XL2

Remote Measurement

Reference Manual V3.00

Refers to XL2 firmware version 3.00 or higher
March 2015

Table of contents

Introduction

Purpose of the XL2 Remote Measurement

The XL2 Remote Measurement option enables you to query your XL2 measurement data from your PC via the USB interface, allowing you to program your own measurement application on your PC, e.g. for sound level monitoring or automated measurement tasks. The following XL2 measurement functions are supported:

- Sound level meter and spectrum analyzer SLMeter
- Audio analyzer RMS/THDN

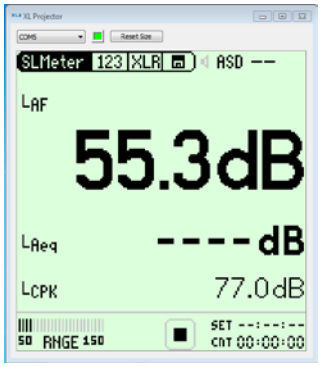
NOTE – The following functions are not supported with the Remote Measurement Option:

- **access to the XL2 file system**
- **setup of data presentation on your XL2 LCD**
- **logging and reporting to the internal SD card**

Furthermore, the current XL2 hardware does not support remote power-up of the XL2. Please contact NTi Audio for available workarounds.

XL2 Projector

The Projector displays your XL2 screen, in real-time via USB, on your PC. The virtual keyboard provides you with control of your XL2 Audio and Acoustic Analyzer from your PC, using mouse clicks.



Your XL2 Analyzer offers the facility to set limits for the maximum permitted sound level e.g. as prescribed by local authorities for live sound monitoring. In case such limits are exceeded, the XL2 Projector background color on your PC turns from green to yellow or red according to your defined limits.

XL2 Projector runs with every XL2, there is no need to have any options installed on the XL2.
The XL2 Projector software is available to you as a free download at www.nti-audio.com/XL2.

NOTE: The XL2 Projector function uses the "COM port" USB mode. Click the SD-Card icon in the XL2 Projector to switch the XL2 to "Mass Storage" mode and access the XL2 files through your PC file system.

Remote Measurement or Type Approved Option required

To query measurement results from the XL2 remotely, the XL2 must be equipped with a Remote Measurement or Type Approved option.

NOTE – If the XL2 Remote Measurement option is not installed, the instrument will respond to some basic commands like *IDN?, but will answer with the error "Parameter not available, license not installed" (Error No. 5).

Driver

The XL2 Analyzer communicates with the PC via the USB interface utilizing a virtual COM port. The required drivers will automatically be installed with the XL2 Projector software.

Getting Started

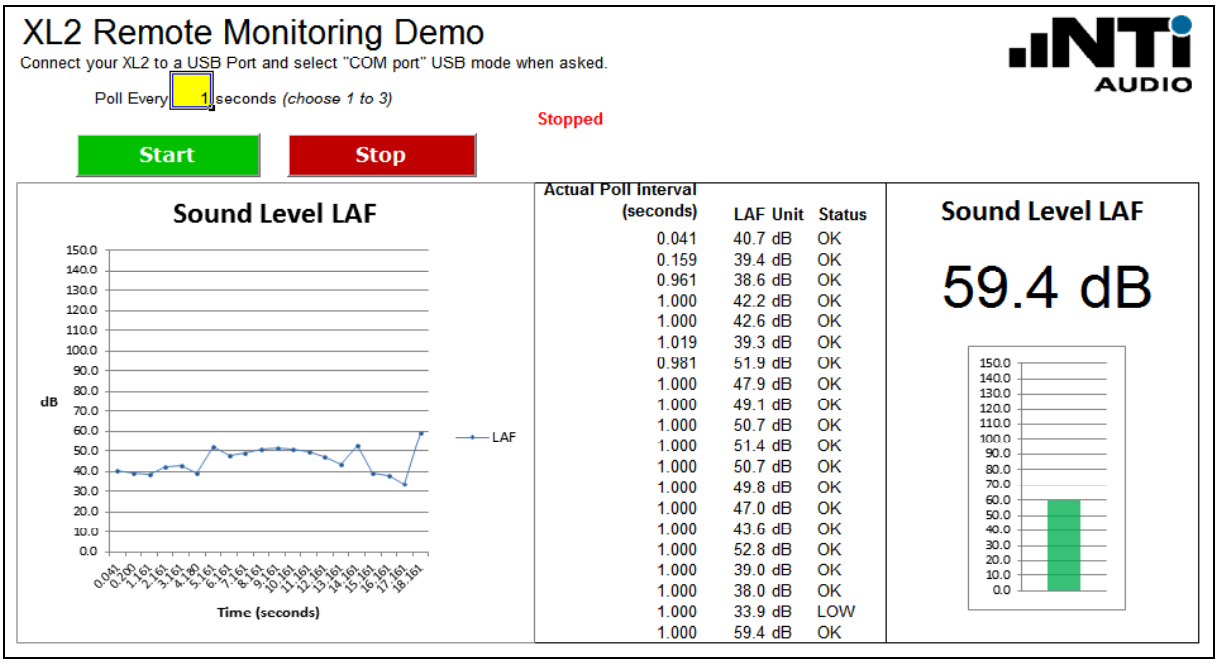
Requirement: Make sure that you have installed the XL2 Projector on your system. Together with the XL2 Projector, the USB serial driver is installed, which is required for the Remote Measurement option of the XL2.

You receive the following "Getting Started" tools:

MICROSOFT EXCEL DEMO APPLICATION

This demo, written in visual basic for applications (VBA), queries XL2 data online into MS Excel and charts the sound level graph.

1. Connect the XL2 to your computer
2. On the XL2, select "COM port" at the pop-up USB Mode
3. On your PC, open the file "XL2 Remote Monitoring Demo.xls"
4. Click the start button in the software -> the XL2 is started and logging starts on the PC screen.
5. The demo stops automatically after 20 log lines.



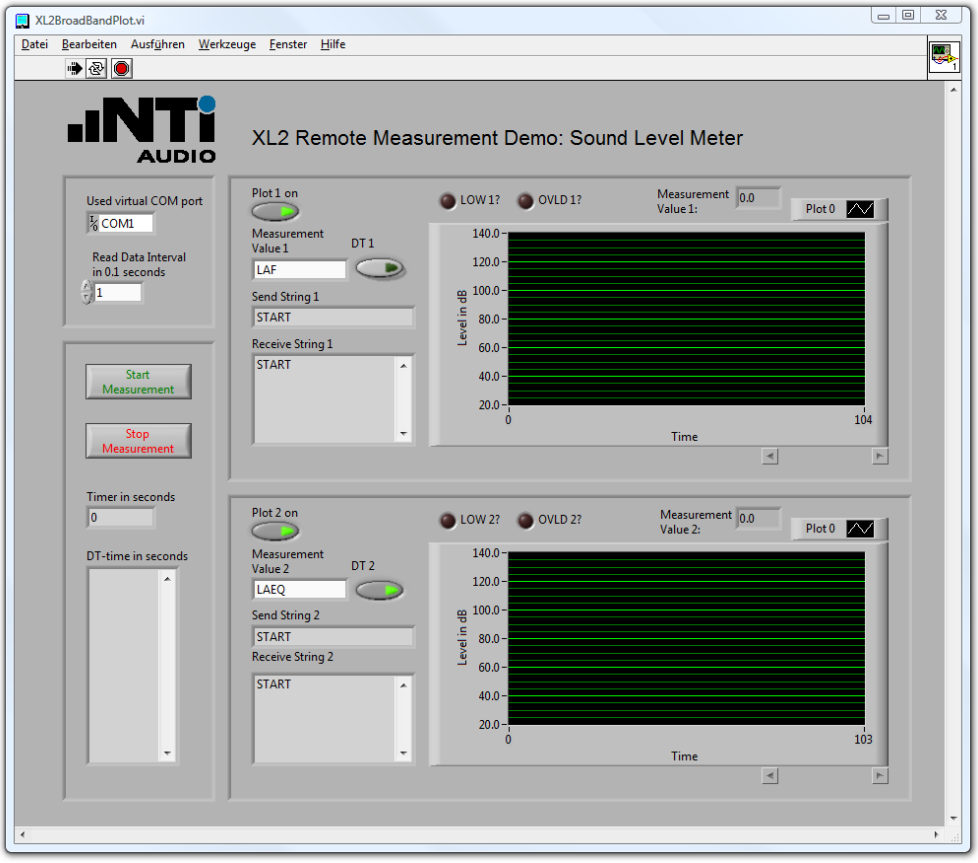
On your PC, press the Alt-F11 keys to access the open source code and extend the functionalities according your individual requirements.

LABVIEW DEMO APPLICATION

You can use the demo application in two ways. Use either the runtime version (runs without LabVIEW), or the LabVIEW source files (requires a LabVIEW 10 basic license).

RUNTIME VERSION

6. Select the folder "LabViewDemo\RuntimeInstall"
7. Start "setup.exe" and follow the instructions on the screen. As soon as the installation has successfully completed, a shortcut is available in the Windows start menu.
8. Start "XL2SLMeterRemote" under "All Programs → NTi Audio".



9. In case any error messages are displayed, simply continue with these instructions.

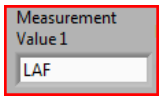
10. Connect your XL2
    a) Connect your XL2 to your PC via USB and power-up the XL2. The XL2 displays the **USB Mode** window.
    b) Select **COM port** on your XL2.
    c) As soon as your XL2 is connected to your PC, the "Used virtual COM port" changes from "COM1" to another COM port.

NOTE – If the COM port used is higher than COM9, then the application will not open the COM port. In this case you need to change the number of the COM port in the windows device manager to a number lower than 10.

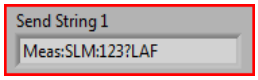
11. Upper plot
    The default measurement value is LAF. The data is immediately displayed in the upper plot area.

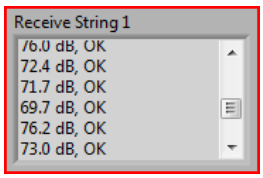
12. Lower plot
    The default measurement value is LAEQ. Press the "Start Measurement" button to display the measurement results in the lower plot area.

13. Change measurement value
    The plotted measurement value can be changed by typing the value name into the "Measurement Value" field of the plot below. A list of valid value names is listed in the section "MEASure:SLM:123?" of this manual. Some measurement values require the Extended Acoustics Pack option of the XL2. To get a dt-value of a measurement, click the displayed dt button. For more details please refer to section "MEASure:SLM:123:dt?" in this manual.

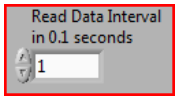


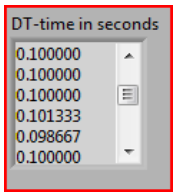
14. Commands
    The "Send String" of a plot shows the complete string, which is sent to your XL2 to acquire the measurement value. The "Receive String" of a plot shows all received measurement values. Use the scrollbar to view all values.





15. Setting time parameter
    In the "Read Data Interval" you can change the speed of acquiring data from your XL2. The minimum interval is 0.1 second, represented by a value of 1. Enter the value 20 to acquire measurement results every two seconds. The “Timer in seconds” shows the current measurement timer value of your XL2. The “dt-time in seconds” shows the measured time between two consecutive measurement value requests. Use the scroll bar to view all request intervals.





LABVIEW SOURCE FILE VERSION

Open the file “XL2BroadBandPlot.vi” in the “LabViewDemo\SourceFiles\SLM” folder to run the application with LabVIEW. The application works as described in the Runtime Version section.

An additional demo is included as a source file, which plots the level RMS , THD+N + frequency time sweep. In order to run the THD demo application, open the File “XL2ThdPlot.vi in the ”LabViewDemo\SourceFiles\THD” folder.

DEMOS WITH TERMINAL PROGRAM

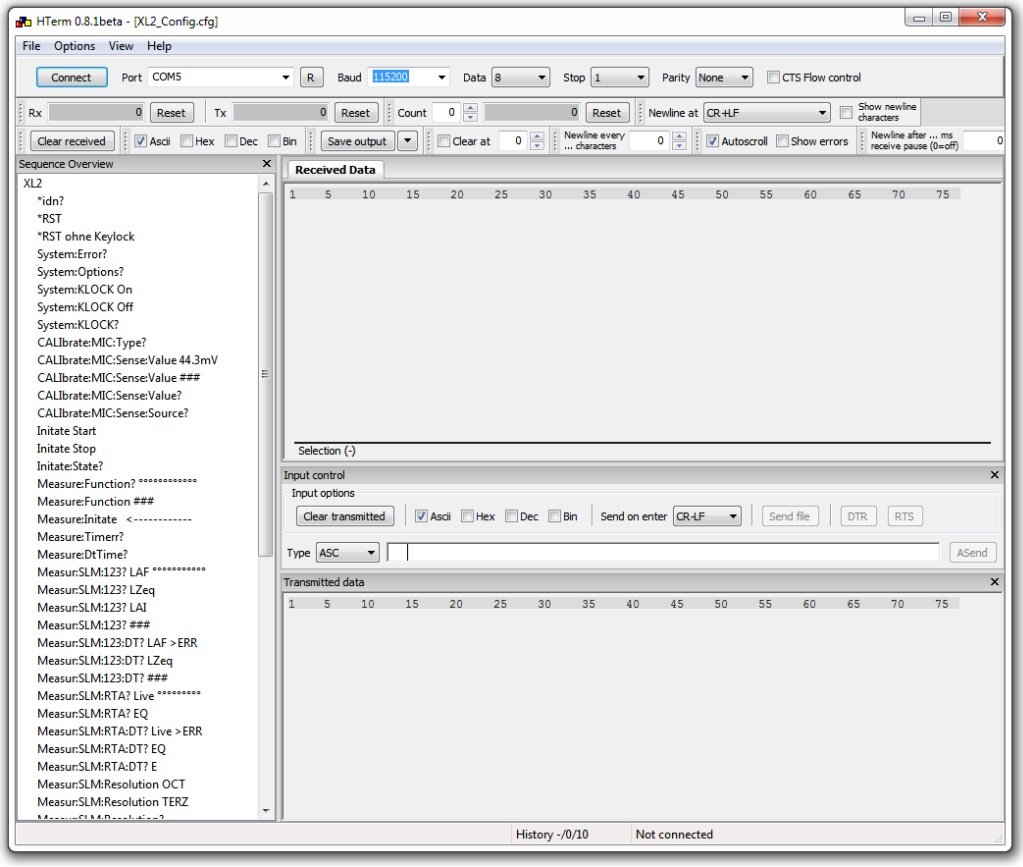
The terminal program "HTerm" with the configuration file "XL2_Config.cfg" offers a quick introduction to handling the XL2 Remote Measurement commands.

Requirement: Make sure that you have installed the XL2 Projector software on your PC, thereby ensuring that the serial driver, required for remote measuring, is available. Verify that the XL2 Projector software functions correctly prior to continuing with the next steps.

1) Start the terminal program HTerm.exe:

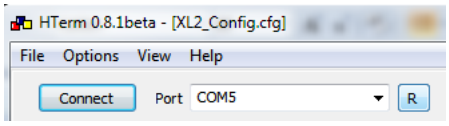
2) Load the configuration File "XL2_Config.cfg"

3) HTerm displays the loaded XL2 configuration file:

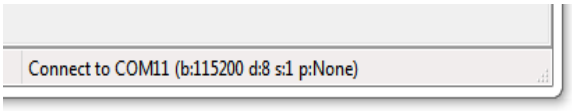


4) Connect your XL2
   a) Connect your XL2 via USB to your PC and power-up the XL2. The XL2 displays the **USB Mode** window.
   b) Select **COM port** on your XL2.

5) Selecting the COM port in HTerm:
   a) Press the R button to refresh the port list



   b) Select the COM port used to communicate with your XL2 (e.g. your PC displays the com port assigned to the XL2 during the initial connection to your XL2).
   c) Press "Connect" and wait for the status information of HTerm in the bottom line; as soon as it is connected successfully, the status line should show something like this:



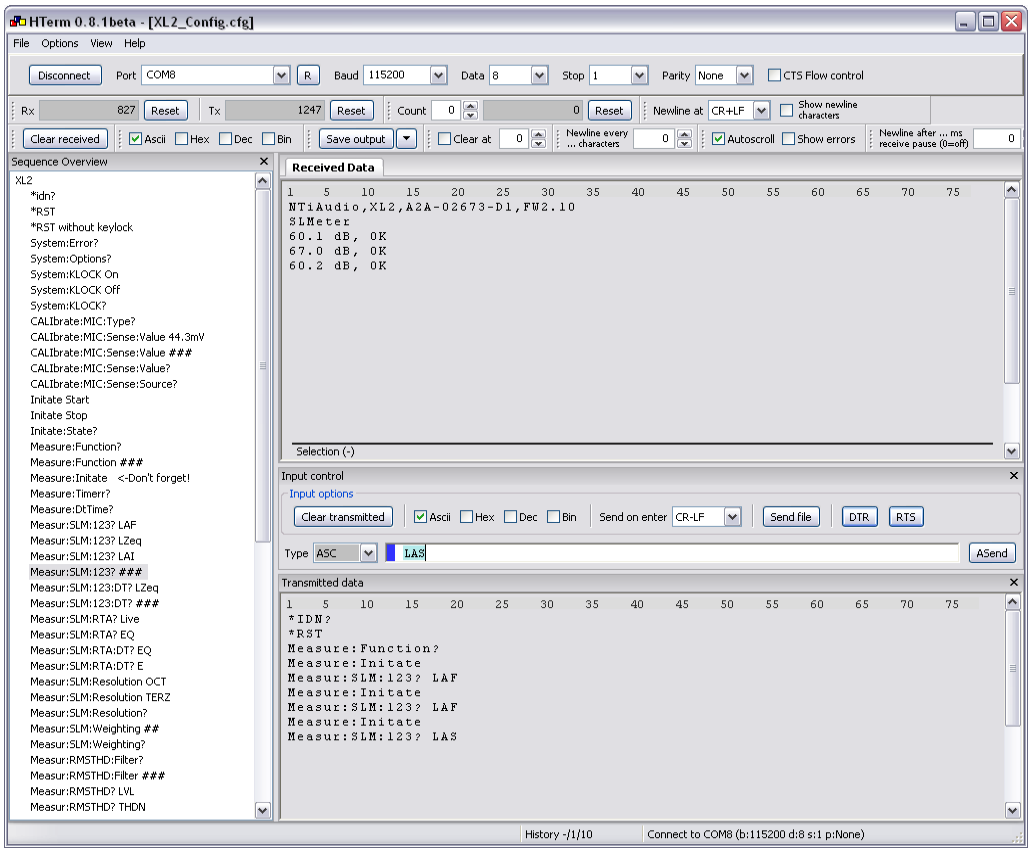
6) First communication with your XL2:
   a) Double-click on "*idn?" in the "Sequence Overview" window on the left hand side.
   b) "*IDN?" is shown on the transmitted data window.
   c) "NTiAudio, XL2, A2A-xxxx-D1, FW2.xx" is shown in the Received Data window.

7) Reset your XL2 to a defined status
   a) Execute the "*RST" command to set your XL2 to a defined state. The RST command
      i) clears the error queue
      ii) stops any running measurement
      iii) exits any active profile
      iv) selects the SLMeter function
      v) resets parameters
      vi) locks the keyboard

8) Read measurement function
   a) Double-click "MEASure:Function?"
   b) The Received Data window shows "SLMeter"

9) Read measurement data
   a) Double-click "MEASure:INITiate", this reads all the actual measurement results for post-processing on your PC.
   b) Double-click "MEASur:SLM:123: LAF?"
   c) The Received Data window shows e.g. "70.1 dB, OK" (= the live sound level from "MEASure:INITiate" before)
   d) Double-click any other parameter to read out further measurement results taken by the "MEASure:INITiate" command.

10) Read measurement data
    a) Any measurement results not listed in the HTerm sequence overview can be read using the commands with "###" (= placeholders for individual input values).
    b) Double-click "MEASure:INITiate"
    c) Double-click "MEASur:SLM:123: ###"
    d) Add the required characters into the "Input control" window and press ENTER. In the example below, the value "LAS" is queried:



11) Stopping the remote measurement
    a) Click the button "Disconnect" in HTerm
    b) Disconnect your XL2 from the USB connection to your PC.

Commands

Command Structure

The commands are send in ASCII format through the virtual COM port to your XL2 Audio and Acoustic Analyzer. Every command transmission from your PC to your XL2 or vice versa must be terminated with "CR LF" (**C**arriage **R**eturn, **L**ine **F**eed).

The measurement commands are divided into six groups (i.e. "subsystems").

Subsystem	***Function***
*	Device status commands
INITiate	Status control for a measurement
MEASurement	Measurement result query commands
INPUt	Settings for Input signal path
CALIbrate	Microphone Calibration commands
SYSTem	System status commands

- The XL2 accepts the *short* or any variant of the *full* form of the commands.
- In the command list, the CAPITAL letters indicate the *short* form. However, the XL2 accepts both lowercase and UPPERCASE letters, i.e. commands are not case-sensitive.
- Multiple commands separated by semi-colons (";") are not supported.
- Errors are stored in an error queue and can be queried with the "SYSTem:ERROr?" command.

Command Notation & Descriptive Symbols

The XL2 command descriptions use headings to divide the syntax information into easily-readable parts. These headings and their meaning are listed below. If a heading does not apply to a command, it does not appear in the command syntax description.

Usage	What the command does
Availability	The mode and system settings that must be active to execute the command
Parameter	The parameters to be set and their types
Answer	The possible answer(s) to a query command
Examples	Command examples are provided here. Short form and lowercase characters are randomly altered to remind the reader that both forms are allowed
Explanation	Additional explanations, hints and notes

The subsequent table lists the symbols that are used for the command description.

Symbol	***Description***
:	Colons separate elements of an XL2 command.
[ ]	Square brackets enclose the *list of available parameters*, out of which 1 parameter must be selected.
¦	A vertical line reads as an "OR", i.e. this sign separates *alternative* parameters.
< >	Triangle brackets enclose the *variable parameters* that must be set for a user-defined value.
{ }	Braces have the same meaning as triangle brackets ("< >"), except that the enclosed parameters can be included *several* times.
,	Commas separate arguments in an arguments list.
?	The question mark indicates a *query* command.
( )	Round brackets enclose comments.
💻 →	The string is sent from your PC to your XL2.
→ 💻	The string is returned from your XL2 to your PC.

NOTE – If a value is undefined, the XL2 returns the message -999.

My first program (a typical program skeleton)

When starting to write a program to query values from the XL2, we suggest using the following skeleton (code is written in Python 2.7):

```
1   import time
2   import serial
3
4   # Query the Device Manager of your Windows PC to find out which COM port the
5   # system assigned to the XL2 and adapt the following line:
6   COM_PORT = "COM15"
7
8   xl2 = serial.Serial(COM_PORT, timeout=1)
9
10  xl2.write('*RST\n')             # Reset the XL2 to default state (SLMeter, ...)
11  xl2.write('INIT START\n')       # Start the measurement
12  time.sleep(3)                   # Allow the XL2 to start the measurement
13
14  for i in range(10):
15      xl2.write('MEAS:INIT\n')              # Triggers a measurement
16      xl2.write('MEAS:SLM:123? LAS\n')      # Query LAS
17      result = xl2.readline()
18      print result,
19      time.sleep(1)
20
21  xl2.write('INIT STOP\n')        # Stop the measurement (optional)
22  xl2.close()
```

Output of the program:

```
36.0 dB, OK
34.8 dB, OK
48.8 dB, OK
44.7 dB, OK
53.4 dB, OK
49.4 dB, OK
45.3 dB, OK
41.8 dB, OK
39.3 dB, OK
38.0 dB, OK
```

Device Status

*IDN?

Shortcut Identification: reads the unique identification of the XL2.

Availability always

Answer

```
<Manufacturer>,        string
<Unit>,
<Serial Number>,
<FW Version>
```

Example

💻→ `*IDN?`
→💻 `NTiAudio,XL2,A2A-12345-D0,FW2.03`

*RST

Shortcut Executes a device reset, and should be the first command when starting a remote session to ensure that all XL2 settings make sense for remote measuring.

Availability always

Example 💻→ `*RST`

Details It is highly recommended to execute this command first to avoid unwanted side effects.
The RST command

- clears the error queue
- stops any running measurement
- stops any running script
- exits any active profile
- selects the SLMeter function
- sets the following parameters
  - Append mode: OFF
  - Auto save: OFF
  - Logging: OFF
  - Events: OFF
  - Timer mode: CONTINUOUS
  - Range: MID
  - RMS/THDN Filter: Z-WEIGHTED
  - Input: XLR
  - Phantom Power: ON
  - RTA Source: LZF
  - RTA Resolution: TERZ
- locks the keyboard
- sets the precision of queried floating-point numbers to 'LCD'

Debug

ECHO

Shortcut Returns the string after the command including separators. It is for debugging purpose only.

Availability always

Parameter `<text>` *string*

Answer `<text>` *string*

Example
```
💻→ ECHO This is an echo, isn't it:
→💻 This is an echo, isn't it:
```

INITiate Subsystem

INITiate

Shortcut Starts/Stops a measurement

Availability SLMeter, FFT, 1/12 Oct

Parameter `[START|STOP]` *string*

Example
```
💻→ INIT START
```

Details Time dependent parameters like LAeq, LAFmax, etc. are undefined until START has been initiated.
The start procedure may last a few seconds. If required, query INIT:STATE? to see, whether the start procedure is finished.
When a measurement is stopped with STOP, the calculation of time dependent parameters is stopped and the result stays constant.

INITiate:STATe?

Shortcut Queries the run status of a measurement

Availability always

Answer `[STOPPED|FROZEN|SETTLING|RUNNING|PAUSED]` *string*

Example
```
💻→ INIT:STATE?
→💻 RUNNING
```

MEASure Subsystem

MEASure:FUNCtion

Shortcut Defines the active measurement function

Availability always

Parameter

`[SLMeter|FFT|RT60|Polarity|Delay|RMS/THD|N.Rating|Scope|1/12Oct|STIPA|Calibrte|System]`	*string*	only the first two characters are necessary

Example 💻→ `MEASURE:FUNCTION SLMETER`

Details Switching between measurement functions may last 1-2 seconds.

MEASure:FUNCtion?

Shortcut Queries the active measurement function

Availability always

Answer

`[SLMeter|FFT|RT60|Polarity|Delay|RMS/THD|N.Rating|Scope|1/12Oct|STIPA|Calibrte|System]`	*string*	If the Type Approved Firmware is running the SLMeter returns "SLM TA" instead of "SLMeter"

Example
💻→ `MEASURE:FUNCTION?`
→💻 `SLMeter`

MEASure:INITiate

Shortcut Triggers a measurement

Availability always

Example 💻→ `MEAS:INIT`

Details All measurements results of the MEASure subsystem are stored synchronously by this command.
Before the first MEAS:INIT has been sent, all measurement values are undefined.
A typical workflow is

```
*RST
INIT START
MEAS:INIT
MEAS:SLM:123? <para1>
MEAS:SLM:123? <para2>
MEAS:INIT
MEAS:SLM:123? <para1>
MEAS:SLM:123? <para2>
...
```

MEASure:TIMEr?

Shortcut Queries the actual measurement timer value.

Availability SLMeter

Answer

`<timer>`	*float*	0.1 seconds resolution (1 decimal)
`sec, [OK|UNDEF]`	*string*	

Example

```
💻→ MEAS:INIT
     MEAS:TIMER?
→💻 3765.4 sec, ok
```

Details This represents the time since initiating START.

MEASure:DTTIme?

Shortcut Queries the time period used for the calculation of dt values. The value is active as long as the measurement is RUNNING, and is reset after each INIT:MEAS or INIT START command.

Availability SLMeter, when RUNNING

Answer

`<timer>`	*float*
`sec, [OK|UNDEF]`	*string*

Example

```
💻→ INIT START
     MEAS:INIT
     MEAS:DTTIme?
→💻 2.156522 sec, ok
```

Details This exact time information is required to correctly combine EQ values acquired with the remote interface. In contrast to the XL2 internal logging, where the time interval between log lines is equidistant and therefore the dt time is not required when combining LEQ_dt values, measurements acquired with the remote interface have a certain time jitter. For combining EQ_dt values that are not equidistant, the accurate dt time of each LEQ_dt value is needed.
RECOMMENDATION: an LEQ of any time period can be calculated by summing up LE values and then calculate the LEQ = LE – 10*log(PERIODE[sec]). For doing so, the dt time is not necessary.

MEASure:DECImals

Shortcut Defines the precision of queried floating-point numbers.

Availability Always

Parameter `[LCD|EXTENDED]` *string* only the first character is necessary

Example 💻→ `MEASURE:DECI EXTENDED`

Details With the default setting 'LCD' all floating point numbers are returned in the same precision as seen on the units LCD. With 'EXTENDED', two additional digits are returned.

MEASure:DECImals?

Shortcut Queries the precision of queried floating-point numbers.

Availability always

Answer `[LCD|EXTENDED]` *string*

Example 💻→ `MEASURE:DECI?`
→💻 `LCD`

MEASure:SLM Subsystem

MEASure:SLM:123?

Shortcut Queries a broad band measurement result of the SLMeter.

Availability SLMeter

Parameter

`[LxS|LxSMAX|LxSMIN|LxF|LxFMAX|LxFMIN|LxEQ|LxPK|LxPKMAX|LAEQt|LAEQtMAX|k1|k2]`	*string*	x = [A|C|Z] LAEQt|LAEQtMAX: Replace t with [5”|10”|15”|30”|1’|5’|10’|15’|30’|60’]
Additional with installed EAP `[LxI|LxIMAX|LxIMIN|LxE|LAFT3|LAFT3EQ|LAFT5|LAFT5EQ|LAFT5EQ-LAEQ|LAIEQ-LAEQ|LCEQ-LAEQ|LN%]`	*string*	LN%: One of the seven statistic values specified on the "Set EQt, L%" page of the XL2, e.g. L90.0% (if the decimal place is zero you can also use L90%)

Answer

`<Level,> dB,`	*float*	
`[OK|UNDEF|LOW|OVLD|OPTION_REQUIRED]`	*string*	

Example

```
💻→ INIT START
    MEAS:INIT
    MEAS:SLM:123? LASMAX
→💻 53.8 dB, OK
```

Details

Returns a broad band result parameter that has been stored by the last MEAS:INIT command.
If the parameter is unknown, a ";" is returned.
Statistic Values:
For custom setting use the custom values to read, e.g. MEAS:SLM:123? L33.3%
Remotely changing/reading the settings is not implemented.
Be aware of the decimal separator. Use the setting from the “System Settings” page.

Call with multiple parameters Queries up to 10 parameters with one command. Parameters have to be separated by a blank character.

Example

```
💻→ INIT START
    MEAS:INIT
    MEAS:SLM:123? LASMAX LAFMAX LZSMAX LZFMAX
→💻 52.1 dB, OK
    54.8 dB, OK
    63.7 dB, OK
    65.3 dB, OK
```

Details This command reduces the XL2 load when reading several values in short intervals (e.g. 0.1sec)

MEASure:SLM:123:dt?

Shortcut Queries a broad band dt measurement result of the SLMeter.

Availability SLMeter

Parameter

`[LxSMAX|LxSMIN|` `|LxFMAX|LxFMIN|` `LxEQ|LxPKMAX|`	*string*	x = [A|C|Z]
Additional with installed EAP `[LxIMAX|LxIMIN|LxE|`	*string*	

Answer

`<Level,> dB,`	*float*
`[OK|UNDEF|LOW|OVLD|` `OPTION_REQUIRED|NO_DT_VALUE]`	*string*

Example

```
💻→ INIT START
     MEAS:INIT
     MEAS:SLM:123:dt? LASMAX
→💻 53.8 dB, OK
```

Details Queries a broad band result parameter of the SLMeter that has been stored with the last MEAS:INIT command. dt measurements are cleared after each MEAS:INIT, so this function returns the e.g. LEQ between two MEAS:INIT commands. The values have the same meaning as the dt values found in XL2 log files.

If the parameter is unknown, a ";" is returned.

Call with multiple parameters Queries up to 10 Parameters with one command. Parameters have to be comma separated.

Example

```
💻→ INIT START
     MEAS:INIT
     MEAS:SLM:123:dt? LASMAX,LAFMAX,LZSMAX,LZFMAX
→💻 52.1 dB, OK
     54.8 dB, OK
     63.7 dB, OK
     65.3 dB, OK
```

Details This command reduces the XL2 CPU load when reading several values in short intervals (e.g. 0.1sec)

MEASure:SLM:RTA?

Shortcut Queries the spectral results of the SLMeter.

Availability SLMeter

Parameter

`[LIVE|MAX|MIN|EQ|CAPT]`	*string*	
Additional with installed EAP `[E|N%]`		N%: One of the seven statistic values specified on the "Set EQt, L%" page of the XL2, e.g. 10.0% (if the decimal place is zero you can also use 10%)

Answer

`{Level`$_n$`,} dB, [OK|UNDEF|OVLD|OPTION_REQUIRED]`	*float* *string*	1/1 Oct: n = 12, f_{start} = 8 Hz 1/3 Oct: n = 36, f_{start} = 6.3 Hz Levels sorted from lowest to highest frequency

Example

```
💻→ INIT START
     MEAS:INIT
     MEAS:SLM:RTA? EQ
→💻 46.3,50.7,34.5,45.4,42.2,37.2,39.0,39.8,32.1,28.5,29.8,
     31.0 dB, LOW
```

Details Queries the spectral results of the SLMeter that have been stored by the last MEAS:INIT command. If the parameter is unknown, a ";" is returned.

MEASure:SLM:RTA:DT?

Shortcut Queries the dt spectral results of the SLMeter.

Availability SLMeter

Parameter

`[EQ|E]`	*string*	

Answer

`{Level`$_n$`,} dB, [OK|UNDEF|OVLD|NO_DT_VALUE]`	*float* *string*	1/1 Oct: n = 12, f_{start} = 8 Hz 1/3 Oct: n = 36, f_{start} = 6.3 Hz Levels sorted from lowest to highest frequency

Example

```
💻→ INIT START
     MEAS:INIT
     MEAS:SLM:RTA:DT? EQ
→💻 46.3,50.7,34.5,45.4,42.2,37.2,39.0,39.8,32.1,28.5,29.8,
     31.0 dB, LOW
```

Details Queries the spectral results parameter of the SLMeter that has been stored by the last MEAS:INIT command. dt measurements are cleared after each MEAS:INIT, so this function returns the LEQ of LE between two MEAS:INIT commands. The values have the same meaning as the dt values found in XL2 log files. If the parameter is unknown, a ";" is returned.

MEASure:SLM:RTA:RESOlution

Shortcut Defines the resolution, in which the RTA results are acquired.

Availability SLMeter

Parameter `[OCT|TERZ]` *string*

Example 💻→ `MEAS:SLM:RTA:RESO TERZ`

Details Command is only accepted when SLMeter is stopped.

MEASure:SLM:RTA:RESOlution?

Shortcut Queries the resolution, in which the RTA results are acquired.

Availability SLMeter

Answer `[OCT|TERZ]` *string*

Example 💻→ `MEAS:SLM:RTA:RESO?`
→💻 `TERZ`

MEASure:SLM:RTA:WEIGhting

Shortcut Defines the frequency and time weighting, in which the RTA results are acquired.

Availability SLMeter

Parameter `[AF|AS|` *string*
`CF|CS|`
`ZF|ZS|`
`XF|XS]`

Example 💻→ `MEAS:SLM:RTA:WEIG ZS`

Details Command is only accepted when SLMeter is stopped.

MEASure:SLM:RTA:WEIGhting?

Shortcut Queries the frequency and time weighting, in which the RTA results are acquired.

Availability SLMeter

Answer `[AF|AS|` *string*
`CF|CS|`
`ZF|ZS|`
`XF|XS]`

Example 💻→ `MEAS:SLM:RTA:WEIG?`
→💻 `ZS`

MEASure:RMSThdn Subsystem

MEASure:RMSThdn?

Shortcut Queries a measurement result from the RMS+THDN meter

Availability RMSTHD

Parameter `[LVL|THDN|THDN,DB|F]` *string*

Answer

`<value>`	*float*

`[V,[OK|UNDEF|OVERLOAD]] |`
`[%,[OK|UNDEF|OVERLOAD]] |`
`[db,[OK|UNDEF|OVERLOAD]]|`
`[Hz,[OK|UNDEF|OVERLOAD]]`
	string

Example

```
💻→ MEAS:INIT
     MEAS:RMST? LVL
→💻 5.184e-6 V,OK
💻→ MEAS:RMST? THDN
→💻 0.0028 %,OK
💻→ MEAS:RMST? THDN,DB
→💻 -94.8 dB, OK
💻→ MEAS:RMST? F
→💻 127.101 Hz,OK
```

MEASure:RMSThdn:FILTer

Shortcut Defines the frequency weighting filter for the RMS/THDN meter.

Availability RMSTHD

Parameter

`[Z-WEIGHTING|A-WEIGHTING|`
`C-WEIGHTING|HP 100Hz|`
`HP 400Hz|HP 19kHz|`
`22.4-22.4k]`
	string	It is sufficient to send the first 5 characters, e.g. "Z-WEI".

Example

```
💻→ MEAS:RMSTHD:FILTER HP 4
```

MEASure:RMSThdn:FILTer?

Shortcut Queries the frequency weighting filter for the RMS/THDN meter.

Availability RMSTHD

Answer

`[Z-WEIGHTING|A-WEIGHTING|`
`C-WEIGHTING|HP 100Hz|`
`HP 400Hz|HP 19kHz|`
`22.4-22.4k]`
	string

Example

```
💻→ MEAS:RMSTHD:FILTER?
→💻 22.4-22.4k
```

MEASure: FFT Subsystem

MEASure:FFT?

Shortcut	Queries a measurement result from the FFT analyzer		
Availability	FFT		
Parameter	`[Live|Max|Min|EQ]`	*string*	
Answer	`{Level`$_n$`,} dB|dBu|dBV|V,` `[OK|UNDEF|OVLD]`	*float* *string*	n = 143 Levels sorted from lowest to highest frequency

Example

```
💻→ MEAS:INIT
     MEAS:FFT? Live
→💻 29.1,24.0,21.6,24.3,24.1,23.2,22.1,20.6,19.6,20.3,18.8,
     19.3,20.7,20.0,21.6,22.6,26.3,29.9,29.5,27.7,25.0,30.8,
     32.4,31.2,27.5,26.4,23.1,25.9,26.6,28.1,29.7,34.7,34.0,
     27.3,27.0,25.9,23.5,21.1,22.0,24.5,25.0,22.1,27.7,28.0,
     24.9,23.4,22.4,21.6,22.6,22.2,22.6,23.5,20.4,17.5,21.7,
     25.6,28.2,31.1,32.0,31.7,26.5,25.2,23.2,22.9,23.5,24.7,
     29.2,33.8,33.9,30.6,27.2,23.4,15.7,22.4,24.1,24.2,23.9,
     26.7,25.9,20.5,15.1,15.0,15.3,23.4,23.8,23.0,23.7,20.4,
     19.0,17.7,12.3,15.2,17.9,18.4,23.1,24.3,23.3,22.6,19.0,
     14.7,12.7,17.4,19.1,18.2,21.2,20.8,17.5,19.4,22.4,19.8,
     17.5,15.1,12.0,11.3,13.9,16.5,17.2,18.2,18.7,18.4,19.6,
     23.2,23.0,20.6,23.4,20.0,15.0,17.6,22.3,22.4,22.7,21.9,
     19.4,17.5,15.4,14.8,14.9,21.9,24.2,21.4,18.7,16.0,12.9
     dB, OK
```

Details

Queries the spectral results of the FFT analyzer that have been stored by the last MEAS:INIT command. If the parameter is unknown, a ";" is returned.

If voltage (V) is selected as reading unit on the XL2, then the result is returned in engineering format (e.g. 1.234e-3)

MEASure:FFT:dt?

Shortcut	Queries a measurement dt result from the FFT analyzer		
Availability	FFT		
Parameter	`EQ`	*string*	
Answer	`{Level`$_n$`,} dB|dBu|dBV|V, [OK|UNDEF|OVLD]`	*float* *string*	n = 143 Levels sorted from lowest to highest frequency

Example

💻→
```
MEAS:INIT
MEAS:FFT:dt? EQ
```
→💻
```
29.1,24.0,21.6,24.3,24.1,23.2,22.1,20.6,19.6,20.3,18.8,
19.3,20.7,20.0,21.6,22.6,26.3,29.9,29.5,27.7,25.0,30.8,
32.4,31.2,27.5,26.4,23.1,25.9,26.6,28.1,29.7,34.7,34.0,
27.3,27.0,25.9,23.5,21.1,22.0,24.5,25.0,22.1,27.7,28.0,
24.9,23.4,22.4,21.6,22.6,22.2,22.6,23.5,20.4,17.5,21.7,
25.6,28.2,31.1,32.0,31.7,26.5,25.2,23.2,22.9,23.5,24.7,
29.2,33.8,33.9,30.6,27.2,23.4,15.7,22.4,24.1,24.2,23.9,
26.7,25.9,20.5,15.1,15.0,15.3,23.4,23.8,23.0,23.7,20.4,
19.0,17.7,12.3,15.2,17.9,18.4,23.1,24.3,23.3,22.6,19.0,
14.7,12.7,17.4,19.1,18.2,21.2,20.8,17.5,19.4,22.4,19.8,
17.5,15.1,12.0,11.3,13.9,16.5,17.2,18.2,18.7,18.4,19.6,
23.2,23.0,20.6,23.4,20.0,15.0,17.6,22.3,22.4,22.7,21.9,
19.4,17.5,15.4,14.8,14.9,21.9,24.2,21.4,18.7,16.0,12.9
dB, OK
```

Details

Queries the spectral results of the FFT analyzer that have been stored by the last MEAS:INIT command. dt measurements are cleared after each MEAS:INIT, so this function returns the LEQ between two MEAS:INIT commands.

If voltage (V) is selected as reading unit on the XL2, then the result is returned in engineering format (e.g. 1.234e-3)

MEASure:FFT:PAGE?

Shortcut Queries the selected frequency range page

Availability FFT

Answer `20k|1k7|200|usr` *string*

Example
💻→ `MEAS:FFT:PAGE?`
→💻 `20k`

MEASure:FFT:PAGE

Shortcut Defines the frequency range page

Availability FFT

Parameter `20k|1k7|200|usr` 'usr' is only available with installed EAP

MEASure:FFT:ZOOM?

Shortcut Queries the current zoom step (the width of the calculated frequency range)

Availability FFT

Answer `0|1|2|...|9` *string*

0: Δ = 20.0 kHz
1: Δ = 13.3 kHz
2: Δ = 6.7 kHz
3: Δ = 3.3 kHz
4: Δ = 1.7 kHz
5: Δ = 832 Hz
6: Δ = 416 Hz
7: Δ = 208 Hz
8: Δ = 104 Hz
9: Δ = 52 Hz

Example
💻→ `MEAS:FFT:ZOOM?`
→💻 `0`

Details With zoom=0, the FFT is set to calculate the spectrum of the whole frequency range, while e.g. with zoom=9 a 52 Hz part of the whole frequency range is calculated and shown. Where the 52 Hz range starts is defined with the command MEASure:FFT:FSTArt

MEASure:FFT:ZOOM

Shortcut Sets the zoom step

Availability FFT with installed EAP or SLI

Parameter `0|1|2|...|9` *string* See table in 'MEAS:FFT:ZOOM?'

Example 💻→ `MEAS:FFT:ZOOM 0`

MEASure:FFT:F?

Shortcut Queries the bin frequencies

Availability FFT

Answer `<f_n,> Hz` *string* n = 143

Example

💻→ `MEAS:FFT:F?`

→💻
```
484.38,625.00,765.63,906.25,1046.88,1187.50,1328.13,
1468.75,1609.38,1750.00,1890.63,2031.25,2171.88,2312.50,
2453.13,2593.75,2734.38,2875.00,3015.63,3156.25,3296.88,
3437.50,3578.13,3718.75,3859.38,4000.00,4140.63,4281.25,
4421.88,4562.50,4703.13,4843.75,4984.38,5125.00,5265.63,
5406.25,5546.88,5687.50,5828.13,5968.75,6109.38,6250.00,
6390.63,6531.25,6671.88,6812.50,6953.13,7093.75,7234.38,
7375.00,7515.63,7656.25,7796.88,7937.50,8078.13,8218.75,
8359.38,8500.00,8640.63,8781.25,8921.88,9062.50,9203.13,
9343.75,9484.38,9625.00,9765.63,9906.25,10046.88,10187.50,
10328.13,10468.75,10609.38,10750.00,10890.63,11031.25,
11171.88,11312.50,11453.13,11593.75,11734.38,11875.00,
12015.63,12156.25,12296.88,12437.50,12578.13,12718.75,
12859.38,13000.00,13140.63,13281.25,13421.88,13562.50,
13703.13,13843.75,13984.38,14125.00,14265.63,14406.25,
14546.88,14687.50,14828.13,14968.75,15109.38,15250.00,
15390.63,15531.25,15671.88,15812.50,15953.13,16093.75,
16234.38,16375.00,16515.63,16656.25,16796.88,16937.50,
17078.13,17218.75,17359.38,17500.00,17640.63,17781.25,
17921.88,18062.50,18203.13,18343.75,18484.38,18625.00,
18765.63,18906.25,19046.88,19187.50,19328.13,19468.75,
19609.38,19750.00,19890.63,20031.25,20171.88,20312.50,
20453.13 Hz
```

MEASure:FFT:FSTArt

Shortcut Defines the frequency of the first bin

Availability FFT with installed EAP or SLI

Parameter `f` *float* in Hz

Example 💻→ `MEAS:FFT:FSTART 58.00`

Details Defines the frequency of the first bin of the Zoom FFT and automatically sets the analyzer to the 'usr' page. Not all combinations of 'zoom level' and 'start frequency' are valid, but the XL2 will choose the settings closest to the parameter. We suggest first using the manual user interface of the XL2 to get a feeling for valid settings.

The MEAS:FFT:F command can be used to verify the setting.

MEASure: 12OCT Subsystem

MEASure:12OCT?

Shortcut Queries the spectral results of the 1/12 Octave analyzer.

Availability 1/12 Oct (requires installed SLI)

Parameter `[Live|Max|Min|EQ]` *string*

Answer `{Level`$_n$`,} dB|dBu|dBV|V, [OK|UNDEF|OVLD| OPTION_REQUIRED]` *float*

1/1 Oct: n = 11 + 2*, f_{start} = 16.0 Hz
1/3 Oct: n = 33 + 2*, f_{start} = 12.5 Hz
1/6 Oct: n = 66 + 2*, f_{start} = 11.8 Hz
1/12 Oct: n = 132 + 2*, f_{start} = 11.5 Hz

* Spectrum + 2 broad band results, as displayed on the XL2 screen.

Example

```
💻→ MEAS:INIT
     MEAS:12OCT? Live
→💻 55.5,34.4,44.0,39.4,34.9,29.4,29.2,27.6,40.1,41.1,38.8,
     44.1,56.2 dB, OK
```

Details Queries the spectral results of the 1/12 Octave analyzer that have been stored by the last MEAS:INIT command. If the parameter is unknown, a ";" is returned.

If voltage (V) is selected as reading unit on the XL2, then the result is returned in engineering format (e.g. 1.234e-3)

MEASure:12OCT:dt?

Shortcut Queries the spectral dt results of the 1/12 Octave analyzer.

Availability 1/12 Oct (requires installed SLI)

Parameter `EQ` *string*

Answer `{Level`$_n$`,} dB|dBu|dBV|V, [OK|UNDEF|OVLD| OPTION_REQUIRED]` *float*

1/1 Oct: n = 11 + 2*, f_{start} = 16.0 Hz
1/3 Oct: n = 33 + 2*, f_{start} = 12.5 Hz
1/6 Oct: n = 66 + 2*, f_{start} = 11.8 Hz
1/12 Oct: n = 132 + 2*, f_{start} = 11.5 Hz

* Spectrum + 2 broad band results, as displayed on the XL2 screen.

Example

```
💻→ MEAS:INIT
     MEAS:12OCT:dt? EQ
→💻 55.5,34.4,44.0,39.4,34.9,29.4,29.2,27.6,40.1,41.1,38.8,
     44.1,56.2 dB, OK
```

Details Queries the spectral results of the 1/12 Octave analyzer that have been stored by the last MEAS:INIT command. dt measurements are cleared after each MEAS:INIT, so this function returns the LEQ between two MEAS:INIT commands.

If voltage (V) is selected as reading unit on the XL2, then the result is returned in engineering format (e.g. 1.234e-3)

MEASure:12OCT:RESOlution

Shortcut Defines the resolution, in which the 1/12 Octave analyzer results are acquired.

Availability 1/12 Oct (requires installed SLI)

Parameter `[1/1|1/3|1/6|1/12]` *string*

Example 💻→ `MEAS:12OCT:RESO 1/3`

MEASure:12OCT:RESOlution?

Shortcut Queries the resolution, in which the 1/12 Octave analyzer results are acquired.

Availability 1/12 Oct (requires installed SLI)

Answer `[1/1|1/3|1/6|1/12]` *string*

Example 💻→ `MEAS:12OCT:RESO?`
→💻 `1/3`

INPUt Subsystem

INPUt:SELEct

Shortcut Configures which input connector is selected.

Availability always

Parameter `[XLR|RCA]` *string*

Example 💻→ `INPUT:SELECT XLR`

INPUt:SELEct?

Shortcut Queries the input range setting.

Availability always

Answer `[XLR|RCA]` *string*

Example 💻→ `INPU:SELE?`
→💻 `XLR`

INPUt:RANGe

Shortcut Configures the input range setting for the SLMeter, Zoom FFT and 1/12 Octave analyzer

Availability SLMeter when stopped, FFT, 1/12 Oct

Parameter `[LOW|MID|HIGH]` *string*

Example 💻→ `INPUT:RANGE MID`

INPUt:RANGe?

Shortcut Queries the input range setting for the SLMeter, Zoom FFT and 1/12 Octave analyzer

Availability always

Answer `[LOW|MID|HIGH]` *string*

Example 💻→ `INPUT:RANGE?`
→💻 `MID`

INPUt: PHANtom

Shortcut Configures the input range setting.

Availability always

Answer `[ON|OFF]` *string*

Example 💻→ `INPUT:PHAN ON`

Details This command is not accepted when an ASD sensor is connected.

INPUt: PHANtom?

Shortcut Queries the phantom power setting.

Availability always

Answer `[ON|OFF|ASD]` *string*

Example 💻→ `INPUT:PHAN?`
→💻 `ON`

CALIBrate Subsystem

CALIbrate:MIC:TYPE?

Shortcut Queries the microphone type recognized by the ASD (**A**utomatic **S**ensor **D**etection) system.

Availability Always

Answer `[M2210|M4260|noASD]` *String*

Example
💻→ `CALI:MIC:TYPE?`
→💻 `M4260`

Details If no ASD microphone is currently connected, the command always returns noASD. In contrast, the command CALIB:MIC:SENS:SOURce returns the ASD microphone that was last connected, as long as the microphone sensitivity has not been changed manually or by remote command.

CALIbrate:MIC:SENS:SOURce?

Shortcut Queries the source of the sensitivity value.

Availability always

Answer
```
[PLEASE CALIBRATE|USER CALIBRATED|MANUALLY|          string
M2210 USER|M2210 FACTORY|M2210 CAL.CENTER|
M4260 USER|M4260 FACTORY|M4260 CAL.CENTER]
```

Example
💻→ `CALI:MIC:SENS:SOURce?`
→💻 `M4260 FACTORY`

Details Returns the ASD microphone that was last connected as long as the microphone sensitivity has not been changed manually or by remote command.
PLEASE CALIBRATE is returned when the sensitivity has never been set since the last factory default setup.

CALIbrate:MIC:SENS:VALUe

Shortcut Defines the microphone sensitivity in V/Pa.

Availability always

Parameter `<sens>` *float* 100e-6 to 9.99 V/Pa

Example
```
💻→ CALIB:MIC:SENS:VALU 0.02
     CALIB:MIC:SENS:VALU 20e-3
```

Details Command is not accepted when an ASD microphone is connected.

CALIbrate:MIC:SENS:VALUe?

Shortcut Queries the microphone sensitivity in V/Pa .

Availability always

Answer `<sens> V,OK` *float* 100e-6 to 9.99 V/Pa

Example
```
💻→ CALIB:MIC:SENS:VALU?
→💻 21.54e-3 V,OK
```

SYSTem Subsystem

SYSTem:ERROr?

Shortcut Queries the error queue

Availability always

Answer {errno$_n$,} *integer* n ≤ 10

SCPI System errors

-350	Error queue full - at least 2 errors lost
-115	Too many parameters in command
-113	Invalid command
-112	Too many characters in one of the command parts
-109	Missing command or parameter
-108	Invalid parameter

XL2 errors

0	no error (queue is empty)
1	Command too long; too many characters without new line
2	UNEXPECTED_PID
3	DSP_TIMEOUT
4	Changing microphone sensitivity is not possible when an ASD microphone is connected to the XL2
5	Parameter not available, license not installed
6	dt value does not exist for this parameter
7	Parameter is not available in the current measurement function
8	Unspecified DSP error
9	Not valid, measurement is running

Example

```
💻→ SYSTem:ERRO?
→💻 -113, -113, -113, -109, -109
💻→ SYSTem:ERRO?
→💻 0
```

Details There are different classes of errors. Some errors refer to the command syntax, others to internal states of the XL2.
Every error is pushed into the error queue that must be queried to get information about any error.

SYSTem:KEY

Shortcut Simulates a key stroke on the XL2.

Availability always

Parameter

`[ESC|NEXT|FNEXT|PREV|FPREV|ENTER|PAGE|START|PAUSE|SPEAKER|LIMIT|LIGHT]`	*string*	Multiple keys may be sent with one command (see example).

Example

```
💻→ SYST:KEY PAGE
→💻 OK
💻→ SYST:KEY ESC ENTER PREV PREV
→💻 OK
```

Details The command returns OK after all keystrokes have been executed by the XL2. Execution of the keys can take a moment, especially if measurement functions are changed using this command.
FNEXT and FPREV are "fast" wheel turn simulations that are necessary to manipulate numbers using key commands.

SYSTem:KLOCk

Shortcut Locks the keyboard of the XL2

Availability always

Parameter `[ON|OFF]` *string* ON: Keyboard is locked

Example

```
💻→ SYST:KLOCK ON
```

Details If the USB cable is disconnected, KLOCK is automatically set to OFF.

SYSTem:KLOCk?

Shortcut Queries the key lock status

Availability always

Answer `[ON|OFF]` *string*

Example

```
💻→ SYST:KLOCK?
→💻 ON
```

SYSTem:SPEAker:ONOFf

Shortcut Switching speaker on and off

Availability always

Parameter `[ON|OFF]` *string*

Example 💻→ `SYST:Speaker:OnOff On`

SYSTem:SPEAker:LEVEl

Shortcut Set the Speaker Level

Availability always

Parameter `<level>` *integer* -80 to 80

Example 💻→ `SYST:Speaker:Level 10`

SYSTem:LIMItled?

Shortcut Queries the limit LED status

Availability always

Answer `[OFF|GREEN|ORANGE|RED]` *string*

Example 💻→ `SYST:LIMI?`
→💻 `GREEN`

SYSTem:MSD

Shortcut Switches the XL2 to the USB mass storage mode

Availability always

Answer `No answer`

Example 💻→ `SYST:MSD`

Details After sending this command, the XL2 drops the COM connection (no more remote commands are possible) and switches to mass storage mode. The host then has full access to the data stored on the SD card of the XL2.

The XL2 returns to the COM mode immediately after the XL2 drive was ejected by the host.

If "safely remove" was chosen by the host, the XL2 returns to the COM mode after a timeout of 2 minutes.

SYSTem:MSDMAC

Shortcut Switches the XL2 to the USB mass storage mode for Mac.

Availability always

Answer `No answer`

Example 💻→ `SYST:MSDMAC`

Details Same as "SYSTem:MSD"
Use this Command on Mac. Otherwise MSD will always timeout after 2 minutes and USB returns to COM mode.

SYSTem:OPTIons?

Shortcut Queries the installed options

Availability always

Answer

`<EAP,>`	*string*
`<STIPA,>`	
`<REMOTE,>`	
`<SLI,>`	
`<CA,>`	
`<TA>`	

Example
💻→ `SYST:OPTI?`
→💻 `EAP,REMOTE`

Supplements

Automatic COM Port Detection

The following procedure automatically detects the COM port assigned for communication to your XL2. Thus, it is not necessary for you to manually set the COM port. The procedure is also used in the XL2 Projector and the Microsoft Excel and LabVIEW XL2 remote demonstration projects.

The XL2 Projector uses the Windows built-in driver "usbser.sys" to communicate with your XL2. This driver provides a virtual COM port over USB. If the XL2 Projector is installed, the usbser.sys driver is available. In order to find the COM port assigned to your XL2, kindly follow these steps in the MS Windows registry:

1. Execute the command "regedit" to open the registry editor.

2. Open the key "HKEY_LOCAL_MACHINE\system\CurrentControlSet\Services\usbser", which is available if the driver is installed.

3. Open the "Enum" subkey that is created as soon as an usbser device is connected to your PC for the first time.

4. Note the hexadecimal value behind the "Count" entry in the Enum subdirectory; it indicates the number of devices that are currently using the usbser driver.
   Example: Count → 0x0000000A (10) means that ten devices are using the usbser driver.

5. Note the strings behind the variables "0", "1" … to the aforementioned number of devices that are using the usbser driver. If the string contains the substring "VID_1A2B&PID_0004", the connected device is an XL2 from NTi Audio.
   Example: the string "USB\VID_1A2B&PID_0004\5&640e942&0&1" indicates that the connected device is an XL2.

6. Open the subkey "HKEY_LOCAL_MACHINE\system\CurrentControlSet\Enum\<xxx>\Device Parameters" under "Computer", whereby <xxx> stands for the result string obtained in step 5.

7. Read the variable "PortName" from the key; it contains the name of the virtual COM port to which your XL2 is connected (e.g. "COM10").

8. Open the given COM port to check if it is already used by another program.

NTi Audio offers an implementation of this algorithm in C++, VBA and LabVIEW upon request.