This document provides additional assistance with wiring your Extron IP Link Pro Control Processor to your device. Different components may require a different wiring scheme than those listed below.

For complete operating instructions, refer to the user's manual for the specific IP Link Pro Control Processor or the documentation supplied by the manufacturer of the controlled device.

For more information on using Global Scripter Modules, refer to the "Guide to Using Scripter Modules" document.

Device Specifications

Device Type: Other
Manufacturer: Shure
Firmware Version: N/A
Model(s): ULX-D, ULX-D1, ULX-D2, QLX-D, ULX-D6, ULX-D8, ULX-D4, ULX-D4D, ULX-D4Q

Tested on the Following Software and Firmware Versions

IP Link Pro Control Processor Firmware	Global Scripter Version
3.16.0000-b013	2.17.0.15

Version History

Module Version	Date	Notes
1_1_5_0	10/25/2022	Removed error message for BatteryBars status when microphone is turned off.
1_1_4_0	9/21/2022	Fixed GroupandChannel and MeterRate control commands.
1_1_2_0	9/17/2019	Added statuses for Antenna, AntennaRFLevel and AudioLevel.
1_1_1_0	4/23/2019	Added InterferenceDetection, EncryptionMismatchWarning, and ChannelName commands.
1_1_0_0	9/25/2018	Added MeterRate command.
1_0_5_0	6/4/2018	Fixed BatteryBars status when transmitter is off or using AA batteries.
1_0_4_0	4/4/2018	Initial Version

Module Notes

- Unidirectional variable must be set to 'True' if status is not required. Default value is 'False'.
  `Example: InterfaceName.Unidirectional = 'True'`

- connectionCounter variable must be set to the number of queries that will be sent to the device before displaying 'Disconnected' if no response is received. Default value is 15.
  `Example: InterfaceName.connectionCounter = 5`

Supported Class and Example

EthernetClass

```
InterfaceName = ModuleName.EthernetClass('192.168.254.254', 2202, Model='ULX-D')
```

Control Commands

Format with Qualifier:

```
InterfaceName.Set(Command, Value, {'Qualifier Key': 'Qualifier Value'})
```

Format with Qualifier:

```
InterfaceName.Set(Command, Value)
```

Command **AudioMute** [1]	Value 'On'	Value 'Off'	Value 'Toggle'
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'	Qualifier Value 'All'	
`# AudioMute example` `InterfaceName.Set('AudioMute', 'On', {'Channel': '1'})`			
Command **Frequency**	Value 450 – 950		
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'	Qualifier Value 'All'	
`# Frequency example` `InterfaceName.Set('Frequency', 950, {'Channel': '1'})`			
Command **GroupandChannel**	Value None		
Qualifier Key **'Receiver Channel'**	Qualifier Value '1' – '4'		
Qualifier Key **'Group'**	Qualifier Value 1 – 99		
Qualifier Key **'Channel'**	Qualifier Value 1 – 99		
`# GroupandChannel example` `InterfaceName.Set('GroupandChannel', None, {'Receiver Channel': '1', 'Group': 99, 'Channel': 99})`			
Command **MeterRate**	Value 0 – 99999		
Qualifier Key **'Channel'** [2]	Qualifier Value '1' – '4'	Qualifier Value 'All'	
`# MeterRate example` `InterfaceName.Set('MeterRate', 99999, {'Channel': '1'})`			
Command **Volume**	Value 0 to 60 in steps of 1		
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'	Qualifier Value 'All'	
`# Volume example` `InterfaceName.Set('Volume', 60, {'Channel': '1'})`			

[1] Not available on the QLX-D model.

[2] Any value less than 100 milliseconds entered will be 0

Status Available

For all commands, call Update to receive the latest status. ConnectionStatus, AntennaRFLevelStatus, AntennaStatus, AudioLevelStatus, and MeterRate do not support the Update function. ConnectionStatus is triggered by the device providing a successful response to other Update function calls.

Format with Qualifier:

```
InterfaceName.Update(Command, {'Qualifier Key': 'Qualifier Value'})
Value = InterfaceName.ReadStatus(Command, {'Qualifier Key': 'Qualifier Value'})
InterfaceName.SubscribeStatus(Command, {'Qualifier Key': 'Qualifier Value'},
FeedbackHandler)
FeedbackHandler will be called only when the specified qualifier gets a new status.
```

Format without Qualifier:

```
InterfaceName.Update(Command)
Value = InterfaceName.ReadStatus(Command)
InterfaceName.SubscribeStatus(Command, None, FeedbackHandler)
FeedbackHandler will be called when any qualifier gets a new status.
```

Command **AntennaRFLevelStatus**	Value -128 – -13		
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'		

`# AntennaRFLevelStatus example`
`Value = InterfaceName.ReadStatus('AntennaRFLevelStatus', {'Channel': '1'})`
`InterfaceName.SubscribeStatus('AntennaRFLevelStatus', None, FeedbackHandler)`			

Command **AntennaStatus**	Value 'A on, B off' 'None'	Value 'A off, B on'	Value 'A off, B off'
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'		

`# AntennaStatus example`
`Value = InterfaceName.ReadStatus('AntennaStatus', {'Channel': '1'})`
`InterfaceName.SubscribeStatus('AntennaStatus', None, FeedbackHandler)`			

Command **AudioLevelStatus**	Value 0 – 50		
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'		

`# AudioLevelStatus example`
`Value = InterfaceName.ReadStatus('AudioLevelStatus', {'Channel': '1'})`
`InterfaceName.SubscribeStatus('AudioLevelStatus', None, FeedbackHandler)`			

Command **AudioMute** [1]	Value 'On'	Value 'Off'	Value 'Toggle'
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'	Qualifier Value 'All'	

`# AudioMute example`
`InterfaceName.Update('AudioMute', {'Channel': '1'})`
`Value = InterfaceName.ReadStatus('AudioMute', {'Channel': '1'})`
`InterfaceName.SubscribeStatus('AudioMute', None, FeedbackHandler)`			

Command **BatteryBars**	Value 0 – 5		
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'		

`# BatteryBars example`
`InterfaceName.Update('BatteryBars', {'Channel': '1'})`
`Value = InterfaceName.ReadStatus('BatteryBars', {'Channel': '1'})`			

```
InterfaceName.SubscribeStatus('BatteryBars', None, FeedbackHandler)
```

Command	Value
BatteryChargeStatus	0 – 100
Qualifier Key	**Qualifier Value**
'Channel'	'1' – '4'

```
# BatteryChargeStatus example
InterfaceName.Update('BatteryChargeStatus', {'Channel': '1'})
Value = InterfaceName.ReadStatus('BatteryChargeStatus', {'Channel': '1'})
InterfaceName.SubscribeStatus('BatteryChargeStatus', None, FeedbackHandler)
```

Command	Value
BatteryRemainingTime [3]	'String'
Qualifier Key	**Qualifier Value**
'Channel'	'1' – '4'

```
# BatteryRemainingTime example
InterfaceName.Update('BatteryRemainingTime', {'Channel': '1'})
Value = InterfaceName.ReadStatus('BatteryRemainingTime', {'Channel': '1'})
InterfaceName.SubscribeStatus('BatteryRemainingTime', None, FeedbackHandler)
```

Command	Value
ChannelName	'String'
Qualifier Key	**Qualifier Value**
'Channel'	'1' – '4'

```
# ChannelName example
InterfaceName.Update('ChannelName', {'Channel': '1'})
Value = InterfaceName.ReadStatus('ChannelName', {'Channel': '1'})
InterfaceName.SubscribeStatus('ChannelName', None, FeedbackHandler)
```

Command	Value
ChannelStatus [4]	1 – 99
Qualifier Key	**Qualifier Value**
'Channel'	'1' – '4'

```
# ChannelStatus example
InterfaceName.Update('ChannelStatus', {'Channel': '1'})
Value = InterfaceName.ReadStatus('ChannelStatus', {'Channel': '1'})
InterfaceName.SubscribeStatus('ChannelStatus', None, FeedbackHandler)
```

Command	Value	Value
ConnectionStatus	'Connected'	'Disconnected'

```
# ConnectionStatus example
Value = InterfaceName.ReadStatus('ConnectionStatus')
InterfaceName.SubscribeStatus('ConnectionStatus', None, FeedbackHandler)
```

Command	Value	Value
EncryptionMismatchWarning	'On'	'Off'
Qualifier Key	**Qualifier Value**	
'Channel'	'1' – '4'	

```
# EncryptionMismatchWarning example
InterfaceName.Update('EncryptionMismatchWarning', {'Channel': '1'})
Value = InterfaceName.ReadStatus('EncryptionMismatchWarning', {'Channel': '1'})
InterfaceName.SubscribeStatus('EncryptionMismatchWarning', None, FeedbackHandler)
```

Command	Value
FirmwareVersion	'String'

```
# FirmwareVersion example
InterfaceName.Update('FirmwareVersion')
Value = InterfaceName.ReadStatus('FirmwareVersion')
InterfaceName.SubscribeStatus('FirmwareVersion', None, FeedbackHandler)
```

Command	Value
Frequency	450 – 950

Qualifier Key	Qualifier Value	Qualifier Value
'Channel'	'1' – '4'	'All'

```
# Frequency example
InterfaceName.Update('Frequency', {'Channel': '1'})
Value = InterfaceName.ReadStatus('Frequency', {'Channel': '1'})
InterfaceName.SubscribeStatus('Frequency', None, FeedbackHandler)
```

Command	Value
GroupStatus [4]	1 – 99
Qualifier Key	**Qualifier Value**
'Channel'	'1' – '4'

```
# GroupStatus example
InterfaceName.Update('GroupStatus', {'Channel': '1'})
Value = InterfaceName.ReadStatus('GroupStatus', {'Channel': '1'})
InterfaceName.SubscribeStatus('GroupStatus', None, FeedbackHandler)
```

Command	Value	Value
InterferenceDetection	'None'	'Critical'
Qualifier Key	**Qualifier Value**	
'Channel'	'1' – '4'	

```
# InterferenceDetection example
InterfaceName.Update('InterferenceDetection', {'Channel': '1'})
Value = InterfaceName.ReadStatus('InterferenceDetection', {'Channel': '1'})
InterfaceName.SubscribeStatus('InterferenceDetection', None, FeedbackHandler)
```

Command	Value
MeterRate [5]	0 – 99999
Qualifier Key	**Qualifier Value**	**Qualifier Value**
'Channel'	'1' – '4'	'All'

```
# MeterRate example
Value = InterfaceName.ReadStatus('MeterRate', {'Channel': '1'})
InterfaceName.SubscribeStatus('MeterRate', None, FeedbackHandler)
```

Command	Value	Value	Value
TransmitterMuteButtonStatus [2]	'Pressed'	'Released'	'Unknown'
Qualifier Key	**Qualifier Value**		
'Channel'	'1' – '4'		

```
# TransmitterMuteButtonStatus example
InterfaceName.Update('TransmitterMuteButtonStatus', {'Channel': '1'})
Value = InterfaceName.ReadStatus('TransmitterMuteButtonStatus', {'Channel': '1'})
InterfaceName.SubscribeStatus('TransmitterMuteButtonStatus', None, FeedbackHandler)
```

Command	Value	Value	Value
TransmitterMuteStatus [2]	'On'	'Off'	'Unknown'
Qualifier Key	**Qualifier Value**		
'Channel'	'1' – '4'		

```
# TransmitterMuteStatus example
InterfaceName.Update('TransmitterMuteStatus', {'Channel': '1'})
Value = InterfaceName.ReadStatus('TransmitterMuteStatus', {'Channel': '1'})
InterfaceName.SubscribeStatus('TransmitterMuteStatus', None, FeedbackHandler)
```

Command	Value	Value	Value
TransmitterPowerSourceStatus	'Battery'	'External'	'Unknown'
Qualifier Key	**Qualifier Value**		
'Channel'	'1' – '4'		

```
# TransmitterPowerSourceStatus example
InterfaceName.Update('TransmitterPowerSourceStatus', {'Channel': '1'})
Value = InterfaceName.ReadStatus('TransmitterPowerSourceStatus', {'Channel': '1'})
InterfaceName.SubscribeStatus('TransmitterPowerSourceStatus', None, FeedbackHandler)
```

Command	Value	Value	Value

TransmitterRFPower	'Low' 'Unknown'	'Normal'	'High'
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'		

```
# TransmitterRFPower example
InterfaceName.Update('TransmitterRFPower', {'Channel': '1'})
Value = InterfaceName.ReadStatus('TransmitterRFPower', {'Channel': '1'})
InterfaceName.SubscribeStatus('TransmitterRFPower', None, FeedbackHandler)
```

Command **TransmitterType**	Value 'ULXD 1' 'ULXD 8' 'Unknown'	Value 'ULXD 2' 'QLXD 1'	Value 'ULXD 6' 'QLXD 2'
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'		

```
# TransmitterType example
InterfaceName.Update('TransmitterType', {'Channel': '1'})
Value = InterfaceName.ReadStatus('TransmitterType', {'Channel': '1'})
InterfaceName.SubscribeStatus('TransmitterType', None, FeedbackHandler)
```

Command **Volume**	Value 0 to 60 in steps of 1	
Qualifier Key **'Channel'**	Qualifier Value '1' – '4'	Qualifier Value 'All'

```
# Volume example
InterfaceName.Update('Volume', {'Channel': '1'})
Value = InterfaceName.ReadStatus('Volume', {'Channel': '1'})
InterfaceName.SubscribeStatus('Volume', None, FeedbackHandler)
```

[1] Not available on the QLX-D model.

[2] Only available on ULX-D6 and ULX-D8

[3] The 'String' is in this format, HH:MM (Hours:Minutes). If the remaining time is unknown, then the 'String' will contain 'Unknown'.

[4] Calling Update method with either ChannelStatus or GroupStatus will provide statuses for both commands.

[5] This status only updates via unsolicited responses. If no changes are made, the status will not update.

Network communication

When configuring the Ethernet module, be sure device settings match those of the Global Scripter ethernet interface.

Port Type:	Ethernet (TCP)
Default Port:	2202
Logon Credentials Supported:	No
Multi-Connection Capabilities:	Undetermined
Port Changeability:	Yes

Ethernet Module Configuration Description

Please refer to user manual for settings and changes to the network communication

Notes for the Device

Appendix A. Set Commands

Audio Mute Off Channel 1	< SET 1 AUDIO_MUTE OFF >
Audio Mute Off Channel 4	< SET 4 AUDIO_MUTE OFF >
Audio Mute Off Channel All	< SET 0 AUDIO_MUTE OFF >
Audio Mute On Channel 1	< SET 1 AUDIO_MUTE ON >
Audio Mute On Channel 4	< SET 4 AUDIO_MUTE ON >
Audio Mute On Channel All	< SET 0 AUDIO_MUTE ON >
Audio Mute Toggle Channel 1	< SET 1 AUDIO_MUTE TOGGLE >
Audio Mute Toggle Channel 4	< SET 4 AUDIO_MUTE TOGGLE >
Audio Mute Toggle Channel All	< SET 0 AUDIO_MUTE TOGGLE >
Frequency 450 Channel 1	< SET 1 FREQUENCY 450000 >
Frequency 450 Channel 4	< SET 4 FREQUENCY 450000 >
Frequency 450 Channel All	< SET 0 FREQUENCY 450000 >
Frequency 950 Channel 1	< SET 1 FREQUENCY 950000 >
Frequency 950 Channel 4	< SET 4 FREQUENCY 950000 >
Frequency 950 Channel All	< SET 0 FREQUENCY 950000 >
Group and Channel None Receiver Channel 1 Group 1 Channel 1	< SET 1 GROUP_CHAN 01,01 >
Group and Channel None Receiver Channel 1 Group 1 Channel 99	< SET 1 GROUP_CHAN 01,99 >
Group and Channel None Receiver Channel 1 Group 99 Channel 1	< SET 1 GROUP_CHAN 99,01 >
Group and Channel None Receiver Channel 1 Group 99 Channel 99	< SET 1 GROUP_CHAN 99,99 >
Group and Channel None Receiver Channel 4 Group 1 Channel 1	< SET 4 GROUP_CHAN 01,01 >
Group and Channel None Receiver Channel 4 Group 1 Channel 99	< SET 4 GROUP_CHAN 01,99 >
Group and Channel None Receiver Channel 4 Group 99 Channel 1	< SET 4 GROUP_CHAN 99,01 >
Group and Channel None Receiver Channel 4 Group 99 Channel 99	< SET 4 GROUP_CHAN 99,99 >
Meter Rate 0 Channel 1	< SET 1 METER_RATE 00000 >
Meter Rate 0 Channel 4	< SET 4 METER_RATE 00000 >
Meter Rate 0 Channel All	< SET 0 METER_RATE 00000 >
Meter Rate 99999 Channel 1	< SET 1 METER_RATE 99999 >
Meter Rate 99999 Channel 4	< SET 4 METER_RATE 99999 >
Meter Rate 99999 Channel All	< SET 0 METER_RATE 99999 >
Volume 0 Channel 1	< SET 1 AUDIO_GAIN 000 >
Volume 0 Channel 4	< SET 4 AUDIO_GAIN 000 >
Volume 0 Channel All	< SET 0 AUDIO_GAIN 000 >
Volume 60 Channel 1	< SET 1 AUDIO_GAIN 060 >
Volume 60 Channel 4	< SET 4 AUDIO_GAIN 060 >
Volume 60 Channel All	< SET 0 AUDIO_GAIN 060 >

Appendix B. Update Commands

Audio Mute Channel 1	< GET 1 AUDIO_MUTE >
Audio Mute Channel 4	< GET 4 AUDIO_MUTE >
Battery Bars Channel 1	< GET 1 BATT_BARS >
Battery Bars Channel 4	< GET 4 BATT_BARS >
Battery Charge Status Channel 1	< GET 1 BATT_CHARGE >
Battery Charge Status Channel 4	< GET 4 BATT_CHARGE >
Battery Remaining Time Channel 1	< GET 1 BATT_RUN_TIME >
Battery Remaining Time Channel 4	< GET 4 BATT_RUN_TIME >
Channel Name Channel 1	< GET 1 CHAN_NAME >
Channel Name Channel 4	< GET 4 CHAN_NAME >
Encryption Mismatch Warning Channel 1	< GET 1 ENCRYPTION_WARNING >
Encryption Mismatch Warning Channel 4	< GET 4 ENCRYPTION_WARNING >
Firmware Version	< GET FW_VER >
Frequency Channel 1	< GET 1 FREQUENCY >
Frequency Channel 4	< GET 4 FREQUENCY >
Group Status Channel 1	< GET 1 GROUP_CHAN >
Group Status Channel 4	< GET 4 GROUP_CHAN >
Interference Detection Channel 1	< GET 1 RF_INT_DET >
Interference Detection Channel 4	< GET 4 RF_INT_DET >
Transmitter Mute Button Status Channel 1	< GET 1 TX_MUTE_BUTTON_STATUS >
Transmitter Mute Button Status Channel 4	< GET 4 TX_MUTE_BUTTON_STATUS >
Transmitter Mute Status Channel 1	< GET 1 TX_MUTE_STATUS >
Transmitter Mute Status Channel 4	< GET 4 TX_MUTE_STATUS >
Transmitter Power Source Status Channel 1	< GET 1 TX_POWER_SOURCE >
Transmitter Power Source Status Channel 4	< GET 4 TX_POWER_SOURCE >
Transmitter RF Power Channel 1	< GET 1 TX_RF_PWR >
Transmitter RF Power Channel 4	< GET 4 TX_RF_PWR >
Transmitter Type Channel 1	< GET 1 TX_TYPE >
Transmitter Type Channel 4	< GET 4 TX_TYPE >
Volume Channel 1	< GET 1 AUDIO_GAIN >
Volume Channel 4	< GET 4 AUDIO_GAIN >