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: | Audio Processor |
| Manufacturer: | Yamaha |
| Firmware Version: | N/A |
| Model(s): | TF5, TF1, TF-Rack |

## Tested on the Following Software and Firmware Versions

| IP Link Pro Control Processor Firmware | Global Scripter Version |
|---|---|
| 2.06.0002-b002 | 1.4.2 |

## Version History

| Module Version | Date | Notes |
|---|---|---|
| 1_0_0_0 | 4/18/2018 | 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 Classes and Examples

**EthernetClass**

```
InterfaceName = ModuleName.EthernetClass('192.168.254.254', 49280, Model='TF5')
```

## Set Commands

Format with Qualifier:

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

Format without Qualifier:

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

| | | |
|---|---|---|
| Command<br>**InputLevel** | Value<br>0 to 1000 in steps of 1 | |
| Qualifier Key<br>**'Channel'** | Qualifier Value<br>'1' – '40' | |
| `# InputLevel example`<br>`InterfaceName.Set('InputLevel', 1000, {'Channel': '1'})` | | |
| Command<br>**InputMute** | Value<br>'On' | Value<br>'Off' |
| Qualifier Key<br>**'Channel'** | Qualifier Value<br>'1' – '40' | |
| `# InputMute example`<br>`InterfaceName.Set('InputMute', 'On', {'Channel': '1'})` | | |
| Command<br>**OutputLevel** | Value<br>0 to 1000 in steps of 1 | |
| Qualifier Key<br>**'Channel'** | Qualifier Value<br>'1' – '20' | |
| `# OutputLevel example`<br>`InterfaceName.Set('OutputLevel', 1000, {'Channel': '1'})` | | |
| Command<br>**OutputMute** | Value<br>'On' | Value<br>'Off' |
| Qualifier Key<br>**'Channel'** | Qualifier Value<br>'1' – '20' | |
| `# OutputMute example`<br>`InterfaceName.Set('OutputMute', 'On', {'Channel': '1'})` | | |
| Command<br>**Preset** | Value<br>'0' – '99' | |
| Qualifier Key<br>**'Scene'** | Qualifier Value<br>'A' | Qualifier Value<br>'B' |
| Qualifier Key<br>**'Action'** | Qualifier Value<br>'Recall' | Qualifier Value<br>'Store' |
| `# Preset example`<br>`InterfaceName.Set('Preset', '0', {'Scene': 'A', 'Action': 'Recall'})` | | |

## Status Available

For all commands, call Update to receive the latest status. 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<br>**ConnectionStatus** | Value<br>'Connected' | Value<br>'Disconnected' |

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

| | |
|---|---|
| Command<br>**Firmware** | Value<br>'String' |

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

| | |
|---|---|
| Command<br>**InputLevel** | Value<br>0 to 1000 in steps of 1 |
| Qualifier Key<br>**'Channel'** | Qualifier Value<br>'1' – '40' |

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

| | | |
|---|---|---|
| Command<br>**InputMute** | Value<br>'On' | Value<br>'Off' |
| Qualifier Key<br>**'Channel'** | Qualifier Value<br>'1' – '40' | |

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

| | |
|---|---|
| Command<br>**OutputLevel** | Value<br>0 to 1000 in steps of 1 |

| | | |
|---|---|---|
| **Qualifier Key**<br>**'Channel'** | Qualifier Value<br>'1' – '20' | |
| `# OutputLevel examples`<br>`InterfaceName.Update('OutputLevel', {'Channel': '1'})`<br>`Value = InterfaceName.ReadStatus('OutputLevel', {'Channel': '1'})`<br>`InterfaceName.SubscribeStatus('OutputLevel', None, FeedbackHandler)` | | |
| **Command**<br>**OutputMute** | Value<br>'On' | Value<br>'Off' |
| **Qualifier Key**<br>**'Channel'** | Qualifier Value<br>'1' – '20' | |
| `# OutputMute examples`<br>`InterfaceName.Update('OutputMute', {'Channel': '1'})`<br>`Value = InterfaceName.ReadStatus('OutputMute', {'Channel': '1'})`<br>`InterfaceName.SubscribeStatus('OutputMute', None, FeedbackHandler)` | | |

## Network communication

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

| | |
|---|---|
| **Port Type:** | Ethernet |
| **Default Port:** | 49280 |
| **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

| | |
|---|---|
| **Input Level 0 Channel 1** | `setn MIXER:Current/InCh/Fader/Level 0 0 0\x0A` |
| **Input Level 1000 Channel 1** | `setn MIXER:Current/InCh/Fader/Level 0 0 1000\x0A` |
| **Input Level 0 Channel 32** | `setn MIXER:Current/InCh/Fader/Level 31 0 0\x0A` |
| **Input Level 1000 Channel 32** | `setn MIXER:Current/InCh/Fader/Level 31 0 1000\x0A` |
| **Input Level 0 Channel 40** | `setn MIXER:Current/InCh/Fader/Level 39 0 0\x0A` |
| **Input Level 1000 Channel 40** | `setn MIXER:Current/InCh/Fader/Level 39 0 1000\x0A` |
| **Input Mute Off Channel 1** | `set MIXER:Current/InCh/Fader/On 0 0 0\x0A` |
| **Input Mute On Channel 1** | `set MIXER:Current/InCh/Fader/On 0 0 1\x0A` |
| **Input Mute Off Channel 32** | `set MIXER:Current/InCh/Fader/On 31 0 0\x0A` |
| **Input Mute On Channel 32** | `set MIXER:Current/InCh/Fader/On 31 0 1\x0A` |
| **Input Mute Off Channel 40** | `set MIXER:Current/InCh/Fader/On 39 0 0\x0A` |
| **Input Mute On Channel 40** | `set MIXER:Current/InCh/Fader/On 39 0 1\x0A` |
| **Output Level 0 Channel 1** | `setn MIXER:Current/Mix/Fader/Level 0 0 0\x0A` |
| **Output Level 1000 Channel 1** | `setn MIXER:Current/Mix/Fader/Level 0 0 1000\x0A` |
| **Output Level 0 Channel 20** | `setn MIXER:Current/Mix/Fader/Level 19 0 0\x0A` |
| **Output Level 1000 Channel 20** | `setn MIXER:Current/Mix/Fader/Level 19 0 1000\x0A` |
| **Output Mute Off Channel 1** | `set MIXER:Current/Mix/Fader/On 0 0 0\x0A` |
| **Output Mute On Channel 1** | `set MIXER:Current/Mix/Fader/On 0 0 1\x0A` |
| **Output Mute Off Channel 20** | `set MIXER:Current/Mix/Fader/On 19 0 0\x0A` |
| **Output Mute On Channel 20** | `set MIXER:Current/Mix/Fader/On 19 0 1\x0A` |
| **Preset 0 Scene A Action Recall** | `ssrecall_ex scene_a 0\x0A` |
| **Preset 99 Scene A Action Recall** | `ssrecall_ex scene_a 99\x0A` |
| **Preset 0 Scene A Action Store** | `ssupdate_ex scene_a 0\x0A` |
| **Preset 99 Scene A Action Store** | `ssupdate_ex scene_a 99\x0A` |
| **Preset 0 Scene B Action Recall** | `ssrecall_ex scene_b 0\x0A` |
| **Preset 99 Scene B Action Recall** | `ssrecall_ex scene_b 99\x0A` |
| **Preset 0 Scene B Action Store** | `ssupdate_ex scene_b 0\x0A` |
| **Preset 99 Scene B Action Store** | `ssupdate_ex scene_b 99\x0A` |

## Appendix B. Update Commands

| | |
|---|---|
| **Firmware** | devinfo version\x0A |
| **Input Level Channel 1** | getn MIXER:Current/InCh/Fader/Level 0 0\x0A |
| **Input Level Channel 32** | getn MIXER:Current/InCh/Fader/Level 31 0\x0A |
| **Input Level Channel 40** | getn MIXER:Current/InCh/Fader/Level 39 0\x0A |
| **Input Mute Channel 1** | get MIXER:Current/InCh/Fader/On 0 0\x0A |
| **Input Mute Channel 32** | get MIXER:Current/InCh/Fader/On 31 0\x0A |
| **Input Mute Channel 40** | get MIXER:Current/InCh/Fader/On 39 0\x0A |
| **Output Level Channel 1** | getn MIXER:Current/Mix/Fader/Level 0 0\x0A |
| **Output Level Channel 20** | getn MIXER:Current/Mix/Fader/Level 19 0\x0A |
| **Output Mute Channel 20** | get MIXER:Current/Mix/Fader/On 19 0\x0A |