Global Scripter Module Communication Sheet

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:	Switcher
Manufacturer:	Extron
Firmware Version:	SW2 HD 4K and SW4 HD 4K: 2.0.2, SW6 HD 4K and SW8 HD 4K: 1.01.0002
Model(s):	SW2 HD 4K, SW4 HD 4K, SW6 HD 4K, SW8 HD 4K

Tested on the Following Software and Firmware Versions

IP Link Pro Control Processor Firmware	Global Scripter Version
2.08.0002-b001	1.4.2

Version History

Module Version	Date	Notes
1_2_0_0	7/24/2018	Added SW4 HD 4K and SW8 HD 4K. Added Auto Switch Mode status. Added 'On with Sync' value to Video Mute. Renamed HDCP Authorization to HDCP Input Authorization. Renamed Signal Status to Input Signal Status and updated values to 'Active' and 'Inactive'. Updated Input value 'No Input' to '0'.
1_1_2_2	9/28/2017	Fixed status issue.
1_1_2_1	8/31/2017	Update status example.
1_1_2_0	8/1/2017	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 Classes and Examples

SerialClass

```
    InterfaceName = ModuleName.SerialClass(ProcessorName, 'COM1', Model='SW2 HD 4K')
```

SerialOverEthernetClass

```
    InterfaceName = ModuleName.SerialOverEthernetClass('192.168.254.254', 2001, Model='SW2 HD 4K')
```

Control Commands

Format with Qualifier:

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

Format without Qualifier:

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

Command	Value	Value
AudioMute	'On'	'Off'

```
# AudioMute example
InterfaceName.Set('AudioMute', 'On')
```

Command	Value	Value
ExecutiveMode	'On'	'Off'

```
# ExecutiveMode example
InterfaceName.Set('ExecutiveMode', 'On')
```

Command	Value	Value
HDCPInputAuthorization	'On'	'Off'
Qualifier Key	Qualifier Value	
'Input'	'1' – '8' [1]	

```
# HDCPInputAuthorization example
InterfaceName.Set('HDCPInputAuthorization', 'On', {'Input': '1'})
```

Command	Value
Input	'0' – '8' [1]

```
# Input example
InterfaceName.Set('Input', '1')
```

Command	Value	Value
IRSensor	'On'	'Off'

```
# IRSensor example
InterfaceName.Set('IRSensor', 'On')
```

Command	Value	Value	Value
VideoMute	'On'	'Off'	'On with Sync'

```
# VideoMute example
InterfaceName.Set('VideoMute', 'On')
```

[1] 'Input' range is proportional to the number of inputs on the device.

Status Available

For all commands except for AutoSwitchMode and InputSignalStatus, Update should be called only once since the command's status will be updated automatically as the device's status changes. ConnectionStatus does not support the Update function and 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	Value	Value
AudioMute	'On'	'Off'

```
# AudioMute examples
InterfaceName.Update('AudioMute')
Value = InterfaceName.ReadStatus('AudioMute')
InterfaceName.SubscribeStatus('AudioMute', None, FeedbackHandler)
```

Command	Value	Value
AutoSwitchMode	'On'	'Off'

```
# AutoSwitchMode examples
InterfaceName.Update('AutoSwitchMode')
Value = InterfaceName.ReadStatus('AutoSwitchMode')
InterfaceName.SubscribeStatus('AutoSwitchMode', None, FeedbackHandler)
```

Command	Value	Value
ConnectionStatus	'Connected'	'Disconnected'

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

Command	Value	Value
ExecutiveMode	'On'	'Off'

```
# ExecutiveMode examples
InterfaceName.Update('ExecutiveMode')
Value = InterfaceName.ReadStatus('ExecutiveMode')
InterfaceName.SubscribeStatus('ExecutiveMode', None, FeedbackHandler)
```

Command	Value	Value
HDCPInputAuthorization	'On'	'Off'
Qualifier Key	Qualifier Value	
'Input' [1]	'1' – '8'	

```
    # HDCPInputAuthorization examples
    InterfaceName.Update('HDCPInputAuthorization', {'Input': '1'})
    Value = InterfaceName.ReadStatus('HDCPInputAuthorization', {'Input': '1'})
    InterfaceName.SubscribeStatus('HDCPInputAuthorization', None, FeedbackHandler)
```

Command **Input** [1]	Value '0' – '8'		

```
    # Input examples
    InterfaceName.Update('Input')
    Value = InterfaceName.ReadStatus('Input')
    InterfaceName.SubscribeStatus('Input', None, FeedbackHandler)
```

Command **InputSignalStatus**	Value 'Active'	Value 'Not Active'	
Qualifier Key **'Input'** [1]	Qualifier Value '1' – '8'		

```
    # InputSignalStatus examples
    InterfaceName.Update('InputSignalStatus', {'Input': '1'})
    Value = InterfaceName.ReadStatus('InputSignalStatus', {'Input': '1'})
    InterfaceName.SubscribeStatus('InputSignalStatus', None, FeedbackHandler)
```

Command **IRSensor**	Value 'On'	Value 'Off'	

```
    # IRSensor examples
    InterfaceName.Update('IRSensor')
    Value = InterfaceName.ReadStatus('IRSensor')
    InterfaceName.SubscribeStatus('IRSensor', None, FeedbackHandler)
```

Command **VideoMute**	Value 'On'	Value 'Off'	Value 'On with Sync'

```
    # VideoMute examples
    InterfaceName.Update('VideoMute')
    Value = InterfaceName.ReadStatus('VideoMute')
    InterfaceName.SubscribeStatus('VideoMute', None, FeedbackHandler)
```

[1] 'Input' range is proportional to the number of inputs on the device.

Cable and Adapter Requirements

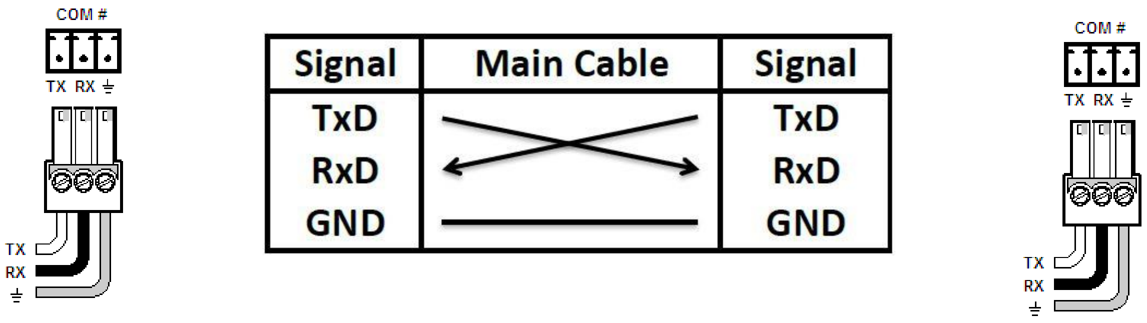
Captive Screw to Captive Screw RS-232 Serial Cable

Notes for the Device

Serial communication

Port Type:	RS-232	**Parity:**	None
Baud Rate:	9600	**Stop Bits:**	One
Data Bits:	8	**Flow Control:**	None

Pin Assignments Diagram



Appendix A. Set Commands

Audio Mute Off	0Z
Audio Mute On	1Z
Executive Mode Off	0X
Executive Mode On	1X
HDCP Input Authorization Off Input 1	wE1*0HDCP\x0D
HDCP Input Authorization On Input 1	wE1*1HDCP\x0D
HDCP Input Authorization Off Input 4	wE4*0HDCP\x0D
HDCP Input Authorization On Input 4	wE4*1HDCP\x0D
HDCP Input Authorization Off Input 2	wE2*0HDCP\x0D
HDCP Input Authorization On Input 2	wE2*1HDCP\x0D
HDCP Input Authorization Off Input 6	wE6*0HDCP\x0D
HDCP Input Authorization On Input 6	wE6*1HDCP\x0D
HDCP Input Authorization Off Input 8	wE8*0HDCP\x0D
HDCP Input Authorization On Input 8	wE8*1HDCP\x0D
IR Sensor Off	1*65#
IR Sensor On	0*65#
Input 0	0!
Input 4	4!
Input 2	2!
Input 6	6!
Input 8	8!
Video Mute Off	0B
Video Mute On	1B
Video Mute On with Sync	2B

Appendix B. Update Commands

Audio Mute	`Z`
Auto Switch Mode	`72#`
Executive Mode	`X`
HDCP Input Authorization Input 1	`wEHDCP\x0D`
HDCP Input Authorization Input 4	`wEHDCP\x0D`
HDCP Input Authorization Input 2	`wEHDCP\x0D`
HDCP Input Authorization Input 6	`wEHDCP\x0D`
HDCP Input Authorization Input 8	`wEHDCP\x0D`
IR Sensor	`65#`
Input	`I`
Input Signal Status Input 1	`\x1BLS\x0D`
Input Signal Status Input 2	`\x1BLS\x0D`
Input Signal Status Input 4	`\x1BLS\x0D`
Input Signal Status Input 6	`\x1BLS\x0D`
Video Mute	`B`