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"](http://www.extron.com/download/files/gsmodules/HowToUseGSModules.pdf) document.

# Device Specifications

|  |  |
| --- | --- |
| Device Type: | Camera |
| Manufacturer: | Panasonic |
| Firmware Version: | N/A |
| Model(s): | AW-HE100, AW-HE120, AW-HE38H, AW-HE38HK, AW-HE38HW, AW-HE40HKE, AW-  HE40HKP, AW-HE40HWE, AW-HE40HWP, AW-HE40SKE, AW-HE40SKP, AW-HE40SWE,  AW-HE40SWP, AW-HE50, AW-HE50H, AW-HE50S, AW-HE60, AW-HE60SE, AWHE65HKMC, AW-HE65HWMC, AW-HE65SKMC, AW-HE65SWMC, AW-HE70HK, AW-  HE70HW, AW-HE70SK, AW-HE70SW, AW-UE70, AW-UE70KE, AW-UE70KP, AWUE70WE, AW-UE70WP, AW-HE40HWPJ9, AW-HE40SWPJ9, AW-HE40HWEJ9, AWHE40SWEJ9, AW-HE40HKPJ9, AW-HE40SKPJ9, AW-HE40HKEJ9, AW-HE40SKEJ9, AWUE150WP, AW-UE150KP, AW-UE150WE, AW-UE150KE |
|  |  |

# Tested on the Following Software and Firmware Versions

|  |  |  |
| --- | --- | --- |
| **IP Link Pro Control Processor Firmware** |  | **Global Scripter Version** |
| 3.04.0001-b002 | 2.3.1 |  |

# Version History

|  |  |  |
| --- | --- | --- |
| **Module Version** | **Date** | **Notes** |
| 1\_6\_1\_1 | 1/30/2020 | Fixed authentication. |
| 1\_6\_1\_0 | 4/22/2019 | Added Scene File Control command for Ethernet control, for all models starting with: AW-HE38, AW-HE40, AW-HE50, AW-HE60, AW-HE-65, AW-HE70 or AW-UE70. |
| 1\_5\_1\_0 | 2/13/2019 | 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='AW-HE100') |
| **SerialOverEthernetClass** |
| InterfaceName = ModuleName.SerialOverEthernetClass('192.168.254.254', 2001, Model='AW-HE100') |
| **HTTPClass** |
| InterfaceName = ModuleName.HTTPClass('192.168.254.254', 80, 'admin', 'password', Model='AW-  HE100') |

# Control Commands (Serial)

Format with Qualifier:

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

Format with Qualifier:

InterfaceName.Set(Command, Value)

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** **AutoFocus** | Value 'On' | Value 'Off' |  |
| # AutoFocus example  InterfaceName.Set('Au | toFocus', 'On') |  |  |
| **Command**  **ExtenderAFControl** | Value 'On' | Value 'Off' |  |
| # ExtenderAFControl e | xample |  |  |
| InterfaceName.Set('Ex | tenderAFControl', 'On') |  |  |
| **Command** **Focus** | Value 'Far' | Value 'Near' | Value 'Stop' |
| **Qualifier Key**  **'Focus Speed'** | Qualifier Value  1 to 49 in steps of 1 |  |  |
| # Focus example  InterfaceName.Set('Fo | cus', 'Far', {'Focus Speed | ': 49}) |  |
| **Command** **Iris** | Value  1 to 99 in steps of 1 |  |  |
| # Iris example  InterfaceName.Set('Ir | is', 99) |  |  |
| **Command** **IrisMode** | Value  'Manual' | Value 'Auto' |  |
| # IrisMode example  InterfaceName.Set('Ir | isMode', 'Manual') |  |  |
| **Command** **Pan** | Value 'Left' | Value 'Right' | Value 'Stop' |
| **Qualifier Key**  **'Pan Speed'** | Qualifier Value  1 to 49 in steps of 1 |  |  |
| # Pan example  InterfaceName.Set('Pa | n', 'Left', {'Pan Speed': | 49}) |  |
| **Command** **Power** | Value 'On' | Value 'Off' |  |
| # Power example  InterfaceName.Set('Po | wer', 'On') |  |  |
| **Command** **Preset** | Value  '1' – '100' |  |  |
| **Qualifier Key**  **'Type'** | Qualifier Value 'Save' | Qualifier Value 'Recall' |  |
| # Preset example  InterfaceName.Set('Preset', '1', {'Type': 'Save'}) | | |  |
| **Command** Value Value **Tally** 'On' 'Off' | | |  |
| # Tally example  InterfaceName.Set('Tally', 'On') | | |  |
| **Command** Value Value **TallyEnable** 'On' 'Off' | | |  |
| # TallyEnable example  InterfaceName.Set('TallyEnable', 'On') | | |  |
| **Command** **Tilt** | Value 'Up' | Value 'Down' | Value 'Stop' |
| **Qualifier Key**  **'Tilt Speed'** | Qualifier Value  1 to 49 in steps of 1 |  |  |
| # Tilt example  InterfaceName.Set('Ti | lt', 'Up', {'Tilt Speed': | 49}) |  |
| **Command** **Zoom** | Value 'Tele' | Value 'Wide' | Value 'Stop' |
| **Qualifier Key**  **'Zoom Speed'** | Qualifier Value  1 to 49 in steps of 1 |  |  |
| # Zoom example  InterfaceName.Set('Zoom', 'Tele', {'Zoom Speed': 49}) | | |  |

# Status Available (Serial)

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**  **ConnectionStatus** | Value  'Connected' | Value  'Disconnected' |  |
| Value = InterfaceName | .ReadStatus('ConnectionSta | tus') |  |
| InterfaceName.Subscri | beStatus('ConnectionStatus | ', None, FeedbackHandler) |  |
| **Command**  **ExtenderAFControl** | Value 'On' | Value 'Off' |  |
| # ExtenderAFControl e | xample |  |  |
| InterfaceName.Update( | 'ExtenderAFControl') |  |  |
| Value = InterfaceName | .ReadStatus('ExtenderAFCon | trol') |  |
| InterfaceName.Subscri | beStatus('ExtenderAFContro | l', None, FeedbackHandle | r) |
| **Command** **Iris** | Value  1 to 99 in steps of 1 |  |  |
| # Iris example  InterfaceName.Update( | 'Iris') |  |  |
| Value = InterfaceName | .ReadStatus('Iris') |  |  |
| InterfaceName.Subscri | beStatus('Iris', None, Fee | dbackHandler) |  |
| **Command** **IrisMode** | Value  'Manual' | Value 'Auto' |  |
| # IrisMode example  InterfaceName.Update( | 'IrisMode') |  |  |
| Value = InterfaceName | .ReadStatus('IrisMode') |  |  |
| InterfaceName.Subscri | beStatus('IrisMode', None, | FeedbackHandler) |  |
| **Command** **Power** | Value 'On' | Value 'Off' | Value  'Starting' |
| # Power example  InterfaceName.Update( | 'Power') |  |  |
| Value = InterfaceName | .ReadStatus('Power') |  |  |
| InterfaceName.Subscri | beStatus('Power', None, Fe | edbackHandler) |  |
| **Command**  **Tally** | Value 'On' | Value 'Off' |  |
| # Tally example  InterfaceName.Update('Tally')  Value = InterfaceName.ReadStatus('Tally')  InterfaceName.SubscribeStatus('Tally', None, FeedbackHandler) | | | |
| **Command** Value Value **TallyEnable** 'On' 'Off' | | | |
| # TallyEnable example  InterfaceName.Update('TallyEnable')  Value = InterfaceName.ReadStatus('TallyEnable')  InterfaceName.SubscribeStatus('TallyEnable', None, FeedbackHandler) | | | |

# Control Commands (Ethernet)

Format with Qualifier:

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

Format with Qualifier:

InterfaceName.Set(Command, Value)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Command** **AutoFocus** | Value 'Auto' | Value  'Manual' |  | |
| # AutoFocus example  InterfaceName.Set('Au | toFocus', 'Auto') |  |  | |
| **Command** **AutoIris** | Value 'On' | Value 'Off' |  | |
| # AutoIris example  InterfaceName.Set('Au | toIris', 'On') |  |  | |
| **Command** **ColorBar** | Value 'On' | Value 'Off' |  | |
| # ColorBar example  InterfaceName.Set('Co | lorBar', 'On') |  |  | |
| **Command**  **Detail** | Value  'On'  'High' | Value  'Off' | Value  'Low' | |
| # Detail example  InterfaceName.Set('De | tail', 'On') |  |  | |
| **Command** **Focus** | Value 'Far' | Value 'Near' | Value 'Stop' | |
| **Qualifier Key**  **'Speed'** | Qualifier Value  1 to 49 in steps of 1 |  |  | |
| # Focus example  InterfaceName.Set('Fo | cus', 'Far', {'Speed': 49} | ) |  | |
| **Command**  **Installation** | Value  'Desktop' | Value  'Hanging' |  | |
| # Installation example  InterfaceName.Set('In | stallation', 'Desktop') |  |  | |
| **Command**  **IrisPosition** | Value  0 to 20 in steps of 1 |  |  | |
| # IrisPosition example  InterfaceName.Set('Ir | isPosition', 20) |  |  | |
| **Command** **PanTilt** | Value 'Left' | Value 'Right' | Value 'Up' | |
|  | 'Down' | 'Stop' |  | |
| **Qualifier Key**  **'Speed'** | Qualifier Value  1 to 49 in steps of 1 |  |  | |
| # PanTilt example  InterfaceName.Set('Pa | nTilt', 'Left', {'Speed': | 49}) |  | |
| **Command** **Power** | Value 'On' | Value  'Standby' |  | |
| # Power example  InterfaceName.Set('Po | wer', 'On') |  |  | |
| **Command** **Preset** | Value  '1' – '100' |  |  | |
| **Qualifier Key** | Qualifier Value | Qualifier Value |  | |
| **'Type'** | 'Save' | 'Recall' | |  |
| # Preset example  InterfaceName.Set('Pr | eset', '1', {'Type': 'Save | '}) | |  |
| **Command**  **ResetPanTiltPosition** | Value 'None' |  | |  |
| # ResetPanTiltPositio | n example |  | |  |
| InterfaceName.Set('Re | setPanTiltPosition', None) |  | |  |
| **Command** **ResetZoom** | Value 'None' |  | |  |
| # ResetZoom example  InterfaceName.Set('Re | setZoom', None) |  | |  |
| **Command**  **SceneFileControl** | Value  'Manual 1'  'Full Auto' | Value  'Manual 2' | | Value  'Manual 3' |
| # SceneFileControl ex | ample |  | |  |
| InterfaceName.Set('Sc | eneFileControl', 'Manual 1 | ') | |  |
| **Command** **Tally** | Value 'On' | Value 'Off' | |  |
| # Tally example  InterfaceName.Set('Ta | lly', 'On') |  | |  |
| **Command** **TallyInput** | Value  'Enable' | Value  'Disable' | |  |
| # TallyInput example  InterfaceName.Set('Ta | llyInput', 'Enable') |  | |  |
| **Command** **Zoom** | Value 'Wide' | Value 'Tele' | | Value 'Stop' |
| **Qualifier Key**  **'Speed'** | Qualifier Value  1 to 49 in steps of 1 |  | |  |
| # Zoom example  InterfaceName.Set('Zoom', 'Wide', {'Speed': 49}) | | | |  |

# Status Available (Ethernet)

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.

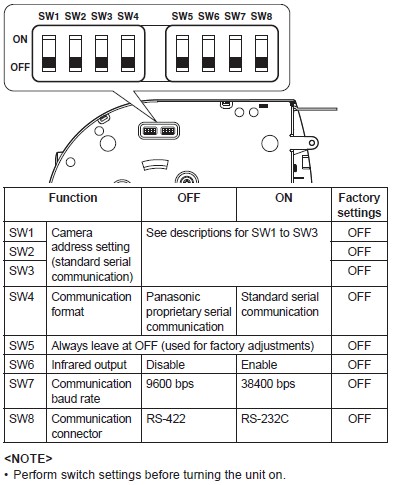
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Command** **AutoFocus** | Value 'Auto' | Value  'Manual' | |  |
| # AutoFocus example  InterfaceName.Update( | 'AutoFocus') |  | |  |
| Value = InterfaceName | .ReadStatus('AutoFocu | s') | |  |
| InterfaceName.Subscri | beStatus('AutoFocus', | None | , FeedbackHandler) |  |
| **Command** **AutoIris** | Value 'On' |  | Value 'Off' |  |
| # AutoIris example  InterfaceName.Update( | 'AutoIris') |  |  |  |
| Value = InterfaceName | .ReadStatus('AutoIris | ') |  |  |
| InterfaceName.Subscri | beStatus('AutoIris', | None, | FeedbackHandler) |  |
| **Command** **ColorBar** | Value 'On' |  | Value 'Off' |  |
| # ColorBar example  InterfaceName.Update( | 'ColorBar') |  |  |  |
| Value = InterfaceName | .ReadStatus('ColorBar | ') |  |  |
| InterfaceName.Subscri | beStatus('ColorBar', | None, | FeedbackHandler) |  |
| **Command**  **ConnectionStatus** | Value  'Connected' | | Value  'Disconnected' |  |
| Value = InterfaceName | .ReadStatus('ConnectionSta | | tus') |  |
| InterfaceName.Subscri | beStatus('ConnectionStatus | | ', None, FeedbackHandler) |  |
| **Command**  **Detail** | Value  'On'  'High' | | Value  'Off' | Value  'Low' |
| # Detail example  InterfaceName.Update( | 'Detail') | |  |  |
| Value = InterfaceName | .ReadStatus('Detail') | |  |  |
| InterfaceName.Subscri | beStatus('Detail', None, F | | eedbackHandler) |  |
| **Command** **Power** | Value 'On' | | Value  'Standby' | Value  'Starting' |
| # Power example  InterfaceName.Update('Power')  Value = InterfaceName.ReadStatus('Power')  InterfaceName.SubscribeStatus('Power', None, FeedbackHandler) | | | | |
| **Command** Value **PresetRecallStatus** 1 – 100 | | | | |
| # PresetRecallStatus example  InterfaceName.Update('PresetRecallStatus') | | | | |
| Value = InterfaceName.ReadStatus('PresetRecallStatus')  InterfaceName.SubscribeStatus('PresetRecallStatus', None, FeedbackHandler) | | | | |
| **Command**  **SceneFileControl** | Value  'Manual 1'  'Full Auto' | | Value  'Manual 2' | Value  'Manual 3' |
| # SceneFileControl ex | ample | |  |  |
| InterfaceName.Update( | 'SceneFileControl') | |  |  |
| Value = InterfaceName | .ReadStatus('SceneFileCont | | rol') |  |
| InterfaceName.Subscri | beStatus('SceneFileControl | | ', None, FeedbackHandler) |  |
| **Command** **Tally** | Value 'On' | | Value 'Off' |  |
| # Tally example  InterfaceName.Update( | 'Tally') | |  |  |
| Value = InterfaceName | .ReadStatus('Tally') | |  |  |
| InterfaceName.Subscri | beStatus('Tally', None, Fe | | edbackHandler) |  |
| **Command** **TallyInput** | Value  'Enable' | | Value  'Disable' |  |
| # TallyInput example  InterfaceName.Update('TallyInput')  Value = InterfaceName.ReadStatus('TallyInput')  InterfaceName.SubscribeStatus('TallyInput', None, FeedbackHandler) | | | | |

# Cable and Adapter Requirements

Captive Screw to RJ-45 jack cable. (Panasonic AWCA20T6G) Captive Screw to Male Mini 8-Pin Din RS232 cable.

# Notes for the Device

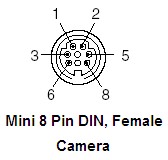
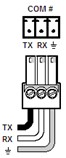
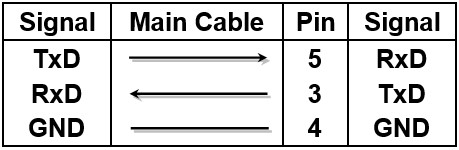
If applicable for the configured model, please set the dip switches according to the diagram below to choose between RS232 and RS422 control.



# Serial communication

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

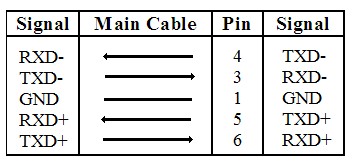
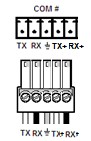
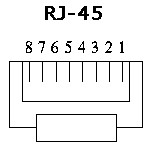
# Pin Assignments Diagram



# Serial communication

|  |  |
| --- | --- |
| **Port Type:** RS-422 | **Parity:** None |
| **Baud Rate:** 9600 | **Stop Bits:** One |
| **Data Bits:** 8 | **Flow Control:** None |

# Pin Assignments Diagram



# Network communication

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

|  |  |
| --- | --- |
| **Port Type:** | Ethernet |
| **Default Port:** | 80 |
| **Logon Credentials Supported:** | Yes |
| **Multi-Connection Capabilities:** | No |
| **Port Changeability:** | Yes |

**Ethernet Module Configuration Description**

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

# Notes for the Device

If using authentication, ensure that Authentication setting is set to Basic on the device.

For the AW-HE40 models, Host and User Authentication settings cannot be set to **On** simultaneously.

# Appendix A. Set Commands (Serial)

|  |  |
| --- | --- |
| **Auto Focus Off** | \x02OAF:0\x03 |
| **Auto Focus On** | \x02OAF:1\x03 |
| **Extender AF Control Off** | #D10\x0D |
| **Extender AF Control On** | #D11\x0D |
| **Focus Far Focus Speed 1** | #F51\x0D |
| **Focus Far Focus Speed 49** | #F99\x0D |
| **Focus Near Focus Speed 1** | #F49\x0D |
| **Focus Near Focus Speed 49** | #F01\x0D |
| **Focus Stop Focus Speed 1** | #F50\x0D |
| **Focus Stop Focus Speed 49** | #F50\x0D |
| **Iris 1** | #I01\x0D |
| **Iris 99** | #I99\x0D |
| **Iris Mode Auto** | #D31\x0D |
| **Iris Mode Manual** | #D30\x0D |
| **Pan Left Pan Speed 1** | #P49\x0D |
| **Pan Left Pan Speed 49** | #P01\x0D |
| **Pan Right Pan Speed 1** | #P51\x0D |
| **Pan Right Pan Speed 49** | #P99\x0D |
| **Pan Stop Pan Speed 1** | #P50\x0D |
| **Pan Stop Pan Speed 49** | #P50\x0D |
| **Power Off** | #O0\x0D |
| **Power On** | #O1\x0D |
| **Preset 1 Type Recall** | #R00\x0D |
| **Preset 1 Type Save** | #M00\x0D |
| **Preset 100 Type Recall** | #R99\x0D |
| **Preset 100 Type Save** | #M99\x0D |
| **Tally Enable Off** | #TAE0\x0D |
| **Tally Enable On** | #TAE1\x0D |
| **Tally Off** | #DA0\x0D |
| **Tally On** | #DA1\x0D |
| **Tilt Down Tilt Speed 1** | #T49\x0D |
| **Tilt Down Tilt Speed 49** | #T01\x0D |
| **Tilt Stop Tilt Speed 1** | #T50\x0D |
| **Tilt Stop Tilt Speed 49** | #T50\x0D |
| **Tilt Up Tilt Speed 1** | #T51\x0D |
| **Tilt Up Tilt Speed 49** | #T99\x0D |
| **Zoom Stop Zoom Speed 1** | #Z50\x0D |
| **Zoom Stop Zoom Speed 49** | #Z50\x0D |
| **Zoom Tele Zoom Speed 1** | #Z51\x0D |
| **Zoom Tele Zoom Speed 49** | #Z99\x0D |
| **Zoom Wide Zoom Speed 1** | #Z49\x0D |
| **Zoom Wide Zoom Speed 49** | #Z01\x0D |

# Appendix B. Update Commands (Serial)

|  |  |
| --- | --- |
| **Extender AF Control** | #D1\x0D |
| **Iris** | #I\x0D |
| **Iris Mode** | #D3\x0D |
| **Power** | #O\x0D |
| **Tally** | #DA\x0D |
| **Tally Enable** | #TAE\x0D |