

ADT773/783/793 SCPI Commands Set

1 Commands Instruction

SCPI means Standard Commands for Programmable Instruments, which defines a set of standard syntax and commands for controlling programmable instruments. It communicates with the instrument through ASCII string format. A command generally consists of a series of keywords, some of which also need have parameters. In the protocol, the command is specified in the following form: * IDN?. In use, it can be the full name or only abbreviations containing uppercase letters. Usually, instrument commands can be divided into controlling commands and query commands. The controlling commands donot have return values, its result can be checked by sending the command SYSTem: ERRor?. The query commands have returned value, and it is also ASCII string.

(1)Format

Each command includes two parts: **keyword** and **parameter**, different keywords are separated by a colon ':', followed by optional parameters. If there is a "?" after the command, it indicates it is query command. The keyword and the first parameter should be separated by a space.

For example:

Command **CHANnel1: BWLimit 20M**. The first keyword is CHANnel, and the second keyword is BWLimit. The keywords are separated by ":", with 20M being the parameter and separated from the keywords by space.

Query command **CHANnel1: BWLimit?** The question mark "?" indicates a query.

(2) About the symbol

The following symbols are not sent with the command.

a. Vertical line |

A vertical line is used to separate multiple parameters, and one of the parameters must be selected when using the command.

b. Square brackets []

The content in square brackets can be omitted.

c. Triangle brackets <>

The parameters in the triangle brackets must be replaced with a valid value.

(3) About the abbreviation

All commands are not case sensitive, you can use all uppercase or lowercase. But if you want to abbreviate, you must enter all uppercase letters in the command format.

For example: **CHANnel: Value?** Can it be abbreviated as **CHAN: Value?**

(4) Terminator

The SCPI command must include a command terminator, which can be one of the follows (excluding double quotation marks): "\r\n", "\r", "\n" or "\0". In some serial communication software, select the option "send line break" means that the

software will automatically send the command terminator.

2 Commands instruction

2.1 IEEE488.2 common commands

| NO. | Command | Explanation | Parameters | Returning values | Example |
|-----|---------|--|------------|-----------------------|--|
| 1 | *CLS | This command eliminates the following registers; a. Standard event register b. Querying event register c. Operating event register d. Status byte register f. Error queue | - | - | *CLS |
| 2 | *IDN? | Device identification query, the returned data is divided into 4 parts: 1. manufacturer 2. model 3. product serial number 4. device ID and software version | - | Device identification | Send: *IDN Return: ADDITEL,,123456789,P25d&MPC V2.0.0.6 |
| 3 | *RST | Main software reset | - | - | *RST |

2.2 Pressure commands

| NO. | Command | Explanation | Parameters | Returning values | Example |
|-----|-------------------------------------|----------------------------|--|-----------------------------|----------------------------------|
| 1. | PRESsure:MODUle:UNIT? <ModuleID> | Read pressure module units | <ModuleID> number of the designated module (1 2 4 6) | Pressure unit of the module | Send: PRESsure:MODUle:UNIT? 1 |

| | | | | | |
|----|---|-------------------------------------|--|---|---|
| | | | 1: control module 2: internal high range module 4: external module 6: barometric module | | Return:MPa |
| 2. | PRESsure:MODule:UNIT <ModuleID>,<UNIT> | Set pressure unit of the module | <ModuleID> number of the designated module (1 2 4 6) 1: control module 2: internal high range module 4: external module 6: barometric module <UNIT> Below are valid units: Pa hPa kPa MPa mbar bar psi mmH2O@4°C cmH2O@20°C inH2O@4°C inH2O@20°C kgf/cm2 torr ftH2O@4°C inH2O@20°C inHg@0°C mmHg@0°C The list also contains custom unit, read custom unit by sending command: PRESsure:MODule:UNIT:LIST? | None | PRESsure:MODule:UNIT 2,Pa |
| 3. | PRESsure:MODule:UNIT:LIST ? | Read the current pressure unit list | None | the list of current pressure unit names, including the unit, availability status, and custom unit status, Pa&1&0, indicates that the unit is available and not a custom unit | Send: PRESsure:MODule:UNIT:LIST? Return: Pa&0&0,hPa&1&0,kPa&1&0,MPa &1&0,psi&1&0,User1&1&1,User2& |

| | | | | | |
|----|---|------------------------|---|---|--|
| | | | | USER1&0&1 indicates that the unit is not available and is a custom unit | 1&1, User3&1&1, User4&1&1, User5&1&1 |
| 4. | PRESsure:MODule:RESolutio n? <ModuleID> | Read module resolution | <ModuleID> module number (1 2 5 6) 1: control module 2: internal high range module 4: external module 6: barometric module | Current resolution | Send: PRESsure:MODule:RESolutio n? 1 Return:5 |
| 5. | PRESsure:MODule:RESolutio n <ModuleID>,<Value> | Set module resolution | <ModuleID> module number (1 2 3 4 6) 1: control module 2: internal high range module 3: internal low range module 4: external module 6: barometric module <Value> Resolution to be set, valid value 5,6,7 | None | PRESsure:MODule:RESolutio n 2,6 |
| 6. | PRESsure:MODule:ZERO <ModuleID> | Zero the module | <ModuleID> module number (1 2 4 6) 1: control module 2: internal high range module 4: external module 6: barometric module | None | PRESsure:MODule:ZERO 2 |
| 7. | PRESsure:MODule:ZERO:CA Ncel <ModuleID> | Cannel zero the module | <ModuleID> module number (1 2 4 6) 1: control module 2: internal high range module | None | PRESsure:MODule:CANCel:ZERO 5 |

| | | | | | |
|-----|--------------------------------------|---|--|---|---|
| | | | 4: external module 6: barometric module | | |
| 8. | PRESsure:MODule:PTYPE? <ModuleID> | Read pressure module pressure type | <ModuleID> module number (1 2 4) 1: control module 2: internal high range module 4: external module | The pressure type of current module: G:gauge A:absolute D:differential | Read internal high range module pressure type Send: PRESsure:MODule:PTYPE? 2 Return:A |
| 9. | PRESsure:MODule:RANGE? <ModuleID> | Read the module range | <ModuleID> module number (1 2 5 6) 1: control module 2: internal high range module 4: external module 6: barometric module | The range and unit of the current module, high and low range is separated by "~", unit is out of the "()". Multiple range is separated by "," | Read internal module range Send: PRESsure:MODule:RANGE? 2 Return:(0 ~ 25) MPa If the multiple ranges are: 0~70MPa 和 0 ~ 25MPa Send: PRESsure:MODule:RANGE? 2 Return:(0 ~ 70) MPa,(0 ~ 25) MPa |
| 10. | PRESsure:RANGE:LIST? | Read the multiple range list, the range index is not start from 0 | None | Get a list of current ranges, including index (the first digit represents modules <2 high range module, 3 low range module, 4 external module>, second digit represents module range numbers), ranges, and units. High and low ranges are separated by '~', and ranges are separated by '&' | Send:PRESsure:RANGE:LIST? Return:21,(0 ~ 70) MPa&22,(0 ~ 25) MPa |
| 11. | PRESsure:RANGE:INDEX? | Read the range index of the controller | None | The index of the current range in the list: | Send: PRESsure:RANGE:INDEX? |

| | | | | | |
|-----|--|---------------------------------------|---|---|--|
| | | | | (the first digit represents modules <2 high range module, 3 low range module, 4 external module>, second digit represents module range numbers, it is not start from 0) | Return:21 |
| 12. | PRESsure:RANGe:INDEX <INDEX> | Set the range index of the controller | <INDEX> The index to be set The value should be within the range list: the first digit represents modules <2 high range module, 3 low range module, 4 external module>, second digit represents module range numbers, it is not start from 0, needs get the data from the list | None | Set the current range of internal module as the range of index 0 in the list Send: PRESsure:RANGe:INDEX 21 |
| 13. | PRESsure:MODUle:MULTiRange? <ModuleID> | Read if the module has multiple range | <ModuleID> module number (1 2 5 6) 1: control module 2: internal high range module 4: external module 6: barometric module | 0:single range 1:multiple range | Read if the internal module has multiple range Send: PRESsure::MODUle:MULTi:RANGe? 2 Return:1 |
| 14. | PRESsure:RANGe:MODE? | Read the current range mode | None | 0>manual range 1:auto range | Send: PRESsure:RANGe:MODE? Return:0 |
| 15. | PRESsure:RANGe:MODE <Mode> | Set the current range mode | <Mode> 0>manual range 1:auto range | | Set the range to manual Send: PRESsure:RANGe:MODE 0 |

| | | | | | |
|-----|---------------------------------------|---|---|--|--|
| | | | | | Return:None |
| 16. | PRESsure:MODule:ONLine? <ModuleID> | Read pressure module online status | <ModuleID> module number (1 2 4 6) 1: control module 2: internal high range module 4: external module 6: barometric module | 0: module is offline 1: module is online | Send: PRESsure:MODule:ONLine? 2 Return:1 |
| 17. | PRESsure:MODule:INFO? <ModuleID> | Read information of the pressure module, which is a comprehensive command. If the module has multiple ranges, the ranges are separated by '&' | <ModuleID> module number (1 2 3 4 6) 1: control module 2: internal high range module 3: internal low range module 4: external module 6: barometric module | Module information, separated by (,.) <ID> serial number <Range>range <Type>pressure type G:gauge A:absolute D:differential <Version>version <Accuracy>accuracy | Read internal module information Send: PRESsure:MODule:INFO? 2 Return:DPSE022480040,(0 ~ 25) MPa,G,DPS-EX V00.00.00.15,6 For multiple ranges module, ranges are separated by '&' Send: PRESsure:MODule:INFO? 2 Return:DPSE022480040,(0 ~ 70) MPa&(0 ~ 25) MPa,G,DPS-EX V00.00.00.15,6 |
| 18. | PRESsure:MODule:FILTer? <ModuleID> | Read module filter information | <ModuleID> module number (1 2 3 4) 1: control module 2: internal high range module 3: internal low range module 4: external module | <Enable> 0:off, 1:on <FilterType> 0:first-order filter 1:average filter <Value>filter parameter: If FilterType is 0, this means the filter coefficient, range is 0-1; If FilterType is 1, this means the sampling time, range is 1-20(s); | Read control module's filter information Send: PRESsure:MODule:FILTer? 1 Return:1,0,0.5 |

| | | | | | |
|-----|--|---|--|--|--|
| 19. | PRESsure:MODUle:FILTer <ModuleID>,<Enable>,<FilterType>,<Value> | Set module filter information | <ModuleID> module number (1 2 3 4) 1: control module 2: internal high range module 3: internal low range module 4: external module <Enable> 0:off, 1:on <FilterType> 0:first-order filter 1:average filter <Value>filter parameter: If FilterType is 0, this means the filter coefficient, range is 0-1; If FilterType is 1, this means the sampling time, range is 1-20(s) | | Send: PRESsure:MODUle:FILTer 1,1,0,0.5 Return: none |
| 20. | PRESsure:MODUle:VALUes? | Read pressure value of all the modules. Before using this command, please ensure that the "DIAGnostic: LIVEdata: START" command has been used to enable the real-time data acquisition mode of the control board. If the real-time data acquisition mode of the control board is not enabled, the PML and PMH pressure values are not guaranteed to be correct 2. When the command is no longer | | ADT793 <PML>,<Unit>&<PMH>,<Unit>&<PctI>,<Unit>&<Pin>,<Unit>&<Acc>,<Unit>&<Baro>,<Unit>&<ExtPM>,<Unit> <PML> internal low range pressure <PMH> internal high range pressure <PctI> PctI pressure <Pin> pump source pressure <Acc> accumulator pressure <Baro> barometric module pressure <ExtPM> external module pressure ADT773/783 | Send: PRESsure:MODUle:VALUes? Return: 0.86974597,MPa&13326.722,MPa &0.0018362,MPa&3.075833,MPa & 2.065,MPa&100.132,MPa&-0.054,MPa If external module is not connected, the value for the external module will be empty(behind the last ",") For barometric module, it is same. |

| | | | | | |
|--|--|---|--|---|--|
| | | used, please use "DIAGnostic: LIVEdata: STOP" to turn off the real-time data acquisition mode of the control board to avoid affecting its operational efficiency 3. When the controller is not connected to the barometric module or external module, this bit is empty, as shown in the following example | | <PML>,<Unit>&<PMH>,<Unit>&<S1>,<Unit>&<S2>,<Unit>&<Baro>,<Unit>&<ExtPM>,<Unit> <PML> internal low range pressure <PMH> internal high range pressure <S1> pressure supply pressure <S2> vacuum supply pressure <Baro> barometric pressure <ExtPM> external module pressure | |
|--|--|---|--|---|--|

2.3 Pressure measurement

| NO. | Command | Explanation | Parameters | Returning values | Example |
|-----|--------------------------------------|--|--|---|---|
| 1. | PRESsure:MODUle:MEASure ? <ModuleID> | Read the measured value of the pressure module | <ModuleID> module number (1 2 3 4 6) 1: selected module 2: internal high range module 3: internal low range module 4: external module 6: barometric module | Measured value and unit, separated by "," | Read the measured value of internal module Send: PRESsure:MODUle:MEASure? 2 Return:0.566, MPa |

2.4 Pressure output

| NO. | Command | Explanation | Parameters | Returning values | Example |
|-----|--------------------------|---|------------|---|--|
| 1. | PRESsure? | Read the current output pressure | | Pressure value and unit | Send: PRESsure? Return: 0.0003, MPa |
| 2. | PRESsure:MODUle:CONTRol? | Read current pressure module controlling status | | The controlling status, one of the following three: | Send: PRESsure:MODUle:MODE? |

| | | | | | |
|----|------------------------------------|---|---|--|---|
| | | | | VENT MEASURE CONTROL | Return: VENT |
| 3. | PRESsure:MODule:CONTRol <STATE> | Set current pressure module controlling status | <STATE>: The controlling status, one of the following three: VENT MEASURE CONTROL | | Set the status to control Send: PRESsure:MODule:CONTRol CONTROL Return: None |
| 4. | PRESsure:MODE? | Read pressure module controlling status | None | The controlling status, one of the following three: VENT MEASURE CONTROL | Send: PRESsure:MODE? Return: VENT |
| 5. | PRESsure:MODE <STATE> | Set pressure module controlling status | <STATE>: The controlling status, one of the following three: VENT MEASURE CONTROL | None | Set the status of current module to Control Send: PRESsure:MODE CONTROL Return: none Set the status of current module to Measure Send: PRESsure:MODE 1 Return: none |
| 6. | PRESsure:TARGet:RANGe? | Read the range of target value | | Low limit, high limit, unit format: <LOW>, <HIGH>, <UNIT> | Send: PRESsure:TARGet:RANGe? Return: 0,73.5,MPa |
| 7. | PRESsure:TARGet? | Read the target value | | Target value and unit | Send: PRESsure:TARGet? Return: 0.10000,MPa |

| | | | | | |
|-----|--------------------------------|---|--|--|---|
| 8. | PRESsure:TARGet <Value> | Set the target value, note that the value should within the range | <Value>target value | | Set the target pressure at 100 Send: PRESsure:TARGet 100 Return: None |
| 9. | PRESsure:RANGe? | Read the current range | | The range index (first digit is the module number: 2 internal high range, 3 internal low range, 4 external module; second digit is range number starts from 1) and range | Send: PRESsure:RANGe? Return: 21,(0 ~ 25) MPa |
| 10. | PRESsure:MODUle? | Read the current control module | | 1: no control module 2: internal high range 3: internal low range 4: external module | Send: PRESsure:MODUle? Return: 2 |
| 11. | PRESsure:MODUle <ModuleID> | Set the control module | <ModuleID> module ID(2 3 4) 2: internal high range 3: internal low range 4: external module | | Send: PRESsure:MODUle 2 Return: None |
| 12. | PRESsure:Vent? | Read the vent pressure | | Vent pressure | Send: PRESsure:Vent? Return: 0.1,MPa |
| 13. | PRESsure:Vent <Value> | Set the vent pressure | <Value> vent value to be set | | Send: PRESsure:Vent 0.2 Return: None |
| 14. | PRESsure:PLIMit:ENABLE? | Read the status of setpoint limit | | Status of setpoint limit 0: disable 1: enable | Send: PRESsure:PLIMit:ENABLE Return: 0 |
| 15. | PRESsure:PLIMit:ENABLE <Value> | Set the status of setpoint limit | <Value> 0: disable | | Send: PRESsure:PLIMit:ENABLE 1 |

| | | | | | |
|-----|---------------------------|---|--|---|---|
| | | | 1: enable | | Return: None |
| 16. | PRESsure:PLIMit? | Read the setpoint limit range | | Lower limit, upper limit and unit | Send: PRESsure:PLIMit? Return: 0.005,70,MPa |
| 17. | PRESsure:PLIMit <Lv>,<Uv> | Set the setpoint limit range | <Lv> lower limit of the setpoint limit <Uv> upper limit of the setpoint limit | | Set the setpoint limit range as 0.005,70 Send: PRESsure:PLIMit 0.005,70 Return: None |
| 18. | PRESsure:TYPE? | Read the current pressure type | | <Type> pressure type G: gauge A: absolute <EnableSwitch> switchable between gauge and absolute or not? 0: not support 1: support | Send: PRESsure:TYPE? Return: G,0 |
| 19. | PRESsure:TYPE <Type> | Set the current pressure type. If it doesn't support switch pressure type, it will return an error. | <Type> G: gauge A: absolute | | Set pressure type as absolute: Send: PRESsure:TYPE A Return: None |
| 20. | PRESsure:STEP? | Read manual step value | | Step value | Send: PRESsure:STEP? Return: 0.5 |
| 21. | PRESsure:STEP <Value> | Set manual step value | <Value>step value to be set | | Set step value as 0.5 Send: PRESsure:STEP 0.5 Return: None |
| 22. | PRESsure:STEP:UP | Perform upward step | | | Send: PRESsure:STEP:UP Return: None |
| 23. | PRESsure:STEP:DOWN | Perform downward step | | | Send: PRESsure:STEP:DOWN |

| | | | | | |
|-----|------------------------|----------------------------------|--|---|---|
| | | | | | Return: None |
| 24. | PRESsure:CONTRol:INFO? | Read current control information | | <p>The control information, items are separated by (,)</p> <p><Value> real time pressure value</p> <p><Target> target value</p> <p><Unit> unit</p> <p><Range> range</p> <p><Type> pressure type</p> <p>G: gauge</p> <p>A: absolute</p> <p>D: differential</p> <p><IsStable> stabilized or not</p> <p>0: not stabilized</p> <p>1: stabilized</p> <p><State>control status</p> <p>VENT</p> <p>MEASURE</p> <p>CONTROL</p> <p><ExtendInfo> extend ports information</p> <p>one byte data lenfth, indicates 8 IO statuses, as below:</p> <p>bit7~0:</p> <p>CPS, DRV1, DRV2, DO1,DO2, DO3, DC24, Switch</p> | <p>Send:</p> <p>PRESsure:CONTRol:INFO?</p> <p>Return:0.0267,2.0,MPa,(0 ~ 25) MPa,G,0,MEASURE,30</p> |
| 25. | PRESsure:CONTRol:MODE? | Read control mode | | <p>Control mode</p> <p>0: fast</p> | <p>Send:</p> <p>PRESsure:CONTRol:MODE?</p> |

| | | | | | |
|-----|---|--|--|---|---|
| | | | | 1: standard 2: custom | Return: 0 |
| 26. | PRESsure:CONTRol:MODE <Value> | Set control mode | <Value> 0: fast 1: standard 2: custom | | Send: PRESsure:CONTRol:MODE 1 Return: None |
| 27. | PRESsure:CONTRol:SLEWrate ? | Read control slew rate | | <type> ,<value>,<unit> <type>: slew rate type; 0=unlimited 1=limited <value>: slew rate value; When the type is 0 without limit, the value is MAX, and when the slew rate is 1, the limit value will be returned <unit>: Unit of slew rate | 1 read slew rate, no limit Send: PRESsure:CONTRol:SLEWrate? Return: 0,MAX,MPa 2 Read slew rate, the limit value is 5MPa Send: PRESsure:CONTRol:SLEWrate? Return:1,5,MPa |
| 28. | PRESsure:CONTRol:SLEWrate :MAX | Set slew rate as unlimited, only available when control mode is custom | | | Send: PRESsure:CONTRol:SLEWrate:MA X Return: None |
| 29. | PRESsure:CONTRol:SLEWrate :LIMIt <Value> | Set a slew rate value, the unit is current pressure unit, only available when control mode is custom | <Value>: slew rate value | | Send: PRESsure:CONTRol:SLEWrate:LI MIt 5 Return: None |
| 30. | PRESsure:CONTRol:STABility? | Read stability setting | | <type>,<value>,<unit>,<second> <type>:stability setting, 0 percentage, 1 stability value <value>: when type=0, it is | Send: PRESsure:CONTRol:STABility? Return: 0,0.05,%FS,15 |

| | | | | | |
|-----|---|---|--|---|---|
| | | | | percentage; when type=1, it is stability value <unit>: when type=0, it is %FS, when type=1, it is stability value <second> stability time | |
| 31. | PRESsure:CONTRol:STABility <type>,<value>,<second> | Set stability, only available when control mode is custom | <type>:stability setting, 0 percentage, 1 stability value <value>: when type=0, it is percentage; when type=1, it is stability value <second> stability time | | PRESsure:CONTRol:STABility 0,0.05,30 |
| 32. | PRESsure:CONTRol:HEIGHt:CORRection? | Read head correction information | <Enable> 0:off, 1:on <UnitType> 0:imperial 1:metric <Height>head correction, metric unit (cm), imperial unit (in) <Density>media density, metric unit (kg/m³), imperial unit (lb/ft³) <Gravity>gravity acceleration metric unit (m/s²), imperial unit (ft/s²) <Temp>temperature, unit is °C | | Send: PRESsure:MODule:HEIGHt:CORRection? Return:0,1,10,1.293,9.8,25 |
| 33. | PRESsure:CONTRol:HEIGHt:CORRection <Enable>,<UnitType>,<Height>,<Density>,<Gravity>,<Temp> | Set head correction information | <Enable> 0:off, 1:on <UnitType> 0:imperial 1:metric <Height>head correction, metric unit (cm), imperial unit (in) <Density>media density, metric unit (kg/m³), imperial unit (lb/ft³) <Gravity>gravity acceleration metric | | PRESsure:MODule:HEIGHt:CORRection 0,1,10,1.293,9.8,25 |

| | | | | | |
|-----|--|--|---|---|--|
| | | | unit (m/s ²), imperial unit (ft/s ²) <Temp>temperature, unit is °C | | |
| 34. | PRESsure:CONTRol:TARE? | Read tare information | | <Enable><Enable> 0:off, 1:on <Value> tare value | Read internal high range module tare value: Send: PRESsure:MODule:TARE? Return:0,0 |
| 35. | PRESsure:CONTRol:TARE <Enable>,<Tare> | Set tare information of designated module | <Enable><Enable> 0:off, 1:on <Value> tare value to be set | | PRESsure:MODule:TARE 1,1 |
| 36. | PRESsure:SWITCh:TYPE? | Read pressure switch type | | 0: mechanical 1:NPN 2:PNP | PRESsure:SWITCh:TYPE? |
| 37. | PRESsure:SWITCh:TYPE <value> | Set pressure switch type | 0: mechanical 1:NPN 2:PNP | | PRESsure:SWITCh:TYPE 1 |
| 38. | PRESsure:SWITCh:VALUe? | Read pressure switch action value | | Close value, pressure unit & open value, pressure unit | Send: PRESsure:SWITCh:VALUe? Return: 20,MPa&18,MPa |
| 39. | PRESsure:SWITCh:VALUe:RE SEt | Reset pressure switch action value | | | PRESsure:SWITCh:VALUe:RESEt |
| 40. | PRESsure:EXTEnd:INTERface :STATe? | Read extend interface status | | CPS,DRV1,DRV2,DO1,DO2,DO3, DC24,Switch 0 inactivated 1 activated | Send: PRESsure:EXTEnd:INTERface:ST ATe? Return: 1,0,0,0,0,0,1,0 |
| 41. | PRESsure:EXTEnd:INTERface :MODE? <type> | Read extend interface output mode | <type> 0:CPS 1:DVR1 2:DVR2 3:DO1 | <mode>&<mode1>,<mode2>,<mode3> Returns a list of current output mode and available output modes, current output mode and available | Send: PRESsure:EXTEnd:INTERface:MO DE? 0 Return: 1&0,1 |

| | | | | | |
|-----|---|---|--|--|--|
| | | | 4:DO2 5:DO3 | output mode are separated by the "&" , and different available output modes are separated by a comma; List of output modes: 0:manual 1:CPS 2:vacuum pump 3:pressure pump 4:remote mode 5: status mode | |
| 42. | PRESsure:EXTEnd:INTERface :MODE <type>,<mode> | Set extend interface output mode | <type> 0:CPS 1:DVR1 2:DVR2 3:DO1 4:DO2 5:DO3 <mode> 0:manual 1:CPS 2:vacuum pump 3:pressure pump 4:remote mode 5: status mode | | Set CPS manual output Send: PRESsure:EXTEnd:INTERface:MODE 0,0 |
| 43. | PRESsure:EXTEnd:INTERface :Remote <type>,<value> | Set extend interface remote mode in on or off | <type> 1:DVR1 2:DVR2 | | Set DVR1 remote mode status Send: PRESsure:EXTEnd:INTERface:Re |

| | | | | | |
|-----|--------------------------------------|----------------------------------|---------------------------------|----------------------------|---|
| | | | <value> 0: off 1: on | | mote 1,1 |
| 44. | PRESsure:AZERo? | Read auto zero status | | 0: off 1: on | Send: PRESsure:AZERo? Return: 1 |
| 45. | PRESsure:AZERo <value> | Set auto zero status | 0: off 1: on | | PRESsure:AZERo1 |
| 46. | PRESsure:ZERO:POINT:STRAtegy? | Read zero point control strategy | | 0 vent 1 control | Send: PRESsure:ZERO:POINT:STRAtegy ? Return: 1 |
| 47. | PRESsure:ZERO:POINT:STRAtegy <value> | Set zero point control strategy | <value>: 0 vent 1 control | | PRESsure:ZERO:POINT:STRAtegy 1 |
| 48. | PRESsure:STABLE? | Read pressure stability status | None | 0= not stable 1= stable | Send: PRESsure:STABLE? Return: 1 |

2.5 Interface

| NO. | Command | Explanation | Parameters | Returning values | Example |
|-----|--------------|---|------------|--------------------------|---------------------------------|
| 1. | SYSTem:HOME | Return to home menu. When using this command, please make sure the unit is idle without running any task or test. | | | |
| 2. | SYSTem:LOCK? | Read the screen lock status | | 0: unlocked 1: locked | Send: SYSTem:LOCK? Return: 1 |

| | | | | | |
|----|---------------------|----------------------------|---|--|--|
| 3. | SYSTem:LOCK <State> | Set the screen lock status | <State> set screen lock or unlock 0: unlocked 1: locked | | Set the screen as locked Send:SYSTem:LOCK 1 Return: None |
|----|---------------------|----------------------------|---|--|--|

2.6 Communication

| NO. | Command | Explanation | Parameters | Returning values | Example |
|-----|------------------------------|--|--------------------------------|----------------------|---|
| 1. | SYSTem:WLAN:STATe? | Read WLAN status | | 0:off 1:on | Send: SYSTem:WLAN:STATe? Return: 1 |
| 2. | SYSTem:WLAN:STATe <State> | Set WLAN status | <State> 0:off 1:on | | Set WLAN as off Send: SYSTem:WLAN:STATe 0 Return: None |
| 3. | SYSTem:WLAN:ADDRes? | Read WLAN IP address | | WLAN IP address | Send: SYSTem:WLAN:ADDRes? Return: 192.168.61.127 |
| 4. | SYSTem:WLAN:ADDRes <IP> | Set WLAN IP address, available only when DHCP is off. | <IP> IP address | | Send: SYSTem:WLAN:ADDRes 192.168.1.23 Return: None |
| 5. | SYSTem:WLAN:MASK? | Read WLAN IP subnet mask | | WLAN IP subnet mask | Send: SYSTem:WLAN:MASK? Return: 255.255.254.0 |
| 6. | SYSTem:WLAN:MASK <Mask> | Set WLAN IP subnet mask, available only when DHCP is off | <State> WLAN IP subnet mask | | |
| 7. | SYSTem:WLAN:GATeway? | Read WLAN gateway address | | WLAN gateway address | Send: SYSTem:WLAN:GATeway? Return: 192.168.60.1 |

| | | | | | |
|-----|--|--------------------------|--|---|--|
| 8. | SYSTem:WLAN:GATeway <Gateway> | Set WLAN gateway address | <Gateway> WLAN gateway address | | Send: SYSTem:WLAN:GATeway 192.168.60.1 Return: None |
| 9. | SYSTem:WLAN:DHCP? | Read WLAN DHCP status | | 0:off 1:on | Send:SYSTem:WLAN:DHCP? Return: 1 |
| 10. | SYSTem:WLAN:DHCP <DHCP> | Set WLAN DHCP status | <DHCP> WLAN DHCP status 0:off 1:on | | Send:SYSTem:WLAN:DHCP 0 Return: None |
| 11. | SYSTem:WLAN:MAC? | Read WLAN MAC address | | MAC address | Send:SYSTem:WLAN:MAC? Return:00:C1:40:76:10:99 |
| 12. | SYSTem:WLAN:SSID? [<ALL>] | Read WLAN SSID | <ALL> Optional: If use ALL, it returns all connectable hotspot names If no parameter: it returns the name of the hotspot that is currently connected When there is no connection, and does not return when connected | The WLAN SSID currently connected, or all connectable hotspot names | Send:SYSTem:WLAN:SSID? Return: ConST If wifi is not connected, adding the 'ALL' parameter and it will return the list of wifi hotspots, separated by ','. If wifi is connected, the parameter will be ignored and the current connected wifi name will be returned. Send:SYSTem:WLAN:SSID? ALL Return: ConST,ESP_844851,CONSTRD,RD TEST,ESP_627851 |
| 13. | SYSTem:WLAN:CONNect <Name>,[<Pswd>] | Connect with hotspot | <Name> Hotspot name <Pswd> | | Send: SYSTem:WLAN:CONNect ConST,123456789 |

| | | | | | |
|-----|---------------------------------|---|--|--------------------------|--|
| | | | Hotspot password, optional, if no password or the hotspot was connected before, this parameter can be ignored. | | Return: if no password or the hotspot was connected before, then: Send: SYSTem:WLAN:CONnect ConST Return: None |
| 14. | SYSTem:WLAN:DISConnect | Disconnect with hotspot | | | |
| 15. | SYSTem:ETHernet:ADDRess? | Read Ethernet IP address | | Ethernet IP address | |
| 16. | SYSTem:ETHernet:ADDRess <IP> | Set Ethernet IP address. If the set IP address is the IP address of the network currently connected to the SCPI, it will cause the SCPI to disconnect, and connecting again requires configuring a new IP address | <IP> IP address | | Set Ethernet IP address as 192.168.1.23 Send: SYSTem:ETHernet:ADDRess 192.168.1.23 Return: None |
| 17. | SYSTem:ETHernet:MASK? | Read Ethernet subnet mask | Ethernet subnet mask | | Send: SYSTem:ETHernet:MASK? Return: 255.255.254.0 |
| 18. | SYSTem:ETHernet:MASK <Mask> | Set Ethernet subnet mask | <Mask> Ethernet subnet mask to be set | | Send: SYSTem:ETHernet:MASK 255.255.254.0 Return: None |
| 19. | SYSTem:ETHernet:GATeway? | Read Ethernet gateway address | | Ethernet gateway address | Send: SYSTem:ETHernet:GATeway? Return: 192.168.40.1 |

| | | | | | |
|-----|--------------------------------------|--------------------------------------|--|--|--|
| 20. | SYSTem:ETHeRnet:GATeway <Gateway> | Set Ethernet gateway address | <Gateway> Ethernet gateway address to be set | | Send: SYSTem:ETHeRnet:GATeway 192.168.40.1 Return: None |
| 21. | SYSTem:ETHeRnet:DHCP? | Read Ethernet DHCP status | | 0:off 1:on | |
| 22. | SYSTem:ETHeRnet:DHCP <DHCP> | Set Ethernet DHCP status | < DHCP > 0:off 1:on | | Send: SYSTem:ETHeRnet:DHCP 0 Return: None |
| 23. | SYSTem:ETHeRnet:MAC? | Read Ethernet MAC address | | Ethernet MAC address | Send: SYSTem:ETHeRnet:MAC? Return: 00:07:32:A9:ED:CF |
| 24. | SYSTem:RS232:Info? | Read RS232 communication information | | The current RS232 parameters, separated by “,” <BaudRate>values can be 9600/19200/38400/57600/115200 <DataBits> 5/6/7/8 <StopBits> None, One,Two, OnePointFive <Parity>parity bit None Odd Even Mark | Send:SYSTem:RS232:Info? Return:9600,8,One,None |

| | | | | | |
|-----|--|-------------------------------------|--|--|--|
| 25. | SYSTem:RS232:Info <BaudRate>,<DataBits>,<StopBits>,<Parity> | Set RS232 communication information | <BaudRate>values can be 9600/19200/38400/57600/115200 <DataBits> 5/6/7/8 <StopBits> None, One,Two, OnePointFive <Parity>parity bit None, Odd, Even, Mark | | Send:SYSTem:RS232:Info 115200,8,One,None Return:None |
|-----|--|-------------------------------------|--|--|--|