XL2
Remote Measurement

Reference Manual V4.80

Refers to XL2 firmware version 4.80 or higher
2022-02-09

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:

- o Sound level meter and spectrum analyzer SLMeter
- o FFT analyzer
- o Audio analyzer RMS/THDN
- o High resolution spectral analyzer 1/12 Oct + Tol

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

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

The access to the XL2 file system is supported with the accessory „NetBox" used in Gateway Mode. See the chapter „Accessing an XL2 over the Internet" in this manual for more details.

The XL2 with serial number ending „E0" or higher may power on automatically upon mains power is available. This auto start functionality is enabled by copying a txt-file with the file name „AutoOn.txt" onto the SD Card. This allows to remotely control the XL2 as soon as power supply is available at the instrument.

XL2 Projector PRO

The XL2 Projector PRO 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.
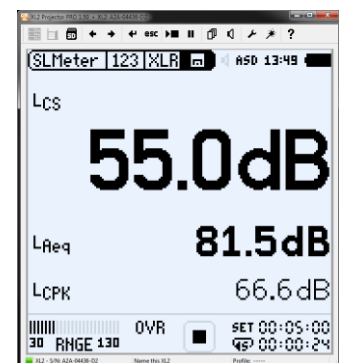
XL2 Projector PRO runs with every XL2, there is no need to have any options installed on the XL2. The XL2 Projector PRO is available to you as a free download at https://my.nti-audio.com/support/xl2.

The optional „XL View" and „Sound Level Predictor" is dedicated for live sound monitoring applications. Its enabled by the Projector PRO Option or Remote Measurement Option upon installation on your XL2.

NOTE: The XL2 Projector function uses the „COM port" USB mode. Click the SD-Card icon in the XL2 Projector PRO to open the XL2 File Explorer.

Remote Measurement Option required

To query measurement results from the XL2 remotely, the XL2 must be equipped with a Remote Measurement Option.

NOTE **–** If the XL2 Remote Measurement option is <u>not</u> 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 PRO software.

Getting Started

Requirement: Make sure that you have installed the XL2 Projector PRO on your system. Together with the XL2 Projector PRO, 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.
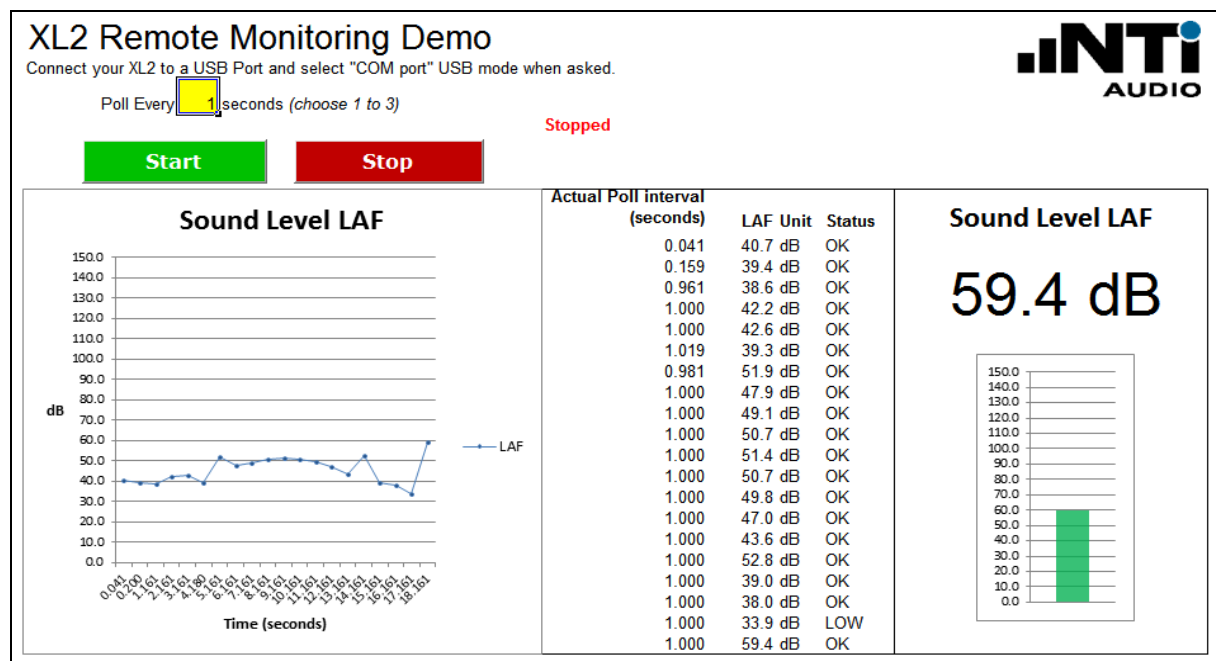
On your PC, press the Alt-F11 keys to access the open source code and extend the functionalities according your individual requirements.

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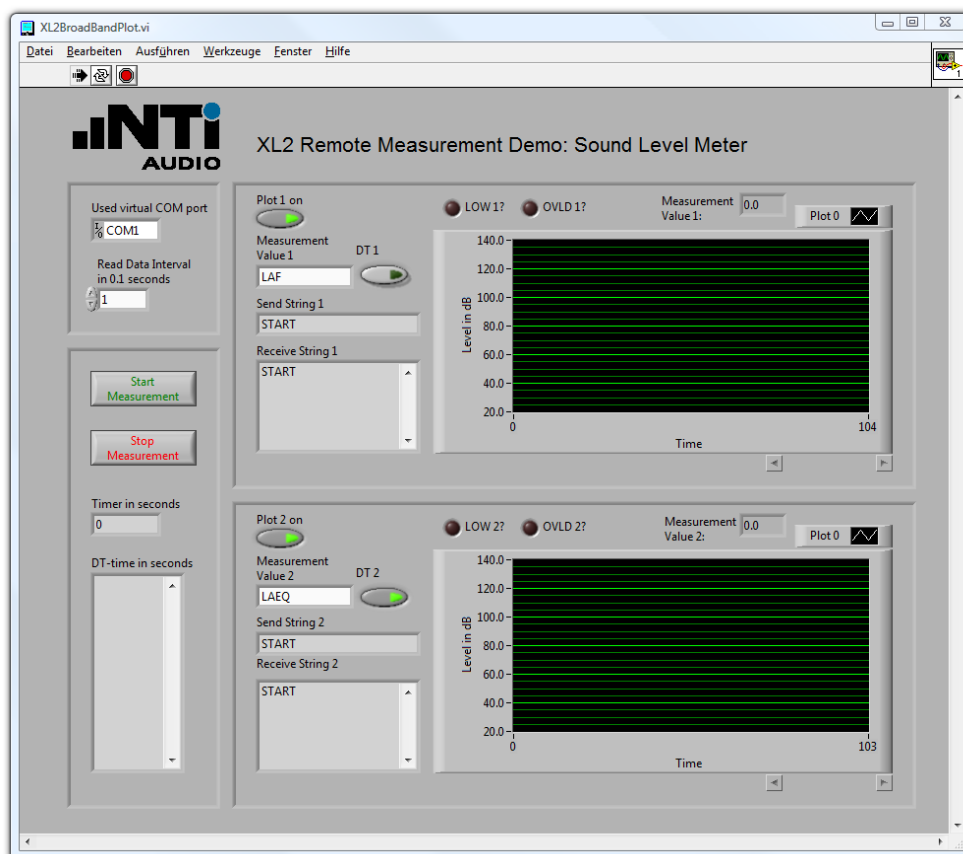
9. In case any error messages are displayed, simply continue with these instructions.

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.
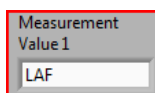
14. Commands
   The "Send String" of a plot shows the complete string, which is sent to your XL2 to acquire the measurement value. The "Receive String" of a plot shows all received measurement values. Use the scrollbar to view all values.
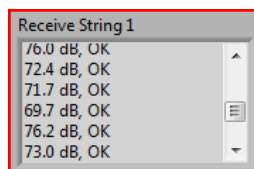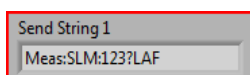
15. Setting time parameter
In the "Read Data Interval" you can change the speed of acquiring data from your XL2. The minimum interval is 0.1 second, represented by a value of 1. Enter the value 20 to acquire measurement results every two seconds. The „Timer in seconds" shows the current measurement timer 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.

DT-time in seconds
0.100000
0.100000
0.100000
0.101333
0.098667
0.100000

Read Data Interval
in 0.1 seconds
1

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 PRO software on your PC, thereby ensuring that the serial driver, required for remote measuring, is available. Verify that the XL2 Projector PRO software functions correctly prior to continuing with the next steps.

1) Start the terminal program HTerm.exe:

HTerm.exe	13.09.2010 11:31
XL2 Keys.hts	04.10.2010 11:15
XL2.hts	14.10.2010 12:22
XL2_Config.cfg	08.11.2010 09:27
XL2_Input.hts	14.10.2010 12:22

2) Load the configuration File „XL2_Config.cfg"
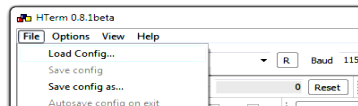
3) HTerm displays the loaded XL2 configuration file:
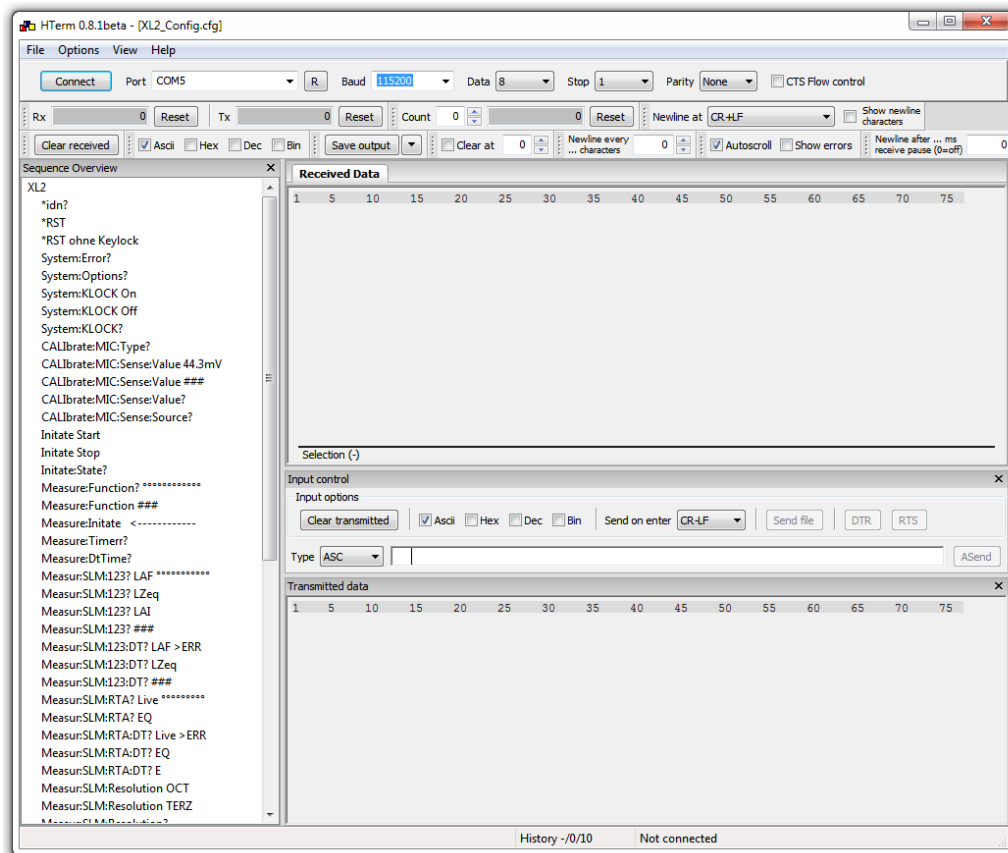
4) Connect your XL2
   a) Connect your XL2 via USB to your PC and power-up the XL2. The XL2 displays the USB Mode window.
   b) Select COM port on your XL2.

5) Selecting the COM port in HTerm:
   a) Press the R button to refresh the port list
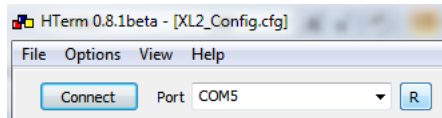
   b) Select the COM port used to communicate with your XL2 (e.g. your PC displays the com port assigned to the XL2 during the initial connection to your XL2).
   c) Press „Connect" and wait for the status information of HTerm in the bottom line; as soon as it is connected successfully, the status line should show something like this:
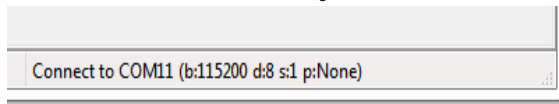
6) First communication with your XL2:
   a) Double-click on „*idn?" in the „Sequence Overview" window on the left hand side.
   b) „*IDN?" is shown on the transmitted data window.
   c) "NTiAudio, XL2, A2A-xxxx-D1, FW2.xx" is shown in the Received Data window.

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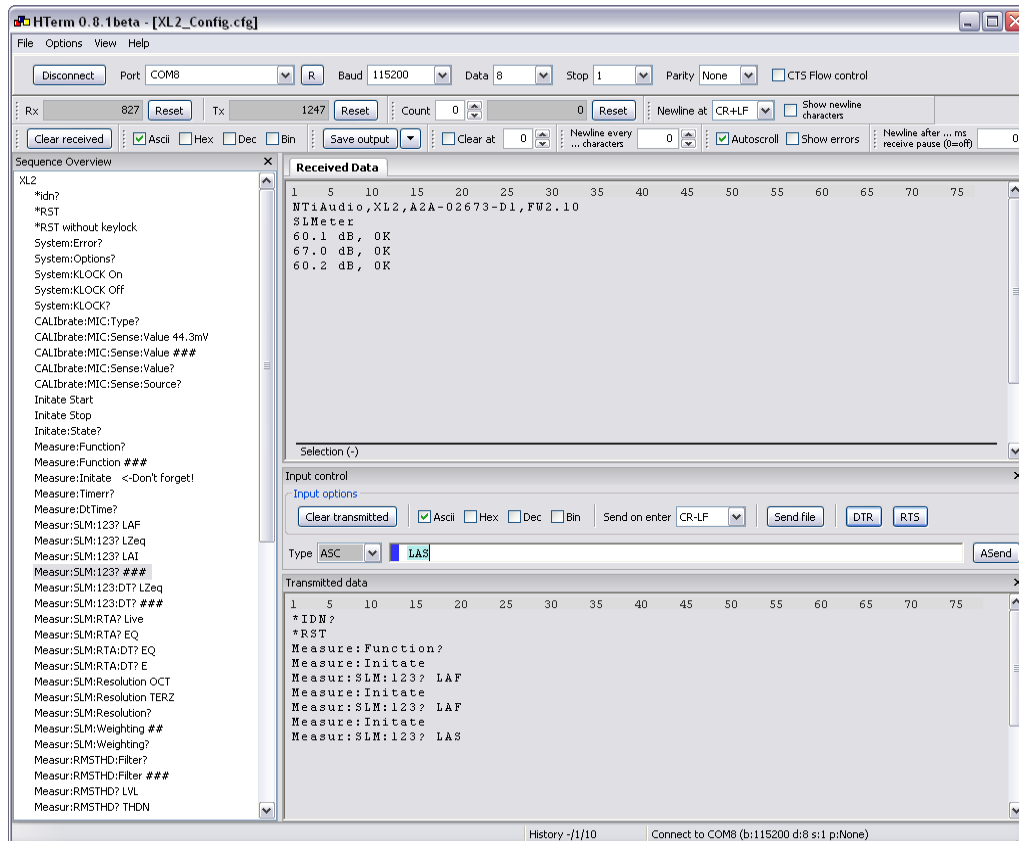
11) Stopping the remote measurement
   a) Click the button „Disconnect" in HTerm
   b) Disconnect your XL2 from the USB connection to your PC.

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" (Carriage Return, Line Feed).

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 3.7):

```python
import time
import serial

# Query the Device Manager of your Windows PC to find out which COM port the
# system assigned to the XL2 and adapt the following line:
COM_PORT = "COM15"

xl2 = serial.Serial(COM_PORT, timeout=1)
xl2.write(b'*RST\n')                # Reset the XL2 to default state (SLMeter, ...)
xl2.write(b'INIT START\n')          # Start the measurement
time.sleep(3)                       # Allow the XL2 to start the measurement

for i in range(10):
    xl2.write(b'MEAS:INIT\n')            # Triggers a measurement
    xl2.write(b'MEAS:SLM:123? LAS\n')    # Query LAS
    result = xl2.readline()
    print(result)
    time.sleep(1)

xl2.write(b'INIT STOP\n')           # Stop the measurement (optional)
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
```

Response Time

The time the XL2 requires for a response depends on the command and the workload of the XL2 (logging settings). A typical pattern for querying measurement data is:

```
xl2.write(b'MEAS:INIT\n')
xl2.write(b'MEAS:SLM:RTA:dt? EQ\n')
result = xl2.readline()
```

This pattern delivers a measurement result within:

Min: 8 ms
Mean: 10 ms
Max: 35 ms

Comment: Tested with XL2 FW 4.50 running 1000 times the above pattern.

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

        o   Append mode: OFF
        o   Auto save: OFF
        o   Logging: OFF
        o   Events: OFF
        o   Timer mode: CONTINUOUS
        o   Range: MID
        o   RMS/THDN Filter: Z-WEIGHTED
        o   Input: XLR
        o   Phantom Power: ON
        o   RTA Source: LZF
        o   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, RT60
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|STOPPING|UNDEFINED]` *string*	UNDEFINED when DSP is not responding
Example	🖥→ `INIT:STATE?` →🖥 `RUNNING`	

INITiate:STATe:SETTlingtime?

Shortcut	Queries the current settling time.
Availability	SLMeter, RT60
Answer	`<timer>` *string* 0.1 seconds resolution (1 decimal) `sec, [OK|UNDEF]`
Example	🖥→ `INIT:STATE:SETT?` →🖥 `5.4 sec, ok`
Details	When the settling time is 0, then a INIT START command will immediately start a measurement. Otherwise, the run status will first go to SETTLING when starting a measurement.

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|`
	`VibMeter*]`
		string	only the first two characters are necessary

			*only in Domain Vibration
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|`
	`VibMeter*]`
		string	If the Type Approved Firmware is running the SLMeter returns „SLM TA" instead of „SLMeter"

			*only in Domain Vibration
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 sec resolution (1 decimal) when
	`sec, [OK|UNDEF]`	*string*	logging with 100ms, else 1.0 sec.

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 MEAS:INIT 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:DOMAin?

Shortcut	Queries the actual Domain of the XL2.
Availability	always
Answer	`[Sound|Vibration]` *string*
Example	⌨→ `MEASURE:Domain?` →⌨ `Sound`
Details	If the Vibration Option is installed, the XL2 can be switched between sound and vibration domain manually or automatically with NTi ASD microphones or ASD vibration Sensors.

MEASure:VIBDbref? (Domain: Vibration)

Shortcut	Queries the dB reference for values acceleration, velocity and displacement.
Availability	Domain: Vibration
Parameter	`none`

Answer	`<ACC dBref Level um/s2,`
	`VEL dBref Level nm/s,` All levels in one Line
	`DIS dBref level pm>`

Example	⌨→`Meas:Vibdbref ?` →⌨`1.00um/s2,1.00nm/s,1.00pm`
Details	Reads all three dBref values from the device system settings / vibration page.

MEASure: SLM Subsystem (Domain: Sound)

MEASure:SLM:123?

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

Parameter	`[LxS|LxSMAX|LxSMIN|`
	`LxF|LxFMAX|LxFMIN|`
	`LxEQ|Prev_LxEQ|`
	`LxPK|LxPKMAX|`
	`LCPk5"|Prev_LcPK|`
	`LyEQt|LyEQtMAX|`
	`k1|k2]`
		string	Replace lowercase letters as follows.
			x = [A|C|Z]
			y = [A|C]
			t = One of the four settings specified on the "Set EQt, L%" page of the XL2, e.g. LAEQ5" or LCEQ15'max

	Additional with installed Extended Acoustic Pack
	`[LxI|LxIEQ|LxIMAX|LxIMIN|`
	`LAFT3|LAFT3EQ|`
	`LAFT5|LAFT5EQ|`
	`LxE|Ln%|`
	`LAFT5EQ-LAEQ|`
	`LAIEQ-LAEQ|LCEQ-LAEQ|`
	`LAF20_200|LAFmax20_200|`
	`LAeq20_200]`
		string	n = 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,`
	`[OK|UNDEF|LOW|OVLD|`
	`OPTION_REQUIRED]`
		float
		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. The parameters must be separated by spaces.		
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 Extended Acoustic Pack `[LxIMAX|LxIMIN|LxE|` ` LAFmax20_200|LAeq20_200]`	*string*
Answer	`<Level,> dB,` `[OK|UNDEF|LOW|OVLD|` `OPTION_REQUIRED|NO_DT_VALUE` `]`	*float* *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. The parameters must be separated by spaces.	
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|`
	`HOLD3|HOLD5|HLD10]`
		string	

	Additional with installed Extended Acoustic Pack `[E|N%]`	*string*	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|dBu|dBV|V,`
	`[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, OK`		

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. The unit (dB, dBu, dBV, V) is adopted by the setting of the user interface.		

MEASure:SLM:RTA:dt?

Shortcut	Queries the dt spectral results of the SLMeter.		
Availability	SLMeter		
Parameter	`[EQ]`	*string*	
	Additional with installed Extended Acoustic Pack `[E]`	*string*	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|dBu|dBV|V,`
	`[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: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, OK
```

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.

The unit (dB, dBu, dBV, V) is adopted by the setting of the user interface.

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: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: RMSThdn Subsystem (Domain: Sound)

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]] |`	*string*
	`[db,[OK|UNDEF|OVERLOAD]]|`	
	`[Hz,[OK|UNDEF|OVERLOAD]]`	
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 (Domain: Sound)

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. The unit (dB, dBu, dBV, V) is adopted by the setting of the user interface.		

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. The unit (dB, dBu, dBV, V) is adopted by the setting of the user interface.

MEASure:FFT:PAGE

Shortcut	Defines the frequency range page
Availability	FFT

Parameter		
	`20k|1k7|200|usr`	'usr' is only available with installed „Extended Acoustic Pack" or „Spectral Limits" option

Example	🖥→ `MEAS:FFT:PAGE 20k`

MEASure:FFT:PAGE?

Shortcut	Queries the selected frequency range page
Availability	FFT
Answer	`20k|1k7|200|usr` *string* 'usr' is only available with installed „Extended Acoustic Pack" or „Spectral Limits" option
Example	🖥→ `MEAS:FFT:PAGE?` →🖥 `20k`

MEASure:FFT:ZOOM

Shortcut	Sets the zoom step
Availability	FFT with installed Extended Acoustic Pack or Spectral Limits Option
Parameter	`0|1|2|...|9` *string* See table in 'MEAS:FFT:ZOOM?'
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	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`

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, 9343.75,9484.38,9625.00,9765.63,9906.25,10046.88,10187.5 0,10328.13,10468.75,10609.38,10750.00,10890.63,11031.25, 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 Extended Acoustic Pack or Spectral Limits Option
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 (Domain: Sound)

MEASure:12OCT?

Shortcut	Queries the spectral results of the 1/12 Octave analyzer.		
Availability	1/12 Oct (requires installed Spectral Limits Option)		
Parameter	`[Live|Max|Min|` `EQ|EQ1"|EQ4"]`	*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 Spectral Limits Option)		
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 Spectral Limits Option)
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 Spectral Limits Option)
Answer	`[1/1|1/3|1/6|1/12]` *string*
Example	🖳→ `MEAS:12OCT:RESO?` →🖳 `1/3`

MEASure: RT60 Subsystem (Domain: Sound)

MEASure:RT60 ?

Shortcut	Queries the results of the RT60 analyzer.
Availability	RTA
Parameter	`[AVG]` *string*

Answer	`{t,} sec,`	*float*	1/1 Oct: n = 8	
	`[OK]`		1/3 Oct: n = 32	

Example	⌨→ `MEAS:RT60? AVG`
	→⌨ `3.2,2.9,2.4,2.2,2.7,2.6,3.2,2.8 sec, OK`
Details	Queries the averaged results of the RT60 Octave analyzer. MEAS:INIT command is not required. If the parameter is unknown, a ";" is returned.

MEASure: Vibration Introduction (Domain: Vibration)

With the Vibration Option installed on your XL2, you can switch the XL2 into the Vibration Domain, thereby turning the XL2 into a device for measuring vibration. There are no functions that cross over between the XL2 as a sound-measuring device and the XL2 as a vibration-measuring device. Each has their own individual settings. There is also no remote command to switch between the Sound and Vibration domains; the switch has to be done on the XL2 itself (you have to change the sensor anyway).

The XL2 can be switched between Sound and Vibration Domain,
a) manually in the Calibration screen. (not possible while an ASD Sensor is connected)
b) automatically when you connect an NTi ASD Microphone or NTi ASD Vibration Sensor.

The current Domain can be read using the remote command `Measure:Domain?` The Vibration Domain provides an additional Vibration Setup page. Use the remote command `Measure:dBref?` to read the reference values for all Vibration dB units. These reference values are valid for the measurement functions VIBM, VFFT and V12OCT.

The Units of the remote output correspond to what you see on the device screen. In the Vibration Setup page, you specify your preference for Metric or Imperial units. Within the measurement function you select individual linear or logarithmic units.

Measure	Acceleration	Velocity	Displacement	Comment
Metric	m/s^2	m/s	m	m, mm, um, nm, pm
Imperial	in/s^2	in/s	in	in, mil, uin, nin, pin
dB	dBa	dBv	dBd	Ref: pm/s^2, nm/s, pm
Gravity	g	--	--	

MEASure: VIBM Subsystem (Domain: Vibration)

MEASure:VIBM:123?

Shortcut	Queries a broad band measurement result of the VibMeter.	
Availability	VibMeter	
Parameter	`[xxxS|xxxSMAX|xxxSMIN` `|xxxF|xxxFMAX|xxxFMIN` `|xxxEQ|xxxPrev_eq` `|xxx1"|xxx2"|xxx4"|xxx8` `"` `|xxxPK|xxxPKMAX` `|xxxPkPk|xxxPkPkMAX]`	*string* Replace as follows: xxx = [acc|vel|dis]
	Additional with installed Extended Acoustic Pack `[xxxE|velFT30|velFT30eq` `]`	*string*
Answer	`<Level,> unit,` `[OK|UNDEF|LOW|OVLD|` `OPTION_REQUIRED]`	*float* *string* Unit see table in Vibration Introduction

Example	💻➔ `INIT START`
	 `MEAS:INIT`
	 `MEAS:VIBM:123? accFMAX`
	 ➔💻 `9.84 m/s2, 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. The unit is always the same as in the user interface.	
Call with multiple parameters	Queries up to 10 parameters with one command. The parameters must be separated by spaces.	

Example	💻➔ `INIT START`
	 `MEAS:INIT`
	 `MEAS:VIBM:123? accFMAX velFMAX disFMAX`
	 ➔💻 `9.84 m/s2, OK`
	 `1.96e-2 m/s, OK`
	 `3.95e-5 m, OK`	

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

MEASure:VIBM:123:dt?

Shortcut	Queries a broad band dt measurement result of the VibMeter.
Availability	VibMeter

Parameter	`[xxxSMAX|xxxSMIN` `|xxxFMAX|xxxFMIN` `|xxxEQ|xxxPkMAX|xxxPkPkMAX]`	*string*	Replace as follows: `xxx = [acc|vel|dis]`
	Additional with installed Extended Acoustic Pack `[xxxE]`	*string*	
Answer	`<Level,> unit,` `[OK|UNDEF|LOW|OVLD|` `OPTION_REQUIRED|NO_DT_VALUE]`	*float* *string*	Unit see table in Vibration Introduction

Example	🖥→ `INIT START` `MEAS:INIT` `MEAS:VIBM:123:dt? accEQ` →🖥 `3.99e-4 m/s2, OK`
Details	Queries a broad band result parameter of the VibMeter 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.
The unit is always the same as in the user interface. |
Call with multiple parameters	Queries up to 10 Parameters with one command. The parameters must be separated by spaces.
Example	🖥→ `INIT START` `MEAS:INIT` `MEAS:VIBM:123:dt?  acceq veleq diseq` →🖥 `4.32e-4 m/s2, OK` `2.51e-5 m/s, OK` `3.37e-6 m, OK`
Details	This command reduces the XL2 CPU load when reading several values in short intervals (e.g. 0.1sec).

MEASure:VIBM:FILTer

Shortcut	Set the actual filter for the VibMeter.
Availability	VibMeter
Parameter	`[Flat|1-80Hz|1-315Hz]` *string*
Answer	`none`
Example	🖥→ `MEAS:VIBM:Filter Flat`
Details	Filter is used for broadband values and can be used for the spectrum with separate command. Command is only accepted when measurement is stopped.

MEASure:VIBM:FILTer?

Shortcut	Queries the actual filter from the VibMeter.
Availability	VibMeter
Parameter	`none`
Answer	`[Flat|1-80Hz|1-315Hz]` *string*
Example	🖥→ `MEAS:VIBM:Filter?` →🖥 `Flat`

MEASure:VIBM:SPECtrum:FILTer

Shortcut	Sets the actual spectrum filter setting On or Off.
Availability	VibMeter
Parameter	`[On|Off]` *string*
Answer	`None` *string*
Example	🖥→ `MEAS:VIBM:SPEC:Filter ON`
Details	Here you decide to use the actual filter for the spectrum or not. The actual filter is set in the broadband screen or with the `MEASure:VIBM:FILTer` command. E.g. when the actual broadband filter is „`1-8Hz`", `On` uses this Filter, `Off` uses Flat for the spectrum. Note: When the broadband filter is „`Flat`", switching `On` and `Off` in the spectrum makes no difference. Command is only accepted when measurement is stopped.

MEASure:VIBM:SPECtrum:FILTer ?

Shortcut	Queries the actual spectrum Filter setting.
Availability	VibMeter
Parameter	`none`
Answer	`[On|Off]` *string*
Example	⌨→ `MEAS:VIBM:SPEC:Filter?` →⌨ `Off`

MEASure:VIBM:SPECtrum?

Shortcut	Queries the spectral result of the VibMeter.		
Availability	VibMeter		

Parameter	`[LIVE|`
	`MAX|MIN|`
	`EQ|CAPT`
	`HOLD3|HOLD5|HLD10]`
		string	

	Additional with installed Extended Acoustic Pack [E]	*string*	

Answer	`{Level`$_n$`,} unit,`
	`[OK|UNDEF|OVLD|`
	`OPTION_REQUIRED]`
		float
		string
			1/1 Oct: n = 12, f_{start} = 1 Hz
			1/3 Oct: n = 36, f_{start} = 0.8 Hz
			Levels sorted from lowest to highest frequency. For the unit, see table in capture „Vibration Introduction"

Example	⌨→ `INIT START`
	`MEAS:INIT`
	`MEAS:VIBM:SPEC? EQ`
	→⌨ `5.76e-6,8.49e-6,3.80e-6,1.11e-5,8.60e-6,4.32e-6,3.07e-6,2.63e-6,1.61e-6,8.54e-7,6.79e-7,5.27e-7,3.63e-7,2.92e-7,2.85e-7,2.11e-7,1.66e-7,1.20e-7,9.03e-8,8.28e-8,5.87e-8,5.74e-8,4.05e-8,3.46e-8,2.73e-8,2.40e-8,1.90e-8,1.53e-8,1.34e-8,1.08e-8,8.91e-9,7.36e-9,6.38e-9,5.15e-9,4.33e-9,3.41e-9 m/s, OK`		

Details	Queries the spectral result of the VibMeter that has been stored by the last MEAS:INIT command. If the parameter is unknown, a ";" is returned. The unit is always the same as in the user interface.		

MEASure:VIBM:SPECtrum:dt?

Shortcut	Queries the dt spectral result of the VibMeter.	
Availability	VibMeter	
Parameter	`[EQ]`	*string*
	Additional with installed Extended Acoustic Pack `[E]`	*string*

Answer	`{Level``,} unit,` `[OK|UNDEF|OVLD|` `OPTION_REQUIRED]`	*float* *string*	1/1 Oct: n = 12, f_{start} = 1 Hz 1/3 Oct: n = 36, f_{start} = 0.8 Hz Levels sorted from lowest to highest frequency. For the unit, see table in capture „Vibration Introduction".

Answer: `{Level_n,} unit, [OK|UNDEF|OVLD|OPTION_REQUIRED]`

Example

⌨→ `INIT START`
`MEAS:INIT`
`MEAS:VIBM:SPEC:DT? EQ`
→⌨ `8.47e-5,5.52e-5,7.85e-5,6.17e-5,5.64e-5,5.88e-5,3.73e-5,4.27e-5,4.01e-5,4.57e-5,4.03e-5,3.37e-5,3.11e-5,3.53e-5,3.31e-5,3.00e-5,3.50e-5,3.53e-5,3.44e-5,3.23e-5,3.49e-5,3.40e-5,3.59e-5,3.63e-5,3.51e-5,3.81e-5,3.72e-5,3.89e-5,4.00e-5,4.07e-5,4.27e-5,4.43e-5,4.78e-5,4.97e-5,5.30e-5,5.34e-5 m/s2, OK`

Details

Queries the spectral result parameters of the VibMeter 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.

The unit is always the same as in the user interface.

MEASure:VIBM:SPECtrum:WEIGhting

Shortcut	Defines the measure and time weighting, in which the spectrum result is acquired.
Availability	VibMeter
Parameter	`[aF|aS|vF|vS|dF|dS]` *string*
Example	🖥→ `MEAS:VIBM:SPEC:WEIG ds`
Details	Command is only accepted when measurement is stopped.

MEASure:VIBM:SPECtrum:WEIGhting?

Shortcut	Queries the measure and time weighting, in which the spectrum result is acquired.
Availability	VibMeter
Answer	`[aF|aS|vF|vS|dF|dS]` *string*
Example	🖥→ `MEAS:VIBM:SPEC:WEIG?` →🖥 `aS`

MEASure:VIBM:SPECtrum:RESOlution

Shortcut	Defines the resolution, in which the spectrum result is acquired.
Availability	VIBMeter
Parameter	`[OCT|TERZ]` *string*
Example	🖥→ `MEAS:VIBM:SPEC:RESO TERZ`
Details	Command is only accepted when measurement is stopped.

## MEASure:VIBM:SPECtrum:RESOlution?

Shortcut	Queries the resolution, in which the spectrum result is acquired.
Availability	VIBMeter
Answer	`[OCT|TERZ]` *string*
Example	🖥→ `MEAS:VIBM:SPECtrum:RESO?` →🖥 `TERZ`

MEASure: VFFT Subsystem (Domain: Vibration)

MEASure:VFFT?

Shortcut	Queries a measurement result from the VFFT analyzer
Availability	VFFT
Parameter	`[Live|Max|Min|EQ]` *string*

Answer	`{Level`$_n$`,} unit,` `[OK|UNDEF|OVLD]`	*float* *string*	n = 143 Levels sorted from lowest to highest frequency. For the unit, see table in capture „Vibration Introduction".

Example	🖥→ `MEAS:INIT`
	 `MEAS:VFFT? Live`
	 →🖥 `1.30e-3,2.58e-3,6.66e-3,2.94e-2,6.02e-1,9.85e-1,4.00e-`
	 `1,1.69e-2,4.58e-3,1.88e-3,9.47e-4,5.42e-4,3.38e-`
	 `4,2.24e-4,1.56e-4,1.13e-4,8.42e-5,6.44e-5,5.03e-`
	 `5,4.01e-5,3.30e-5,2.70e-5,2.18e-5,1.97e-5,1.58e-`
	 `5,1.40e-5,1.21e-5,1.06e-5,9.67e-6,8.82e-6,8.49e-`
	 `6,7.01e-6,6.07e-6,6.26e-6,5.62e-6,4.78e-6,5.51e-`
	 `6,5.65e-6,4.93e-6,4.38e-6,4.34e-6,4.35e-6,3.81e-`
	 `6,3.20e-6,4.40e-6,3.76e-6,3.80e-6,3.80e-6,4.97e-`
	 `6,4.67e-6,2.88e-6,2.89e-6,2.85e-6,2.48e-6,2.54e-`
	 `6,2.18e-6,2.92e-6,2.14e-6,2.28e-6,3.56e-6,2.40e-`
	 `6,2.31e-6,2.18e-6,2.45e-6,2.37e-6,2.77e-6,2.69e-`
	 `6,2.44e-6,2.29e-6,2.54e-6,3.04e-6,2.41e-6,2.76e-`
	 `6,1.97e-6,1.91e-6,2.49e-6,2.57e-6,2.43e-6,2.14e-`
	 `6,2.38e-6,2.37e-6,2.43e-6,3.56e-6,3.55e-6,3.02e-`
	 `6,3.05e-6,1.71e-6,2.31e-6,2.88e-6,2.81e-6,2.65e-`
	 `6,1.77e-6,1.57e-6,1.66e-6,1.87e-6,1.99e-6,1.87e-`
	 `6,1.94e-6,1.83e-6,2.22e-6,2.08e-6,2.42e-6,2.44e-`
	 `6,1.91e-6,2.49e-6,2.17e-6,1.79e-6,1.76e-6,1.98e-`
	 `6,2.62e-6,2.62e-6,2.04e-6,2.08e-6,2.63e-6,2.21e-`
	 `6,1.94e-6,2.68e-6,2.53e-6,1.92e-6,1.97e-6,1.85e-`
	 `6,1.88e-6,1.77e-6,1.73e-6,2.14e-6,1.84e-6,2.34e-`
	 `6,1.62e-6,1.67e-6,2.18e-6,1.64e-6,1.46e-6,1.61e-`
	 `6,2.76e-6,2.95e-6,2.17e-6,2.04e-6,2.53e-6,2.50e-`
	 `6,2.79e-6,2.67e-6,1.82e-6,1.80e-6 g, OK`

Details	Queries the spectral results of the VFFT analyzer that have been stored by the last MEAS:INIT command. If the parameter is unknown, a ";" is returned. The unit is always the same as in the user interface.

MEASure:VFFT:dt?

Shortcut	Queries a measurement dt result from the VFFT analyzer
Availability	VFFT
Parameter	EQ *string*

Answer		
`{Level`$_n$`,} unit,` `[OK│UNDEF│OVLD]`	*float* *string*	n = 143 Levels sorted from lowest to highest frequency. For the unit, see table in capture „Vibration Introduction".

Example

  💻→ `MEAS:INIT`
        `MEAS:VFFT:dt? EQ`
→💻 `2.65e-2,2.63e-2,2.71e-2,2.64e-2,2.72e-2,2.78e-2,2.84e-`
`2,2.84e-2,2.95e-2,2.88e-2,2.85e-2,2.86e-2,2.90e-`
`2,3.00e-2,2.88e-2,2.67e-2,2.59e-2,2.61e-2,2.43e-`
`2,2.42e-2,2.50e-2,2.43e-2,2.52e-2,2.58e-2,2.83e-`
`2,2.94e-2,2.78e-2,2.88e-2,2.87e-2,2.71e-2,2.48e-`
`2,2.43e-2,2.38e-2,2.37e-2,2.61e-2,2.53e-2,2.65e-`
`2,2.62e-2,2.53e-2,2.76e-2,2.74e-2,2.52e-2,2.63e-`
`2,2.54e-2,2.57e-2,2.74e-2,2.73e-2,2.66e-2,2.85e-`
`2,2.82e-2,2.84e-2,2.95e-2,2.60e-2,2.43e-2,2.58e-`
`2,2.78e-2,2.84e-2,2.94e-2,2.85e-2,3.01e-2,2.89e-`
`2,2.70e-2,2.66e-2,2.74e-2,2.64e-2,2.75e-2,2.77e-`
`2,2.81e-2,2.97e-2,2.76e-2,2.59e-2,2.61e-2,2.78e-`
`2,2.76e-2,2.74e-2,2.74e-2,2.90e-2,2.84e-2,2.61e-`
`2,2.83e-2,2.64e-2,2.44e-2,2.63e-2,2.46e-2,2.59e-`
`2,2.72e-2,2.81e-2,2.80e-2,2.80e-2,2.68e-2,2.74e-`
`2,2.78e-2,2.52e-2,2.48e-2,2.70e-2,2.73e-2,2.49e-`
`2,2.77e-2,2.88e-2,2.74e-2,2.64e-2,2.64e-2,2.71e-`
`2,2.64e-2,2.55e-2,2.64e-2,2.77e-2,2.61e-2,2.70e-`
`2,3.15e-2,3.07e-2,2.64e-2,2.82e-2,3.10e-2,2.96e-`
`2,2.84e-2,2.63e-2,2.83e-2,2.88e-2,2.68e-2,2.89e-`
`2,2.83e-2,2.72e-2,2.88e-2,2.77e-2,2.49e-2,2.92e-`
`2,2.98e-2,2.81e-2,2.72e-2,2.72e-2,2.63e-2,2.80e-`
`2,2.93e-2,2.65e-2,2.60e-2,2.81e-2,2.75e-2,2.96e-`
`2,3.14e-2,3.16e-2,2.78e-2,2.74e-2 g, OK`

Details   Queries the spectral results of the VFFT 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.

The unit is always the same as in the user interface.

MEASure:VFFT:PAGE

Shortcut	Defines the frequency range page
Availability	VFFT
Parameter	`1k7|200|100|usr` *string* 'usr' is only available with installed „Extended Acoustic" Pack or „Spectral Limits option"
Example	⌨→ `MEAS:VFFT:PAGE 1k7`

MEASure:VFFT:PAGE?

|	
Shortcut	Queries the selected frequency range page
Availability	VFFT
Answer	`1k7|200|100|usr` *string* 'usr' is only available with installed „Extended Acoustic Pack" or „Spectral Limits" option
Example	⌨→ `MEAS:VFFT:PAGE?` →⌨ `1k7`

MEASure:VFFT:ZOOM

Shortcut	Sets the zoom step
Availability	VFFT with installed Extended Acoustic Pack or Spectral Limits Option
Parameter	`0|1|2|...|9` *string* See table in 'MEAS:VFFT:ZOOM?'
Example	⌨→ `MEAS:VFFT:ZOOM 0`
Details	With zoom=0, the VFFT 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:VFFT:FSTArt

MEASure:VFFT:ZOOM?

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

<div align="right">

7: Δ = 208 Hz
8: Δ = 104 Hz
9: Δ = 52 Hz

</div>

| Example | ⌨→ | `MEAS:VFFT:ZOOM?` |
| | →⌨ | `0` |

MEASure:VFFT:F?

Shortcut	Queries the bin frequencies
Availability	VFFT
Answer	`<f_n,> Hz` *string* n = 143
Example	⌨→ `MEAS:VFFT:F?`

→⌨ `23.00,23.37,23.73,24.10,24.46,24.83,25.20,25.56,`
`25.93,26.30,26.66,27.03,27.39,27.76,28.13,28.49,`
`28.86,29.23,29.59,29.96,30.32,30.69,31.06,31.42,`
`31.79,32.16,32.52,32.89,33.25,33.62,33.99,34.35,`
`34.72,35.08,35.45,35.82,36.18,36.55,36.92,37.28,`
`37.65,38.01,38.38,38.75,39.11,39.48,39.85,40.21,`
`40.58,40.94,41.31,41.68,42.04,42.41,42.78,43.14,`
`43.51,43.87,44.24,44.61,44.97,45.34,45.71,46.07,`
`46.44,46.80,47.17,47.54,47.90,48.27,48.63,49.00,`
`49.37,49.73,50.10,50.47,50.83,51.20,51.56,51.93,`
`52.30,52.66,53.03,53.40,53.76,54.13,54.49,54.86,`
`55.23,55.59,55.96,56.33,56.69,57.06,57.42,57.79,`
`58.16,58.52,58.89,59.25,59.62,59.99,60.35,60.72,`
`61.09,61.45,61.82,62.18,62.55,62.92,63.28,63.65,`
`64.02,64.38,64.75,65.11,65.48,65.85,66.21,66.58,`
`66.95,67.31,67.68,68.04,68.41,68.78,69.14,69.51,`
`69.88,70.24,70.61,70.97,71.34,71.71,72.07,72.44,`
`72.80,73.17,73.54,73.90,74.27,74.64,75.00 Hz`

MEASure:VFFT:FSTArt

Shortcut	Defines the frequency of the first bin
Availability	VFFT with installed Extended Acoustic Pack or Spectral Limits Option
Parameter	`f` *float* in Hz
Example	⌨→ `MEAS:VFFT:FSTART 58.00`
Details	Defines the frequency of the first bin of the Zoom VFFT 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 Range of FSTART is 1Hz…19948Hz with full zoom in
	The MEAS:VFFT:F command can be used to verify the setting.

MEASure: V12OCT Subsystem (Domain: Vibration)

MEASure:V12OCT?

Shortcut	Queries the spectral results of the 1/12 Octave analyzer.
Availability	1/12 Oct (requires installed Spectral Limits Option)
Parameter	`[Live|Max|Min|` `EQ|EQ1"|EQ4"]` *string*

Answer		*float*	

	`{Level`$_n$`,} unit,`
	`[OK|UNDEF|OVLD|`
	`OPTION_REQUIRED]`
			1/1 Oct: n = 11 + 2*, f_{start} = 1.00 Hz
			1/3 Oct: n = 33 + 2*, f_{start} = 0.80 Hz
			1/6 Oct: n = 66 + 2*, f_{start} = 0.75 Hz
			1/12 Oct: n = 132 + 2*, f_{start} = 0.73 Hz

* Spectrum + 2 broad band results, as displayed on the XL2 screen. For the unit, see table in capture „Vibration Introduction".

Example	🖥→ `MEAS:INIT`
	      `MEAS:V12OCT? Live`
	→🖥 `8.66e-5,3.24e-5,1.08e-4,8.72e-5,8.71e-5,5.97e-5,6.66e-`
	      `5,6.28e-5,6.75e-5,7.33e-5,7.50e-5,2.51e-4,-999 m/s2, 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.

MEASure:V12OCT:dt?

Shortcut	Queries the spectral dt results of the 1/12 Octave analyzer.
Availability	1/12 Oct (requires installed Spectral Limits Option)
Parameter	`EQ` *string*

Answer		*float*	

	`{Level`$_n$`,} unit,`
	`[OK|UNDEF|OVLD|`
	`OPTION_REQUIRED]`
			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. For the unit, see table in capt. „Vibration Introduction".

Example	🖥→ `MEAS:INIT`
	      `MEAS:V12OCT:dt? EQ`
	→🖥 `9.15e-5,7.04e-5,6.41e-5,6.82e-5,5.53e-5,5.68e-5,6.04e-`
	      `5,6.07e-5,6.30e-5,6.80e-5,7.47e-5,2.23e-4,-999 m/s2, 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.

MEASure:V12OCT:RESOlution

Shortcut	Defines the resolution, in which the 1/12 Octave analyzer results are acquired.
Availability	1/12 Oct (requires installed Spectral Limits Option)
Parameter	`[1/1|1/3|1/6|1/12]` *string*
Example	🖥→ `MEAS:V12OCT:RESO 1/3`

MEASure:V12OCT:RESOlution?

Shortcut	Queries the resolution, in which the 1/12 Octave analyzer results are acquired.
Availability	1/12 Oct (requires installed Spectral Limits Option)
Answer	`[1/1|1/3|1/6|1/12]` *string*
Example	🖥→ `MEAS:V12OCT:RESO?` →🖥 `1/3`

INPUt Subsystem   51

INPUt:SELEct

Shortcut	Configures which input connector is selected.
Availability	always
Parameter	`[XLR|RCA]` *string*
Example	🖥→ `INPUT:SELECT XLR`

INPUt:SELEct?

Shortcut	Queries which input connector is selected.
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 <u>not</u> 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`

CALIbration Subsystem

CALIbration:MIC:TYPE?

Shortcut	Queries the microphone type recognized by the ASD (Automatic Sensor Detection) 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.

CALIbration: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.

CALIbration :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.

CALIbration: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`

CALIbration:MIC:CIC

Shortcut	Switches M2340 microphones (and MA230 mic preamps) into self-test mode. This means, the microphone generates a reference tone (square wave with 31.25 Hz or 1 kHz) which can be measured by the XL2.
Availability	Always
Parameter	OFF | F1 | F2

	OFF	*Normal microphone mode*
	F1	Microphone generates 31.25 Hz square wave
	F2	Microphone generates 1 kHz square wave

Example	🖥→ `CALIBRATION:MIC:CIC F1`

Details   To avoid CIC staying on (for example, if remote connection breaks during CIC measurement), CIC turns off automatically after 60 seconds. If you need the CIC signal for a longer time, retrigger the signal by sending the F1 or F2 command again within 60 seconds.

There are three error conditions handled by the system:
- Error No. 10: There is no microphone connected to the XL2
- Error No. 15: The connected microphone does not support CIC
- Error No. 16: Switching to the selected mode failed because of a bad hardware condition

To read the errors, use the system:error? command.

SYSTem Subsystem

SYSTem:ERROr?

Shortcut	Queries the error queue
Availability	always
Answer	{errno$_n$,} *integer* $n \le 10$

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 or switching off Phantom Power 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|LSTART|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. LSTART (long start) is used to enter/exit the locked run mode and to exit the XL2 scheduler.

SYSTem:KLOCk

Shortcut	Locks the keyboard of the XL2 or only Start,Pause,Page keys
Availability	always

Parameter	`[ON` `{key,}|OFF]`	*string*	ON: Keyboard is locked key = [START|PAUSE|PAGE]

Example	🖥→ `SYST:KLOCK ON` 🖥→ `SYST:KLOCK ON START` 🖥→ `SYST:KLOCK ON START Page`

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:OPTIons?

Shortcut	Queries the installed options
Availability	always

Answer	`<EAP,>` *string*
	`<STIPA,>`
	`<REMOTE,>`
	`<SLI,> <CA,> <TA,>`
	`<DX,> <SI,> <LW>`

Example	🖥➔ `SYST:OPTI?` ➔🖥 `EAP,REMOTE`
Details	These are the used abbreviations for the available Options: Extended Acoustic Pack (EAP), Spectral Limits Option (SLI), Cinema Assistant (CA), Type Approved (TA), Data Explorer (DX), Data Explorer (DX), Sound Insulation (SI), Sound Power (LW)

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.

To return to COM mode eject the XL2 drive from the host computer.

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 and Linux.
Availability	always
Answer	`No answer`
Example	💻→ `SYST:MSDMAC`

Details   Use this Command on Mac and Linux instead of „SYSTem:MSD", otherwise MSD will timeout after 2 minutes and the XL2 returns to COM mode.

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.

To return to COM mode eject the XL2 drive from the host computer.

Attention: If you unmount the XL2 drive by the host, the XL2 will not return to COM mode.

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 PRO and the Microsoft Excel and LabVIEW XL2 remote demonstration projects.

The XL2 Projector PRO 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 PRO 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.

Accessing an XL2 over the Internet

The NetBox adds internet access to the XL2 and allows SFTP (Secure File Transfer Protocol) and remote COM port access to the device. Mobil communication, LAN of Wi-Fi can be uses to connect the NetBox to the internet.

For an easy connection setup NTi Audio provides a gateway server on the internet[1]. From your PC, you can login to the server via SFTP to access the XL2 files. Also, the server provides access to the XL2 COM port from your software using a socket.
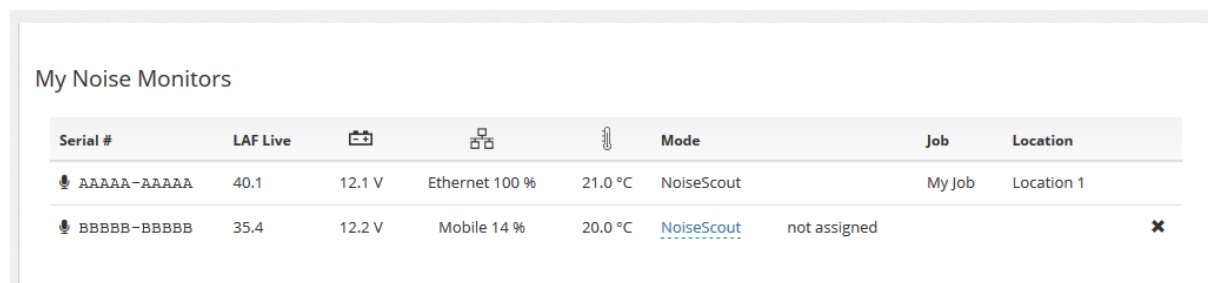
NETBOX GATEWAY MODE

A NetBox can either be used
- to connect the XL2 to the NoiseScout System www.noisescout.com, for fully-managed noise monitoring (for this the NetBox is set to NoiseScout mode) or
- to grant SFTP and COM port access (for this the NetBox is set to Gateway mode)

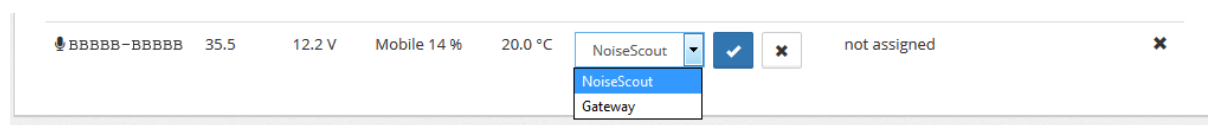### SWITCHING BETWEEN NOISESCOUT AND GATEWAY MODE

To activate the Gateway Mode for your NetBox you need to contact NTi Audio via email at info@nti-audio.com. Once activated, you can switch between NoiseScout and Gateway Mode on www.noisescout.com.

Picture 1: List of Noise Monitors[2] on the Dashboard in www.noisescout.com

In the image above, NetBox AAAAA-AAAAA is assigned to a Job and a Location. It is not available for switching into Gateway Mode. NetBox BBBBB-BBBBB is not assigned and can thus be switched to Gateway Mode:

Picture 2: Switching between NoiseScout and Gateway Mode

After switching the Mode, the NetBox will reboot. A NetBox in Gateway Mode is still connected to the NoiseScout server, but is not available anymore for assignment to a Location.

---

[1] This overcomes the limitations of local area networks and avoids the challenging router setup required for a direct connection from the NetBox to your PC.
[2] A Noise Monitor consists of a NetBox plus an XL2.

SFTP ACCESS

For details about the SFTP access please refer to the appropriate manual.

The connection details in brief:
- Server: xl2gateway.nti-audio.com
- Username: <xxxxx-xxxxx> (NetBox serial number)
- Password: <Your Password at my.nti-audio.com>
- Port: 22

COM PORT ACCESS

The gateway server xl2gateway.nti-audio.com listens on port 8432 for incoming, encrypted TLS connections. Once connected, the server expects a NetBox serial number and a password separated by a comma and terminated by a new line or by a carriage return and a new line:

`'XXXXX-XXXXX,MyPassWord<CR><LF>'.`

If your login is correct, the NetBox is online and the XL2 is connected to the NetBox, the server responds with the following answer:

`'Login OK, NetBox OK, XL2 OK<CR><LF>'.`

> Hint: In case of any error the server returns an error message and terminates the connection. The following errors are possible:

Error	Message sent by server
Serial number or password is incorrect	`Login incorrect<CR><LF>`
NetBox is offline	`Login OK, NetBox offline<CR><LF>`
NetBox is already in use by another remote connection	`Login OK, NetBox already in use<CR><LF>`
XL2 is not connected to the NetBox	`Login OK, NetBox OK, XL2 not connected<CR><LF>`

After a successful login, the server switches the connection to the related Noise Monitor and you can send any remote command to the XL2 in the same way you do with a local USB connection. Only one connection per NetBox is possible at the time.

COM PORT ACCESS SAMPLE CODE

Python 3.7, usage: **python TestXL2Gateway.py**

```python
import socket
import select
import ssl

Response = str()

def readline(Conn):
    global Response
    Line = ''
    while True:
        rd, wr, er = select.select([Conn], [], [], 0.5)
        if rd:
            Packet = Conn.recv(200)
            if not Packet:
                Conn.close()
                break
            Response += Packet.decode("utf-8")
            if '\n' in Response:
                Lines = Response.split('\n', 1)
                Line = Lines[0]
                Response = Lines[1]
                break
    return Line.strip('\r')

Serial = 'XXXXX-XXXXX'
Passwd = 'MyPassWord'
USE_GATEWAY = True

if USE_GATEWAY:
    Login = ','.join([Serial, Passwd]) + '\r\n'

    _s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    Conn = ssl.wrap_socket(_s, ssl_version=ssl.PROTOCOL_TLSv1)
    Conn.connect(('xl2gateway.nti-audio.com', 8432))
else:
    Login = Passwd + '\r\n'

    Conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    Conn.connect(('XXX.XXX.XXX.XXX', 50505))   # Local IP Address

Conn.send(str.encode(Login))
LoginResult = readline(Conn)
print(LoginResult)

if 'XL2 OK' in LoginResult:
    Conn.send(b'*idn?\r\n')
    IdnResult = readline(Conn)
    print(IdnResult)

Conn.close()
```