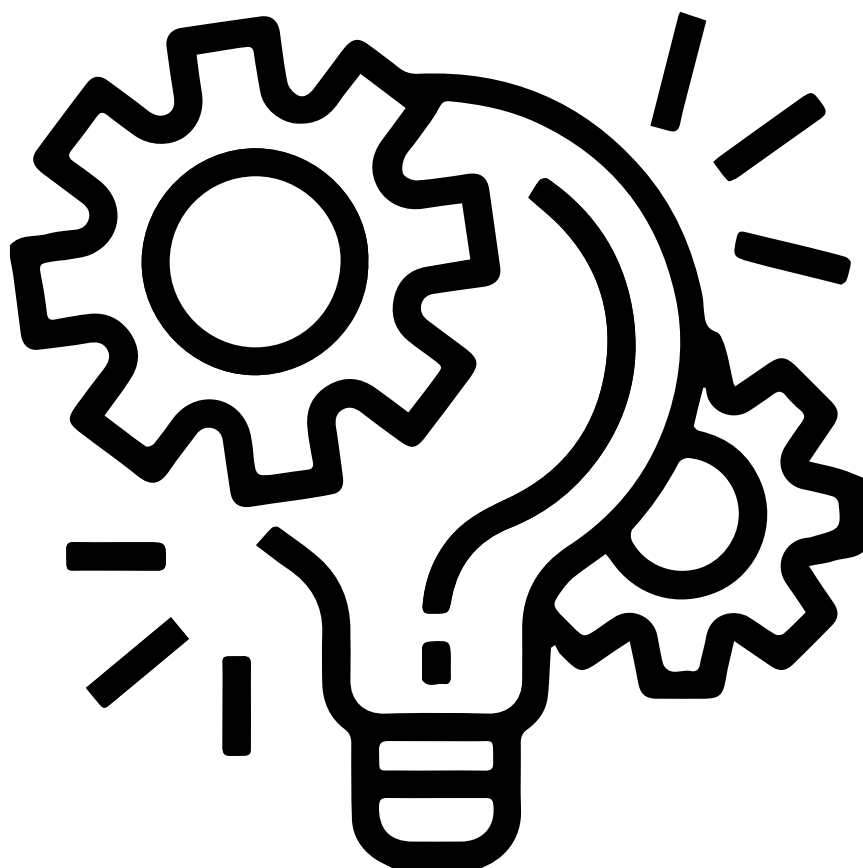


CA 834X

Q2ControlPanel



CONTENTS

1. INTRODUCTION	3
2. CONNECTION	3
2.1. Ethernet	3
2.2. Wifi.....	3
2.3. USB	3
3. COMMUNICATION.....	3
3.1. Communication port	3
3.2. SCPI principles	3
3.3. Data transfer (IEEE 488.2 block format).....	4
3.4. SCPI implementation in CA834X.....	4
4. COMMAND REFERENCE	5
4.1. IEEE 488.2 Mandated commands.....	5
4.2. CONTrol subsystem	6
4.3. HCOPY subsystem	8
4.4. HELP subsystem	8
4.5. MEASurement subsystem.....	8
4.6. MMEMory subsystem.....	13
4.7. SYSTem subsystem	14
5. REFERENCES.....	14

1. INTRODUCTION

You can remotely control CA 834X using TCP/IP (port 23) via :

- ETHERNET
- WIFI
- USB

Programming language is SCPI. This language is defined by the IVI Foundation and you can find a reference guide to SCPI here: <http://www.ivifoundation.org/docs/SCPI-99.PDF>.

This instrument uses a subset of that reference guide and other commands not defined in that guide, in respecting the SCPI semantic.

2. CONNECTION

2.1. ETHERNET

Connect the CA 834X to your local network or to your PC directly.

In the Network configuration menu, you can select DHCP automatic connection and press the RUN key.

Wait up to 30 seconds, note your IP address.

2.2. WIFI

After selecting SSID and entering Passphrase, select DHCP and press the RUN key.

Wait up to 30 seconds, note your IP address.

2.3. USB

You can also communicate through USB using a serial communication driver in your PC.

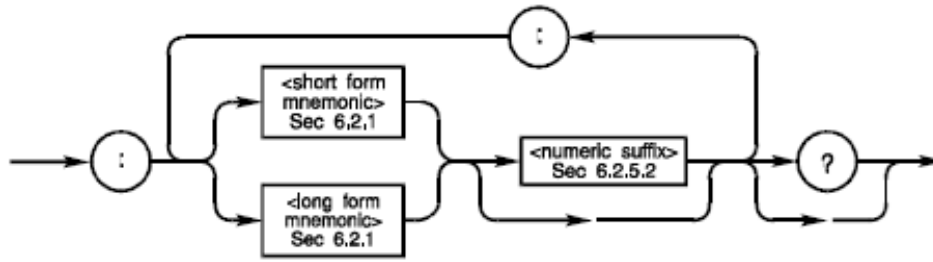
3. COMMUNICATION

3.1. COMMUNICATION PORT

- Remote control is accessible on port 23 over ETHERNET, WIFI.
- Remote control is also accessible using USB port with a serial profile.

3.2. SCPI PRINCIPLES

- SCPI is based on a client/model scheme. Instrument is the server, PC is the client.
- Server (Instrument) never emits data without a request from the client (PC).
- A request from the client may be response-less.
- SCPI commands are a suite of keywords separated by the COLON character ":".



- Keyword can be written using a long form or a short form which is a subset of the long form. In the tables below UPPERCASE characters describe the short form and UPPERCASE+LOWERCASE characters describe the long form.
- Parameters can be added to the SCPI command line and separated from the command by the SPACE character.
- Terminator char is "\n" or "\r".
- If the request execution fails, an error code is stored in a FIFO queue. The client can read and consume the error code by sending the command **SYST:ERR?**. Error queue may be cleared with ***CLS** command.

3.3. DATA TRANSFER (IEEE 488.2 BLOCK FORMAT)

Every data transfer uses the IEEE 488.2 block format.

In this format, a block of bytes is preceded by a #AN sequence, where A equals the number of digits of N (non null numeric digit), and N equals the length of a data block (#AN is coded in ASCII).

E.g. To send/receive 1234 bytes of a binary block of data, the transmitted data will be: #41234 followed by 1234 bytes.

3.4. SCPI IMPLEMENTATION IN CA 834X

- SCPI is used to transfer configuration sets.
- Each configuration set is coded using JSON format.
- Binary format is used for result files. A dedicated JSON format file can be used to interpret the content of this binary file.

4. COMMAND REFERENCE

In this tables below:

- UPPERCASE characters describe the short form
- UPPERCASE+LOWERCASE characters describe the long form.
- Parameters between [] are optional

4.1. IEEE 488.2 MANDATED COMMANDS

Command	Description	Reference	Comments
*CLS	Clear Status Command	SCPI-99.PDF IEEE 488.2	To clean error stack
*ESE	Standard Event Status Enable Command	SCPI-99.PDF IEEE 488.2	
*ESE?	Standard Event Status Enable Query	SCPI-99.PDF IEEE 488.2	
*ESR?	Standard Event Status Register Query	SCPI-99.PDF IEEE 488.2	
*IDN?	Identification Query	SCPI-99.PDF IEEE 488.2	Response format: <model>, <serial number>, <Firmware version>, <HPS version>, <FPGA version> Used to test the presence of CA 834X
*OPC	Operation Complete Query	SCPI-99.PDF IEEE 488.2	
*OPC?	Operation Complete Query	SCPI-99.PDF IEEE 488.2	
*RST	Reset Command	SCPI-99.PDF IEEE 488.2	
*SRE	Service Request Enable Command	SCPI-99.PDF IEEE 488.2	
*SRE?	Service Request Enable Query	SCPI-99.PDF IEEE 488.2	
*STB?	Self-Test Query	SCPI-99.PDF IEEE 488.2	
*WAI	Wait-to-Continue Command	SCPI-99.PDF IEEE 488.2	

4.2. CONTrol subsystem

In this section, the content of the field <data>, is coded using the IEEE 488.2 block format (#AN...).
The response to all requests terminated by "?", is also coded using the IEEE 488.2 block format (#AN...).

Command	Description	Comments
CONTrol:ALARM:SCHEDule #an<data>	Set Alarm Campaign Schedule parameters (see .acs file).	<Data> is sent or received using IEEE488.2 block format.
CONTrol:ALARM:SCHEDule?	Get Alarm Campaign Schedule parameters (see .acs file).	<Data> is formatted using JSON standard.
CONTrol:ALARM:SETUP #an<data>	Set Alarm Campaign User Profile parameters (see .acup file).	<Data> is sent or received using IEEE488.2 block format.
CONTrol:ALARM:SETUP?	Get Alarm Campaign User Profile parameters (see .acup file).	<Data> is formatted using JSON standard.
CONTrol:ALARM:SETUP:QUICKstart #an<data>	Set Alarm Quickstart setup parameters	<Data> is sent or received using IEEE488.2 block format.
CONTrol:ALARM:SETUP:QUICKstart?	Get Alarm Quickstart setup parameters	<Data> is formatted using JSON standard.
CONTrol:ALARM:STATE <STOP START QUICKstart>	Set state of Alarm mode	<Data> is sent or received using IEEE488.2 block format.
CONTrol:ALARM:STATE?	Get state of Alarm mode (STOP, WAIT STOP, INIT, WAIT or RECORDING)	<Data> is formatted using JSON standard.
CONTrol[:CALCulate]:DISPlay #an<data>	Set real time calculation parameters	<Data> is sent or received using IEEE488.2 block format.
CONTrol[:CALCulate]:DISPlay?	Get real time calculation parameters	<Data> is formatted using JSON standard.
CONTrol[:CALCulate]: HARMonic #an<data>	Set harmonic calculation parameters	<Data> is sent or received using IEEE488.2 block format.
CONTrol[:CALCulate]:HARMonic?	Get harmonic calculation parameters	<Data> is formatted using JSON standard.
CONTrol[:CALCulate]: NOMInal #an<data>	Set nominal calculation parameters	<Data> is sent or received using IEEE488.2 block format.
CONTrol[:CALCulate]:NOMInal?	Get nominal calculation parameters	<Data> is formatted using JSON standard.
CONTrol[:CALCulate]:WAVEform?	Get waveform calculation parameters	<Data> is sent or received using IEEE488.2 block format.
CONTrol[:CALCulate]:WAVEform #an<data>	Set waveform calculation parameters	<Data> is formatted using JSON standard.
CONTrol:ENERGY:SETUP #an<data>	Set Configuration of Energy counting	<Data> is sent or received using IEEE488.2 block format.
CONTrol:ENERGY:SETUP?	Get Configuration of Energy counting	<Data> is formatted using JSON standard.
CONTrol:ENERGY:STATE <STOP START RESET>	Set state of Energy counting	<Data> is sent or received using IEEE488.2 block format.
CONTrol:ENERGY:STATE?	Get state of Energy counting	<Data> is formatted using JSON standard.
CONTrol:FLAGGING #an<data>	Set Flagging configuration	<Data> is sent or received using IEEE488.2 block format.
CONTrol:FLAGGING?	Get Flagging configuration	<Data> is formatted using JSON standard.
CONTrol:HOOKUP #an<data>	Set the Configuration of the hookup	<Data> is sent or received using IEEE488.2 block format.
CONTrol:HOOKUP?	Get the Configuration of the hookup	<Data> is formatted using JSON standard.
CONTrol:ICP?	Instrument Control Panel	<Data> is received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTrol:INRUSH:SCHEDule <data>	Set Inrush Search Schedule parameters (see .iss file)	<Data> is sent or received using IEEE488.2 block format.
CONTrol:INRUSH:SCHEDule?	Get Inrush Search Schedule parameters (see .iss file)	<Data> is formatted using JSON standard.
CONTrol:INRUSH:SETUP #an<data>	Set Inrush Search User Profile parameters (see .isup file)	<Data> is sent or received using IEEE488.2 block format.
CONTrol:INRUSH:SETUP?	Get Inrush Search User Profile parameters (see .isup file)	<Data> is formatted using JSON standard.
CONTrol:INRUSH:SETUP:QUICKstart #an<data>	Set Inrush Quickstart setup parameters	<Data> is sent or received using IEEE488.2 block format.
CONTrol:INRUSH:SETUP:QUICKstart?	Get Inrush Quickstart setup parameters	<Data> is formatted using JSON standard.
CONTrol:INRUSH:STATE <STOP START QUICKstart>	Set state of Inrush mode	<Data> is sent or received using IEEE488.2 block format.
CONTrol:INRUSH:STATE?	Get state of Inrush mode (STOP, WAIT STOP, INIT, WAIT or RECORDING)	<Data> is formatted using JSON standard.
CONTrol:LANGUage <language>	Set current language	<Data> is sent or received using IEEE488.2 block format.
CONTrol:LANGUage?	Get current language	<Data> is formatted using JSON standard.
CONTrol:LCD:BRIGHtness #an<data>	Set LCD brightness	<Data> is sent or received using IEEE488.2 block format.
CONTrol:LCD:BRIGHtness?	Get LCD brightness	<Data> is formatted using JSON standard.

Command	Description	Comments
CONTRol:LCD:COLors #an<data>	Set color of the waveforms	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:LCD:COLors?	Get color of the waveforms	
CONTRol:MONIToring:SCHEdUle #an<data>	Set Monitoring Schedule parameters (see .mns file)	Monitor mode is a macromode grouping TREND, TRANSIENT and ALARM modes. Setup of monitor mode is done using a complete setup for a specific user. See "CONT:USER MONITOR" command.
CONTRol:MONIToring:SCHEdUle?	Get Monitoring Schedule parameters (see .mns)	
CONTRol:MONIToring:SETUP:QUICKstart <data>	Set Monitoring Quickstart setup parameters	
CONTRol:MONIToring:SETUP:QUICKstart?	Get Monitoring Quickstart setup parameters	
CONTRol:MONIToring:STATE <STOP START QUICKstart>	Set state of monitor mode	
CONTRol:MONIToring:STATE?	Get state of monitor mode (STOP, WAIT STOP, INIT, WAIT or RECORDING)	
CONTRol:NETWork:EMAIL #an<data>	Set email configuration parameters	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:NETWork:EMAIL?	Get email configuration parameters	
CONTRol:NETWork:ETHERnet #an<data>	Set ethernet configuration parameters	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:NETWork:ETHERnet?	Get ethernet configuration parameters	
CONTRol:NETWork:IRD #an<data>	Set IRD configuration parameters	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:NETWork:IRD?	Get IRD configuration parameters	
CONTRol:NETWork:WAP #an<data>	Set Wifi Access Point configuration parameters	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:NETWork:WAP?	Get Wifi Acces Point configuration parameters	
CONTRol:NETWork:WIFI #an<data>	Set WIFI configuration parameters	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:NETWork:WIFI?	Get WIFI configuration parameters	
CONTRol:RATIOsensors #an<data>	Set the current sensors and current/voltage ratios configuration	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:RATIOsensors?	Get the current sensors and current/voltage ratios configuration	
CONTRol:SSI?	System Static Informations	<Data> is received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TIME #an<data>	Set the configuration of date/time	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TIME?	Get the configuration of date/time	
CONTRol:TRANSient:SCHEdUle #an <data>	Set Transient Search Schedule parameters (see .tss file).	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TRANSient:SCHEdUle?	Get Transient Search Schedule parameters (see .tss file).	
CONTRol:TRANSient:SETUP #an<data>	Set Transient Search User Profile parameters (see .tsup file).	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TRANSient:SETUP?	Get Transient Search User Profile parameters (see .tsup file).	
CONTRol:TRANSient:SETUP:QUICKstart #an<data>	Set Transient Quickstart setup parameters	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TRANSient:SETUP:QUICKstart?	Get Transient Quickstart setup parameters	
CONTRol:TRANSient:STATE <STOP START QUICKstart>	Set Transient state	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TRANSient:STATE?	Get Transient state (STOP, WAIT STOP, INIT, WAIT or RECORDING)	
CONTRol:TREND:SCHEdUle #an<data>	Set Trend Recording Schedule parameters (see .trs file).	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TREND:SCHEdUle?	Get Trend Recording Schedule parameters (see .trs file).	
CONTRol:TREND:SETUP{[1] 2 3 4} #an<data>	Set Trend Recording User Profile parameters (see .trup file).	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TREND:SETUP{[1] 2 3 4}?	Get Trend Recording User Profile parameters (see .trup file).	
CONTRol:TREND:SETUP:QUICKstart #an<data>	Set Trend Quickstart setup parameters	<Data> is sent or received using IEEE488.2 block format. <Data> is formatted using JSON standard.
CONTRol:TREND:SETUP:QUICKstart?	Get Trend Quickstart setup parameters	

CONTRol:TREND:STATE <STOP START QUICKstart>	Set Trend Recording state	<Data> is sent or received using IEEE488.2 block format.
CONTRol:TREND:STATE?	Get Trend Recording state (STOP, WAIT STOP, INIT, WAIT or RECORDING)	<Data> is formatted using JSON standard.
CONTRol:USER[:PROFILE] <CURRENT USER1 USER2 USER3 MONITOR>, #an<data>	Set the complete configuration for user specified. User named "MONITOR" is used to set monitor mode setup.	<Data> is sent or received using IEEE488.2 block format.
CONTRol:USER[:PROFILE]? <CURRENT USER1 USER2 USER3 MONITOR>	Get the complete configuration for user specified.	<Data> is formatted using JSON standard.
CONTRol:USER:SET < USER1 USER2 USER3>	Set current user	<Data> is sent or received using IEEE488.2 block format.
CONTRol:USER:SET?	Get current user	<Data> is formatted using JSON standard.

4.3. HCOpy subsystem

Command	Description	Comments
HCOpy:SDUMp[:IMMediate] "filename"	Create a Screenshot file named "filename" in the directory "Screenshot"	

4.4. HELP subsystem

Command	Description	Comments
HELP[?] [<subsystem>]	Display content of a subsystem	

4.5. MEASurement subsystem

The last argument of each command defined here, is a Boolean used to format output stream:

- If 0, expended format: indentation with SPACE characters.
- If 1, compressed format: no indentation.

Command	Description	SCPI-99	Comments
MEASurement[:STANDard] <meas "all">,<0 1>	Value of a standard measurement.	no	See note ¹
MEASurement:ENERGY <meas "all">,<0 1>	Value of an energy measurement.	no	See note ²
MEASurement:HARMonic <meas "all">,<0 1>	Value of an harmonic measurement.	no	See note ³
MEASurement:INTERharmonic <meas "all">,<0 1>	Value of an inter-harmonic measurement.	no	See note ⁴
MEASurement:POWER <meas "all">,<0 1>	Value of a power measurement	no	See note ⁵
MEASurement:WAVEform <meas "all">,<0 1>	Value of a waveform measurement	no	See note ⁶

Notes:

Reply contains all the measurements corresponding to the request:

- the different channels,

1 -<meas> in MEASurement[:STANDard] command:

"ACF", "AD", "ADC", "APKM", "APKMMIN", "APKP", "APKPMAX", "ARMS", "AUNBN", "AUNBZ", "FHL", "FK", "HZ", "KF", "PHIA", "PHIU", "PHIV", "PLT", "PST", "PSTI", "UCF", "UD", "UDC", "UMS1", "UMS2", "UPKP", "UPKM", "UPKMMIN", "UPKPMAX", "URMS", "UUNBN", "VCF", "VD", "VDC", "VMS1", "VMS2", "VPKP", "VPKM", "VPKMMIN", "VPKPMAX", "VRMS", "VUNBN", "VUNBZ".

2 -<meas> in MEASurement:ENERGY command: "DHC", "DHG", "NHC", "NHG", "PDCHC", "PDCHG", "PHC", "PHG", "Q1HCC", "Q1HCG", "Q1HIC", "Q1HIG", "SHC", "SHG".

3 -<meas> in MEASurement:HARMonic command: "AH", "AHANG", "ATHDE", "ATHDF", "ATHDR", "UBIN", "UBINMAX", "UH", "UHANG", "UTHDF", "UTHDR", "VHANG", "VH", "VTHDF", "VTHDR", "VTHDE",.

4 -<meas> in MEASurement:INTERharmonic command: "AIH", "UIH", "VIH".

5 -<meas> in MEASurement:POWER command: "D", "DH", "DI", "DV", "LUNB", "N", "P", "P1", "P1P", "PDC", "PF", "PF1", "PF1P", "PFNF", "PH", "PHIVA", "Q1", "Q1P", "S", "S1", "S1P", "S1U", "SH", "SNF", "TAN".

6 -<meas> in MEASurement:WAVEform command: "ASAMPLES", "USAMPLES", "VSAMPLES".

■ min and max values,

■ Measurement values acquired in different conditions (10.12c, 150/180c, ...)

Every measurement values available at the time the request is issued, are inserted in the reply.

Parameter "all" may be used to request the entire set of measurements supported by the command.

To the command:
Meas "VRMS", 0

CA 834XX replies:

```
#46280{  
  "Vrms": {  
    "L1": {  
      "Format": "float",  
      "Unit": "V",  
      "Value": {  
        "0.5c": [  
          56.779167175292969,  
          56.779727935791016,  
          56.779239654541016,  
          56.781730651855469,  
          56.783107757568359,  
          56.779945373535156,  
          56.778038024902344,  
          56.779514312744141,  
          56.780746459960938,  
          56.734840393066406,  
          56.778610229492188,  
          56.781661987304688,  
          56.779499053955078,  
          56.777797698974609,  
          56.777206420898438,  
          56.776588439941406,  
          56.776500701904297,  
          56.777824401855469,  
          56.778484344482422,  
          56.779712677001953,  
          56.738048553466797,  
          56.780677795410156,  
          56.779262542724609,  
          56.780769348144531  
        ],  
        "10/12c": 56.774085998535156,  
        "150/180c": 56.773025512695312  
      }  
    },  
    "L2": {  
      "Format": "float",  
      "Unit": "V",  
      "Value": {  
        "0.5c": [  
          54.825035095214844,  
          54.823825836181641,  
          54.826683044433594,  
          54.827682495117188,  
          54.783878326416016,  
          54.822734832763672,  
          54.822456359863281,  
          54.824199676513672,  
          54.823902130126953,  
          54.822734832763672,  
          54.823078155517578,  
          54.825008392333984,  
          54.824607849121094,  
          54.822994232177734,  
          54.821887969970703,  
          54.821372985839844,  
          54.821834564208984,  
          54.823970794677734,  
          54.822711944580078,  
          54.823249816894531,  
          54.822734832763672,  
          54.825424194335938,  
          54.827865600585938,  
        ]  
      }  
    }  
  }  
}
```

The value entry can contain different measurements.
Each measurement is done with a particular setup:

- 0.5 c: half cycle (array of 24 items)
- 200 ms: 200 milliseconds
- 10/12 c: 10 cycles 50 Hz or 12 cycles 60 Hz
- 150/180 c: 150 cycles 50 Hz or 180 cycles 60 Hz
- 3 s: 3 seconds
- 10 s: 10 seconds
- 10 min: 10 minutes
- 2 h: 2 hours

If a specific value is not available, the corresponding entry is not present.

```

54.825309753417969
],
"10/12c": 54.819549560546875,
"150/180c": 54.817050933837891
}
},
"L3": {
  "Format": "float",
  "Unit": "V",
  "Value": {
    "0.5c": [
      54.617996215820312,
      54.618431091308594,
      54.620468139648438,
      54.580772399902344,
      54.62103271484375,
      54.617988586425781,
      54.617000579833984,
      54.618938446044922,
      54.619003295898438,
      54.617263793945312,
      54.617404937744141,
      54.619277954101562,
      54.618003845214844,
      54.617473602294922,
      54.618869781494141,
      54.616916656494141,
      54.616424560546875,
      54.617504119873047,
      54.618007659912109,
      54.620368957519531,
      54.620456695556641,
      54.619064331054688,
      54.619041442871094,
      54.618568420410156
    ],
    "10/12c": 54.614303588867188,
    "150/180c": 54.611812591552734
  }
},
"N": {
  "Format": "float",
  "Unit": "V",
  "Value": {
    "0.5c": [
      0.14312368631362915,
      0.13556264340877533,
      0.14063639938831329,
      0.13865818083286285,
      0.13893310725688934,
      0.14214582741260529,
      0.13507276773452759,
      0.14171944558620453,
      0.13909289240837097,
      0.13703462481498718,
      0.13978621363639832,
      0.13364340364933014,
      0.14280261099338531,
      0.14027078449726105,
      0.13718436658382416,
      0.14159682393074036,
      0.13494111597537994,
      0.14283508062362671,
      0.14014379680156708,
      0.13619793951511383,
      0.13944767415523529,
      0.13287004828453064,
      0.14159274101257324,

```

```
        0.13876959681510925
      ],
      "10/12c": 0.13956716656684875,
      "150/180c": 0.13931198418140411
    }
  },
  "VrmsMax": {
    "L1": {
      "Format": "float",
      "Unit": "V",
      "Value": {
        "10/12c": 56.91571044921875
      }
    },
    "L2": {
      "Format": "float",
      "Unit": "V",
      "Value": {
        "10/12c": 54.910678863525391
      }
    },
    "L3": {
      "Format": "float",
      "Unit": "V",
      "Value": {
        "10/12c": 54.709587097167969
      }
    },
    "N": {
      "Format": "float",
      "Unit": "V",
      "Value": {
        "10/12c": 0.12526048719882965
      }
    }
  },
  "VrmsMin": {
    "L1": {
      "Format": "float",
      "Unit": "V",
      "Value": {
        "10/12c": 56.477264404296875
      }
    },
    "L2": {
      "Format": "float",
      "Unit": "V",
      "Value": {
        "10/12c": 54.613021850585938
      }
    },
    "L3": {
      "Format": "float",
      "Unit": "V",
      "Value": {
        "10/12c": 54.402042388916016
      }
    },
    "N": {
      "Format": "float",
      "Unit": "V",
      "Value": {
        "10/12c": 0.12526048719882965
      }
    }
  }
}
```

4.6. MMEMory subsystem

Command	Description	SCPI-99	Comments
MMEMory:CATalog? ["directory_name"]	<p>Returns information about the contents of a directory.</p> <ul style="list-style-type: none"> The argument "directory_name" is optional. If not defined, the command uses the current working directory defined with "MMEM:CDIR". If the first character of "directory_name" is '/' then path is absolute; if not, path is relative to current working directory. <p>CA834X returns its directory information in the following format: <i><numeric value>, <numeric value> {<file_entry>}</i>.</p> <ul style="list-style-type: none"> The first parameter indicates the total amount of storage currently used (bytes). The second parameter the total amount of storage available on the SDCARD (bytes). Each <i><file_entry></i> indicates the name, type and size of one file in the directory list: <i><file_name>, <file_type>, <file_size></i>. <p>The <i><file_type></i> can be one of the following:</p> <ul style="list-style-type: none"> ASCIi: An ASCII text file, BINary: A binary file, DIRectory: Name of a directory. In this case, <i><file_size></i> is the total size of the directory on the mass storage (not the sum of the size of each file in the directory). In case a record is running, this size is not computed (fixed to 0). 	Inspired from §14.1	The value "amount of storage available on the device" is computed from the real free space and the memory reserved to store TREND record if it has been programmed.
MMEMory:CDIRectory ["directory_name"]	<p>Changes the default directory name for a mass memory file system.</p> <p>"directory_name" indicates:</p> <ul style="list-style-type: none"> an absolute pass if the first character is '/'. a relative path to current default directory if first character is not '/'. 	§14.2	"directory_name" is optional. If not defined, its value is fixed to ""
MMEMory:CDIRectory?	Returns the default directory for a mass memory file system.	no	
MMEMory:DATA "filename",#an<data>	<p>Loads <i><data></i> into the file "<i>filename</i>" on the default mass storage.</p> <p><i><data></i> is coded in IEEE 488.2 block format.</p> <p>"filename" can be an absolute pass to a file ex: <i>/Qualistar_2/Alarm/Alarm.icp</i> or a relative pass from current working directory ex: <i>Alarm/Alarm.icp</i> if current directory is <i>/Qualistar-2</i></p>	§14.5	
MMEMory:DATA? "filename"	<p>Returns the content of the file <i><filename></i> in IEEE 488.2 block format</p> <p>"filename" indicates an absolute pass to a file or a relative pass from current directory;</p>	§14.5	
MMEMory:DELeTe "filename"	<p>Removes a file form the specified mass storage;</p> <p>"filename" indicates an absolute pass to a file or a relative pass from current working directory;</p> <p>if "<i>filename</i>" is a directory, every element in this directory are suppressed.</p>	§14.6	
MMEMory:STATus? <sdcard usb>	<p>Return informations about device:</p> <p>CA 834X returns its informations in the following format: <i><boolean value>, <numeric value>, <numeric value></i>.</p> <p>The first parameter value is '1' if the device is mounted,</p> <p>The second parameter indicates the total free space on the device(bytes),</p> <p>The third parameter indicates the amount of storage available on the device (bytes)</p>	-	The value "total free space on the device" doesn't take into account the memory necessary to store TREND record if it has been programmed.

4.7. SYSTEM subsystem

Command	Description	SCPI-99	Comments
SYSTEM:DATE?	Get date of internal RTC	§21.7	
SYSTEM:ERROR[:NEXT]?	Queries the error/event queue for the next item and removes it from the queue	§21.8.5.2	
SYSTEM:PASSWORD	A password can be set to prevent modification of the instrument configuration (from the front panel and from the remote control). The command used without an argument unlocks the access to the configuration.	§21.15	
SYSTEM:PASSWORD <password>			
SYSTEM:PASSWORD:STATE?	Returns "true" if a password has been defined to prevent modification of the instrument configuration.	§21.15,2,1	
SYSTEM:TIME?	Get time of internal RTC	§21.19	

5. REFERENCES

- <https://www.ivifoundation.org/docs/SCPI-99.PDF>

FRANCE

Chauvin Arnoux
12-16 rue Sarah Bernhardt
92600 Asnières-sur-Seine
Tél : +33 1 44 85 44 85
Fax : +33 1 46 27 73 89
info@chauvin-arnoux.com
www.chauvin-arnoux.com

INTERNATIONAL

Chauvin Arnoux
Tél : +33 1 44 85 44 38
Fax : +33 1 46 27 95 69

Our international contacts
www.chauvin-arnoux.com/contacts

