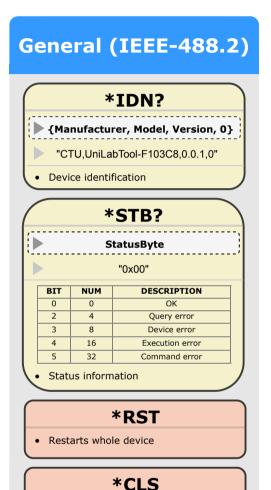
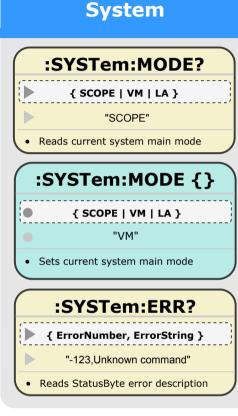
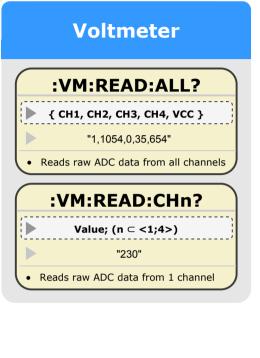
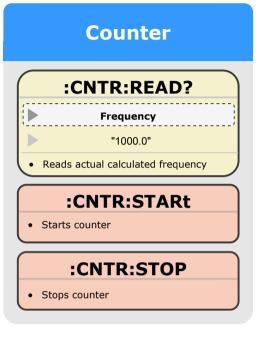
UniLabTool Protocol v0.1

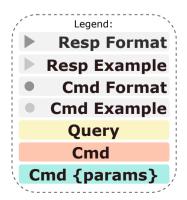


• Resets StatusByte to default value 0x00









Oscilloscope



{BIT,MEM,FS,CH, {TCH,TL,TE,TM}}

"12, 1K, 5M, TFFF, 1, 0.0, R, A"

	PRM	RANGE	DESCRIPTION
	BIT	12 / 8	Each channel's sample bitness
	MEM	0 - 32K	Each channel 's memory depth
	FS	0 - 5M	Sampling frequency in Hz
	CH	4x T / F	Enabled channels (True / False)
	TCH	1 - 4	Trigger source channel number
	TL	0 - 100%	Trigger level (percentage)
	TE	R/F	Trigger edge: R ising / F alling
	TM	A/N/S	Trig. mode: Auto / Normal / Single

• Oscilloscope settings - all

:SCOPe:SET:BIT {}

{12 | 8}

"12"

• Scope settings - sample bitness

:SCOPe:SET:MEM {}

{<1;32K>}

"1K"

• Scope settings - memory depth

:SCOPe:SET:FS {}

{<1;5M>}

"5M"

Scope settings - sampling frequency

:SCOPe:SET:CH {}

BBBB}; (B ⊂ {T | F})

"TTFF"

• Enabled/disabled 4 channels settings

:SCOPe:GET:ALL?

{BIT,MEM,FS,CH, {TCH,TL,TE,TM}}

"12, 1K, 5M, TFFF, 1, 0.0, R, A"

• Reads all settings (viz. :SET:ALL)

:SCOPe:READ:ALL?

#DN{CH1-B0 ..}{CH2-B0 ..}}

"#104 <0x01><0x02> <0x01><0x02>"

- Reads all channels (GPIB binary syntax)
- Either 8-bit or 16-bit, first channel first
- Reenables DMA

:SCOPe:SET:TRIG {}

● {<1;4>,<0;100>,{R|F},{A|N|S}}

"1, 0.0, R, A"

• Oscilloscope settings - trigger

:SCOPe:READ:CHn?

"#102 <0x01><0x02>"

- Reads 1 channel (GPIB binary syntax)
- Either 8-bit or 16-bit raw ADC data
- · Reenables DMA

:SCOPe:RST

- Resets scope buffers
- Reenables DMA

Logic Analyzer

:LA:SET:ALL {}

{MEM,CH, {TCH,TE,TM}}

"1K, TFFF, 1, R, A"

PRM	RANGE	DESCRIPTION
MEM	0 - 32K	Each channel's memory depth
СН	4x T / F	Enabled channels (True / False)
TCH	1 - 4	Trigger source channel number
TE	R/F	Trigger edge: R ising / F alling
TM	A/N/S	Trig. mode: Auto / Normal / Single

• Logic analyzer settings - all

:LA:SET:MEM {}

{<1;32K>}

"1K"

• LA settings - memory depth

:LA:SET:CH {}

 $\{BBBB\}; (B \subset \{T \mid F\})$

"TTFF"

• LA settings - trigger

:LA:SET:TRIG {}

{<1;4>,{R|F},{A|N|S}}

"1, R, A"

• Enabled/disabled 4 channels settings

:LA:RST

- Resets scope buffers
- Enables DMA

:LA:SET?

{MEM,CH, {TCH,TE,TM}}

"1K, TFFF, 1, R, A"

• Reads all settings (viz. :SET:ALL)

:LA:READ:ALL?

#DN{CH1-B0 ..}{CH2-B0 ..}}

"#104 <0x01><0x02> <0x01><0x02>"

- Reads all channels (GPIB binary syntax)
- Bits packed into bytes, first chann first
- Reenables DMA

:LA:READ:CHn?

{#DN B0 B1 ..}; (n ⊂ <1;4>)

"#102 <0x01><0x02>"

- Reads 1 channel (GPIB binary syntax)
- Bits packed into bytes
- · Reenables DMA

Signal Generator

:SGEN:SET:CHn?

FREQ, AMPL, TYPE}; (n ⊂ <1;2>)

"1000,50,SINE"

PRM	RANGE	DESCRIPTION
FREQ	1 - 1M	Generated signal frequency in Hz
AMPL	0 - 100	Generated signal amplitude in %
TYPE		SINE SAW SQUARE TRIA NOISE

• Reads signal generator settings

:SGEN:SET:CHn {}

{ FREQ, AMPL, TYPE }

"800,40,SQUARE"

• Sets signal generator parameters

:SGEN:STARt

• Starts signal generator

:SGEN:STOP

• Stops signal generator

PWM Generator

:PWM:SET:CHn?

FREQ, DUTY }; (n ⊂ <1;2>)

"1100,60"

l	PRM	RANGE	DESCRIPTION
	FREQ	1 - 1M	Generated PWM frequency in Hz
	DUTY	0 - 100	Generated PWM duty in %

• Reads PWM generator settings

:PWM:SET:CHn {}

{ FREQ, DUTY }

"900,30"

• Sets PWM generator parameters

:PWM:STARt

• Starts PWM generator

:PWM:STOP

• Stops PWM generator

Overview

```
*IDN?
                                      :SCOPe:RST
*STB?
                                                       { BIT, MEM, FS, CH, {TCH, TL, TE, TM} }
                                      :SCOPe:SET:ALL
*RST
                                      :SCOPe:SET:BIT
                                                       { 12 | 8 }
*CLS
                                      :SCOPe:SET:MEM { <1;32K> }
                                      :SCOPe:SET:FS
                                                       { <1;1M> }
                                      :SCOPe:SET:CH
                                                       { BBBB; (B ⊂ {T | F}) }
:SYSTem:MODE?
                                      :SCOPe:SET:TRIG
                                                       { <1;4>,<0;100>,{R|F},{A|N|S} }
:SYSTem:MODE { SCOPE | VM | LA }
                                      :SCOPe:GET:ALL?
:SYSTem:ERR?
                                      :SCOPe:READ:ALL?
                                      :SCOPe:READ:CHn?
:VM:READ:ALL?
                                      :LA:RST
:VM:READ:CHn?
                                                       { MEM,CH, {TCH,TE,TM} }
                                      :LA:SET:ALL
                                                       { <1;32K> }
                                      :LA:SET:MEM
                                                       { BBBB; (B ⊂ {T | F}) }
                                      :LA:SET:CH
:SGEN:SET:CHn?
                                                       { <1;4>,{R|F},{A|N|S} }
                                      :LA:SET:TRIG
:SGEN:SET:CHn { FREQ, AMPL, TYPE }
                                      :LA:SET?
:SGEN:STARt
                                      :LA:READ:ALL?
:SGEN:STOP
                                      :LA:READ:CHn?
:PWM:SET:CHn?
                                      :CNTR:READ?
:PWM:SET:CHn { FREQ, DUTY }
                                      :CNTR:STARt
:PWM:STARt
                                      :CNTR:STOP
:PWM:STOP
```

Examples

```
// Identify after connect
> :IDN?
$ CTU,UniLabTool-F103C8,0.0.2,0
// Restart, set mode to Voltmeter, check Status, and read all channels
> *RST
> :SYST:MODE VM
> *STB?
$ 0x00
> :VM:READ:ALL?
$ 1,2024,0,35,654
// Try to reset Scope, check Status, failed so read error message, and reset status
> :SCOP:RST
> *STB?
$ 0x16
> :SYST:ERR?
$ -2, Invalid mode
> *CLS
// Switch mode to Scope, set up, check Status, optionally reset and read buffer of CH3
>:SYST:MODE SCOPE
> :SCOP:SET:ALL 8,2,1K,FFTF,3,50,F,S
> *STB?
$ 0x00
> :SCOP:RST
> :SCOP:READ:CH3?
$ #3002 <0x01><0x02>
```