform bank that you specify by its number to sine wave.

This command is not valid when the synchronization function (FREQ:SYNC ON) is on.

### **Command**

```
WAVE:DATA:SINusoid <NR1>
```

### **Parameter**

Value: 1 to 256    that you want to execute

### **Example**

```
WAVE:DATA:SIN 5
```



**WAVE:DATA:TYPE**

Queries the waveform type at the waveform bank that you specify by its number.

**Command**

WAVE:DATA:TYPE? <NR1>

Response: character

Parameter <NRf>

Value: 1 to 256 Waveform bank number

Response value

|        |           |                       |
|--------|-----------|-----------------------|
| Value: | SINusoid  | Sine                  |
|        | CLIP      | Peak-clipped waveform |
|        | IECPclip  | Flat curve waveform   |
|        | ARBitrary | User-defined waveform |

# List of Errors

## Command errors

An error in the range [-199, -100] indicates that an IEEE 488.2 syntax error has been detected by the instrument's parser. The occurrence of any error in this class causes the Command Error bit (bit 5) in the event status register to be set.

| Error code | Error message description  |
|------------|--|
| -100       | Command error<br>Command error. Generic syntax error.  |
| -101       | Invalid character<br>An invalid character exists. A data element different than those allowed was recognized.  |
| -102       | Syntax error<br>Syntax error. An unrecognized command or data type was encountered.  |
| -103       | Invalid separator<br>Invalid separator The parser was expecting a separator and encountered an illegal character.                                    |
| -104       | Data type error<br>Data type error. The parser recognized a data element different than one allowed.   |
| -105       | GET not allowed<br>Get not allowed. A Group Execute Trigger was received in a program message.   |
| -108       | Parameter not allowed<br>Parameter not allowed More parameters were received than expected for the header.   |
| -109       | Missing parameter<br>Missing parameter Fewer parameters were received than required for the header.  |
| -110       | Command header error<br>Command header error. An error was detected in the header.   |
| -112       | Program mnemonic too long<br>Mnemonic too long. The number of characters in the command header exceeds 12 characters.                                |
| -113       | Undefined header<br>Undefined header. Inappropriate for the product.   |
| -114       | Header suffix out of range<br>Invalid suffix exists in the header.   |
| -115       | Unexpected number of parameters<br>Unexpected parameters were received in the header.  |
| -120       | Numeric data error<br>Numeric data error. Generated when parsing a data element which appears to be numeric, including the nondecimal numeric types. |
| -128       | Numeric data not allowed<br>Numeric data is not allowed.   |
| -130       | Suffix error<br>Suffix error. Generated when parsing a suffix.   |
| -131       | Invalid suffix<br>A suffix is invalid. The suffix does not follow the syntax, or the suffix is inappropriate for the product.                        |
| -134       | Suffix too long<br>Suffix too long. The suffix contains too many characters.   |
| -138       | Suffix not allowed<br>A suffix was encountered after a numeric parameter that does not allow suffixes.   |
| -140       | Character data error<br>Character data error. Generated when parsing a character data element.   |

| Error code | Error message description   |
|------------|---|
| -141       | Invalid character data<br>Either the character data element contains an invalid character, or the element is not valid. |
| -144       | Character data too Long<br>Character data too long. The character data element contains too many characters.            |
| -148       | Character data not allowed<br>Character data is not allowed.  |
| -150       | String data error<br>String data error. Generated when parsing a string data element.                                   |
| -151       | Invalid string data<br>Invalid string data.   |
| -158       | String data not allowed<br>String data is not allowed.  |
| -160       | Block data error<br>Block data error. Generated when parsing a block data element.                                      |
| -170       | Expression error<br>Expression error. Generated when parsing an expression data element.                                |
| -180       | Macro error<br>Generated when defining a macro or executing a macro.  |

## Execution errors

An error in the range [-299, -200] indicates that an error has been detected by the instrument's execution control block. The occurrence of any error in this class causes the Execution Error bit (bit 4) in the event status register to be set.

| Error code | Error message description   |
|------------|---|
| -200       | Execution error (generic)<br>Execution error. A generic product error.                                |
| -203       | Command protected<br>Password protected program or query command cannot be executed.                  |
| -210       | Trigger error<br>Trigger error.   |
| -211       | Trigger ignored<br>A trigger was received but ignored.  |
| -213       | Init ignored<br>A measurement initiate operation was ignored because measurement is in progress.      |
| -214       | Trigger deadlock<br>A deadlock occurred because a query was received before the software trigger.     |
| -220       | Parameter error<br>Invalid parameter.   |
| -221       | Settings conflict<br>A command was received that the product cannot execute in its present condition. |
| -222       | Data out of range<br>Parameter was out of range.  |
| -223       | Too much data<br>Too many parameters were received for the requirements.                              |
| -224       | Illegal parameter value<br>Received invalid parameter data.   |
| -230       | Data corrupt or stale<br>Received a data query before the measurement completed.                      |
| -241       | Hardware missing<br>Cannot be executed because the optional hardware is not installed.                |

## Product-specific errors

The occurrence of any error in this class causes the Device Dependent Error bit (bit 3) in the event status register to be set.

| Error code | Error message description        |
|------------|----------------------------------|
| -310       | System error                     |
| -311       | Memory error                     |
| -313       | Calibration memory lost          |
| -314       | Save/recall memory lost          |
| -315       | Configuration memory lost        |
| -330       | Self-test failed                 |
| -350       | Queue overflow                   |
| -360       | Communication error              |
| -362       | Framing error in program message |
| -363       | Input buffer overrun             |
| -365       | Time out error                   |

## Query errors

An error in the range [-499, -400] indicates that the output queue control of the instrument has detected a problem with the message exchange protocol described in IEEE 488.2, chapter 6. The occurrence of any error in this class causes the Query Error bit (bit 2) in the event status register to be set.

| Error code | Error message description                    |
|------------|--|
| -400       | Query error (generic)                        |
| -410       | Query INTERRUPTED                            |
| -420       | Query UNTERMINATED                           |
| -430       | Query DEADLOCKED                             |
| -440       | Query UNTERMINATED after indefinite response |

## Operation complete event errors

An error in the range [-899, -800] is used when the product wants to report an IEEE 488.2 operation complete event. This event occurs when the instrument's synchronization protocol, having been enabled by an \*OPC command, completes all selected pending operations.

The occurrence of any error in this class causes the Operation Complete bit (bit 0) in the event status register to be set.

| Error code | Error message description  |
|------------|--|
| -800       | Operation complete<br>All selected pending operations in accordance with the IEEE 488.2, 12.5.2 synchronization protocol have completed. |

## Product-dependent errors

The occurrence of any error in this class causes the Device Dependent Error bit (bit 3) in the event status register to be set.

### ■ Configuration change rejection errors

These errors occur when the specified configuration changes cannot be permitted.

| Error code | Error message   |
|------------|---|
| +100       | Setting change denied while OUTPut ON state                 |
| +101       | Setting change denied while TRANsient trigger in progress   |
| +102       | Setting change denied while SIMulation trigger in progress  |
| +103       | Setting change denied while PROGram trigger in progress     |
| +104       | Setting change denied while frequency synchronous.          |
| +105       | Setting change denied while BUSY                            |
| +106       | Setting change denied while SIMulation in progress          |
| +107       | Setting change denied while PROGram in progress             |
| +108       | Setting change denied while SLEEp mode                      |
| +109       | Setting change denied while EXT PROGram SOURce selected     |
| +110       | Setting change denied while INT+EXT PROGram SOURce selected |
| +111       | Setting change denied while V PROGram SOURce selected       |
| +112       | Setting change denied while SStart in progress              |
| +113       | Setting change denied while SStop in progress               |

### ■ Configuration conflict errors

These errors occur settings are in conflict with the existing settings.

| Error code | Error message                   |
|------------|---------------------------------|
| +201       | Conflicts with OUTPut OFF state |

| Error code | Error message  |
|------------|--|
| +202       | Conflicts with PROTection state                            |
| +203       | Conflicts with WIRing configuration                        |
| +204       | Conflicts with BUSY state                                  |
| +205       | Conflicts with CURRent PROTection LIMit selected           |
| +206       | Conflicts with CURRent PROTection TRIP selected            |
| +207       | Conflicts with SStart function enabled                     |
| +208       | Conflicts with Remote Inhibit operation                    |
| +209       | Conflicts with EXTeRnal PROGRaM SOURce selected            |
| +210       | Conflicts with INTeRnal PROGRaM SOURce selected            |
| +211       | Conflicts with unbalanced PHASe configuration              |
| +212       | Conflicts with unbalanced VOLTage configuration            |
| +213       | Conflicts with TRANsient trigger in progress               |
| +214       | Conflicts with SIMulation trigger in progress              |
| +215       | Conflicts with PROGRaM trigger in progress                 |
| +216       | Conflicts with external digital input                      |
| +217       | Conflicts with OUTPut IMPedance REAL                       |
| +218       | Conflicts with OUTPut IMPedance REACtive                   |
| +219       | Conflicts with SStop function enabled                      |
| +220       | Conflicts with V PROGRaM SOURce selected                   |
| +225       | Conflicts with Out of RANGE(VOLT)                          |
| +226       | Conflicts with Out of RANGE(FREQ)                          |
| +227       | Conflicts with Out of RANGE(IMP)                           |
| +228       | Conflicts with Out of RANGE(TIME)                          |
| +230       | Conflicts with existing AC VOLTage (IMMediate)             |
| +231       | Conflicts with existing AC VOLTage (TRIGgered)             |
| +232       | Conflicts with existing DC VOLTage (IMMediate)             |
| +233       | Conflicts with existing DC VOLTage (TRIGgered)             |
| +234       | Conflicts with existing AC+DC overlaid VOLTage (IMMediate) |
| +235       | Conflicts with existing AC+DC overlaid VOLTage (TRIGgered) |
| +236       | Overlaying too much AC+DC voltage                          |
| +237       | Conflicts with non-zero DC VOLTage                         |
| +238       | Conflicts with high-ranged T3 VOLTage                      |
| +239       | Conflicts with existing FREQuency (IMMediate)              |
| +240       | Conflicts with existing FREQuency (IMMediate)              |
| +241       | Conflicts with soft VOLTage LIMit settings                 |
| +242       | Conflicts with soft VOLTage OFFSet LIMit settings          |
| +243       | Conflicts with soft FREQuency LIMit settings               |
| +244       | Conflicts with TRIP in DISabled state                      |
| +245       | Conflicts with non-zero AC VOLTage                         |
| +250       | Conflicts with VOLTage COMPensate not in DISabled state    |
| +251       | Conflicts with HARD voltage compensation                   |
| +252       | Conflicts with SOFT voltage compensation                   |
| +253       | Conflicts with REG-ADJ voltage compensation                |
| +254       | Conflicts with CV RESPonse MEDium or FAST                  |
| +255       | Conflicts with FREQuency SYNChronize function enabled      |
| +256       | Conflicts with OUTPut IMPedance function enabled           |
| +257       | Conflicts with non-zero waveform BANK active               |

| Error code | Error message                      |
|------------|------------------------------------|
| +258       | Conflicts with PHASe ON            |
| +259       | Conflicts with PHASe OFF           |
| +260       | Conflicts with PHAS.CHG in PROGram |

## ■ Operation errors

These errors occur when invalid or incorrect settings are specified.

| Error code | Error message                   |
|------------|---------------------------------|
| +300       | Invalid phase number            |
| +301       | Invalid WAVE BANK name          |
| +302       | Name already used by other BANK |
| +303       | Block data is too long          |
| +304       | Block data is too short         |
| +306       | Channel list is forbidden       |
| +311       | Illegal PROGram nane            |
| +312       | PROGram nane already exists     |
| +313       | PROGram not found               |
| +314       | PROGram not selected            |
| +315       | PROGram not running             |
| +316       | Cannot delete selected PROGram  |
| +317       | Invalid STEP index              |
| +318       | Power saver unsupported         |

## ■ Security errors

| Error code | Error message           |
|------------|-------------------------|
| +501       | Wrong password          |
| +502       | Illegal password format |

## ■ Errors related to the self-test function

These errors occur as results of self-tests executed with \*TST? queries.

| Error code | Error message                        |
|------------|--------------------------------------|
| +901       | Detected empty power module slot     |
| +902       | Detected malfunctioning power module |
| +903       | Invalid master/slave configuration   |

## ■ EIOC errors

The +1000 error is an internal error of the product.

## Command processing time

A certain amount of time is required before the commands shown in the following table are received by the product.

The processing times shown here are standard values, not guaranteed values.

The processing times vary depending on the settings and the measurement conditions.

The values shown below do not include hardware response times.

| Command      | GPIB* <sup>1</sup><br>processing<br>time (ms) | USB<br>processing<br>time (ms) | RS232C* <sup>2</sup><br>processing<br>time (ms) | LAN* <sup>3</sup><br>processing<br>time (ms) | Description                               |
|--------------|---|--------------------------------|---|--|---|
| VOLT         | 7   | 7                              | 5   | 6  | Sets the voltage                          |
| MEAS:VOLT?   | 111   | 110                            | 100   | 120  | Queries the<br>measured output<br>voltage |
| FREQ         | 6   | 6                              | 6   | 6  | Sets the frequency                        |
| MEAS:CURREN? | 111   | 110                            | 100   | 120  | Queries the<br>measured output<br>current |
| *RST         | 600   | 500                            | 500   | 520  | Resets the device                         |

\*<sup>1</sup>: Using a USB-GPIB by National Instruments

\*<sup>2</sup>: Baud rate setting: 19200bps

\*<sup>3</sup>: 100BASE-TX Ethernet



# Tutorial

## Programming AC Output

---

### ■ Setting the AC voltage and frequency

The AC voltage and frequency are controlled by the VOLTage and FREQuency commands. First, set the voltage range.

VOLTage:RANGe 150 'Sets the voltage range to L

VOLTage 110 'Sets the AC voltage to 110 V

FREQuency 55 'Sets the frequency to 55 Hz

OUTPUt ON 'Turns the output on

### ■ Setting the AC voltage and frequency limits

The maximum AC voltage value varies depending on the voltage range setting.

Further, the maximum and minimum AC voltage and frequency values may vary depending on the limit settings. The AC voltage and frequency must be set within the range defined by the specified limits.

The voltage and frequency limits are safety interlock functions to avoid operation errors and programming errors. They are not output limit functions.

VOLTage:RANGe 150 'Sets the voltage range to L

VOLTage:LIMit:UPPer MAX 'Sets the upper voltage limit to the maximum

VOLTage:LIMit:LOWer MIN 'Sets the lower voltage limit to the minimum

VOLTage 110 'Sets the AC voltage to 110 V

FREQuency:LIMit:UPPer MAX 'Sets the upper frequency limit to the maximum

FREQuency:LIMit:LOWer MIN 'Sets the lower frequency limit to the minimum.

FREQuency 55 'Sets the frequency to 55 Hz

In the above example, the limits are set to the maximum so that any AC voltage and frequency within the range can be specified.

If the voltage range is set to L, the AC voltage cannot be set greater than 157.5 V.

### ■ Query

To query the maximum or minimum value of the AC voltage and frequency, include the MINimum or MAXimum parameter in the query.

VOLTage? MINimum  
VOLTage? MAXimum  
FREQuency? MINimum  
FREQuency? MAXimum

## Programming DC Output

---

### ■ Setting the DC voltage

The output voltage is controlled with the `VOLTage:OFFSet` command. First, set the voltage range.

`VOLTage:RANGe 150` 'Sets the voltage range to L

`VOLTage:OFFSet 40` 'Sets the DC voltage to 40 V

`OUTPut ON` 'Turns the output on

The voltage range cannot be changed when the output is on. The DC voltage can be changed when the output is on.

### ■ Setting the DC voltage limits

The maximum DC voltage value varies depending on the voltage range setting. Further, the maximum and minimum DC voltage values may vary depending on the limit settings.

`VOLTage:RANGe 150` 'Sets the voltage range to L

`VOLTage:OFFSet:LIMit:UPPer MAX` 'Sets the upper voltage limit to the maximum

`VOLTage:OFFSet:LIMit:LOWer MIN` 'Sets the lower voltage limit to the minimum

`VOLTage:OFFSet 40` 'Sets the DC voltage to 40 V

In the above example, the limits are set to the maximum so that any DC voltage within the range can be specified.

For the DC voltage, specify the voltage range using an AC voltage expression. The DC voltage setting range is  $\pm 219.0$  V for the L range and  $\pm 438.0$  V for the H range.

### ■ Query

To query the maximum or minimum value, include the `MINimum` or `MAXimum` parameter in the query as shown below.

`VOLTage:OFFSet? MINimum`

`VOLTage:OFFSet? MAXimum`

## Trigger Subsystem

---

This product has five different trigger subsystems.

- TRANSient

This subsystem is used to change the voltage and frequency settings.

- ACQUIRE

This subsystem is used to measure voltage, current, and power.

- SIMulation

This subsystem executes power line abnormality simulations.

- PROGRAM

This subsystem executes sequences.

This command cannot be executed simultaneously with the TRANSient, SIMulation, or PROGRAM subsystem.

The TRANSient, ACQuire, SIMulation, and PROGram trigger subsystems have three states (IDLE state, INITiated state, WTG state).

- IDLE state

When the product is turned on, all trigger subsystems are in the IDLE state. In this state, the trigger subsystem ignores all triggers. If you send any of the following commands, the trigger subsystem is switched to the IDLE state, regardless of its current state.

ABORt

\*RST

\*RCL

IEEE488.1 sdc (Selected Device Clear) or dcl (Device Clear)

- INITiated state

When you send the INIT command while the product is in the IDLE state, the trigger function begins operating, and the tester switches to the INITiated state.

If the trigger source is set to IMMEDIATE, the settings are changed immediately, or the measurement, power line abnormality simulation, or sequence starts immediately.

If the trigger source is set to BUS, the product switches to the WTG (Waiting for Trigger) state.

- WTG (Waiting for Trigger) state

If a trigger is received in the WTG state, the settings are changed, or the measurement, power line abnormality simulation, or sequence starts.

## Changing the output with triggers (TRANSient)

---

The TRANSient group is a trigger subsystem for changing settings. This subsystem is used to change the voltage and frequency settings.

### ■ Output change control

You can use the TRIGger:TRANSient subsystem to synchronize the changes in the output with triggers. This is useful when you want to synchronize the changes in the output to the operation of external devices, such as DC power supplies and electronic loads.

Use the VOLTage:TRIGgered and FREQuency:TRIGgered command to reserve trigger settings.

VOLTage 110 'Sets the voltage to 110 V

FREQuency 60 'Sets the frequency to 60 Hz

VOLTage:TRIGgered 100 'Sets the voltage that will be set when a trigger is received to 100 V

FREQuency:TRIGgered 50 'Sets the frequency that will be set when a trigger is received to 50 Hz

TRIGger:TRANSient:SOURce BUS 'Sets the trigger source to BUS

INITiate:TRANSient 'Initiates the TRANSient group (starts the trigger function)

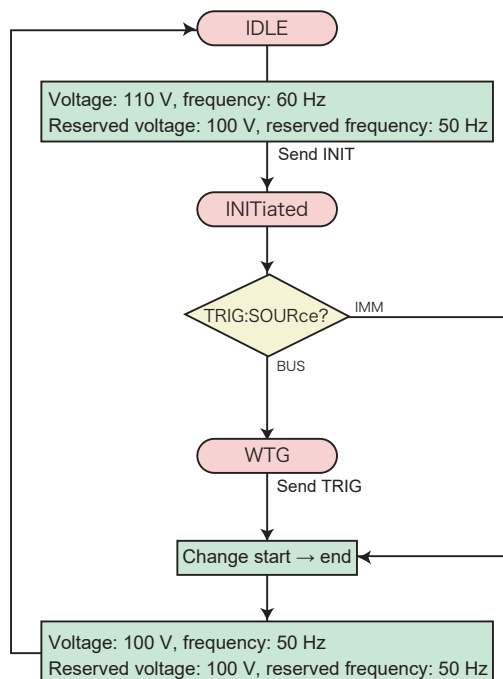
TRIGger:TRANSient 'Applies a software trigger to the TRANSient group

If you repeatedly change the output, a trigger error (-210) may occur.\* By using the \*OPC command, you can prevent this error.-> [“Waiting for Operation Complete”\(p. 332\)\(p. 332\)](#)

TRIGger:TRANSient;\*OPC 'Applies a trigger and waits until the output change is complete

### States

Trigger operation has three states: IDLE, INITiated, and WTG.



Use the TRIGger:TRANSient:SOURce command to set the trigger source to BUS or IMMEDIATE.

If the trigger source is set to IMMEDIATE, an INITiate command immediately executes changes and sets the voltage and frequency to their new values. If the trigger source is set to BUS, the trigger subsystem enters the WTG (Waiting For Trigger) state. When a software trigger is received (through the TRIGger:TRANSient:IMMEDIATE command or \*TRG command), the changes begin.

When the operation is complete, the trigger subsystem returns to the IDLE state again. If the ABORt command or an equivalent command is received instead of a trigger, the changes are canceled, and the trigger subsystem returns to the IDLE state.

The programmable parameters of the TRANSient group are AC voltage, DC voltage, and frequency. The current limit setting cannot be changed using triggers.

TRIGger:TRANSient:IMMEDIATE only applies a software trigger to the TRANSient group.

You can also use the \*TRG command or the IEEE488.1 get (Group Execute Trigger)

command for the same purpose. This command applies a software trigger to all trigger subsystems, if there are other trigger subsystems in the initiated state, their trigger operations will also be executed at the same time.

### How the product operates when triggers are used

When an ABOR command is received, INIT:TRAN is cancelled. The VOLT:TRIG value does not change.

The following table shows the responses when the voltage is set to 20 V (VOLT 20) and the voltage set by the trigger is 10 V (VOLT:TRIG 10).

|  | Response |               |
|--|----------|---------------|
|  | VOLT?    | VOLT:TRIG?    |
| Immediately after the command is set                   | 20 V     | 10 V          |
| After the trigger is sent                              | 10 V     | 10 V          |
| After a *RST is sent                                   | 0 V      | 0 V           |
| Voltage change VOLT 30 sent before the trigger is sent | 30 V     | 30 V (cancel) |



## Measurement (ACQuire)

---

The ACQuire group is the measurement trigger subsystem. This subsystem is used to measure voltage, current, and power.

There are easy measurements and advanced measurements.

### ■ Simple measurement

This product has functions for returning the measured voltage, current, and power. The easiest measurement method is using the MEASure command.

The MEASure command starts a new measurement. Because this query starts a new measurement each time that it is sent, you cannot use it to synchronize the measurement of multiple items. The measurement method explained in “Advanced measurement” allows you to separate the measurement start operation and the data query operation.

### Measuring the voltage and current

MEASure:VOLTage:ACDC? 'Queries the rms voltage

MEASure:CURREnt:ACDC? 'Queries the rms current

MEASure:VOLTage:DC? 'Queries the average voltage

MEASure:CURREnt:DC? 'Queries the average current

The current measurement function has the following additional parameters.

MEASure:CURREnt:AMPLitude:MAXimum? 'Queries the peak current

MEASure:CURREnt:AMPLitude:MAXimum:HOLD? 'Queries the peak current (the held value)

MEASure:CURREnt:CREStfactor? 'Queries crest factor

MEASure:CURREnt:AMPLitude:MAXimum:HOLD queries the maximum peak current after the product is turned on or after the peak current is cleared explicitly. Use the SENSE:CURREnt:PEAK:CLEar command to clear the peak current (the held value).

SENSe:CURREnt:PEAK:CLEar

The \*RST or \*RCL command does not clear the peak current (the held value).

Normally, it takes approximately 110 ms for a single measurement to complete. If you send the MEASure query multiple times, data acquisition will take a long time. If you want to acquire the data of multiple parameters, measure using the method explained

in “Advanced measurement.”

This product also supports the READ command, which starts a new measurement and queries the data. READ and MEASure are aliases. They operate exactly the same.

### **Power measurement**

MEASure:POWer:ACDC? 'Queries the AC power

MEASure:POWer:ACDC:APParent? 'Queries the apparent power

MEASure:POWer:ACDC:REACtive? 'Queries the reactive power

MEASure:POWer:ACDC:PFACTOR? 'Queries the power factor

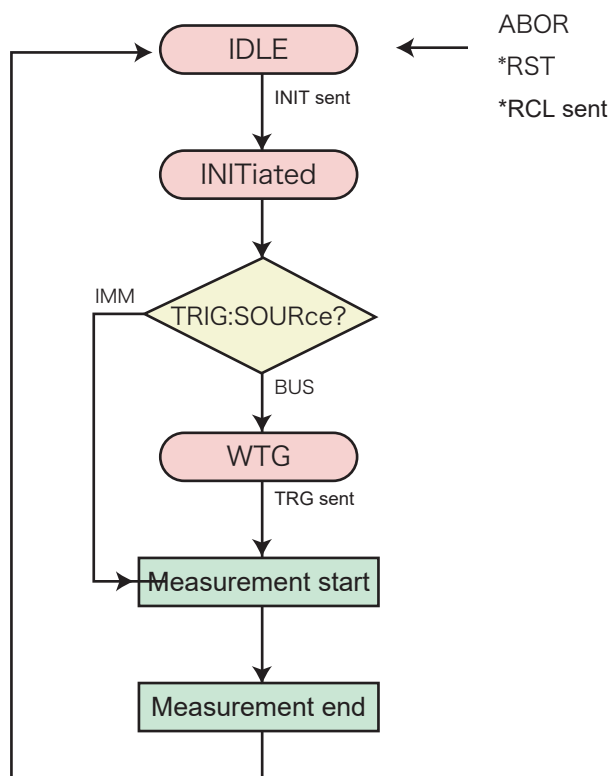
MEASure:POWer:DC? 'Queries the average power

### **■ Advanced measurement**

In advanced measurement, you can separate and control the starting of measurement and the referencing of data.

### **States**

Trigger operation has three states: IDLE, INITiated, and WTG.



## Measuring voltage, current, and power

To start a new measurement, set the trigger source to IMMEDIATE, and then use the INITiate command.

TRIGger:ACQuire:SOURce IMMEDIATE 'Sets the trigger source to IMMEDIATE

INITiate:ACQuire 'Initiates the ACQuire group

To use software triggers to start the measurement on the ACQuire group, change the trigger source to BUS.

TRIGger:ACQuire:SOURce BUS 'Sets the trigger source to BUS

INITiate:ACQuire 'Initiates the ACQuire group

TRIGger:ACQuire 'Applies a software trigger to the ACQuire group

When the measurement finishes, you can use the FETCh query to retrieve the measured data.

FETCh:VOLTage:ACDC? 'Queries the rms voltage

FETCh:CURRent:ACDC? 'Queries the rms current

FETCh:POWer:ACDC? 'Queries the power

FETCh:POWer:ACDC:APParent? 'Queries the apparent power

If you send a FETCh command before the measurement is complete, correct measurement data will not be obtained.\* By using the \*OPC command, you can obtain correct measurement data.-> [“Waiting for Operation Complete”\(p. 332\)\(p. 332\)](#)

INITiate:ACQuire;\*OPC 'Initiates the ACQuire group and waits for the measurement to complete.

Use the TRIGger:ACQuire:SOURce command to set the trigger source to BUS or IMMEDIATE. INITiate:ACQuire pulls the TRIGger subsystem out of the IDLE state and starts (initiates) the trigger function.

If the trigger source is set to IMMEDIATE, the measurement starts immediately. If the trigger source is set to BUS, the TRIGger subsystem enters the WTG (Waiting For Trigger) state. When a software trigger is received (through the TRIGger:ACQuire command or \*TRG command), the measurement starts. When the measurement finishes, the TRIGger subsystem enters the IDLE state again. If the ABORt command or an equivalent command is received instead of a trigger, the measurement is canceled, and the TRIGger subsystem returns to the IDLE state.

The ABORt command and IEEE488.1 sdc/dcl commands abort measurements that are in progress. These commands do not invalidate measured data that has already been retrieved. On the other hand, the \*RST and \*RCL common commands not only abort a measurement that is in progress but also invalidate the acquired measured data. If you send \*RST;:FETC:VOLT?, an error will occur because there is no measured data that the FETCh query can retrieve and there is no new measurement that is going to be performed.

The difference between the MEASure (or READ) command and the FETCh command is as follows. The MEASure command starts a new measurement and then queries the measured data. The FETCh command queries the measured data without first starting a new measurement. The valid measurement parameters are exactly the same between MEASure and FETCh.

## Power line abnormality simulations (SIMulation)

---

The SIMulation group is used to perform power line abnormality simulations.

### ■ Configuring parameters

First, configure the parameters.

SIMulation:T1:PHASe:STATe OFF 'Sets T1 using time

SIMulation:T5:CYCLe:STATe OFF 'Sets T5 using time

SIMulation:T1:TIME 5MS 'Sets the voltage regulation starting time

SIMulation:T2:TIME 10000MS 'Sets slope time 1

SIMulation:T3:TIME 100MS 'Sets the voltage regulation time

SIMulation:T3:VOLTage 50V 'Sets the regulated voltage

SIMulation:T4:TIME 10000MS 'Sets slope time 2

SIMulation:T5:TIME 10000MS 'Sets the return time

Next, set the number of repetitions.

SIMulation:REPeat:COUNT 5 'Sets the number of repetitions

After you have finished configuring the settings, execute the power line abnormality simulation.

### ■ Execute power line abnormality simulations.

Send the following command to execute a power line abnormality simulation.

OUTP ON

SIM:STAT RUN

You can query the execution status of power line abnormality simulations.

SIM:EXEC?

If the execution has been stopped, "STOP" is returned. If the sequence is being executed, "RUN" and the present repetition number is returned.

To stop a power line abnormality simulation that is being executed, send the following command.

SIM:STAT STOP

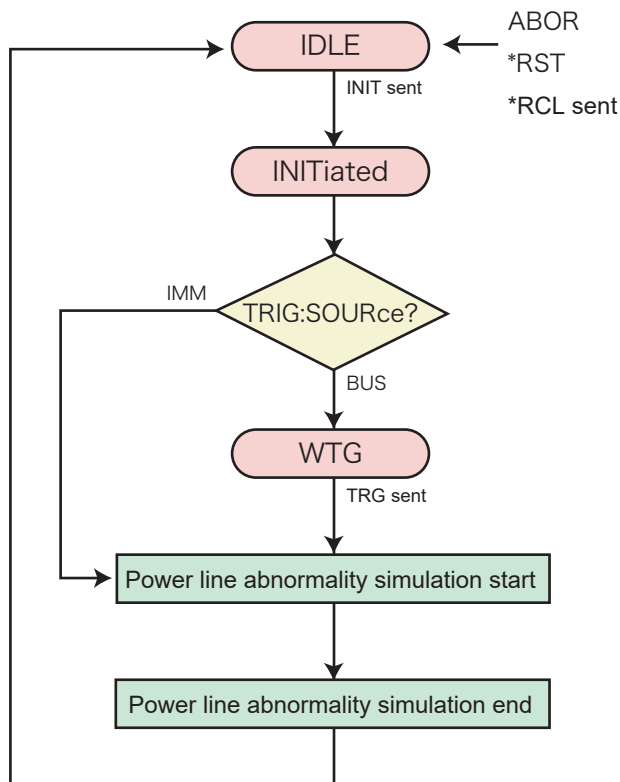
You can use triggers to execute power line abnormality simulations.

### Execution using triggers

You can use triggers to synchronize power line abnormality simulations by using the

TRIGger:SIMulation subsystem.

Trigger operation has three states: IDLE, INITiated, and WTG.



To start a simulation immediately, set the trigger source to IMMEDIATE, and then use the INITiate command.

TRIGger:SIMulation:SOURce IMMEDIATE 'Sets the trigger source to IMM

INITiate:SIMulation 'Initiates the SIMulation group. The power line abnormality simulation begins.

To use software triggers to start the power line abnormality simulation on the SIMulation group, change the trigger source to BUS.

TRIGger:SIMulation:SOURce BUS 'Sets the trigger source to BUS

INITiate:SIMulation 'Initiates the SIMulation group

TRIGger:SIMulation 'Applies a software trigger to the SIMulation group. The power line abnormality simulation begins.

Use the TRIGger:SIMulation:SOURce command to set the trigger source to BUS or IMMEDIATE. INITiate:SIMulation pulls the TRIGger subsystem out of the IDLE state and starts (initiates) the trigger function.

If the trigger source is set to IMMEDIATE, the power line abnormality simulation starts immediately. If the trigger source is set to BUS, the TRIGger subsystem enters the WTG (Waiting For Trigger) state. When a software trigger is received (through the TRIGger:SIMulation command or \*TRG command), the power line abnormality simulation starts. When the simulation finishes, the TRIGger subsystem enters the IDLE state again. If the ABORt command or an equivalent command is received in the WTG state or when a simulation is being executed, the simulation is canceled, and the TRIGger subsystem returns to the IDLE state.

Send the \*RST command to reset all the parameters of the power line abnormality simulation.

TRIGger:SIMulation:IMMEDIATE only applies a software trigger to the SIMulation group.

You can also use the \*TRG command or the IEEE488.1 get (Group Execute Trigger) command for the same purpose. This command applies a software trigger to all trigger subsystems, if there are other trigger subsystems in the initiated state, their trigger operations will also be executed at the same time.



## Sequence Operation (PROGram)

---

The PROGram group runs sequences.

### ■ Configuring step and sequence settings

First, configure the steps.

Use the PROG:EDIT command to set the number of the step that you want to configure, frequency signal change, frequency, AC voltage signal change, AC voltage, DC voltage signal change, DC voltage, step execution time, waveform bank number, status output, trigger output, trigger input, and whether output is ON or OFF.

```
PROG:Edit 1,OFF,50HZ,OFF,100V,OFF,0V,10S,0,OFF,ON,OFF,ON
```

```
PROG:Edit 2,OFF,60HZ,ON,200V,OFF,0V,1MIN,1,OFF,OFF,OFF,ON
```

```
PROG:Edit 3,ON,400HZ,ON,230V,OFF,50V,1HR,2,ON,OFF,OFF,ON
```

Next, configure the sequence conditions.

```
PROG:STEP:START 1 'Sets the starting step number
```

```
PROG:STEP:END 3 'Sets the ending step number
```

```
PROG:LOOP 10 'Sets the number of repetitions
```

After you have finished configuring the sequence, execute it.

### ■ Executing sequences

Send the following command to execute a sequence.

```
PROG:STAT RUN
```

You can query the execution state of the sequence.

```
PROG:EXEC?
```

The execution state—STOP (stopped), RUN (running), or PAUSE (paused), elapsed step time, present repetition number, and step number are returned as a comma-separated list.

To stop a sequence that is being executed, send the following command.

```
PROG:STAT STOP
```

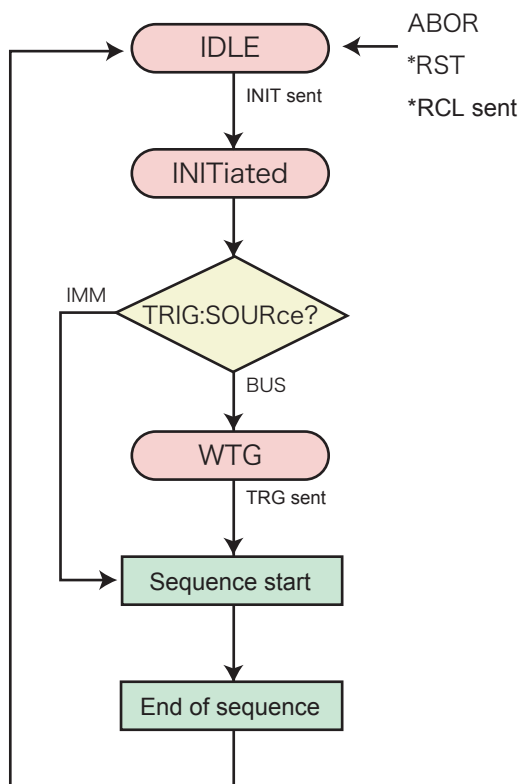
You can use triggers to execute sequences.

### Execution using triggers

You can use triggers to synchronize sequences by using the TRIGger:PROGram sub-

system.

A sequence has three states: IDLE, INITiated, and WTG.



To start a sequence immediately, set the trigger source to IMMEDIATE, and then use the INITiate command.

TRIGger:PROGrama:SOURce IMMEDIATE 'Sets the trigger source to IMM

INITiate:PROGrama 'Initiates the PROGrama group. The sequence begins.

To use software triggers to start the sequence on the PROGrama group, change the trigger source to BUS.

TRIGger:PROGrama:SOURce BUS 'Sets the trigger source to BUS

INITiate:PROGrama 'Initiates the PROGrama group

TRIGger:PROGrama 'Applies a software trigger to the PROGrama group. The sequence begins.

Use the TRIGger:PROGrama:SOURce command to set the trigger source to BUS or

IMMediate. INITiate:PROGrama pulls the TRIGger subsystem out of the IDLE state and starts (initiates) the trigger function.

If the trigger source is set to IMMediate, the sequence starts immediately. If the trigger source is set to BUS, the TRIGger subsystem enters the WTG (Waiting For Trigger) state. When a software trigger is received (through the TRIGger:PROGrama command or \*TRG command), the sequence starts. When the sequence finishes, the TRIGger subsystem enters the IDLE state again. If the ABORt command or an equivalent command is received in the WTG state or when a sequence is being executed, the sequence is canceled, and the TRIGger subsystem returns to the IDLE state.

TRIGger:PROGrama:IMMediate only applies a software trigger to the PROGrama group.

You can also use the \*TRG command or the IEEE488.1 get (Group Execute Trigger) command for the same purpose. This command applies a software trigger to all trigger subsystems, if there are other trigger subsystems in the initiated state, their trigger operations will also be executed at the same time.

When all the PROGrama processes are complete, the product's settings are those of the last step.

If the output is on in the last step of the sequence, the output will remain on when the sequence is completed.

## Waiting for Operation Complete

---

The \*OPC command has a function for waiting for operations to complete. Operation complete means that there are no operations that are waiting for a response from the PCR-WE. Measurement completion requires about 110 ms. The PCR-WE is not in the operation complete state while a measurement is ongoing. When the measurement completes, if there are no other operations waiting to be completed, the PCR-WE enters the operation complete state.

When an \*OPC command is received, the product transitions to the Operation Complete Command Active State (OCAS). If a measurement is completed and there are no operations standing by, the product returns to the Operation Complete Command Idle State (OCIS) and sets the OPC bit (bit 0) of the event status register to TRUE (1). This information can be determined by checking the OPC bit (bit 0) of the \*ESR? query.

Next, we will show an example that starts a new measurement and sends an \*OPC command. Because the event status enable register and service request enable register are configured to generate a service request (SRQ) in response to an operation complete event, an SRQ is generated when a measurement is completed. The SRQ function cannot be used if you are using the RS232 interface.

```
*ESE 1;*SRE 32;*CLS;:INITiate:IMMediate:ACQuire;*OPC  
<Generates a service request>
```

If you use the \*OPC? query command in place of the \*OPC command, the product transitions to the Operation Complete Query Active State (OQAS). If a measurement is completed and there are no operations standing by, the product returns to the Operation Complete Query Idle State (OQIS) and sets response data "1" (in NR1 format) in the output queue.

```
INITiate:IMMediate:ACQuire;*OPC?  
<Reads the response>
```

At power-on, if you send an IEEE488 sdc/dcl, \*RST, or \*RCL, this product switches to the OCIS and OQIS states.

## Status Monitoring

---

The product has two mandatory SCPI standard registers, STATus:OPERation and STATus:QUEStionable, in addition to the IEEE488.2 standard registers.

### ■ Register basics

All SCPI registers have a standard architecture that uses events/filters. CONDition, EVENT, and ENABLE and optionally PTRansition and NTRansition can be used. CONDition and EVENT are read-only registers working as status indicators. ENABLE, PTRansition and NTRansition are read-write registers working as event and summary filters.

### ■ STATus:OPERation

The OPERATION Status register is used to record events and notifications that occur during normal operations.

To check whether CV output is being performed, check the CV bit (bit 8) of the STATus:OPERation register.

STATus:OPERation? 'Checks whether the CV bit is set.

### ■ STATus:QUEStionable

The QUEStionable Status register is used to record events and notifications that occur during abnormal operations.

To check whether a protection function has been activated, check the OV bit (bit 0) of the STATus:QUEStionable register.

STATus:QUEStionable? 'Checks whether the OV bit is set.

### ■ Monitoring status for single-phase three-wire output and three-phase output

#### STATus:OPERation

The OPERATION Status register is used to record events and notifications that occur during normal operations.

To check whether CV output is being performed, check the CV bit (bit 8) of the STATus:OPERation:INSTRument:ISUMmary{1|2|3} subregister.

Of the parameters {1|2|3}, 1 represents U phase, 2 represents V phase, and 3 represents W phase.

STAT:OPER:INST:ISUM2? 'Check whether the CV bit of the V phase is set.

### STATus:QUESTionable

The QUESTionable Status register is used to record events and notifications that occur during abnormal operations.

To check whether the overvoltage protection function has been activated, check the OV bit (bit 0) of the STATus:QUESTionable register.

STAT:QUES? 'Check whether the OV bit is set.

Even if bit 0 is true, you cannot tell on which phase the overvoltage protection function has been activated. To check which phase is operating abnormally, check the STATus:QUESTionable:INSTrument subregister.

STAT:QUES:INST? 'Check which phase is operating abnormally.

All channels whose corresponding bits are true are operating abnormally. You can determine how the specified phase is operating abnormally by checking the STATus:QUESTionable:INSTrument:ISUMmary{1|2|3} subregister of the channel.

Of the parameters {1|2|3}, 1 represents U phase, 2 represents V phase, and 3 represents W phase.

STAT:OPER:INST:ISUM2? 'Check whether the OV bit of the V phase is set.

### ■ PON (Power ON) bit

The PON bit (bit 7) of the event status register is always set when the product is turned on. To generate a power-on SRQ to track power failures and power supply line errors, use PON as follows.

1

#### **Set \*PSC (Power-on Status Clear ) to 0 (or OFF).**

Enable the backup functions for event status enable register and service request enable register settings. (\*PSC 0)

2

#### **Set the PON bit (bit 7) of the event status enable register.**

This enables the transmission of power-on events to the higher layer. (\*ESE 128)

3

#### **Set the ESB bit (bit 5) of the status byte enable register.**

This enables the generation of SRQs based on standard events. (\*SRE 32)

\*PSC 0;\*ESE 128;\*SRE 32

When you use the RS232C interface, the PON bit cannot be assigned to a service request because SRQs are not generated.

When you use the USB or LAN (VXI-11/HiSLIP) interface, even though the SRQ function itself is supported by the communication protocol, a connection lost error occurs in the VISA I/O session immediately before the power-on event. It appears that handling PON events would be difficult.

## Error Checking

---

### ■ Error/event queue

The SCPI specifications define a standard error reporting scheme, Error/Event Queue. This is a FIFO (First In First Out) queue, which records errors and events. The maximum number of errors/events that the product can record is 16. Each error/event can be read with the SYSTem:ERRor query.

SYSTem:ERRor?

The response to this query contains a numeric part (error/event number) and a textual description, such as:

-222,"Data out of range"

The error/event queue becomes empty when the \*CLS common command is sent, when the last item in the queue is read, and when the product is turned on. When the error/event queue is empty, the query returns the following:

0,"No error"

### ■ Displaying communication errors

The product has a debug trace function.

The product can display the oldest item among the errors and events (if there are errors or events). This is convenient for debugging remote control.

When an error or event item is displayed on the panel, the normal voltmeter and ammeter are void.

When the error/event queue is empty, the debug trace function does not display communication errors.\* When you send a CLS command, the communication error display clears.

In local mode, the debug trace function is temporarily disabled.



## Visual Basic 2017

### ■ Configuring a project

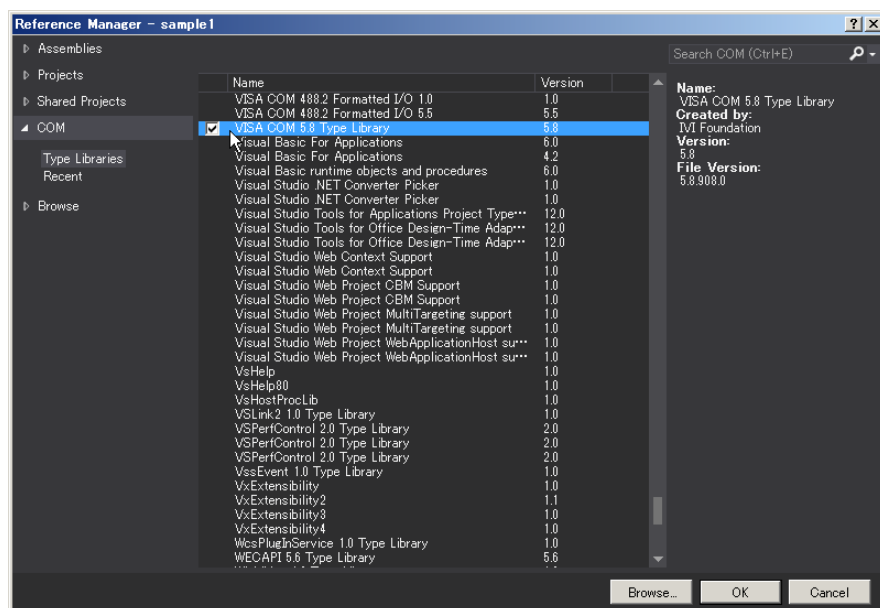
First, add the communication middleware (VISA library) to the project.

Click References on the Project menu to open the Reference Manager window.

On the navigation pane, click COM and then Type Libraries.

From the list in the center of the window, select “VISA COM \*. \* Type Library” (where \*. \* is the VISA library version number), and select the check box.

Click OK to close the dialog box.



## ■ Communicating via GPIB, RS232C, USB, or LAN

### Opening VISA

Before you can use the VISA library to communicate with GPIB, RS232C, USB, and LAN devices, you have to open VISA. Specify an I/O resource to open VISA.

Example: Opening VISA when using USB on the PCR-WE

```
Set rm = CreateObject("VISA.GlobalRM")  
  
Set msg = rm.Open("USB::0x0B3E::0x104E::00000001::INSTR", NO_LOCK, 0, "")
```

"USB::0x0B3E::0x104E::00000001::INSTR" is an I/O resource.

The I/O resource syntax is shown below. The parts surrounded by square brackets ([ ]) can be omitted. Enter the appropriate values in the parts written in *italics*.

|                    |          |   |
|--------------------|----------|---|
| GPIB               |          | GPIB[ <i>board</i> ][:PrimaryAddress][:SecondaryAddress][:INSTR]<br>Example: Measuring instrument with primary address 3 connected to GPIB0<br>GPIB0::3::INSTR  |
| Serial<br>(RS232C) |          | ASRL[ <i>board</i> ][:INSTR]<br>Example: A measuring instrument connected to serial port COM1<br>ASRL1::INSTR   |
| USB                |          | USB[ <i>board</i> ][:VendorID::ProductID::SerialNumber][:InterfaceNumber][:INSTR]<br>Example: A USBTMC measuring instrument whose vendor ID (VID) is 2878, product ID (PID) is 4174, and serial number is 00000001<br>USB0::0x0B3E::0x104E::00000001::INSTR                     |
| LAN <sup>*1</sup>  | VXI-11   | TCPIP[ <i>board</i> ][:hostname][:inst0][:INSTR]<br>Example: Measuring instrument whose IP address (hostname) is 169.254.7.8<br>TCPIP::169.254.7.8::INSTR<br>You can also specify the host name for the hostname parameter.   |
|                    | HiSLIP   | TCPIP[ <i>board</i> ][:hostname::hislip0][:INSTR]<br>Example: Measuring instrument whose IP address (hostname) is 169.254.7.8<br>TCPIP::169.254.7.8::hislip0::INSTR<br>You can also specify the host name for the hostname parameter.   |
|                    | SCPI-RAW | TCPIP[ <i>board</i> ][:hostname::portno::SOCKET]<br>Example: Measuring instrument whose IP address (hostname) is 169.254.7.8 (the product's port number is fixed to 5025)<br>TCPIP::169.254.7.8::5025::SOCKET<br>You can also specify the host name for the hostname parameter. |

<sup>\*1</sup>: The hostname must be a valid mDNS hostname (a Bonjour hostname that ends in ".local") or a DNS hostname that is managed by an external DNS server (a full-qualified domain name—

FQDN). If you are using an mDNS hostname, Apple Bonjour (alternatively, iTunes or Safari) must be installed on your PC.

In VISA, you can use aliases for I/O resources.

If you use an alias for an I/O resource, even if the alias name is hard-coded in the application, the I/O resource name can still be changed to an appropriate value when the application runs.

Example: Using an alias (MYDEV1) for an I/O resource

```
Set msg = rm.Open("MYDEV1", NO_LOCK, 0, "")
```

When you use aliases, specify the actual I/O resources through an external configuration table or similar tool. Refer to the VISA manual.

## Controlling the instrument

Next, we will use commands such as read and write commands to control the instrument. You must include line-feed codes in the command strings.

Examples:

```
msg.WriteString ("VOLT 110" & vbLF) 'Sets the AC voltage to 110 V
msg.WriteString ("FREQ 60" & vbLF)  'Sets the frequency to 60.0 Hz
msg.WriteString ("OUTP 1" & vbLF)   'Turns the output on
```

## Closing VISA

Finally, close VISA.

In a sequence of operations, you only have to open and close VISA once.

```
msg.Close
```

## ■ Sample program

```
Imports Ivi.Visa.Interop
```

```
Public Class Form1
```

```
Dim rm As ResourceManager
```

```
Dim msg As IMessage
```

```
Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
rm = CreateObject("VISA.GlobalRM")
```

```
msg = rm.Open("USB0::0x0B3E::0x104E::00000001::INSTR", AccessMode.NO_LOCK, 0, "")
```

'Version using a VISA alias

```
'msg = rm.Open("MYDEV1", AccessMode.NO_LOCK, 0, "")
```

'Version using LAN (SCPI-RAW)

```
'msg = rm.Open("TCP/IP::169.254.7.8::5025::SOCKET", AccessMode.NO_LOCK, 0, "")
```

'Version using GPIB

```
'msg = rm.Open("GPIB0::1::INSTR", AccessMode.NO_LOCK, 0, "")
```

```
msg.TerminationCharacterEnabled = True
```

End Sub

'Query the ID

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
msg.WriteString("SYST:COMM:RLST REM" & vbCrLf)
```

```
msg.WriteString("**IDN?" & vbCrLf)
```

```
TextBox1.Text = msg.ReadString(256)
```

End Sub

'Set the voltage, frequency, and output.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
```

```
msg.WriteString("OUTP 0" & vbCrLf)
```

```
msg.WriteString("VOLT 110" & vbCrLf)
```

```
msg.WriteString("FREQ 60" & vbCrLf)
```

```
msg.WriteString("OUTP 1" & vbCrLf)
```

End Sub

'Queries the measured voltage

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
```

```
msg.WriteString("MEAS:VOLT:AC?" & vbCrLf)
```

```
TextBox1.Text = msg.ReadString(256)
```

End Sub

```
Private Sub Form1_Disposed(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Disposed
```

```
msg.Close()
```

End Sub

End Class

If you find any misplaced or missing pages in the manuals, they will be replaced. If the manual gets lost or soiled, a new copy can be provided for a fee. In either case, please contact your Kikusui agent or distributor. At that time, inform your agent or distributor of the "Part No." written on the front cover of this manual.

Every effort has been made to ensure the accuracy of this manual. However, if you have any questions or find any errors or omissions, please contact your Kikusui agent or distributor.

After you have finished reading this manual, store it so that you can use it for reference at any time.

## **KIKUSUI ELECTRONICS CORP.**

---

1-1-3 Higashiyamata, Tsuzuki-ku, Yokohama,  
224-0023, Japan

Tel: +81-45-482-6353

Fax: +81-45-482-6261

**[www.kikusui.co.jp/en](http://www.kikusui.co.jp/en)**

