LUNA

v

User Manual | MPX 2010

	Many regions prohibit the disposal of WEEE (Waste Electrical and Electronic Equipment) in the normal waste stream, to comply with the Restriction of Hazardous Substances (RoHS) released into the environment. Please contact your local waste authority for instructions on proper recycling of the electronic product(s) described in this *User Manual*.

MPX 2010 User Manual

Luna Innovations Inc.
3155 State Street
Blacksburg, VA 24060
Phone: (866) 586-2682
Fax: (540) 961-5191
E-mail: solutions@lunainc.com
Web: www.lunainc.com

Table of Contents

1. Safety

The advisory words **Danger**, **Warning** and **Caution** used in this manual identify the level of hazard that may be encountered by the user.

- **DANGER** means if the danger is not avoided, it will cause death or serious injury.
- **WARNING** means if the warning is not heeded, it can cause death or serious injury.
- **CAUTION** means if the precaution is not taken, it may cause minor or moderate injury.

Warning

The protection provided by the equipment may be impaired if the equipment is used in a manner not specified by the manufacturer, resulting in serious injury or death.

The power cord is the main electrical disconnect for this equipment. If it is necessary to ensure no power to the unit, remove the power cord.

The use of controls, adjustments, performance, or procedures other than those specified herein may result in hazardous laser radiation exposure and one or more safety protections may be impaired or rendered ineffective.

Attention

La protection fournie par l'équipement peut être compromise si l'équipement est utilisé d'une manière non spécifiée par le fabricant, entraînant des blessures graves ou la mort.

Le cordon d'alimentation est le principal disjoncteur électrique de cet équipement. S'il est nécessaire de ne pas mettre l'appareil hors tension, retirez le cordon d'alimentation.

L'utilisation de commandes, d'ajustements, de performances ou de procédures autres que celles spécifiées ici peut entraîner une exposition dangereuse au rayonnement laser et une ou plusieurs protections de sécurité peuvent être altérées ou rendues inefficaces.

2. Technical Support

For technical support please visit our webpage Customer Support Portal

2.1. Mail

Luna Innovations Inc.
3155 State Street
Blacksburg, VA 24060

2.2. Phone

Main Phone: +1 (540) 961-5190, +1 (540) 552-5128
Toll-Free Support: +1.866.LUNA OVA (+1.866.586.2682)

2.3. E-mail

We encourage users to create an account and sign-in to the help center to be able to send support tickets.

3. Overview

Luna Innovations Multifunction Polarization Controller (MPX-2010) is a general-purpose, all-fiber design, lossless polarization controller. With its versatile operation modes, the MPX-2010 platform provides comprehensive control over polarization for a wide array of applications. It features four distinct polarization scrambling profiles, each tailored for specific needs:

Tornado Scrambling: This mode allows the State of Polarization (SOP) to trace a spiral pattern around a static or rotating axis, ensuring a nearly uniform SOP variation rate.

Rayleigh Scrambling: This mode creates a continuous trace with a Rayleigh distribution of SOP variation rates, mimicking the SOP variations typically seen in fiber links.

Triangle Scrambling: This mode produces a continuous trace that covers the sphere uniformly, making it ideal for Polarization Dependent Loss (PDL) measurements.

Discrete Scrambling: This mode evenly distributes discrete, random points over the Poincaré sphere at a consistent rate, perfect for randomizing SOP and reducing polarization sensitivity.

In SOP Modulation mode, each polarization control axis can be individually controlled using sine, square, or triangle waves with user-defined frequencies and amplitudes. Additionally, Manual Control is available by adjusting the input voltage via the front panel controls or a remote-control interface.

For applications requiring synchronization with other devices, the MPX-2010 offers an externally triggered scrambling mode. In this mode, discrete, random SOPs are generated in response to a trigger input, making it especially suitable for recirculating loop applications.

There are 2 configurations available:

MPX2010-1-FC/APC: Multifunction Polarization Controller, 1260 nm to 1620 nm, FC/APC. MPX-2010 main frame for C and L band. Power cable, USB 2.0 A to B cable, USB drive for GUI and documents.

MPX2010-2-FC/APC: Multifunction Polarization Controller, 980 nm to 1310 nm, FC/APC. MPX-2010 main frame for 1um band. Power cable, USB 2.0 A to B cable, USB drive for GUI and documents.

3.1. Principle of Operation

A polarization controller can be constructed either from a series of rotatable fixed retardation plates or a series of variable retardation plates of fixed orientation. An example of this second construction, with four variable retardation wave plates, is shown in Figure 3.1 a). The MPX-2010 is built around a fiber-based version of this design, the four-axis PolaRite™ III polarization controller, which consists of four piezoelectric actuator-driven fiber squeezers oriented at 0°-45°- 0°-45°. This structure is shown in Figure 3.2 b). Each fiber squeezer is driven by an applied voltage signal. Squeezing the optical fiber produces a linear birefringence in the fiber, which alters the state of polarization of a light signal passing through it.

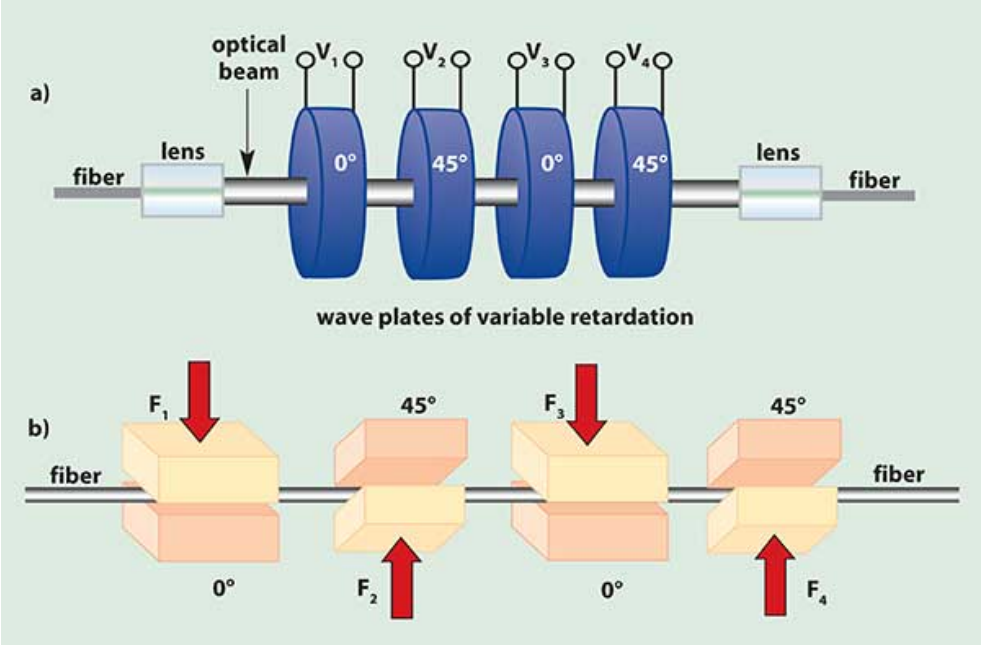
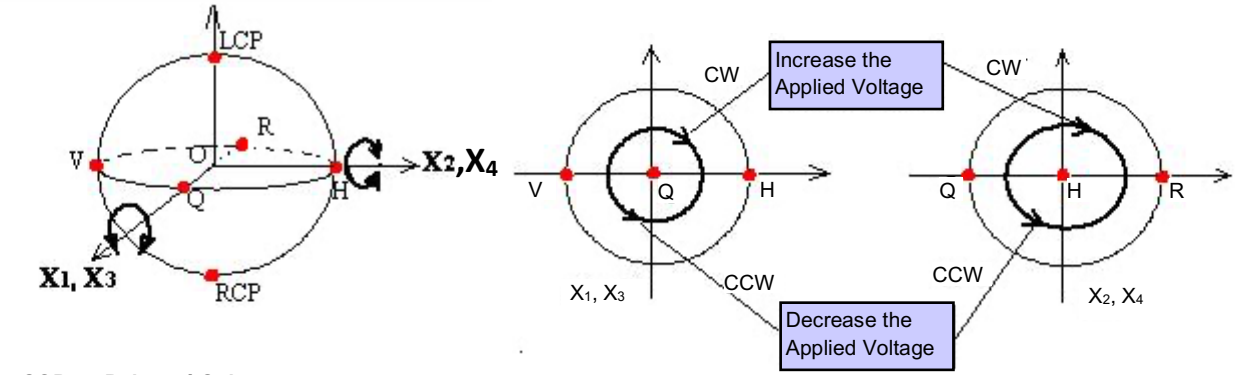


Figure 3.1 a) Free space construction of variable retardation plate polarization controller. b) Fiber squeezer construction of variable retardation plate polarization controller.

An arbitrary polarization state of monochromatic light can be represented by a single point on the Poincaré Sphere, as shown in Figure 3.2. Increasing the voltage on one fiber squeezer (X1 or X3) increases the pressure from that squeezer, causing the polarization state to rotate clockwise about the OQ axis. Decreasing the voltage causes the point to rotate counterclockwise. Similarly, increasing or decreasing the voltage on a second fiber squeezer (X2 or X4) oriented 45° from the first one causes the polarization state to rotate clockwise or counterclockwise about an axis (OH axis) orthogonal to the first one.



SOP on Poincaré Sphere

H: Horizontal Linear Polarization V: Vertical Linear Polarization Q: +45 degree Linear Polarization

R: −45 degree Linear Polarization LCP: Left Circular Polarization RCP: Right Circular Polarization

Figure 3.2 Poincaré Sphere SOP representation and illustration of effects of fiber squeezers

Using different modes of control of the drive voltages, the polarization controller can perform various functions. In Manual Control mode, the user has direct control over the output polarization state. By adjusting the DC drive voltage to each of the 4 fiber squeezers via the user interface, the output polarization can be changed from any initial state to the desired final state.

If the output polarization is required to change with time in a controlled fashion, the built-in waveform generator can generate a sine, triangle, or square wave of user-controlled amplitude, offset, and frequency on each of the 4 channels. This function can be used to accomplish anything from a simple periodic switching between two polarization states to a periodic sweeping out of a complicated pattern on the Poincaré sphere.

When acting as a Polarization Scrambler, the MPX-2010 varies the output polarization state for even coverage of the Poincaré sphere using either a series of discontinuous random points (discrete scrambling) or a continuous trace (triangle, Rayleigh, and Tornado scrambling). The continuous scrambling methods create different SOP trace patterns and SOP variation rate distributions. The MPX-2010 also has a triggered random state generation function, in which it generates a random polarization state triggered by the rising edge of an input TTL trigger signal.

4. Hardware and Physical Setup

4.1. Unpacking

Inspect the MPX 2010 for any physical damage resulting from shipping and transportation. If any damage is found, please contact the carrier immediately. Additionally, carefully review the packing list to verify if any parts or accessories are missing.

Item	Packing list Description
1	MPX 2010
2	Power Cord
3	USB2.0 Cable
4	Ethernet Cable
6	USB key containing User Interface Installation Package, User Manual, Packing lists

4.2. Front Panel Features

The front panel of the MPX 2010 is shown in Figure 4.1. Front panel features:

Power button: Power on/off button

Input: Adapter for optical fiber input

Output: Adapter for optical fiber output

Figure 4.1 MPX 2010 front panel

4.3. Rear Panel Features

The rear panel of the MPX 2010 is shown in Figure 4.2. Rear Panel Features:

ETHERNET: Ethernet interface port

USB2.0: USB 2.0 interface port

SYNC BUS: Instruments communication sync bus port.

TRIGGER IN: SMA for input trigger

TRIGGER OUT: SMA for output trigger

POWER ENTRY: External AC input connector with fuse case

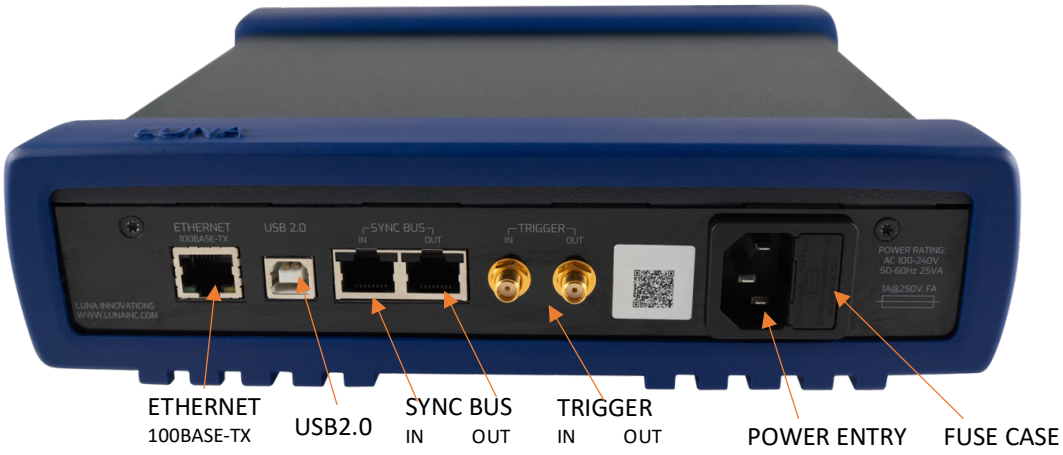


Figure 4.2 MPX 2010 rear panel

4.4. Communication Connection

4.4.1. USB Communications

To communicate with the MPX 2010 over an USB connection, connect a USB cable with a USB-B male connector to the USB-B female connector on the back panel. The other end of the USB cable must then be directly connected to the PC.

4.4.2. Ethernet Communications

To communicate with the MPX 2010 over an Ethernet connection, connect a Cat-5e or greater cable to the Ethernet female connector on the back panel (distinguished with the tab pointing down). Be careful not to connect to one of the two LVDS female connectors (distinguished with the tab pointing up).

The other end of the cable can then be connected to an Ethernet router that a host PC is also connected to or be connected directly to a PC. When connected to an Ethernet router, the MPX 2010 will have the static IP address of 192.168.1.150 by default. Please refer to the SCPI commands section, on how to change the IP address or enable dynamic IP addressing.

When connecting directly to a PC, one must set the IP address strategy to DHCP to allow the MPX 2010 to acquire a link-local IP address.

4.5. Starting the system

To start the system, press the power button on the front panel. The power button will light. The system is ready for use.

4.6. Shutting the system down

To turn off the system, press and hold the power button for about 0.5 second. If the system does not begin to shut down, press and hold the power button for about 2 seconds to force the system to turn off.

5. Working Modes

5.1. Polarization Scrambling Mode

The MPX-2010 features several built-in polarization scrambling functions. Three of these functions—Triangle, Rayleigh, and Tornado—create continuous state-of-polarization (SOP) traces on the Poincaré sphere. The fourth function, Discrete, produces a sequence of randomly distributed, discrete SOPs. The fifth function, Trigger, generates discrete random states in response to an external trigger signal instead of operating at a predetermined rate.

Scrambling Type	Scrambling Rate Unit	Set Value	Range
Triangle	Multiples of 2π/s	Highest frequency used to drive a channel of the polarization controller	0.00 – 4,000
Rayleigh	Rad/s	Mean of Rayleigh distribution	0.00 – 4,000
Tornado	Revolutions/s	# of rotations per second	0.00 – 4,000
Discrete	Points/s	# of points/s generated	0.00 – 40,000

5.2. Manual Polarization Control Mode

This mode enables the user to transform any input polarization state into any desired output polarization state by adjusting the DC control voltage signals applied to each fiber squeezer. By varying the voltage on each channel, the position of a point on the Poincaré sphere can be rotated about a specific axis. There are a total of 4 channels, and each channel can achieve a rotation of up to 4π on the Poincaré sphere.

5.3. Polarization Modulation Mode

In this mode, an internal function generator supplies the drive voltage signals to each of the four channels of the polarization controller. For each channel, the user can select a sine, triangle, or square waveform, and then set the frequency, amplitude, and offset for that waveform. Since the piezoelectric actuators require the application of positive voltages only, the amplitude setting is always less than the offset; as the amplitude is increased, the offset adjusts accordingly.

5.4. ‘Hello world’ test connection

For optimal light polarization monitoring, pair the Luna POD 2000 with MPX-2010. Connect both devices to your PC via USB. Configure the Trigger-In and Trigger-Out functions through the user interface program to achieve synchronized operation. This ensures precision in observing and analyzing polarization states for efficient monitoring while MPX-2010 is working at discrete scrambling or modulation mode. See Figure 5.1.

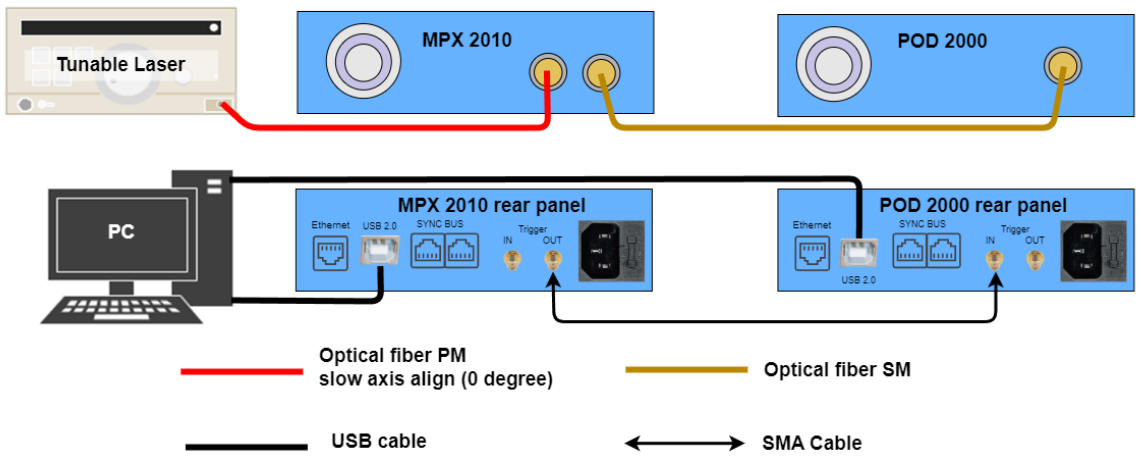


Figure 5.1 'Hello world' test connection

6. PolaX PC software

The PolaX Desktop Application enables communication with an MPX-2010 and a computer via Ethernet and USB. We provide a convenient software installation package to ensure smooth software installation.

Let us take a tour of the application's User Interface (UI). The tour will start with an introduction to the UI layout, then further explain each UI component.

6.1. Installation Package

6.1.1. System Requirements

Minimum:

OSG: Windows 10;

CPU: i3 5th Generation; R

AM: 1 available Gigabyte

Recommended:

CPU: i5 5th Generation or better

RAM: 2 available Gigabytes or better

6.1.2. Software Installation

To install the PolaX-2000 software, run the installer.

Specify the installation folder.

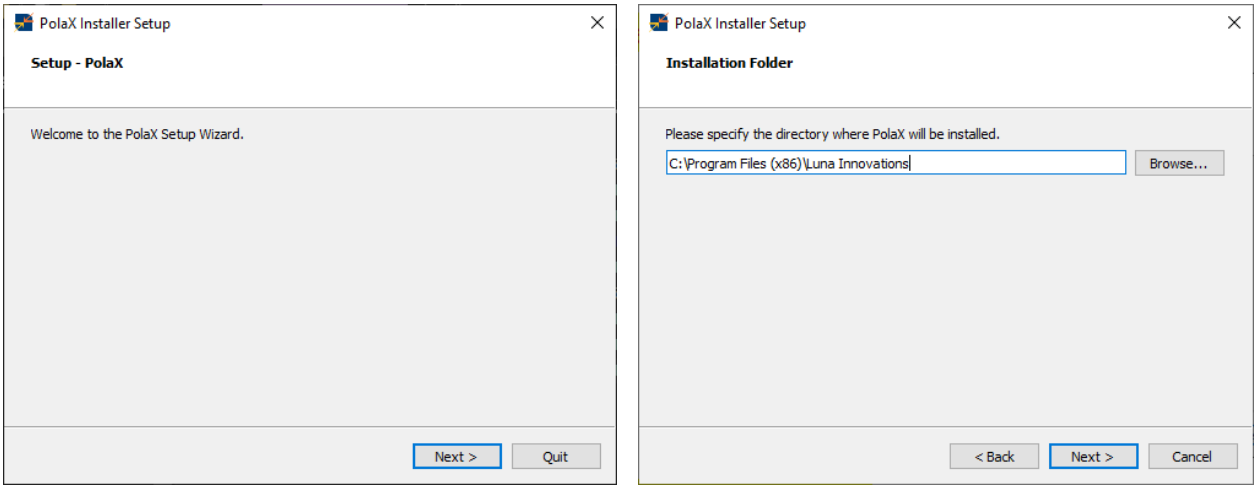


Figure 6.1 Polax Installer Setup

6.1.3. Software Removal

To Uninstall the software, run the Uninstaller from the installation folder. Select “Remove all components” and then click “Next” and uninstall to continue through the process.

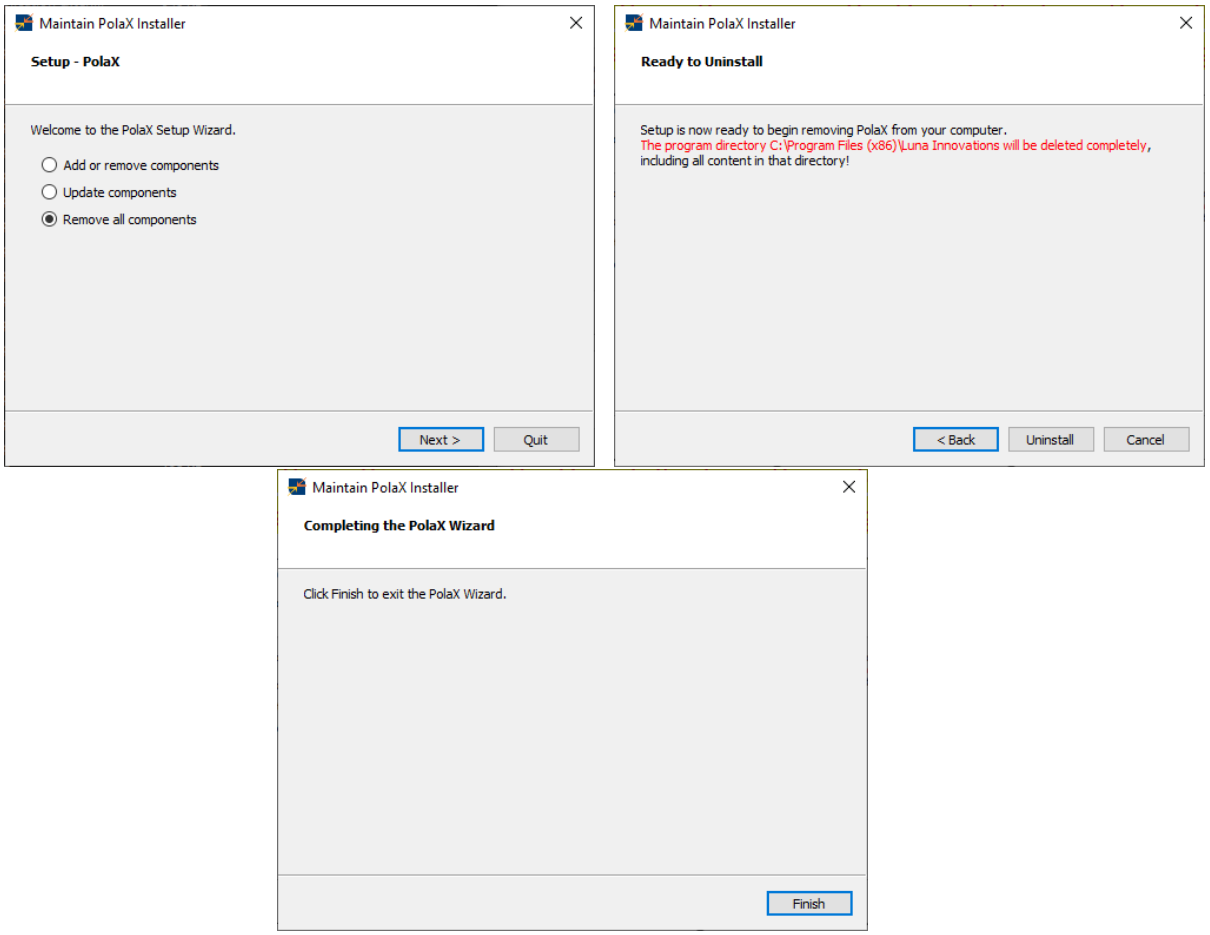


Figure 6.2 Polax Installer pages during uninstallation.

6.2. User interface overview

After starting the PolaX desktop application, the application UI screen will appear on the PC screen. The image below (see Figure 6.3) is an illustration of a typical UI screen layout with all optional displays turned on for a MPX-2010 instrument.

The central section of the UI is the Main Page Area.

The upper section of the UI screen is for the display of the Menu Ribbon.

The left section of the UI screen is the optional Device Selection Area where the user can select the current active device.

Each of the above UI components is further explained in the following titled sections, respectively.

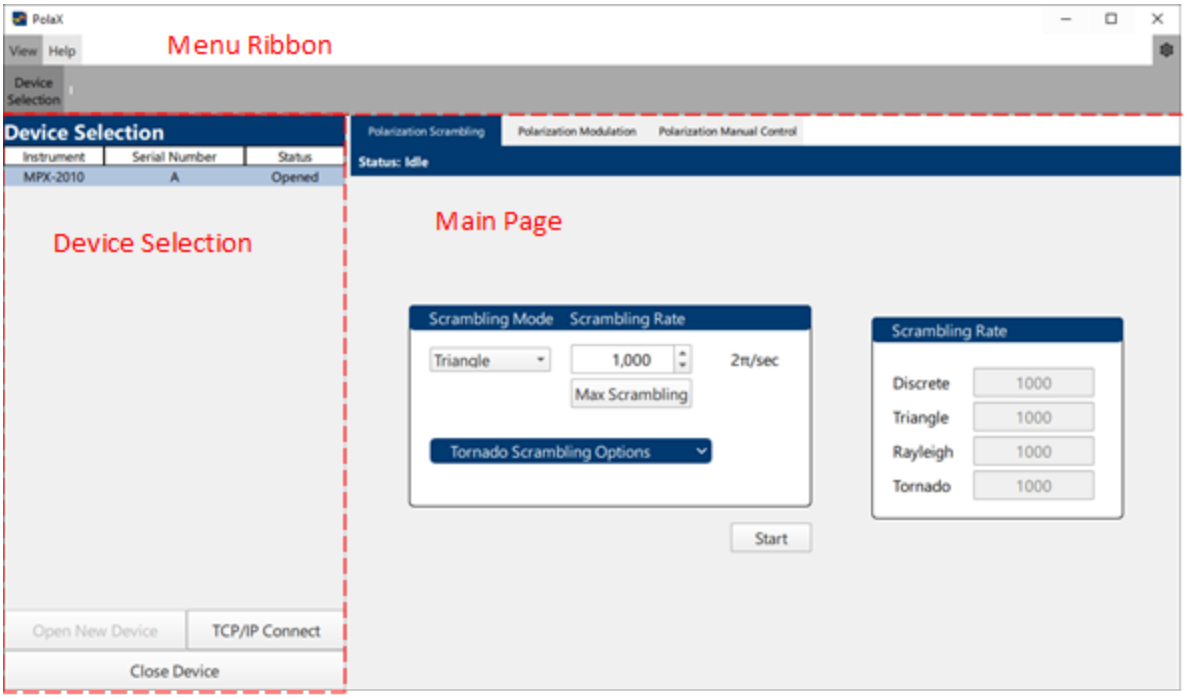


Figure 6.3 UI Screen Layout

6.2.1. Opening and Closing Devices

Device Selection is open by default when opening the application. Device Selection can be toggled on/off from the View Tab in the Menu Ribbon.

6.2.2. Setting the Active Device

Figure 6.4 Device Selection

Devices can be opened either via USB or TCP/IP Connection.

USB connected devices will automatically populate in the list and can be selected and opened via the “Open New Device” button.

TCP/IP connected devices must be connected via the “TCP/IP Connect” button.

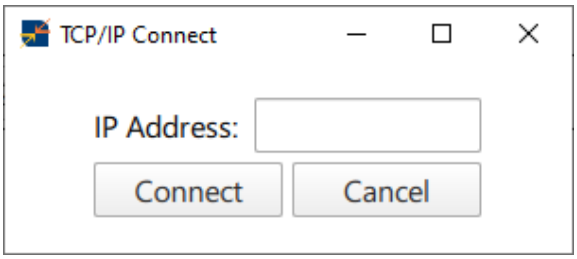


Figure 6.5 TCP/IP Connect Dialog Box

To complete the TCP/IP connection, enter the device’s IP Address and click connect. If it successfully connects, the device will then be listed in the Device Selection area.

6.2.3. Closing Devices

Devices can be closed by selecting them in the Device Selection area and then pressing the “Close Device” button.

6.3. Polarization Scrambling

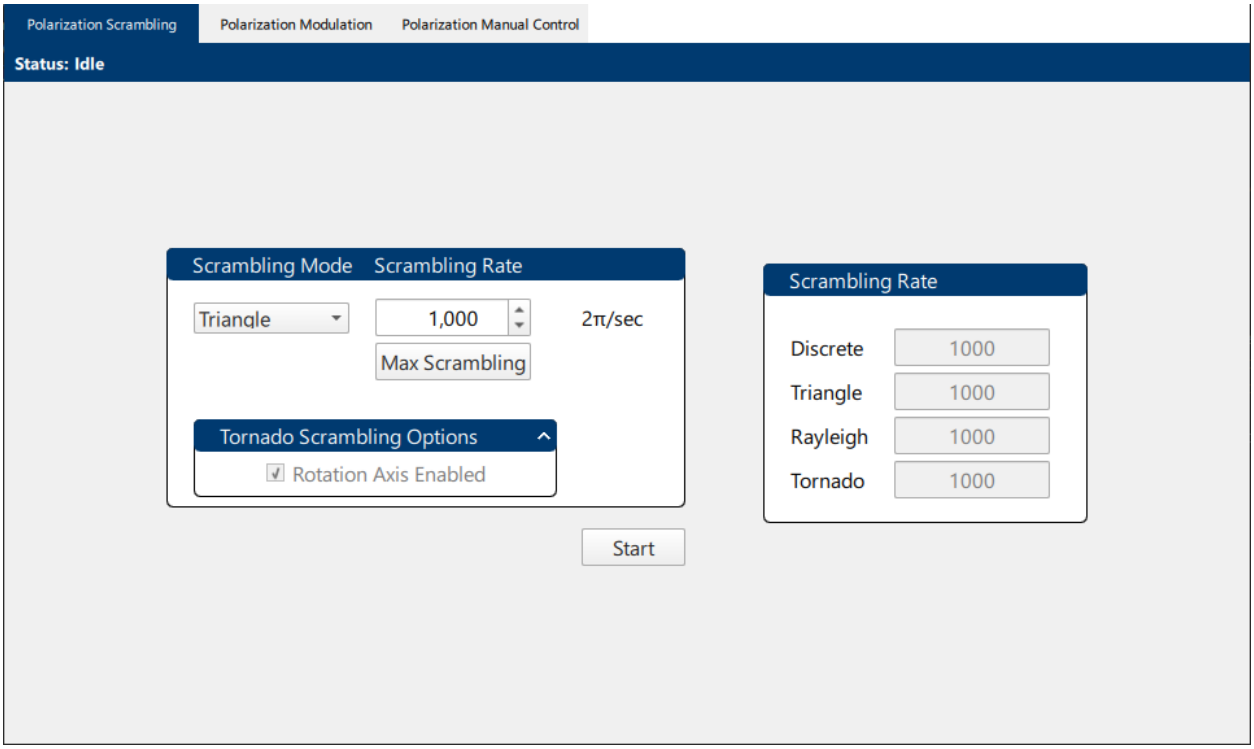


Figure 6.6 Polarization Scrambling

The Polarization Scrambling screen allows for setting the MPX-2010 Scrambling Modes and rates.

The Scrambling Modes available are Discrete, Triangle, Rayleigh, and Tornado

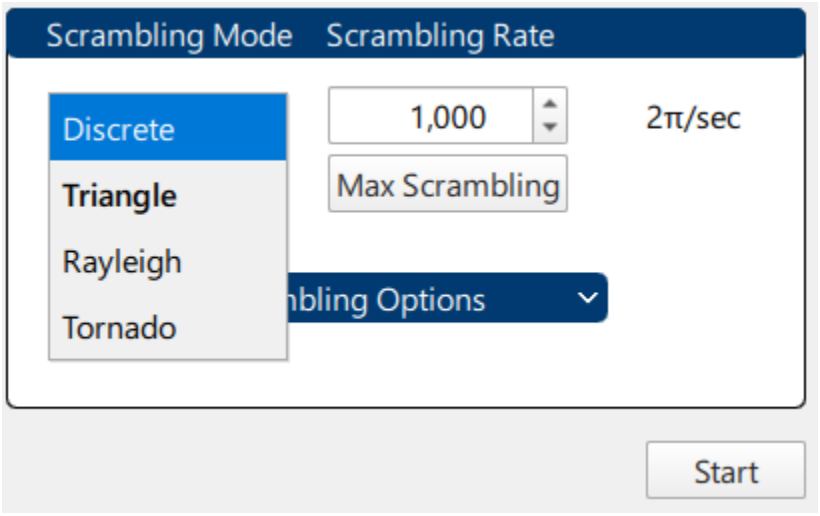


Figure 6.7 Polarization Scrambling Modes

Each Scrambling Mode has its own Scrambling Rate and range associated with it.

Discrete Scrambling: 0 - 40,000 Points/sec
Triangle Scrambling: 0 - 4000 2π/sec

Rayleigh Scrambling: 0 - 4000 Radians/sec
Tornado Scrambling: 0 - 4000 Revolutions/sec

The Scrambling Rate can be set to the maximum value by pressing the “Max Scrambling” button.

Under Tornado Scrambling Options, the Rotation Axis can be enabled or disabled for Tornado Scrambling. By enabling/disabling the Rotation Axis it enables/disables the fourth channel when using Tornado Scrambling.

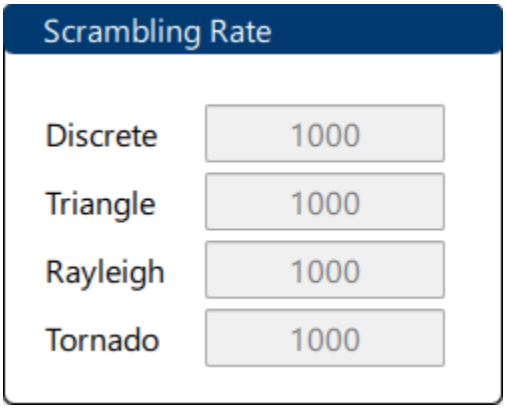


Figure 6.8 Scrambling Rate Table

The Scramble Rate Table shows the last set Scrambling Rate of each mode. Swapping between Scrambling Modes will automatically load the last set Scrambling Rate.

Finally, to initiate Polarization Scrambling, press the “Start” button after setting the Scrambling Mode and Rate.

6.4. Polarization Modulation

The Polarization Modulation screen allows for setting a waveform with its Amplitude, Offset, and Frequency for each channel. A preview of the expected waveform for each channel is shown and updated as parameters are set.

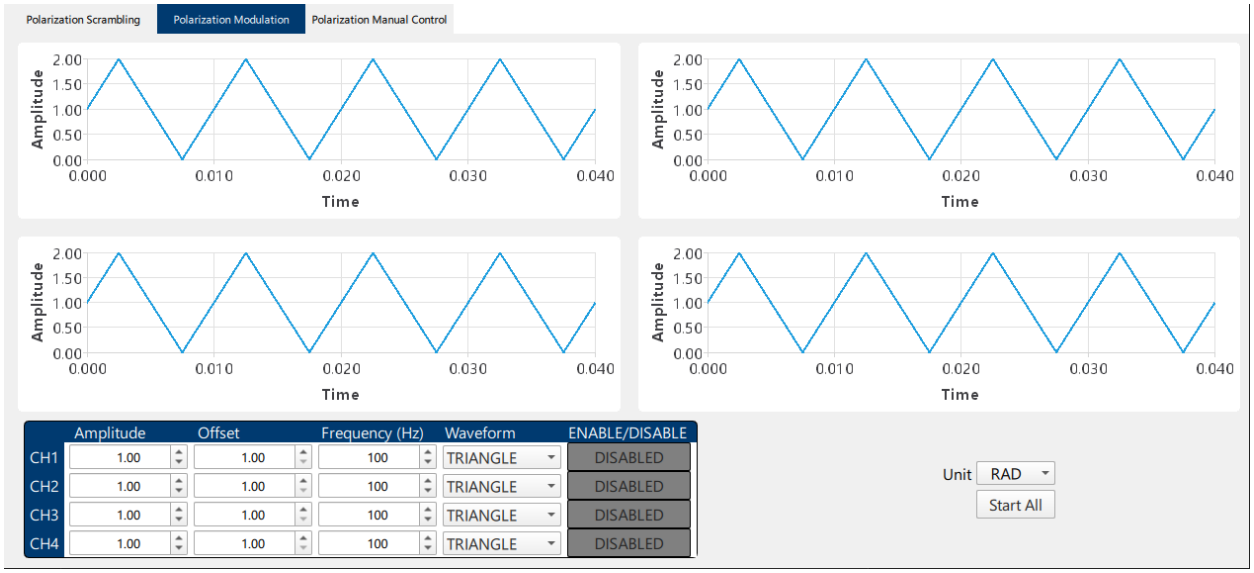


Figure 6.9 Polarization Modulation

Parameters are set in the bottom area using spin-boxes and dropdown menus. The individual parameters are defined below:

Amplitude: 0-1.5 Radians or 0-4π
Offset: 0-1.5 Radians or 0-4π
Frequency: 0-1000 Hz
Waveform: SINE, TRIANGLE, or SQUARE

Each channel can be enabled or disabled clicking the button next to the Waveform selection. If you want to enable/disable all channels at once, you can press the “Start All”/”Pause All” button on the right hand side.

The Unit used for setting parameters can also be changed from RAD and PI using the dropdown above the “Start All”/”Pause All” button.

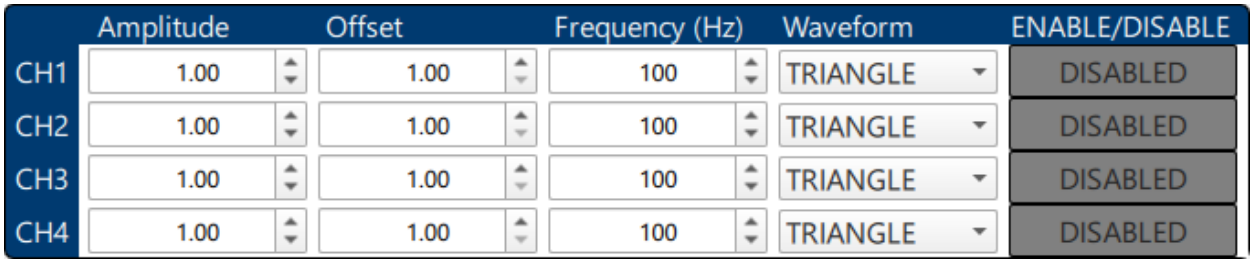


Figure 6.10 Polarization Modulation Parameter Table

6.5. Polarization Manual Control

The Polarization Manual Control screen allows for the user to convert any input polarization state into any desired output polarization state by tuning the Amplitude to each fiber squeezer. Changing the amplitude on each channel rotates the position of a point on the Poincaré sphere about some axis.

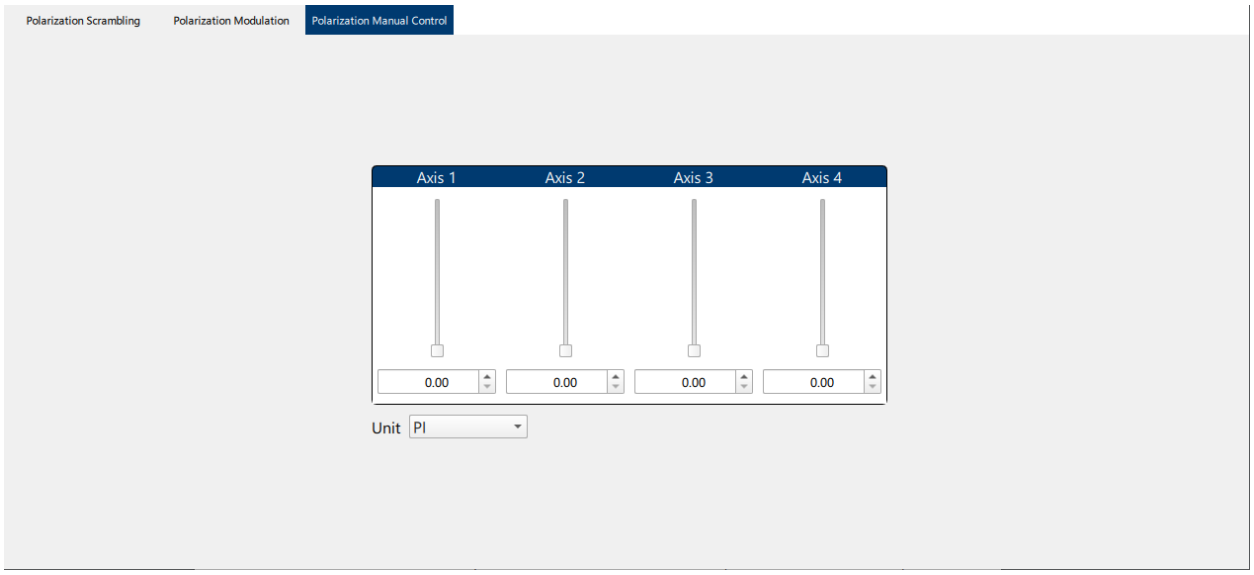


Figure 6.11 Polarization Manual Control

6.6. User interface menu and settings

Figure 1.12 Menu Ribbon

6.6.1. View Tab

The View Tab has one function: Device Selection.

Device Selection: Opens the Device Selection Area

Figure 6.13 View Tab

6.6.2. Help Tab

The Help Tab has three functions: Help, About, and What's New.

Help: Opens the Help Guide for the PolaX Desktop Application.

About: Opens the About Window which contains details about the application itself.

What's New: Opens a window to show what new features were implemented between versions.

Figure 6.14 Help Tab

6.6.3. Settings

The Settings Menu opens by pressing the gear icon in the top right of the application. Any changes made are not final until the Confirm button is pressed. Otherwise, if the gear icon or the Cancel Button is pressed, all changes are discarded.

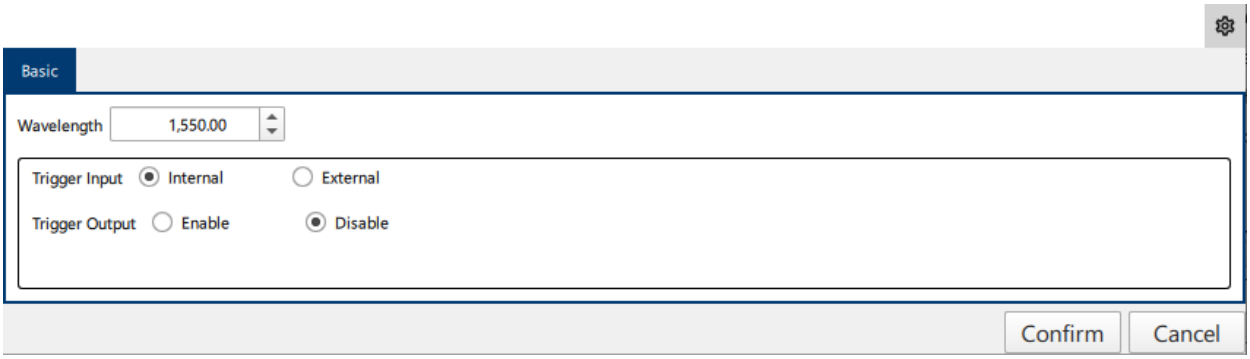


Figure 6.15 Settings

6.6.4. Basic Settings

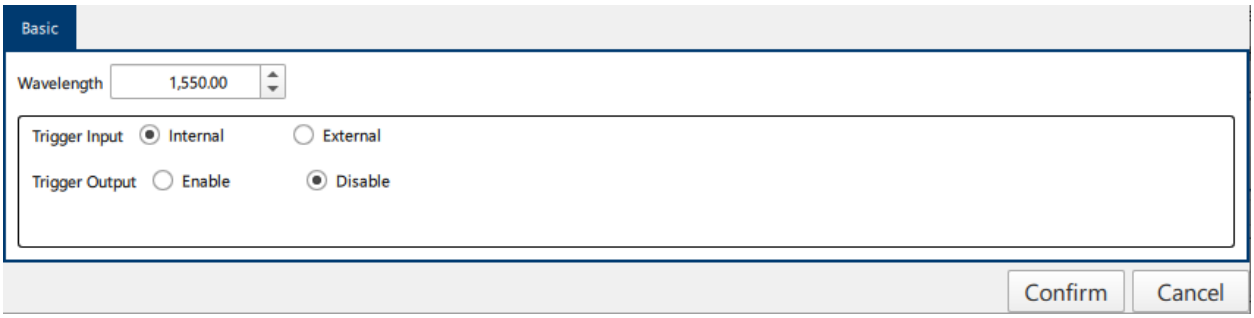


Figure 6.16 Basic Settings

The Basic Tab has three functions: Wavelength, Trigger Input, and Trigger Output.

Wavelength: Changes the active wavelength of the device.

Trigger Input:

Internal: Sets the device to use its internal timers as a trigger source.

External: Sets the device to use the signal from the TTL input port as the trigger source.

Trigger Output:

Enable: An active-high pulse will be emitted from the TTL output port when the polarization state changes.

Disable: An active-high pulse will not be emitted from the TTL output port when the polarization state changes.

Pulse Width: Sets the width of the trigger output pulse. The unit of the pulse width is in microseconds

7. Remote Control Configuration

7.1. USB Configuration and Programming Guide

Custom PC applications that wish to communicate with the MPX-2010 through a USB connection will need to use a language appropriate library for USB communication. For C and C++ application, developers will need to use libusb. For Python, one will need to use PyUSB (with a libusb backend).

SCPI commands and queries will be sent over endpoint 0x01 and query responses will be received over endpoint 0x81.

7.2. TCP/IP Configuration and Programming Guide

Custom PC applications that wish to communicate with the MPX-2010 through a TCP/IP connection can do using the language appropriate libraries for socket communication. When creating client sockets to connect to the MPX-2010, one must use port 5025 for commands and queries.

8. Standard Commands for Programmable Instruments (SCPI) Interface

8.1. Overview

	Subsystem/Register	Commonly Used Commands/Bits
Commands	IEEE Commands	• *IDN? • *RST • *CLS • *SAV • *RCL
	:SYSTEM	• :SYSTem:ERRor? • :SYSTem:VERSion • :SYSTem:COMMunicate:LAN:ADDRess
	:STATUS	• :STATus:OPERation[:EVENt]? • :STATus:OPERation:CONDition? • :STATus:QUEStionable[:EVENt]? • :STATus:QUEStionable:CONDition?
	:CONFigure	• :CONFigure:WAVElength
	:OUTPut	• :OUTPut:SOP • :OUTPut:SWEep:PATTern:SHAPe • :OUTPut:SWEep:PATTern:USRData • :OUTPut:SWEep:PATTern:DATA? • :OUTPut:SWEep:DWELl[:VALue]? • :OUTPut:SWEep:DWELl:UNIT • :OUTPut:SWEep:COUNt • :OUTPut:TRIGger[:STATe] • :OUTPut:TRIGger:PWIDth
	Trigger	• :TRIGger:SOURce • :TRIGger:CHANnel
Status Structure	Status Byte	• Error/Event Queue Summary • Questionable Status Register Summary • Operation Status Register Summary • Standard Event Status
	Standard Event Status Register	• Operation Complete • Query Error • Device Dependent Error • Execution Error • Command Error • Power On
	OPERation Status Register	• Sweeping
	QUEStionable Status Register	• Command Warning

	Subsystem/Register	Commonly Used Commands/Bits
Error/Event	Command Errors	• Invalid Character • Invalid Separator • Data type error • Parameter not allowed • Missing parameter
	Execution Errors	• Parameter error • Settings conflict • Data out of range
	Query	• Query error • Query INTERRUPTED • Query UNTERMINATED • Query DEADLOCKED
	Device-Specific Errors	• Queue overflow • Communication error • Input buffer overrun
	Events	• Power on • User request • Request control • Operation complete

8.2. Commands Reference

8.2.1. IEEE 488.2 Commands

Because the SCPI standard is based on the IEEE 488.2 standard, it inherits certain required commands defined in the IEEE 488.2 standard.

8.2.1.1. *IDN?

```
*IDN?
```

Retrieves the device's identification information in a string of the following format:

```
LUNA,MPX-2010,<Serial Number>,<Firmware Version>
```

8.2.1.2. *RST

```
*RST
```

Resets the device to factory default settings.

8.2.1.3. *CLS

```
*CLS
```

Clears the device's status registers.

8.2.1.4. *STB?

```
*STB
```

Reads the value of the status byte register.

8.2.1.5. *SRE

```
*SRE
```

Sets the service request enable mask of the status byte register.

Note: Due to the device connection design, this command has no effect on the device's operation.

8.2.1.6. *SRE?

```
*SRE?
```

Queries the service request enable mask of the status byte register.

8.2.1.7. *ESR?

```
*ESR?
```

Queries the event values of the Standard Event Status Register.

8.2.1.8. *ESE

```
*ESE
```

Sets the enable mask of the Standard Event Status Register.

8.2.1.9. *ESE?

```
*ESE?
```

Sets the enable mask of the Standard Event Status Register.

8.2.1.10. *WAI

```
*WAI
```

The Wait-to-Continue has no effect on this device.

8.2.1.11. *OPC

```
*OPC
```

The Operation Complete Command will set bit 0 of the Standard Event Status register once all device operations are finished and the device returns to idle.

8.2.1.12. *OPC?

```
*OPC?
```

The Operation Complete Query will result in the integer 1 being sent to the host once all device operations are finished and the device returns to idle.

8.2.1.13. *TST?

```
*TST?
```

The device self-test command. In this implementation of the device, this command has no effect on the device.

8.2.1.14. *SAV

```
*SAV <integer>
```

Saves the current device settings to an area of non-volatile memory specified by the integer parameter.

Note: The only acceptable value of the parameter is the integer 1.

8.2.1.15. *RCL

```
*RCL <integer>
```

Loads the saved device settings from non-volatile memory

Note: The only acceptable value of the parameter is the integer 1.

8.2.1.16. *TRG

```
*TRG
```

Sends a software trigger to the device.

8.2.2. :SYSTem Subsystem

The commands under SYSTem subsystem pertain to settings that relate to the instrument itself as opposed to any sort of instrument behavior.

8.2.2.1. :SYSTem:COMMunicate:LAN:ADDRess

```
:SYSTEM:COMMunicate:LAN:ADDRess <string>

:SYSTEM:COMMunicate:LAN:ADDRess? [CURRent|STATic]
```

Sets the static IP address used by the device when used as a command. The IP address is to be provided in the parameter to this command as a quote string of the address in dot-decimal notation.

When used as a query, one can provide an optional choice parameter to get either the current IP address using CURRent or the static IP address using STATic. Omitting the parameter is equivalent to providing CURRent as the parameter.

- **Command Example:**
  - `:SYST:COMM:LAN:ADDR "192.168.1.101"`
- **Query Response Example:**
  - `"192.168.1.101"`

8.2.2.2. :SYSTem:COMMunicate:LAN:SUBN

```
:SYSTem:COMMunicate:LAN:SUBNet <string>

:SYSTem:COMMunicate:LAN:SUBNet? [CURRent|STATic]
```

Sets the static subnet mask used by the device when used as a command. The subnet mask is to be provided in the parameter to this command as a quote string of the address in dot-decimal notation.

When used as a query, one can provide an optional choice parameter to get either the current subnet mask using CURRent or the static subnet mask using STATic. Omitting the parameter is equivalent to providing CURRent as the parameter.

- **Command Example:**
  - `:SYST:COMM:LAN:SUBN "255.255.255.0"`
- **Query Response Example:**
  - `"255.255.255.0"`

8.2.2.3. :SYStem:COMMunicate:LAN:GATEway

```
:SYStem:COMMunicate:LAN:GATEway <string>

:SYStem:COMMunicate:LAN:GATEway? [CURRent|STATic]
```

Sets the static network gateway address used by the device when used as a command. The address of the network gateway is to be provided in the parameter to this command as a quote string of the address in dot-decimal notation.

When used as a query, one can provide an optional choice parameter to get either the current gateway address using CURRent or the static gateway address using STATic. Omitting the parameter is equivalent to providing CURRent as the parameter.

- **Command Example:**
  - `:SYST:COMM:LAN:GATE "192.168.1.1"`
- **Query Response Example:**
  - `"192.168.1.1"`

8.2.2.4. :SYStem:COMMunicate:LAN:DHCP

```
:SYStem:COMMunicate:LAN:DHCP 0|1|ON|OFF

:SYStem:COMMunicate:LAN:DHCP?
```

Enable or disable the use of DHCP in acquiring a network address for the device. If enabled, the device will use a DHCP server to get an address. If disabled, the device will use the static IP address defined in :SYSTem:COMMunicate:LAN:ADDRess.

- **Command Example:**
  - `:SYS:COMM:LAN:DHCP ON`
- **Query Response Example:**
  - `0`

8.2.2.5. :SYStem:ERRor[:NEXT]?

```
:SYStem:ERRor[:NEXT]?
```

Get the next error in the device's error queue. When there are no errors, the response is 0, "No error".

- **Query Response Example:**
  - `-110, "Command header error"`

8.2.2.6. :SYStem:VERSion?

```
:SYStem:VERSion?
```

Gets the version of the SCPI standard to which this device adheres to. The response is always “1999.0”.

- **Query Response Example:**
  - `“1999.0”`

8.2.3. :STATus Subsystem

The commands under STATus subsystem are used for querying the status structures of the instrument.

8.2.3.1. :STATus:OPERation[:EVENt]?

```
:STATus:OPERation[:EVENt]?
```

Queries the event value of the Operation Status Register.

Note: Querying the event value will clear it from the register.

- **Query Response Example:**
  - `8`

8.2.3.2. :STATus:OPERation:CONDition?

```
:STATus:OPERation:CONDition?
```

Queries the condition value of the Operation Status Register.

- **Query Response Example:**
  - `8`

8.2.3.3. :STATus:OPERation:ENABle

```
:STATus:OPERation:ENABle <integer>

:STATus:OPERation:ENABle?
```

Sets the enable mask of the OPERation Status Register. When the bits in the enable mask value match that of the event value, bit 7 in the Status Byte register will be set for this register.

- **Command Example:**
  - `:STAT:OPER:ENAB 4`
- **Query Response Example:**
  - `8`

8.2.3.4. :STATus:QUEStionable[:EVENt]?

```
:STATus:QUEStionable[:EVENt]?
```

Queries the event value of the Questionable Status Register.

Note: Querying the event value will clear it from the register.

- **Query Response Example:**
  - 8

8.2.3.5. :STATus:QUEStionable:CONDition?

```
:STATus:QUEStionable:CONDition?
```

Queries the condition value of the Questionable Status Register.

- **Query Response Example:**
  - 8

8.2.3.6. :STATus:QUEStionable:ENABle

```
:STATus:QUEStionable:ENABle <integer>

:STATus:QUEStionable:ENABle?
```

Sets the enable mask of the Questionable Status Register. When the bits in the enable mask value match that of the event value, bit 3 in the Status Byte register will be set for this register.

- **Command Example:**
  - :STAT:QUES:ENAB 4
- **Query Response Example:**
  - 8

8.2.3.7. :STATus:PRESet

```
:STATus:PRESet
```

Clears all the status registers and resets the enable masks to pre-defined values. This command has no query form.

8.2.4. :CONFigure Subsystem

8.2.4.1. :CONFigure:WAVElength[:VALue]

```
:CONFigure:WAVElength[:VALue]

:CONFigure:GAIN[:VALue]?
```

Sets the working wavelength of the device. The units are in nanometers.

- **Minimum Value**
  - 1260
- **Maximum Value**
  - 1680
- **Default Value**
  - 1550
- **Command Example:**
  - :CONF:WAV 1550

- **Query Response Example:**
  - 1550

8.2.5. :OUTPut Subsystem

8.2.5.1. OUTPut:MODulation<n>[:STATe]

```
OUTPut:MODulation<n>[:STATe] ON|OFF|1|0

OUTPut:MODulation<n>[:STATe]?
```

Enable or disable the modulation activity of the polarization channel. The numeric suffix in the program header section, MODulation<n>, ranges from 1 to 4 and defaults to 1 when no suffix value is specified.

- **Default Value:**
  - OFF
- **Command Example:**
  - OUTP:MOD2 ON
- **Query Response Example**:
  - 1

8.2.5.2. OUTPut:MODulation<n>:FREQuency

```
OUTPut:MODulation<n>:FREQuency <float>

OUTPut:MODulation<n>:FREQuency?
```

Sets the frequency of the polarization channel modulation. The numeric suffix in the program header section, MODulation<n>, ranges from 1 to 4 and defaults to 1 when no suffix value is specified. The default unit used for frequency is hertz. The values returned by query responses are in units of hertz.

- **Minimum Value:**
  - 0
- **Maximum Value:**
  - 2000
- **Default Value:**
  - 100
- **Command Example:**
  - OUTP:MOD2:FREQ 500
- **Query Response Example:**
  - 1000

8.2.5.3. OUTPut:MODulation<n>:AMPlitude

```
OUTPut:MODulation<n>:AMPlitude <float>

OUTPut:MODulation<n>:AMPlitude?
```

Sets the rotational amplitude of the polarization channel modulation. The value returned from the query version of this command varies depending on the default unit of rotation defined by UNIT:ROTAtion. The numeric suffix in the program header section, MODulation<n>, ranges from 1 to 4 and defaults

to 1 when no suffix value is specified. Acceptable units of rotation are `RADians` and `PI`. The default unit used for amplitude is `RADian`.

When querying the rotational amplitude, the value is returned in the unit of time specified in `UNIT:ROTAtion`.

- **Minimum Value:**
  - `0` (regardless of default unit)
- **Maximum Value:**
  - `1.5` (when the default unit is PI)
  - `4.712390` (when the default unit is radians)
- **Default Value:**
  - `0.318310` (when the default unit is PI)
  - `1` (when the default unit is radians)
- **Command Example:**
  - `OUTP:MOD2:AMP 0.8PI`
  - `OUTP:MOD2:AMP 1.1RAD`
- **Query Response Example**:
  - `0.8`

8.2.5.4. OUTPut:MODulation<n>:OFFSet

```
OUTPut:MODulation<n>:OFFSet <float>

OUTPut:MODulation<n>:OFFSet?
```

Sets the rotational offset of the polarization channel modulation. The maximum limit of the offset value is defined as 3 PI minus the value of the amplitude in PI for the same channel. The minimum limit of the offset value is defined as 0 plus the value of the amplitude in PI for the same channel. The value returned from the query version of this command varies depending on the default unit of rotation defined by `UNIT:ROTAtion`. The numeric suffix in the program header section, `MODulation<n>`, ranges from 1 to 4 and defaults to 1 when no suffix value is specified. Acceptable units of rotation are `RADians` and `PI`. The default unit used for offset is `RADian`.

When querying the rotational offset, the value is returned in the unit of time specified in `UNIT:ROTAtion`.

- **Minimum Value:**
  - `0 + <value of OUTPut:MODulation<n>:AMPlitude in PI>`
- **Maximum Value:**
  - `3.0PI – <value of OUTPut:MODulation<n>:AMPlitude in PI>`
- **Default Value:**
  - `1` (when the default unit is radians)
  - `0.318309` (when the default unit is PI)
- **Command Example:**
  - `OUTP:MOD2:OFFS 0.8PI`

- OUTP:MOD2:OFFS 1.1RAD

- **Query Response Example**:
  - 100.5

8.2.5.5. OUTPut:MODulation<n>:WForm

```
OUTPut:MODulation<n>:WForm TRIangle|SINe|SQUare

OUTPut:MODulation<n>:WForm?
```

Sets the waveform of the modulation.

- **Default Value:**
  - TRIangle
- **Command Example:**
  - OUTP:MOD2:WF SIN
- **Query Response Example**:
  - SQUare

8.2.5.6. OUTPut:SCRAmble[:STATe]

```
OUTPut:SCRAmble[:STATe] 0|1|ON|OFF

OUTPut:SCRAmble[:STATe]?
```

Enable or disable polarization scrambling.

- **Default Value**
  - OFF
- **Command Example**
  - OUTP:SCRAmble ON
- **Query Response Example**
  - 0

8.2.5.7. OUTPut:SCRAmble:PATTern

```
OUTPut:SCRAmble:PATTern RANDom|RAYLeigh|TORNado|TRIangle

OUTPut:SCRAmble:PATTern?
```

Sets the scrambling pattern.

- **Default Value**
  - TRIangle
- **Command Example**
  - OUTP:SCRA:PATT TORN
- **Query Response Example**
  - TORNado

8.2.5.8. OUTPut:SCRAmble:RANDom:FREQuency

```
OUTPut:SCRAmble:RANDom:FREQuency <float>

OUTPut:SCRAmble:RANDom:FREQuency?
```

Sets the frequency of the Random scrambling pattern.

- **Minimum Value:**
  - 0
- **Maximum Value:**
  - 40000
- **Default Value:**
  - 1000
- **Command Example:**
  - OUTP:SCRA:RAND:FREQ 2000.5
- **Query Response Example**:
  - 2000.5

8.2.5.9. OUTPut:SCRAmble:RAYLeigh:FREQuency

```
OUTPut:SCRAmble:RAYLeigh:FREQuency <float>

OUTPut:SCRAmble:RAYLeigh:FREQuency?
```

Sets the frequency of the Rayleigh scrambling pattern.

- **Minimum Value:**
  - 0
- **Maximum Value:**
  - 4000
- **Default Value:**
  - 1000
- **Command Example:**
  - OUTP:SCRA:RAYL:FREQ 2000.5
- **Query Response Example**:
  - 2000.5

8.2.5.10. OUTPut:SCRAmble:TORNado:FREQuency

```
OUTPut:SCRAmble:TORNado:FREQuency <float>

OUTPut:SCRAmble:TORNado:FREQuency?
```

Sets the frequency of the Tornado scrambling pattern.

- **Minimum Value:**
  - 0
- **Maximum Value:**
  - 4000

- **Default Value:**
  - 1000
- **Command Example:**
  - OUTP:SCRA:TORN:FREQ 2000.5
- **Query Response Example**:
  - 2000.5

8.2.5.11. OUTPut:SCRAmble:TORNado:AXIS

```
OUTPut:SCRAmble:TORNado:AXIS MOVing:FIXed

OUTPut:SCRAmble:TORNado:AXIS?
```

Sets whether the axis of rotation of the Tornado scrambling pattern is moving or is fixed.

- **Default Value**
  - MOVing
- **Command Example**
  - OUTP:SCRA:TORN:AXIS FIXed
- **Query Response Example**
  - FIXed

8.2.5.12. OUTPut:SCRAmble:TRIangle:FREQuency

```
OUTPut:SCRAmble:TRIangle:FREQuency <float>

OUTPut:SCRAmble:TRIangle:FREQuency?
```

Sets the frequency of the Triangle scrambling pattern.

- **Minimum Value:**
  - 0
- **Maximum Value:**
  - 4000
- **Default Value:**
  - 1000
- **Command Example:**
  - OUTP:SCRA:TRI:FREQ 2000.5
- **Query Response Example**:
  - 2000.5

8.2.5.13. OUTPut:ROTAtion<n>

```
OUTPut:ROTAtion<n> <float>

OUTPut:ROTAtion<n>?
```

Sets the manual rotation position of the polarization channel. The numeric suffix in the program header section, ROTAtion<n>, ranges from 1 to 4 and defaults to 1 when no suffix value is specified. Values can be given in units of radians or PI.

8.2.5.14. OUTPut:TRIGger:PWIDth

```
OUTPut:TRIGger:PWIDth <float>

OUTPut:TRIGger:PWIDth?
```

Sets the pulse width of the output trigger signal. The pulse width is measured in microseconds.

- **Minimum Value**
  - 0.1
- **Maximum Value**
  - 5
- **Default Value**
  - 2
- **Command Example:**
  - OUTP:TRIG:PWID 1
- **Query Response Example**:
  - 2

8.2.6. Trigger Subsystem

8.2.6.1. TRIGger:SOURce

```
:TRIGger:SOURce INTernal|EXTernal|BUS|HOLD

:TRIGger:SOURce?
```

Selects the trigger source used in the sweep for determining the moment at which the polarization state changes.

The INTernal source is an internal timer that runs at a constant rate. Once the timer period duration has passed, the polarization state will change. The rate at which the timer runs can be adjusted using the OUTPut:SWEep:DWELl[:VALue] command.

The EXTernal source is the signal coming from the TTL trigger input port. Whenever there is a rising edge on the trigger input port, the polarization state will change. However, the maximum frequency of polarization state changes is 5 KHz. If the trigger input signal is higher than 5 KHz, the device will ignore all subsequent rising edges on the trigger input signal until the polarization state change is complete.

The BUS source is the occurrence of the *TRG command. Whenever the *TRG command is received the polarization state will change.

The HOLD source is only useful for pausing a sweep as it is defined as the absence of any trigger source.

- **Command Example:**
  - :TRIG:SOUR EXT
- **Query Response Example**:

- INTernal

8.2.6.2. TRIGger:CHANnel

```
:TRIGger:CHANnel CH1|CH2|CH3|CH4

:TRIGger:CHANnel?
```

Sets which polarization channel to monitor when emitting trigger pulses during modulation. When a channel modulation reaches either a minimum or maximum, the device will emit a pulse on the trigger output port.

- **Default Value**
  - CH1
- **Command Example**
  - TRIG:CHAN CH2
- **Query Response Example**
  - CH3

8.2.7. Unit Subsystem

8.2.7.1. UNIT:ROTAtion

```
:UNIT:ROTAtion RADian|PI

:UNIT:ROTAtion
```

Sets the default unit of rotation used by channel modulation and manual rotation commands. 1 PI is approximately equal to 3.1415 radians.

- **Default Value**
  - RADian
- **Command Example**
  - UNIT:ROTA PI
- **Query Response Example**
  - RADian

8.3. Status Structures

8.3.1. Status Byte

Bit #	Name
0	Unused
1	Unused
2	Error/Event Queue Summary
3	Questionable Status Register Summary

Bit #	Name
4	Output Queue Summary
5	Standard Event Status
6	Master Summary Status
7	Operation Status Register Summary

8.3.2. Standard Event Status Register

Bit #	Name
0	Operation Complete
1	Request Control
2	Query Error
3	Device Dependent Error
4	Execution Error
5	Command Error
6	User Request
7	Power On

8.3.3. OPERation Status Register

Bit #	Name
0	Unused
1	Unused
2	Unused
3	Unused
4	Unused
5	Unused
6	Unused
7	Unused

Bit #	Name
8	Unused
9	Unused
10	Unused
11	Unused
12	Unused
13	Unused
14	Unused

8.3.4. QUEStionable Status Register

Bit #	Name
0	Unused
1	Unused
2	Unused
3	Unused
4	Unused
5	Unused
6	Unused
7	Unused
8	Unused
9	Unused
10	Unused
11	Unused
12	Unused
13	Unused
14	Command Warning

8.4. Event/Error Codes

8.4.1. Command Errors

Error Code	Error String
-101	Invalid Character
-103	Invalid Separator
-104	Data type error
-108	Parameter not allowed
-109	Missing parameter
-110	Command header error
-113	Undefined header
-120	Numeric data error
-130	Suffix error
-140	Character data error
-150	String data error
-151	Invalid string data
-170	Expression error

8.4.2. Execution Errors

Error Code	Error String
-200	Execution error
-210	Trigger error
-220	Parameter error
-221	Settings conflict
-222	Data out of range
-223	Too much data
-224	Illegal parameter value
-230	Data corrupt or stale
-240	Hardware error
-250	Mass storage error
-260	Expression error
-280	Program error
-290	Memory use error

8.4.3. Query Errors

Error Code	Error String
-400	Query error
-410	Query INTERRUPTED
-420	Query UNTERMINATED
-430	Query DEADLOCKED
-440	Query UNTERMINATED after indefinite response

8.4.4. Device-Specific Errors

Error Code	Error String
-300	Device specific error
-310	System error
-313	Calibration memory lost
-314	Save/recall memory lost
-315	Configuration memory lost
-320	Storage fault
-330	Self-test failed
-340	Calibration failed
-350	Queue overflow
-360	Communication error
-363	Input buffer overrun

8.4.5. Events

Event Code	Event String
-500	Power on
-600	User request
-700	Request control
-800	Operation complete

9. PolaX API – MPX-2010 Instrument

Part of the PolaX platform includes the PolaX Api as a C++ library. This library can be used to connect to any supported device of the PolaX family, such as the MPX-2010.

9.1. PolaXDevice

9.1.1. Connecting to an MPX-2010

Using the PolaX library, one can programmatically connect to a MPX-2010 instrument by calling the PolaXDevice constructor or using the static function openDevice. To connect to an MPX-2010 via USB, the first parameter must be the serial number of the instrument. To connect via Ethernet, the first parameter must be the IP address of the device. Finally, to connect to a MPX-2010, the second parameter must be the MPX-2010's product ID, which is stored in an enum for convenience.

```
PolaXDevice mpx = PolaXDevice::openDevice("DUMMY", PolaXDevice::MPX2010);

PolaXDevice mpx2 = PolaXDevice("192.168.1.150", PolaXDevice::MPX2010);
```

9.1.2. Sending Commands

Once connected, to send commands to the MPX-2010, utilize the sendCommand function. The sendCommand function takes in the desired SCPI command as a string as its parameter.

```
mpx.sendCommand(":OUTP:SCRA:PATT RAND\n");
```

9.1.3. Querying Commands

To send query commands, utilize the query function. The query function takes in two parameters, the SCPI query and the output variable.

```
std::string output;

mpx.query(":CONF:WAV?\n", &output);
```

9.1.4. Reading

The MPX-2010 does not support using the read function and does not return any value.

9.1.5. Closing the MPX-2010

When done with the MPX-2010 connection, call the close function to properly close out the connection.

```
mpx.close();
```