



## CALYS 150 / CALYS 1500

### Multifunction Process Calibrator



### SCPI Commands

Version	Date	Changes
1.0	25/03/2013	Original version
1.1	15/10/2015	Update document ref DE/15/160
1.2	17/12/2015	Ohm simulation : range 1mA 4mA added
1.3	31/03/2016	Integrating new calibrator (CALYS 1500). Various typographical corrections.

# 1. General Protocol

## 1.1 Communication format

Transmission format: 8 bits, 1 stop, No parity, no flow control  
 Baud rate: 115200

Commands format: type IEEE 488-2, 'SCPI-like'

- Last character command line is '\n' (decimal code 10, hexadecimal 0x0A). This end of line character can be preceded or followed by '\r' (decimal code 13, hexadecimal 0x0D) which is ignored.
- It can have one or more commands separated by character ';'.
- Each command has a header followed by one or more spaces and 0 or several arguments, separated by character ','.
- Command header can be built by one or more keywords separated by character ':', without space. Succession of these keywords are defining a tree diagram allowing classifying all the commands in subsets.
- All command headers asking for a response are finished by character '?'. This character belongs to the header (no space required)
- When wrong command is sent, instrument does not reply (for commands requiring a response).
- Instrument can propose a table of the last 5 error codes, managed in a dedicated FIFO. Command « ERR? » allows the extraction of the oldest error code. « \*CLS » erases the FIFO content.
- The keywords constituting a control header are shaped 'short' of 4 characters or less and form a 'long' generally comprehensible reading. In the description of the following keywords, shape 'short' is indicated in capital letters, shape 'long' is the set of upper and lower case characters. The command line can use either one or the other form. For example, to place the instrument in remote control, the control can be either "REM" or "REMOTE" (but not "REMO" nor "REMOT")
- In the list of commands, the first part of keywords, written in capital letters, is mandatory. The lower case is optional. Commands or command keywords can be entered in upper or lower case, in their short or long form. However, the mix of upper / lower case in the same keyword is not accepted.
- Keywords in brackets [ ] are optional.
- Keywords in braces { } are required.
- Required separation spaces characters are represented by a symbol '␣' (not to be confused with the underscore-character '\_').
- Adding spaces '␣' before / after required separator characters ',' (comma) to improve the clarity of the entry is permitted (these non-required spaces are simply ignored by the command interpreter).

### Examples (with replies, if relevant):

<pre>REM\n *IDN?\n  AOIP_SAS,CALYS1500,1234,A00\r\n SENS:VOLT:RANG 100MV\n SENS:FUNC VOLT\n MEAS:VOLT?\n  34.8492,mV\r\n  LOC\n</pre>	<p>Place instrument as remote use (lock keypad)</p> <p>Instrument identification request. The command is finished by '?', therefore a response is needed.</p> <p>Identification reply from CALYS 1500 (Serial Nr. 1234, vA.00).</p> <p>Define range of function VOLT</p> <p>Select function VOLT</p> <p>Read request measurement VOLT function. As the command ends with '?', it implies a reply.</p> <p>Read result function measurement VOLT: 34.8492 measured value in Anglo-Saxon digital format (decimal point), followed by comma and range (here: mV)</p> <p>Place instrument as local use (unlock keypad)</p>
---	---

Instrument responses end with « \r\n »

## 1.2 General communication principles

- Switch the instrument in Remote mode by sending « REM » before sending other commands.
- Erase FIFO of old error codes using command « \*CLS »
- Send a command
- If command is not a request, verify it has been performed successfully sending command « ERR? » then waiting for the response.
- Before closing the connection, switch the instrument in Local mode (command « LOC »)

Example :

```
REM\n
*CLS;_other command(s);_ERR?\n
LOC\n
```

- Some commands may be long to run, it is necessary to provide a long response waiting timeout (1 to 2 minutes for the slowest commands).

## 2. Versions

SCPI command set changes with the versions of internal software of CALYS 150 and CALYS 1500 calibrators. Below are specified these developments.

The argument {1MA | 4MA} is present from firmware version B.00 of CALYS 150 and firmware version A.00 of CALYS 1500, it is used in the following commands:

```
SOURce:RESistance:RANGe
SOURce:RESistance:CURRent
SOURce:RTD:CURRent
SOURce:THERmistor:CURRent
CALibration:SOURce:RESsistance
CALibration:SOURce:RESsistance:ADJust
CALibration:SOURce:RESsistance:GAIN
CALibration:SOURce:RESsistance:OFFSet
CALibration:SOURce:RESsistance:DATE
PROCedure:SOURce:RESistance:CURRent
PROCedure:SOURce:RTD:CURRent
PROCedure:SOURce:THERmistance:CURRent
```

### 3. General commands for instrument control

#### REMOte

Switch the instrument as remote (keypad is locked).

This command has no argument when user management function is disabled.

When user management function is activated, only users with administrator rights can control instrument.

The syntax is the following:

`REMOte_ "<user name>" ,<passcode>`

Example: `REM_ "Admin" ,1234`

#### LOCa1

Switch the instrument to LOCAL and unlock the keypad

#### \*CLS

Erase FIFO of old error codes.

#### ERRor?

Extract from FIFO the oldest error code.

Instrument response:

`<error code number>,<error text>`

Example:

`0, "No error"`

#### \*IDN?

Identify instrument

Instrument response:

`<Manufacturer>,<Model>,<Serial number>,<Software version>`

Example:

`AOIP_SAS,CALYS_150,SN_1234,A00`

## 4. Setting and exploitation of measurement channels (IN and IN-OUT)

Measurement channel is indicated adding '1' or '2' without space at the end of 'SENSE' keyword.

"SENS", "SENS1", "SENSE" ou "SENSE1" : IN channel (left-side channel on CALYS 150).

"SENS2" or "SENSE2" : IN-OUT channel, used as measurement channel (right-side channel on CALYS 150).

### 4.1 Measurement function setting selection

```
SENSE[ {1|2} ]
:VOLTage
    :RANGE_ {100MV | 1V | 10V | 50V}
    :AUTO_ {ON | OFF}
:CURRent
    :RANGE_ {0MA | 4MA | 25MA | 100MA}
    :SUPPLY_ {ON | OFF}
    :SCALE_ {LINEAR | QUADRATIC}
    :HART_ {ON | OFF}
:RESistance
    :RANGE_ {400OHM | 3600OHM | 100KOHM}
    :AUTO_ {ON | OFF}
    :WIRES_ {AUTO | WIRES_2 | WIRES_3 | WIRES_4}
:TCouple
    :TYPE_ <TC or calibrated sensor mnemonic>
    :DISPlay_ {MV | CEL | K | FAR}
    :RJUNCTION_ <Value>
        :TYPE_ {INTERNAL | DISABLED | FIXED}
:RTD
    :TYPE_ <RTD or calibrated sensor mnemonic>
    :DISPlay_ {OHM | CEL | K | FAR}
    :WIRES_ {AUTO | WIRES_2 | WIRES_3 | WIRES_4}
:THERmistor
    :STEinhart_ <val. A>, <val. B>, <val. C>
    :WIRES_ {AUTO | WIRES_2 | WIRES_3 | WIRES_4}
:FREQuency
    :RANGE_ {10KHZ | 100KHZ}
    :INPut_ {VOLTage | SWITCh}
    :UNIT_ {HZ | CPM}
:COUNTER
    :INPut_ {VOLTage | SWITCh}
    :DURation_ <counting time in seconds>
:PRESSure
    :UNIT_ {BAR | PSI | PA | ATM | KGCM2 | CMHG | MMHG | INHG
           | MH2O | FTH2O | INH2O}
:HART
    :ADDRESS_ <Manufacturer ID>, <Device type>, <UI MSB>, <UI ISB>, <UI LSB>
    :VARIABLE_ {PV | AO | SV | TV | QV}
```

Thermocouples mnemonic list:

K, T, J, E, N, U, L, S, R, B, C, PL, MO, XA\_K, XK\_L, XK68

RTD mnemonic list :

PT50, PT100, PT200, PT500, PT1000,  
 PT100\_3916, PT100\_3926,  
 NI100, NI120, NI1000,  
 CU10, CU50,  
 PTP46\_1\_3910, PTP50\_1\_3911, P\_50P\_1\_3911, PTP100\_1\_3910, P\_100P\_1\_3911, PTP500\_1\_3910,  
 CU50\_1\_4260, CUP50\_1\_4280, P\_50M\_1\_4280, CU53\_1\_4260, CU100\_1\_4260, CUP100\_1\_4280,  
 P\_100M\_1\_4280

Mnemonics of the 5 calibrated temperature sensors eventually declared :

CSENSOR1, CSENSOR2, CSENSOR3, CSENSOR4, CSENSOR5

Notes:

Functions FREQUency and COUNTer are only implemented on channel « IN ».

Address of Hart device is its 'long' address that has to be specified as the following:

- 1st argument = 'Manufacturer ID' : number between 0 and 63
- 2<sup>nd</sup> argument = 'Device type' : number between 0 and 255
- 3<sup>rd</sup> to 5th arguments : 'Device UI' : 3 numbers between 0 and 255

#### 4.2 Selection of measuring function

```
SENSe[{1|2}]
:FUNCTION_{VOLTag | CURRent | RESistance | TCouple | RTD | THERmistor
          | CONTinuity | FREQUency | COUNTer | PRESsure | HART}
```

Note: FREQUency and COUNTer functions are only implemented on channel « IN ».

#### 4.3 Annex functions settings

```
SENSe[{1|2}]
:FILTer_{ON | OFF}
:COUNT_<number of measurements>

:NULL_{ON | OFF}
:AMPLitude_{<value>}
:TARE // -> Display to 0

:SCALing_{ON | OFF}
:SIZE_<count of scaling points : 2 to 10>
:POINT_<num(1 to 10)>,<Xvalue Nrf>,<Yvalue Nrf>
:POINT?_<num> // Response : Value of X, Value of Y
:UNIT_"<string of 1 to 4 characters (between double quote marks)>"
:ACCuracy_<number of decimal places>

:STATistics
:INITialize
:MAXimum?
:MINimum?
:AVERage?
```

#### 4.4 Exploitation Commands

```
SENSe[{1|2}]
:HOLD
:TRIG
:RUN
```

#### 4.5 Measurement request

Measurement channel is indicated by suffix 1 or 2 of command 'MEASure'.

Channel « IN » : suffix '1' or no suffix

Channel « IN-OUT » : suffix '2'

**MEASure**[{1|2}]?[\_<number of measures to be averaged>]  
 (Using the currently selected measurement function and range)

**MEASure**[{1|2}]  
 :VOLTage?[\_[{78MV | 100MV | 1V | 10V | 50V} [,<number of meas to be averaged>]]  
 :CURRent?[\_[{0MA | 4MA | 25MA | 100MA} [,<number of meas to be averaged>]]  
 :RESistance?[\_[{400 OHM | 3600 OHM | 100KOHM} [,<number of meas to be averaged>]]  
 :FREQuency?[\_[{10KHZ | 100KHZ} [,<number of meas to be averaged>]]  
 :PRESSure?[\_ [<number of meas to be averaged>]  
 :TEMPerature?[\_{TC | RTD} [,<probe type> [,<number of meas to be averaged>]]  
 :RJUNction?

Frequency measurement can only be performed by channel « IN »

#### Instrument

##### The Instrument

- Switches in Hold mode.
- If the command indicates the measurement function, it selects it, otherwise it uses the currently selected function.
- If there is a range indicated, it selects it, otherwise it uses the currently selected range for the indicated measurement function.
- Performs indicated number of measures (or only one if nothing indicated).
- Sends the average of measurements and the range unit.
- Returns to the Run mode, if it was already in that mode.

Examples:

<i>Command</i>	<i>Response from instrument</i>
MEAS?	100.25,CEL
MEAS2?	235.123,OHM
MEAS:VOLT?_100mV,8	95.123,mV
MEAS:VOLT?	95.123,mV (or 0.09512,V if range 1V)
MEAS:CURR?	20.123,mA
MEAS:RES?	300.123,Ohm
MEAS:FREQ?	1234.567,Hz
MEAS:TEMP?_TC,K	100.25,CEL
MEAS:TEMP?_TC	100.25,CEL (Couple type not indicated -> current couple)
MEAS:TEMP?_RTD,PT100	100.25,CEL
MEAS:TEMP?_RTD	100.25,CEL
MEAS:PRES?	30.123,BAR

## 5. Declaration of calibrated sensors

(Calibrated Sensors)

### CSEnsor

**:LOAD** *<number (1 to 5) of the probe>* (load the current setting into working memory)  
**:NAME** *"<name of 15 characters maximum>"*  
**:CDATE** *<year>,<month>,<day>* (calibration date)  
**:TYPE** *{RTD | TC},<sensor type>*  
**:SIZE** *<number of points in the calibration table : 1 to 4>*  
**:UNIT** *{TEMPerature | VOLTage | RESistance}*  
**:POINT** *<n° of point (1 to SIZE)>,<True temperature in °C>,<read value (in °C, Ohm or V according to UNIT)>*  
  
**:SAVE** *<number (1 to 5) of the sensor>* (save working memory)

Examples:

Declaration of thermocouple K calibrated in 3 points:

```
CSEN:NAME "K_CAL" ; _CDATE_2007,2,15 ; _TYPE_TC,K ; _UNIT_VOLT ; _SIZE_3
CSEN:POINT_1,100.5_CEL,4.120_MV ; _POINT_2,200.6_CEL,8.170_MV ;
      _POINT_3,300.1_CEL,_12.209mV
CSEN:SAVE_1
```

Declaration of Pt100 calibrated in 2 points:

```
CSEN:NAME "MY_PT100" ; _CDATE_2007,2,15 ; _TYPE_RTD,PT100 ; _UNIT_RESISTANCE ; _SIZE_2
CSEN:POINT_1,0_CEL,101.234_OHM ; _POINT_2,100.5_CEL,140.162_OHM
CSEN:SAVE_2
```

To use one of the above mentioned sensor, indicate its mnemonic in the command `SENSe:TYPE` used to declare the type of sensor.

Mnemonic of the 5 sensors: CSENSOR1, CSENSOR2, CSENSOR3, CSENSOR4, CSENSOR5

Example :

```
SENS:RTD:TYPE_CSENSOR2
SENS2:TC:TYPE_CSENSOR1
```

### CSEnsor?

Returns the settings of the current loaded sensor in the form of an indefinite length block (that is, a first line '# 0' followed by n lines and a blank line)

Example:

```
#0\r\n
NAME_K_CAL\r\n
CDATE_2007,2,15\r\n
TYPE_TC,K\r\n
SIZE_3\r\n
UNIT_VOLTAGE\r\n
POINT_1,_100.5_CEL,_4.120_MV\r\n
POINT_2,_200.6_CEL,_8.170_MV\r\n
POINT_3,_300.1_CEL,_12.209_MV\r\n
\r\n
```

## 6. Setting and exploitation of emission channel IN-OUT

### 6.1 Definition of each parameter

#### SOURCE

```

:VOLTage
    :RANGe_{100MV | 1V | 10V | 50V}

:CURRent
    :RANGe_{0mA | 4mA | 25mA}           // ranges 0-20mA, 4-20mA and 25mA
    :SUPPLy_{ON | OFF}
    :SCALe_{LINEAR | QUADRADIC}

:RESistance
    :RANGe_{400OHM | 3600OHM | 100KOHM}[,{PULSed | CONTInuous}](1)[,{1MA | 4MA}](2)
    :CURREnt_{PULSed | CONTInuous}[,{1MA | 4MA}](2),(3)

:TCouple
    :TYPE_<thermocouple mnemonic>
    :DISPlay_{MV | CEL | K | FAR}
    :RJUNction
        :TYPE_{INTernal | DISabled | FIXEd}
        :RJUNction_<Value>           // when TYPE = FIXED, in °C if not specified

:RTD
    :TYPE_<RTD mnemonic>
    :DISPlay_{OHM | CEL | K | FAR}
    :CURREnt_{PULSed | CONTInuous}[,{1MA | 4MA}](2)
        (The argument {1MA | 4MA} is optional)
        (For backward compatibility, if {argument 1MA | 4MA} is missing, the 1MA value is taken into account)

:THERmistor
    :STEinhart_<val. of A>,<val. of B>,<val. of C>
    :CURREnt_{PULSed | CONTInuous}[,{1MA | 4MA}](2)
        (The argument {1MA | 4MA} is optional)
        (For backward compatibility, if {argument 1MA | 4MA} is missing, the 1MA value is taken into account)

:FREQuency
    :RANGe_{1000Hz | 100KHZ}
    :OUTPut_{VOLTage | SWITCh}
    :UNIT_{HZ | CPM}
    :DCYClE_<pwm 0.05 to 0.95>           // 0.05 to 0.95, or 5% to 95%
    :LEVel_<amplitude in V>

:PULSe                                     // pulse period > 10Hz
    :OUTPut_{VOLTage | SWITCh}
    :DCYClE_<pwm 0.05 to 0.95>           // 0.05 to 0.95, or 5% to 95%
    :LEVel_<amplitude in V>

:LPULSe                                     // pulse period < 10Hz
    :OUTPut_{ VOLTage | SWITCh }
    :LEVel_<amplitude en V>

```

(1) The argument {Pulsed | CONTInuous} is optional, if it is absent, the previously programmed value is retained.

(2) The optional argument {1MA | 4MA} is applicable only for Calys 150 of WEM41012 type and Calys 1500 of WEM41050 type. If absent, the 1MA value is taken into account by default. For previous generation WEM41010 kind of Calys 150, it is appropriate not to use the optional argument which is not supported in such case.

### 6.2 Selection of emission function

#### SOURCE

```

:FUNCTION_{VOLTage | CURRent | RESistance | TCouple | RTD | THERmistor
           | FREQuency | PULSe | LPULse}

```

### 6.3 Setting of annex functions

**SOURce**

```
:SCALing_{ON | OFF}
:SIZE_{number of point of scaling : 2 to 10}
:POINT_{num (1 to 10)},<Xvalue Nrf>,<Yvalue Nrf>
:POINT?_{num} // Response : X value,Y Value_Unit
:UNIT_{string of 4 characters (between double quote marks)}
:ACCuracy_{number of decimal places}
```

### 6.4 Sending a value

**SOURce\_{value}**

(Uses current emission function and range, not applicable for function PULSE and LPULSE)

**<value> must not have any unit. This one is implicitly expressed in the unit of the current range.**

Example:

SOUR\_0.5

- If instrument is on range 100mV, it sends 0.5mV.
- If instrument is on range 1, 10 or 50V, it sends 0.5V.
- If instrument is on range 25mA, it sends 0.5mA.
- If instrument is on temperature range and current unit is °F, it simulates 0.5°F.
- ...
- If scaling is on, scaling is applied at the specified value.

**SOURce**

```
:VOLTage_{voltage value} // in V if no unit specified
:CURRent_{currentvalue} // in A if no unit specified
:RESistance_{resistance value} // in Ohm if no unit specified
:TCouple_{temperature value} // in °C if no unit specified
:RTD_{temperature value} // in °C if no unit specified
:THERmistor_{temperature value} // in °C if no unit specified
:FREQuency_{frequency value} // in Hz if no unit specified
:PULSe_{number of pulses},<total time (in seconds)>
```

Examples:

SOUR:VOLT\_0.08

SOUR:VOLT\_80 mV

Send voltage 80 mV

SOUR:CURR\_0.005

SOUR:CURR\_5 mA

Send current 5 mA

SOUR:RTD\_123

Simulate temperature 123°C (whatever is the unit defined in setup)

If temperature instrument unit is °F, value 123°C is converted to °F for display.

SOUR:RTD\_123\_FAR

Simulate temperature 123°F (whatever is the unit defined in setup)

## 7. Setting and exploitation of emission mode on channel « IN-OUT »

### 7.1 Mode 'steps'

#### STEPS

```
:LOW_<low value>
:HIGH_<high value>
:INCRement_<step value>
:TIME_<step time (in seconds)>
:DElay_<start delay after START>

:PLAY_{UP | DOWN}
:HOLD
:NEXT
:PREVious
:CONTInue
:STOP
```

### 7.2 Mode 'simple ramp'

#### RAMP

```
:LOW_<Low value>
:HIGH_<High value>
:TIME_<ramp total time (in seconds)>
:DElay_<start delay after START>

:PLAY_{UP | DOWN}
:HOLD
:CONTInue
:STOP
```

### 7.3 Mode 'cyclic ramp'

#### CRAMP

// cyclic ramp

```
:LOW_<low value>
:HIGH_<high value>
:LTIME_<Low value step time (in seconds)>
:RTIME_<rising time (in seconds)>
:HTIME_<High value step time (in seconds)>
:FTIME_<falling time (in seconds)>
:REPeat_<number of occurrences>
:DElay_<start delay after START>

:PLAY_{UP | DOWN}{[,<number of occurrences>]}
:HOLD
:CONTInue
:STOP
```

### 7.4 Mode 'Synthesizer'

#### SYNThetizer

```
:POINT_<num(1 to 100)>,<Value Nrf>
:TIME_<step time (in seconds)>
:REPeat_<number of occurrences>
:DElay_<start delay after START>

:PLAY[_<1st point>[,<last point>[,<number of occurrences>]]]
:HOLD
:NEXT
:PREVious
:CONTInue
:STOP
```

## 8. EEPROM saving and configuration reading

### CONFig

```
:SAVE_<number between 1 and 9>[,<optional name (19 characters max.)>]  
:LOAD_<number between 1 and 9>
```

## 9. Setting and exploitation of data memory

### 9.1 Setting

#### TRACe[ {1|2} ]

<b>:SIZE</b> <number of measurements>	memory size
<b>:TIMer</b> <period>	in seconds by default
<b>:TRIGger</b>	
<b>:SOURce</b> { <b>IMM</b> ediate   <b>MAN</b> ual   <b>INT</b> ernal}	Event triggering POST counting (If <b>SOURce</b> = <b>INT</b> ernal)
<b>:LEVel</b> <level>	
<b>:SLOPe</b> { <b>POS</b> itive   <b>NEG</b> ative}	
<b>:POST</b> <number of measurements>	N° of measurement counted after the Trig

To set measurement memory of channel 1 (channel IN): **TRACE** or **TRACE1**

To set measurement memory of channel 2 (channel IN-OUT): **TRACE2**

TIMer period can only have the following values:

0.5s, 1s, 2s, 5s, 10s, 20s, 30s, 1mn, 2mn, 5mn, 10mn, 20mn, 30mn

If another value is indicated, it will be the lower valid value that will be used.

Example: **TRACe:TIMer\_3mn** -> Used period will be 2mn and not 3

LEVel and SLOPe are taken into account only if **SOURce** = **INT**ernal.

LEVel is then the input level (in current unit) corresponding to the trigger.

If **SLOPE** = **POS**itive, trigger is performed when measurement is upper or = to the **LEVel** value

If **SLOPE** = **NEG**ative, trigger is performed when measurement is lower or = to the **LEVel** value

**POST** is taken into account when **SOURce** = **MAN**ual or **INT**ernal

Programmed value is the number of measurements to be stored after **TRIG** detection.

Example:

**TRAC:SIZE\_100;\_TIM\_0.5s;\_TRIG:SOUR\_INT;\_LEV\_100.5;\_SLOP\_POS;\_POST\_50**

- Buffer size is 100 measurements
- A value is stored every 0.5 seconds
- Trigger is performed when measurement is upper than 100.5
- 50 measurements have to be stored before stopping the record.

### 9.2 Acquisition

#### INITiate[ {1|2} ]

Erase trace and start recording and trigger monitoring (if **TRIG:SOUR** = **MAN** or **INT**)

If **TRIG:SOUR** = **IMM**, **POST** value is not taken into account and trace records exactly **SIZE** measures before stop.

(Equal to keyboard command **Measures | Run**)

#### ABORT[ {1|2} ]

Stop recording in the trace buffer

(Equal to keyboard command **Measures | Stop**)

#### \*TRG[ {1|2} ]

If **TRIGger:SOURce** = **MAN**ual, start counting **POST** records before stop.

### 9.3 Read the measurements

**DATA[ {1 | 2} ] : POINTs?** Give the number of measurements available in the trace buffer.

**DATA[ {1 | 2} ] : HEADer?**  
Give the trace header as binary block with a defined length.

Example:

#297\n	297 characters
W/O_Name\n	Burst name (15 characters max)
300_POINTS\n	Number of values
PROG\n	Recording type : PROG or FREE
10/05/2005_14:40:00\n	Date and time of 1st record
10/05/2005_14:45:00\n	Date and time of last record
TC_K\n	Measurement function and range
°C\n	Measurement unit (to be confirmed line after line if Vdc Auto)
2\n	Number of decimal places (to be confirmed by the range if Vdc Auto)
SCALING_OFF\n	Scaling: SCALING ON or SCALING OFF
TARE_OFF\n	TARE ON or TARE OFF
\n	

**DATA[ {1 | 2} ] ? \_ [ <1st measurement index> , [ <number of measurements> ] ]**  
 <1st measurement index> : between 1 and the number returned by DATA:POINTs? (1 if not specified)  
 <number of measurements> : number of measurements to do (1 if nothing specified)

Return timed and dated measurements as a binary block with a defined length.

Example:

```
#273\n
000000.0\t123.56789\tUNIT\n
000000.5\t123.56789UNIT\n
000001.0\t123.56789UNIT\n
\n
```

Each measurement with time and date is 24 characters (bytes):

- 8 characters (bytes) for time in seconds : 0.1 to 999999.9s
- 1 tabulation
- 9 characters (bytes) for measurement value
- 1 tabulation
- 4 characters (bytes) for unit
- 1 end of line (single line feed)

1 character 'end of line' is sent between size indicator #ddd and first measurement result. This character is counted in the indicated number ddd.

Character 'End of line' 'eventually sent after the last measurement (in case instrument has nothing else to send) is not counted in the number ddd.

#### 9.4 Recording in eeprom and reading

**MEMory:DATA[ {1 | 2} ]:SAVE\_ "<Name>"**  
 Record Trace in EEPROM as a specified name.

**MEMory:DATA[ {1 | 2} ]:COUNT?**  
 Return number of traces in memory.

**MEMory:DATA[ {1 | 2} ]:HEADer?\_<number>**  
 Return trace header position <number>  
 <number> is a recording rank number in the memory trace.  
 Most recent trace is number 1.  
 The older trace has the number returned by command MEMory:DATA?

Trace header is transmitted as a binary block with a defined length with the same structure as the one returned in response to DATA:HEADer?

**MEMory:DATA[ {1 | 2} ]:LOAD\_<number>**  
 Load the trace with the number indicated in the trace buffer.  
 <number> is a recording rank number in the memory trace.  
 Most recent trace is number 1.  
 The older trace has the number returned by command MEMory:DATA?  
 Once loaded in the trace buffer, burst is read using command DATA as described in chapter 9.3

**MEMory:FREE?**  
 Return the number of free bytes in the memory and the number of occupied bytes.  
 <Free Bytes>,<occupied bytes>

**MEMory:DATA[ {1 | 2} ]:DELETE\_<number>**  
 Delete from memory the trace with this number.  
 (Rank number of all burst is then decremented.)

**MEMory:DATA[ {1 | 2} ]:DELETE:ALL**  
 Delete all records of the memory traces.

## 10. Setting and exploitation of calibration procedures

### 10.1 Setting

#### PROCEDURE

```

:NEW                Initialisation of the fields in default value. 0 calibration points.

:HART
  :TYPE_{NONE | LOOP | TRANsducer | DEVICE} Type of Hart procedure (NONE if non HART)
  :MANUFACTURER_<manufacturer ID between 0 and 255>
  :DEVICE_<device ID between 0 and 255>

:NAME_<"chain with 1 to 15 characters between double quote marks">
:MANUFACTURER_<"chain with 1 to 15 characters between dpuble quote marks">

:METHod_{COMParison | REFGenerator}
:MEASure_{CH1 | EXTERNAL}
:REFERence_{CH2 | EXTERNAL}
:GENERator_{CH2 | FURNace | PRESSure | OTHER}

:SENSE[{1|2}]
  :FUNCTION_{VOLTage | CURRent | RESistance | TCouple | RTD | THERmistor
             | FREQuency | PRESSure}

:VOLTage
  :RANGE_{100MV | 1V | 10V | 50V}
  :AUTO_{ON | OFF}

:CURRent
  :RANGE_{0MA | 4MA | 25MA | 100MA}
  :SUPPLY_{ON | OFF}
  :SCALE_{LINEAR | QUADRADIC}
  :HART_{ON | OFF}

:RESistance
  :RANGE_{400 OHM | 3600OHM | 100KOHM}
  :AUTO_{ON | OFF}
  :WIRES_{AUTO | WIRES_2 | WIRES_3 | WIRES_4}

:TCouple
  :TYPE_<TC or calibrated sensor mnemonic>
  :DISPlay_{MV | CEL | K | FAR}
  :RJUNction_<Value>
             :TYPE_{INTernal | DISabled | FIXed}

:RTD
  :TYPE_<RTD or calibrated sensor mnemonic>
  :DISPlay_{OHM | CEL | K | FAR}
  :WIRES_{AUTO | WIRES_2 | WIRES_3 | WIRES_4}

:THERmistor
  :STEinhart_<val. of A>,<val. of B>,<val. of C>
  :WIRES_{AUTO | WIRES_2 | WIRES_3 | WIRES_4}

:FREQuency
  :RANGE_{10KHZ | 100KHZ}
  :INPut_{VOLTage | SWITCh}
  :UNIT_{HZ | CPM}

:PRESSure
  :UNIT_{BAR | PSI | PA | ATM | KGCM2 | CMHG | MMHG | INHG
        | MH2O | FTH2O | INH2O}

:SCALing_{ON | OFF}
  :SIZE_<number of points in scaling table : 2 to 10>
  :POINT_<num(1 to 10)>,<Xvalue Nrf>,<Yvalue Nrf>
  :UNIT_<"string of 1 to 4 characters (between double quote marks)">
  :ACCuracy <resolution>

```

**PROCEDURE**

**:SOURCE**

```

:FUNCTION_{VOLTage | CURRENT | RESistance
           | TCouple | RTD | THERmistor | FREQuency}

:VOLTage
:RANGE_{100MV | 1V | 10V | 50V}
:CURRENT
:RANGE_{0mA | 4mA | 25mA} // ranges 0-20mA, 4-20mA and 25mA
:SUPPLY_{ON | OFF}
:SCALE_{LINEAR | QUADRADIC}
:RESistance
:RANGE_{400OHM | 3600OHM | 100KOHM}
:CURRENT_{PULSed | CONTinuous}[,{1MA | 4MA}](1)
:TCouple
:TYPE_{TC mnemonic}
:DISPlay_{MV | CEL | K | FAR}
:RJUNction
:TYPE_{INTernal | DISabled | FIXed}
:RJUNction_{Value} // when type = FIXED, in °C if not specified
:RTD
:TYPE_{RTD mnemonic}
:DISPlay_{OHM | CEL | K | FAR}
:CURRENT_{PULSed | CONTinuous}[,{1MA | 4MA}](1)
:THERmistor
:STEinhart_{val. of A},<val. of B>,<val. of C>
:CURRENT_{PULSed | CONTinuous}
:FREQuency
:RANGE_{1000Hz | 100KHZ}
:OUTPut_{VOLTage | SWITCh}
:UNIT_{HZ | CPM}
:DCYCLE_{pwm 0.05 to 0.95} // 0.05 to 0.95, or 5% to 95%
:LEVel_{amplitude in V}

:SCALing_{ON | OFF}
:SIZE_{number of point for scaling : 2 to 10}
:POINT_{num(1 to 10)},<Xvalue Nrf>,<Yvalue Nrf>
:UNIT_{string of 1 to 4 characters between double quote marks}
:ACCuracy_{resolution}

```

(1) The argument {1MA | 4MA} is optional, if it is absent, the 1MA value is taken into account by default.

**:TABLE**

```

:SIZE_{Number of points}
:POINT_{Number of the point: 1 to SIZE},<value>
:EXECution_{UP | UPDown | MANual}
:REST <rest value of the generator at the end of calibration>

```

**:STABILITY**

```

:TIME
:BEFORE_{Number of seconds before stability detection starting}
:INTO_{minimum number of seconds in the stability period}
:DELTA_{max difference between set point and reference value}
:VARiation_{maximum variation of the reference in the stability period}

```

```

:VERDict_{ON | OFF} // display (or not) a verdict OK or KO
:RLIMit_{Maximum relative error in %}
:ALIMit_{Maximum absolute error}

```

Explanations about stability detection before measurement at a calibration set point:

Stability can be detected manually or automatically. In manual mode, the operator validates the stability criteria. In automatic mode, the instrument detects the stability by analyzing changes in the reference channel.

Whatever the calibration method (comparison to a reference or standard generator), the manual mode is programmed applying value 0 to the parameter TIME:INTO.

Automatic detection when using comparison calibration (with channel 2 as reference)

TIME : BEFORE defines the number of seconds before stability detection starting.

DELTA defines the max difference between setpoint and reference value

VARIATION maximum variation of the reference in the stability period

TIME : INTO minimal time when measurement of reference channel is within the DELTA value of the set point and within the VARIATION value.

Automatic detection when using calibration with generator :

TIME : INTO is not used (but must not be null to select the automatic mode).

TIME : BEFORE defines the time to wait between applying the set point and running first measurement

DELTA and VARIATION are not used.

## 10.2 Saving in eeprom and readings

- MEMory: PROCedure: SAVE** Save current procedure in memory.
- MEMory: PROCedure: COUNT?** Return number of procedure saved in memory.
- MEMory: PROCedure: SUMMary?**

Return a summary of saved procedures, as a block with an undefined length, with for each procedure, manufacturer name and number of associated reports.

Example:

```
#0\n
<Order number>\t<instrument name>\t<manufacturer name>\t<number of reports>\n
...
<Order number>\t<instrument name>\t<manufacturer name>\t<number of reports>\n
\r\n
```

Length of each line is exactly 39 bytes + \n = 40  
 $(3+1+15+1+15+1+3+1 = 40)$

Example:

```
#0\n
001\tINSTRUMENT_0001\tMANUFACTURER_01\t000\n
002\tINSTRUMENT_0002\tMANUFACTURER_02\t005\n
003\tINSTRUMENT_0003\tMANUFACTURER_03\t010\n
004\tINSTRUMENT_0004\tMANUFACTURER_04\t002\n
\r\n
```

**MEMory: PROCedure?\_<number>**

Return procedure configuration as a block of bytes with undefined length.

Example:

```
#0\n
NAME_ "<string with 1 to 15 characters between double quote marks>"\n
MANUFACTURER_ "<string with 1 to 15 characters between double quote marks>"\n
HART:TYPE_NONE\n
METHOD_COMPARISON\n
MEASURE_CH1\n
REFERENCE_CH2\n
GENERATOR_FURNACE\n
SENSE1:FUNCTION RTD\n
SENSE1:RTD:TYPE_PT100;_DISPLAY_CEL;_WIRES_AUTO\n
SENSE1:SCALING_ON;SIZE_2;UNIT_"CEL";ACCURACY_2\n
SENSE1:SCALING:POINT_1,_10,_10.5\n
SENSE1:SCALING:POINT_2,_100.5,_101.6\n
SENSE2:FUNCTION RTD\n
SENSE2:RTD:TYPE_PT100;_DISPLAY_CEL;_WIRES_AUTO\n
SENSE2:SCALING_ON;SIZE_2;UNIT_"CEL";ACCURACY_2\n
SENSE2:SCALING:POINT_1,_10,_10.5\n
SENSE2:SCALING:POINT_2,_100.5,_101.6\n
TABLE:SIZE_2;EXECUTION_UPD;REST_23\n
TABLE:POINT_1,_50;POINT_2,_100\n
STABILITY:DELTA_0.2;VARIATION_0.1\n
STABILITY:TIME:BEFORE_600;INTO_100\n
VERDICT_ON\n
RLIMIT_0.5\n
ALIMIT_1.5\n
\r\n
```

Length of block undefined

End of block = empty line

Setting of SENSE1 is indicated only if disposal is measured on channel 1 (i.e. MEASURE CH1).  
 Setting of SENSE2 is indicated only if calibration method is programmed as a comparison and reference is set on channel 2.  
 The setting of SOURCE is indicated only if calibration method is a method using standard generator and generator is channel 2.

**MEMory: PROCedure: DElete\_<number>**

Delete the mentioned procedure.

**MEMory: PROCedure: DElete: ALL**

### 10.3 Calibration procedure of HART disposal

Calibration can be performed on the detector, the current loop or the entire disposal (detector+loop). The commands are applied as same as above mentioned with the following differences:

- No setting of the channel 1 (SENSE1)  
No selection of measurement input (always MEASURE\_CH1)
- When calibrating the current loop:  
No setting of the reference channel (SENSE2 or SOURCE)  
No selection of calibration method or generator
- Additional commands

#### PROCEDURE

:HART

```

:TYPE_{NONE | LOOP | TRANSDUCER | DEVICE}           Type of procedure
:MANUFACTURER_{manufacturer ID between 0 and 255}
:DEVICE_{disposal ID between 0 and 255}
    
```

Example of HART procedure setting for current loop testing:

```

NAME_{string of 1 to 15 characters (between double quote marks)}\n
MANUFACTURER_{string with 1 to 15 characters between double quote marks}\n
HART:TYPE_LOOP;MANUFACTURER_125;DEVICE_3\n
TABLE:SIZE_5;EXECUTION_UP;REST_4\n
TABLE:POINT_1,_4;POINT_2,_8;POINT_3,_12;POINT_4,_16;POINT_5,_20\n
STABILITY:TIME:BEFORE_2;INTO_1\n
VERDICT_ON\n
RLIMIT_0.5\n                               As 0.5% of reading
ALIMIT_0.01\n                             So 0.01mA
    
```

HART procedure for detector testing:

```

NAME_{string of 1 to 15 characters (between double quote marks)}\n
MANUFACTURER_{string of 1 to 15 characters between double quote marks}\n
HART:TYPE_TRANSDUCER;MANUFACTURER_125;DEVICE_3\n
METHOD_COMPARISON\n
REFERENCE_CH2\n
GENERATOR_FURNACE\n
SENSE2:FUNCTION_RTD\n
SENSE2:RTD:TYPE_PT100;DISPLAY_CEL;WIRES_AUTO\n
SENSE2:SCALING_OFF\n
TABLE:SIZE_2;EXECUTION_UPD;REST_23\n
TABLE:POINT_1,_50;POINT_2,_100\n
STABILITY:DELTA_0.2;VARIATION_0.1\n
STABILITY:TIME:BEFORE_600;INTO_100\n
VERDICT_ON\n
RLIMIT_0.5\n                               0.5% of reading
ALIMIT_0.01\n                             1.5°C
    
```

HART procedure of entire disposal testing

```

NAME_{string of 1 to 15 characters (between double quote marks)}\n
MANUFACTURER_{string of 1 to 15 characters between double quote marks}\n
HART:TYPE_DEVICE;MANUFACTURER_125;DEVICE_3\n
METHOD_COMPARISON\n
REFERENCE_CH2\n
GENERATOR_FURNACE\n
SENSE2:FUNCTION_RTD\n
SENSE2:RTD:TYPE_PT100;DISPLAY_CEL;WIRES_AUTO\n
SENSE2:SCALING_OFF\n
TABLE:SIZE_2;EXECUTION_UPD;REST_23\n
TABLE:POINT_1,_50;POINT_2,_100\n
STABILITY:DELTA_0.2;VARIATION_0.1\n
STABILITY:TIME:BEFORE_600;INTO_100\n
VERDICT_OFF\n
    
```

### 10.4 Reading reports associated to a procedure

**MEMory:PROCedure:PV?** <procedure number>, <report number>

Return report as a bytes block with undefined length.

Example:

#0\n	block with undefined length
INSTRUMENT_0001\n	instrument name (15 characters max)
MANUFACTURER_01\n	Manufacturer name (15 characters max)
1458045\n	Serial number (15 characters max)
Calys150\n	Calys model number (15 characters max)
1001\n	Calys serial number (15 characters max)
10/01/2015_10:35:00\n	Calys adjustment date
10/05/2015_14:40:00\n	Calys calibration date
AHZ45012\n	Calys Calibration certificate number (50 characters max)
581475\n	Keller sensor serial number (15 characters max)
DUPONT\n	User name (15 characters max)
Instability\n	Comment (15 characters max)
AS_FOUND\n	Step before/after adjustment
10/06/2015_16:25:00\n	Performing date
OK\n	Result
5\n	Number of calibrated points
10.0\t10.2\n	Point #1 (true value / read value)
50.0\t48.8\n	Point #2
80.0\t80.5\n	Point #3
50.0\t50.1\n	Point #4
10.0\t9.9\n	Point #5

**MEMory:PROCedure:PV:DELete** <procedure number>, <report number>

**MEMory:PROCedure:PV:DELete:ALL**

## 11. Adjustment

### 11.1 « IN » and « IN-OUT » channels adjustment

The channel to be adjusted is indicated by suffix 1 or 2 in the command header 'SENSE'.

No space between suffix and command header.

Channel « IN » : suffix '1' or no suffix

Channel « IN-OUT » : suffix '2'

#### CALibration

```
[ :SENSE[ { 1 | 2 } ] ]
  :VOLTage
    :MEASure?_ { 78MV | 100MV | 1V | 10V | 50V } [ , <number of measurements
                                                    to average > ]
    :GAIN_ { 78MV | 100MV | 1V | 10V | 50V } , <gain value >
    :GAIN?_ { 78MV | 100MV | 1V | 10V | 50V }
    :OFFSet_ { 78MV | 100MV | 1V | 10V | 50V } , <offset value >
    :OFFSet?_ { 78MV | 100MV | 1V | 10V | 50V }
    :DATE?_ { 78MV | 100MV | 1V | 10V | 50V } // Last adjustment date of range

  :CURRent
    :MEASure?_ { 25MA | 100MA } , <number of measurements to average >
    :GAIN_ { 25MA | 100MA } , <gain value >
    :GAIN?_ { 25MA | 100MA }
    :OFFSet_ { 25MA | 100MA } , <offset value >
    :OFFSet?_ { 25MA | 100MA }
    :DATE?_ { 25MA | 100MA } // Last adjustment date of range

  :RESistance
    :MEASure?_ { 400OHM | 3600OHM | 100KOHM } [ , <number of measurements
                                                    to average > ]
    :GAIN_ { 400OHM | 3600OHM | 100KOHM } , <gain value >
    :GAIN?_ { 400OHM | 3600OHM | 100KOHM }
    :OFFSet_ { 400OHM | 3600OHM | 100KOHM } , <offset value >
    :OFFSet?_ { 400OHM | 3600OHM | 100KOHM }
    :DATE?_ { 400OHM | 3600OHM | 100KOHM }

  :FREQuency
    :MEASure?_ { 10KHZ | 100KHZ } [ , <number of measurements to average > ]
    :GAIN_ { 10KHZ | 100KHZ } , <gain value > // Range : mandatory, but w/o any effect
    :GAIN?_ { 10KHZ | 100KHZ } // Range : mandatory, but w/o any effect
    :DATE?_ { 10KHZ | 100KHZ } // Range : mandatory, but w/o any effect

  :TCCouple
    :MEASure?_ <couple type> [ , <number of measurements to average > ]
    (performs the measurement of TC with internal RJ, but without applying the RJ correction offset)

  :RJUNction
    :OFFSet_ <offset value >
    :OFFSet?
```

## 11.2 Setting of emission channel « IN-OUT »

### CALibration

```

:SOURce
:VOLTage_{78MV | 100MV | 1V | 10V | 50V},<value to send with unit>
:VOLTage
:ADJust?_{78MV | 100MV | 1V | 10V | 50V} // Auto calibration
:GAIN_{78MV | 100MV | 1V | 10V | 50V},<gain value>
:GAIN?_{78MV | 100MV | 1V | 10V | 50V}
:OFFSet_{78MV | 100MV | 1V | 10V | 50V},<offset value>
:OFFSet?_{78MV | 100MV | 1V | 10V | 50V}
:DATE?_{78MV | 100MV | 1V | 10V | 50V} // Last adjustment date of range
:DAcline?_{78MV | 100MV | 1V | 10V | 50V} // -> 0 and mid-range of DACs
:DACPoints?_{1st point}>[,<number of points>] // calibration points of DACs

:CURRENT_25MA,<value to send with unit>
:CURRENT
:ADJust?_25MA // Auto calibration
:GAIN_25MA,<gain value>
:GAIN?_25MA
:OFFSet_25MA,<offset value>
:OFFSet?_25MA
:DATE?_25MA // Last adjustment date of range
:DAcline? // -> 0 and mid-range of DACS

:RESistance_{400OHM | 3600OHM | 100KOHM},{PULSed | CONTinuous}
[, {1MA | 4MA}](1),<value to send>
:RESistance
:ADJust:OFFSet? // Adjustment of offsets 1 and 2
:ADJust?_{400OHM | 3600OHM | 100KOHM}[, {1MA | 4MA}](1) // Auto calibration
:GAIN_{400OHM | 3600OHM | 100KOHM},{PULSed | CONTinuous}
[, {1MA | 4MA}](1),<gain value>
:GAIN?_{400OHM | 3600OHM | 100KOHM},{PULSed | CONTinuous}
[, {1MA | 4MA}](1)
:OFFSet_{400OHM | 3600OHM | 100KOHM},{PULSed | CONTinuous}
[, {1MA | 4MA}](1),<offset value>
:OFFSet?_{400OHM | 3600OHM | 100KOHM},{PULSed | CONTinuous}
[, {1MA | 4MA}](1)
:DATE?_{400OHM | 3600OHM | 100KOHM},{PULSed | CONTinuous}
[, {1MA | 4MA}](1)

:FREQuency_<value to send with unit>
:FREQuency
:GAIN_<gain value>
:GAIN?
:DATE?

MEASure?

```

(1) The optional argument {1MA | 4MA} is applicable only for WEM41012 kind of Calys 150 and WEM41050 kind of Calys 1500. If absent, the 1MA value is taken into account by default. For previous generation WEM41010 kind of Calys 150, the 400 Ohm - 4 mA range does not exist, so it is not appropriate to use the optional argument (not supported in this case).

**Commands for offset control:**

(see Calys150/1500 NTM47224-390A1 maintenance manual for the description and use of relevant registers)

o1_400_cont_<offset value>	
o1_400_cont?	
o1_400_puls_<offset value>	
o1_400_puls?	
o1_3k6_cont_<offset value>	
o1_3k6_cont?	
o1_3k6_puls_<offset value>	
o1_3k6_puls?	
o1_400_4mA_<offset value>	For WEM41012 kind of Calys 150 and WEM41050 kind of Calys 1500
o1_400_4mA?	For WEM41012 kind of Calys 150 and WEM41050 kind of Calys 1500
o1_100k_<offset value>	For older WEM41010 kind of Calys 150 (or o1_50K_<offset value>)
o1_100k?	For older WEM41010 kind of Calys 150 (or o1_50K?)
o2_450_<offset value>	
o2_450?	
o2_3K6_<offset value>	
o2_3K6?	
o2_400_4mA_<offset value>	For WEM41012 kind of Calys 150 and WEM41050 kind of Calys 1500
o2_400_4mA?	For WEM41012 kind of Calys 150 and WEM41050 kind of Calys 1500
o2_100K_<offset value>	For older WEM41010 kind of Calys 150 (or o2_50K_<offset value>)
o2_100K?	For older WEM41010 kind of Calys 150 (or o2_50K?)

**11.3 Other commands relative to instrument adjustment**

**CALibration**

```

:DATE_<year>,<month>,<day> // Calibration date
:DATE?
:REPort_<string of 0 to 50 chars (between double quote marks)>" // Reference of
calibration certificate
:REPort?
:SECure:STATE_{OFF | ON},<code 5930> // lock/unlock eeprom
:SECure:STATE?

```

## 12. User management and connection log

### 12.1 User management principle

When user management function is on, only users having all rights (administrators) can control the instrument. This can be performed declaring user name and passcode in the command `REMOte`.

### 12.2 Declare a user and read declared user

#### `USER:COUNT?`

Return number of users declared (0-10)  
This commands does not require to be in mode `REMOte`

#### `USER:MANagement_{ON | OFF}`

Switch on/off user management.  
User management can be switched on only if one user at least has all the rights.

#### `USER:MANagement?`

Return user management function on/off status.  
Response: 0 (out of use) or 1 (in use).  
This commands does not require to be in mode `REMOte`

#### `USER:SUMMARY?`

Return list of users (as a block of undefined length)  
Response:  
#0\n  
01\tDUPONT\tALL\n  
02\tDURAND\tRESTRICTED\n  
03\tMARTIN\tEXTENDED\n  
\r\n // End of list

Remark: list of users is alphabetically ordered, without considering lower and upper case

#### `USER:DELEte_<order number 1 to 10>`

Delete the user relative to the specified number. The order number of users that follow in the list is decremented. In the above list example, the command `USER:DELETE_1` removes the user DUPONT. The DURAND user number becomes 1 and that of MARTIN becomes 2.  
The command does not remove the connected user.

#### `USER:DELEte:ALL`

Delete all the users.

#### `USER:ADD_ "<UserName>" ,<PIN Code> ,<Rights>`

<UserName>: user name (between double quote marks), without space, max 15 characters  
<PIN Code>: pass code (integer number from 0 to  $2^{32}-1$ )  
<Rights>: {RESTRicted | EXTended | ALL}

Add a user to the list and siwth off user management.

Example: `USER:ADD_ "ADMIN" , 0 , ALL`

#### `USER:TIME_<Automatic disconnection delay (in minutes)>`

Value 0 switch off automatic disconnection function.

#### `USER:TIME?`

Return the delay of automatic disconnection function in minutes.

### 12.3 Reading logfile

**LOGFile:COUNT?**

Return log file size (count of lines).

**LOGFile:DATA?\_<1st line>,<Number of lines>**

Return the requested lines in the form of an undefined length block.

Response:

#0\n

<Date-time>\t<Event>\t"<UserName>"\t"<Comment>\n

<Date-time>\t<Event>\t"<UserName>"\t"<Comment>\n

...

\r\n

**LOGFile:DElete**

Delete log file.

### 13. Setting SOURCE or SENSE mode for channel CH2

Setting the mode:

```
CH2:MODE_{SOURCE | SENSE}
```

Reading of mode:

```
CH2:MODE?
```

Response: SOURCE or SENSE

## 14. LCD Display information request

Command:

```
DISPlay?[_{BANNER | CH1 | CH2 | ALL}][,{LEFT | CENTER | RIGHT}]]
```

Response: data block of undefined length (# 0, then data and blank line)

Examples:

```
DISP?
DISP?_CH1
DISP?_CH1,CENTER
```

Format of complete response (command 'DISP?' or 'DISP?\_ALL'):

```
#0\r\n
<DISP>
<DATE>27/11/2015_12:40:40</DATE>
<TEMP>25.3°C</TEMP>
<FILE>name</FILE>
<BATT>75%</BATT>
<CH1>
  <LEFT>
    <L1>Text</L1>
    <L2>Text</L2>
    <L3>Text</L3>
    <L4>Text</L4>
    <L5>Text</L5>
  </LEFT>
  <CENTER>
    <INFOS>V1:IN_K</INFOS>
    <STD>
      <REC>
        <RUN>100/150</RUN>
        <STOP>150</STOP>
      </REC>
      <UNIT>°C</UNIT>
      <MEAS>123.45</MEAS>
      <AUX>text</AUX>
    </STD>
    <HART>
      <PV>value and unit</PV>
      <PVAO>value and unit</PVAO>
      <MEAS>value and unit</MEAS>
    </HART>
    <ERROR>text</ERROR>
  </CENTER>
  <RIGHT>
    <FILTER/><SCALING/><NULL/><HOLD/><ERROR/>
  </RIGHT>
</CH1>
<CH2>
  <SENSE/>
  <SOURCE/>
  (same as CH1 for tags <LEFT> and <CENTER>)
  <RIGHT>
    <FILTER/><SCALING/><NULL/><HOLD/><ERROR/>
    <EDIT/><EDITINCR/><PREDEF/><PREDEFPCY/><INCR/><RAMP/><RAMPCYC/>
    <SYNTHE/><PULS/><TRANS/>
  </RIGHT>
</CH2>
</DISP>
\r\n
```

Text transmitted between tags <INFOS> and </INFOS> gives information on name and channel configuration.

It can contain special characters such as 0xA0 to 0xA7, with following meanings:

- 0xA0: current loop powered by Calys
- 0xA1: square scaling (current loop)
- 0xA2: '2 wires' connection
- 0xA3: '3 wires' connection
- 0xA4: '4 wires' connection
- 0xA5: Unit Ohm
- 0xA6: Current loop not powered by Calys
- 0xA7: Hart compatibility resistance switched in the electrical circuit.

**Note:** the indentation and blank lines shown above are only to clarify the presentation. They are not transmitted: no blank line before the last one, and no blank space at beginning of lines.

Some tags may not be transmitted. Generally speaking, only tags containing data are transmitted.

This concerns:

- Tags <L1> to <L6>
- Tag <REC> is transmitted only if a burst is running, or is not yet saved
- Tags <AUX> and <ERROR>
- Tags <FILTER/><SCALING/><NULL/><HOLD/><ERROR/> are transmitted only if necessary

Tag <STD> corresponds to the "Standard" mode of channel 1, tag <HART> to Hart mode. Only one of the 2 tags can be transmitted.

For channel 2, <SENSE/> or <SOURCE/> are transmitted, according to the mode.

The 2 optional command arguments are filters limiting transmitted information.

CH1: to send only channel 1

CH2: to send only channel 2

LEFT: to send only content of left views of windows CH1 or CH2

CENTER: to send only content of central views

RIGHT: to send only content of right views