

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 Scriptor Modules, refer to the "[Guide to Using Scriptor Modules](#)" document.

Device Specifications

Device Type: Audio Processor
Manufacturer: QSC
Firmware Version: 9.2.0-2109.002
Model(s): Q-Sys Core 250i, Q-Sys Core 500i, Q-Sys Core 1000, Q-Sys Core 1100, Q-Sys Core 3000, Q-Sys Core 4000, Q-Sys Core 3100, Q-Sys Core 110f, Q-Sys Core 510i, Q-Sys Cinema Core 110c, Q-Sys Core 8 Flex, Q-Sys Core Nano

Tested on the Following Software and Firmware Versions

IP Link Pro Control Processor Firmware	Global Scriptor Version
3.16.0000-b013	2.18.0

Version History

Module Version	Date	Notes
1_13_2_1	3/7/2023	Fixed communication sheet regarding GetStatusString status.
1_13_2_0	2/2/2023	Added Q-Sys Core Nano model.
1_13_0_3	5/2/2022	Updated ReceiveData function.
1_13_0_2	3/16/2022	Fixed communication sheet notes regarding ControlSetString command.
1_13_0_1	3/1/2022	Added command: ControlSetString.
1_13_0_0	1/13/2022	Added Q-Sys Core 8 Flex model. Removed port changeability. Changed range to 0.001 to 1.0 in steps of 0.001 for PTZ Speed.

**Global Scripter Module
Communication Sheet**

1_12_8_2	7/29/2021	Fixed login authentication.
1_12_8_1	6/17/2021	Updated MatchError function to parse the error that is returned from the device.
1_12_8_0	1/20/2021	Added Gain Expansion command and fixed a regular expression.
1_12_6_0	9/9/2020	Fixed Design Name status.
1_12_4_0	7/10/2020	Changed range of Control Set Value to 0 – 100.
1_12_1_0	10/22/2019	Updated Gain Value Range. Fixed Design Name status. Added Call History Result Set. Added the following Commands: <ul style="list-style-type: none">- PhonebookControl- PhonebookListUpdate- PhonebookNavigation- PhonebookResult Set- PhonebookResults- PhonebookSearch- PhonebookSelected- PhonebookUpdate CallHistoryResultSet.
1_10_4_1	8/19/2019	Fixed DesignName status.
1_10_4_0	10/9/2018	Updated module to revision B1.
1_10_3_0	8/21/2018	Added model: Q-Sys Cinema Core 110c, Added Status command: Level Meter.
1_10_0_1	5/10/2018	Fixed Call History navigation. Added missing command GetStatusString
1_10_0_0	4/17/2018	Updated Camera router command to Router command, to account for audio routing functionality. Added status to Router command. Added Camera Control and Call History commands. Fixed Snapshot Load Command. Added Serial Control.
1_6_5_0	4/4/2017	Added Snapshot Save, Snapshot Load, updated module, added model Q-Sys Core 510i
1_3_0_0	9/12/2016	Update to standard. Fixed Mute.
1_1_0_0	8/24/2016	Initial Version

Module Notes

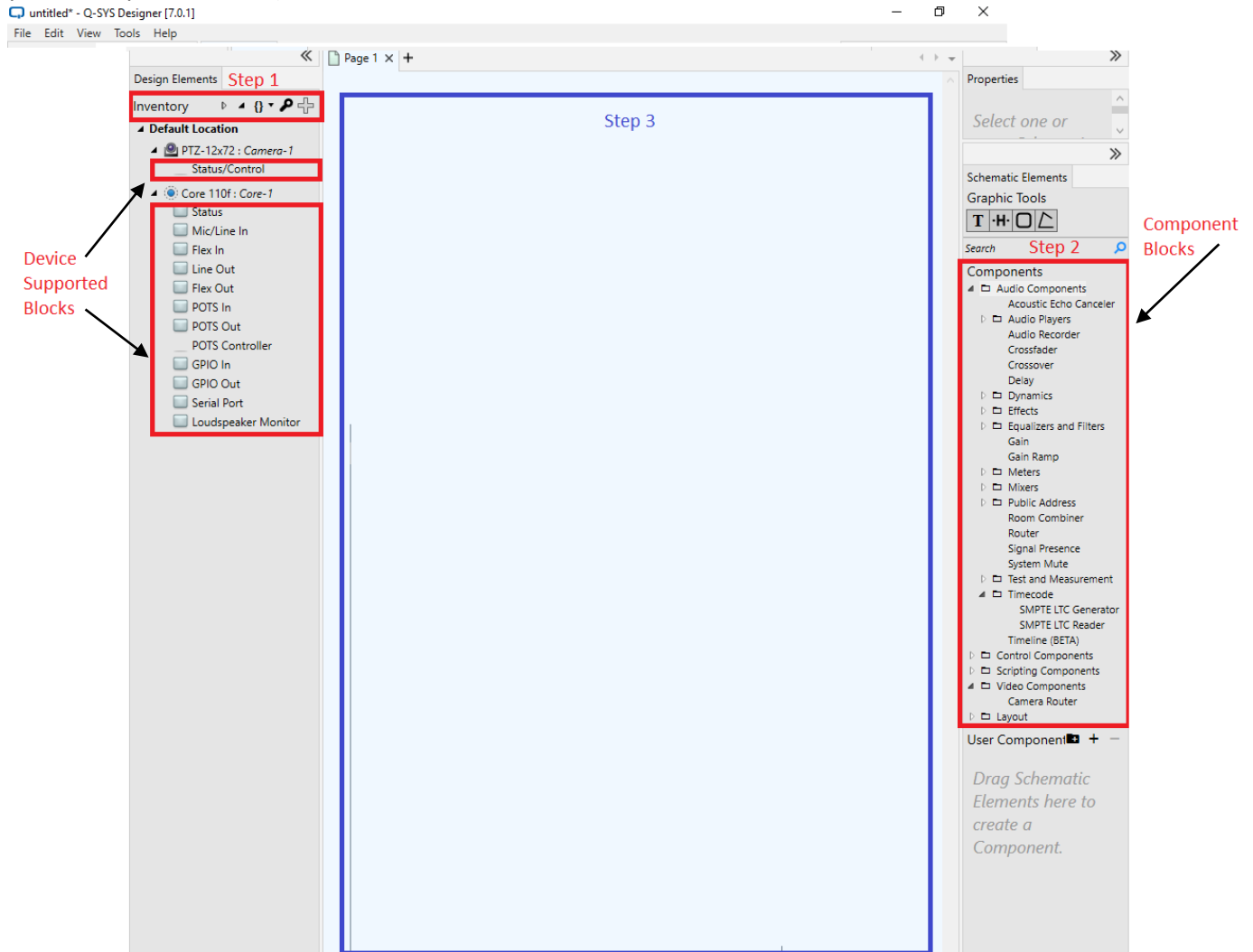
- Unidirectional variable must be set to 'True' if status is not required. Default value is 'False'.
Example: `InterfaceName.Unidirectional = 'True'`
- connectionCounter variable must be set to the number of queries that will be sent to the device before displaying 'Disconnected' if no response is received. Default value is 15.
Example: `InterfaceName.connectionCounter = 5`
- If login credentials are required, devicePassword and deviceUsername must be set accordingly.
Example: `InterfaceName.devicePassword = 'extron'`
- NumberOfCallHistoryResults variable must be set accordingly. Default value is '5'.
NumberOfCallHistoryResults ranges from 1 to 10.
Example: `InterfaceName.NumberofCallHistoryResults = '5'`
- NumberOfPhonebookResults variable must be set accordingly. Default value is '5'.
NumberOfPhonebookResults ranges from 1 to 10.
Example: `InterfaceName.NumberofPhonebookResults = '5'`
- The Control ID command parameters are determined and obtained by the Q-SYS Designer project.
- For Serial control, the RS232 port must be enabled through Q-SYS Designer software. Please contact QSC for additional information on RS232 setup.

Supported Classes and Examples

SerialClass
<code>InterfaceName = ModuleName.SerialClass(ProcessorName, 'COM1', Model='Q-Sys Core 250i')</code>
SerialOverEthernetClass
<code>InterfaceName = ModuleName.SerialOverEthernetClass('192.168.254.254', 2001, Model='Q-Sys Core 250i')</code>
EthernetClass
<code>InterfaceName = ModuleName.EthernetClass('192.168.254.254', 1702, Model='Q-Sys Core 250i')</code>

Q-SYS Designer Software

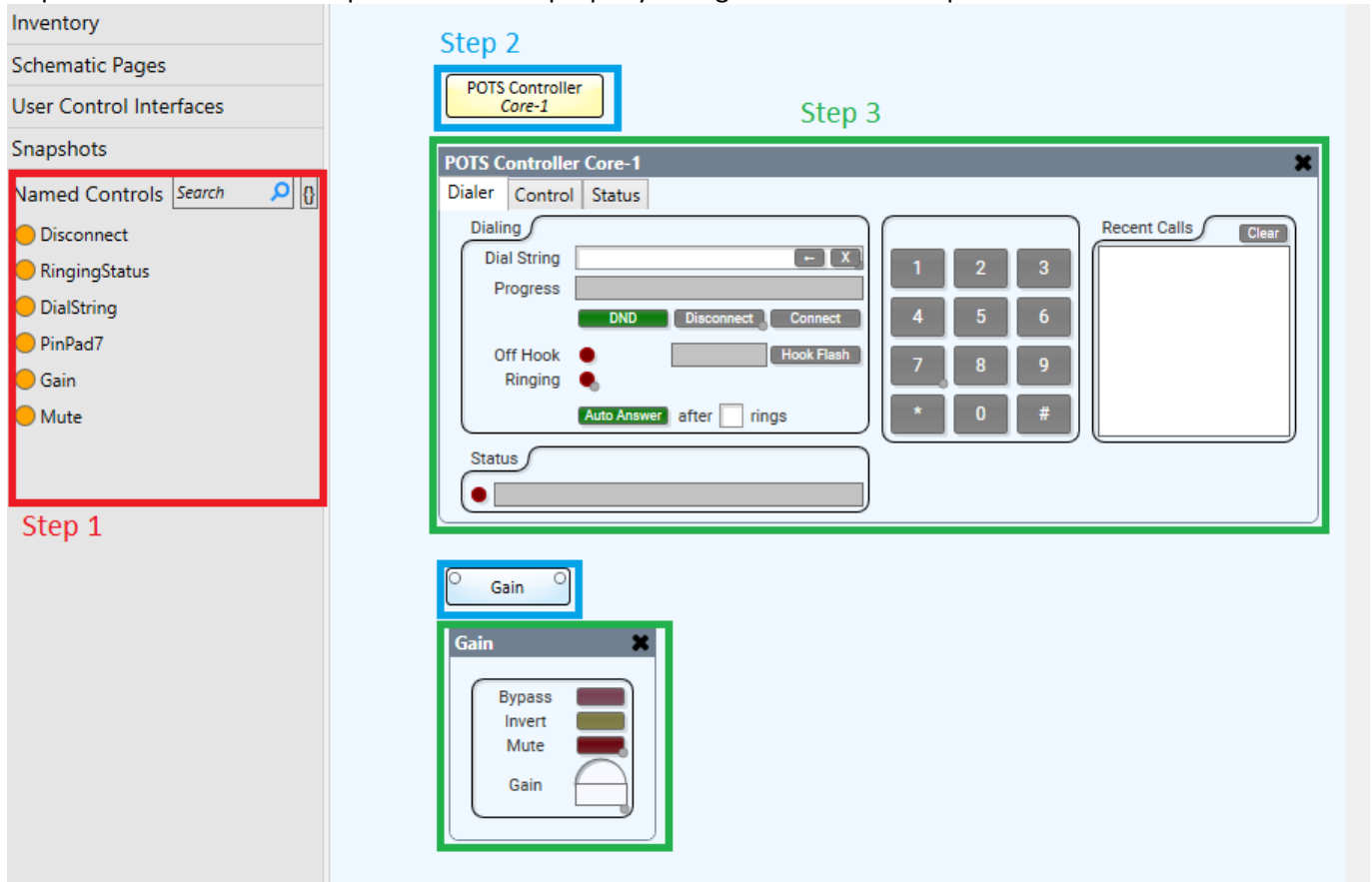
The Q-SYS Designer Software is used to design configurations for QSC devices. This section goes over the first steps to take to design a configuration using the software. (If you have already designed a Q-SYS configuration, you can skip this section.)



1. Open Q-SYS Designer and locate the **Inventory** section. This section is used to add and view the devices to be used in the Q-SYS configuration. It also displays the blocks the devices support.
2. Locate the **Components** section. This section is used to add and view the component blocks to be used in the Q-SYS configuration.
3. Once these sections have been located, you can start designing the Q-SYS configuration by dragging Device blocks and Component blocks into the middle section of the screen boxed in blue.

Obtaining the Control ID

Q-SYS Designer configurations consist of individual functions that correspond to a unique Control ID. These Control IDs are used in this module for control and status functionality. Most commands in this module require a Control ID module parameter to be properly configured in Global Scripter.



1. Locate the **Named Controls** section. This is where the list of Control IDs will be displayed.
2. Select or click on one of the Device blocks or Component blocks in your Q-SYS configuration.
3. A window will pop up once you select or click a block. The window will display the control and status functions available for the selected block.

Obtaining the Control ID (continued)

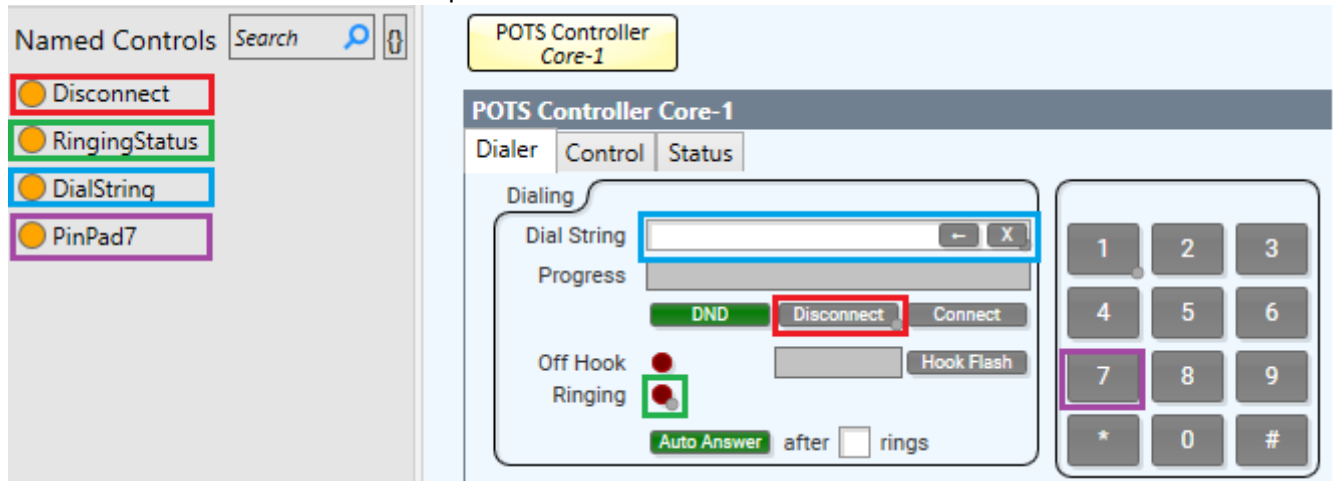
Example Q-SYS Configuration:

The screenshot displays the Q-SYS configuration software interface. On the left, a sidebar contains a 'Named Controls' list with various control IDs. A yellow box highlights the 'Bridge1Camera1Select' control, and a red box highlights the 'FocusManualSpeed' control. The main workspace shows two windows: 'PTZ-12x72 Camera-1' and 'Camera Router 2x1'. The 'PTZ-12x72 Camera-1' window is divided into several sections: 'Pan / Tilt / Zoom' (containing directional buttons and speed sliders), 'Focus' (containing focus mode buttons and speed sliders), and 'Preview' (showing a camera feed). A red box highlights the 'FocusManualSpeed' control in the 'Focus' section. The 'Camera Router 2x1' window shows a diagram of the camera router with a red box highlighting the 'Camera 1' port.

4. Within each window you can click on each of the control and status functions.
5. Select one of the displayed functions and drag it to the **Named Controls** section. The Control ID will then be displayed in this section. Control IDs may be renamed by double clicking it within the list.

Configuring Commands

Once you obtained the Control ID of a function from a specific block, keep in mind what type of function it is. The commands in this module correspond to these different functions.



Control Set String

Fields and list entry selections can be controlled by using Control Set String with Control String set to the desired string. For example, issuing Control Set String(Control ID: DialString, Control String: 123) will set Dial String to 123.

Control Set Value

Single-press functions can be controlled with Control Value set to 1. For example, issuing Control Set Value(Control ID: Disconnect, Control Value: 1) will disconnect the call.

Dial String

Keypad functions can be configured by using Dial String with Control Value of 1. For example, digit 7 corresponds to the Control ID PinPad7. Issuing Dial String(Dial String ID: DialString, Control ID: PinPad7, Control Value: 1) will press 7 once.

To view emulated and live status for Dial String, configure Get Status String(Control ID: DialString). The Control ID of Get Status String must be configured as the Dial String ID used for Dial String. Once this requirement is met, Dial String commands will update emulated status of Get Status String for dial string feedback. However, to update this status regularly, configure live status for Get Status String.

Get Status String

Functions with a string can have its status displayed. For example, to view the string in the Dial String function that corresponds to Control ID DialString, configure Get Status String(Control ID: DialString)

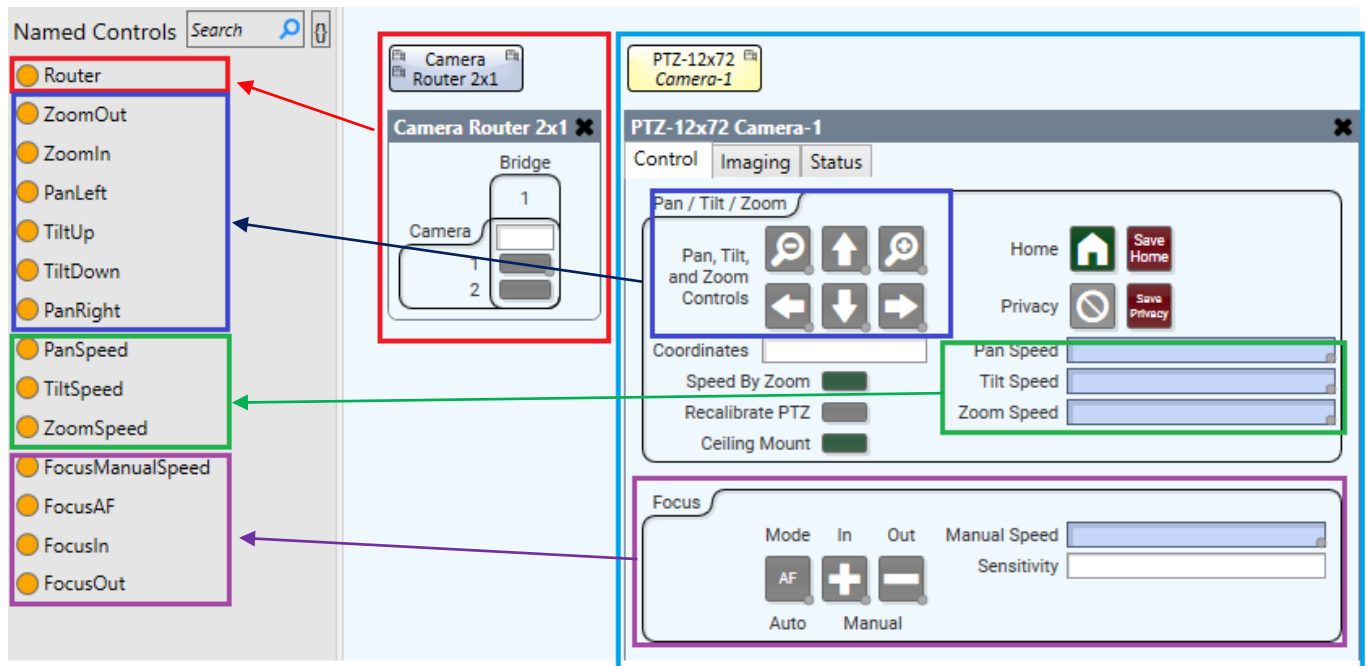
Get Status Value

Functions with binary states such as On/Off can have its status displayed. For example, to view the status of the Ringing function that corresponds to Control ID RingingStatus, configure Get Status Value(Control ID: RingingStatus).

Configuring Commands (continued)

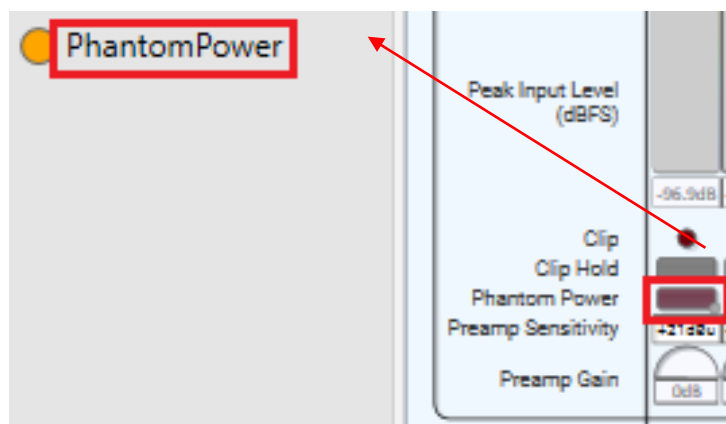
Focus, PTZ, and Router Commands

Focus commands (Control, Mode, and Speed) and PTZ commands (Control and Speed) correspond to the functions in the PTZ Camera block shown below. The Router command corresponds to the functions in the Camera Router block shown below. For example, configure PTZ Control(Control ID: TiltUp) to enable and disable tilt up control.



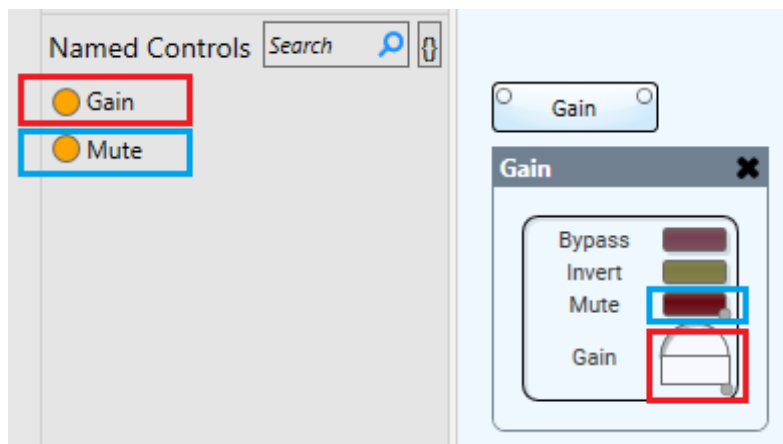
Function

The Function command corresponds to any control that functions *specifically* as an enable/disable such as Phantom Power. For example, configure Function(Control ID: PhantomPower) for control and status of the Phantom Power function shown below.



Gain, Gain Expansion, and Mute

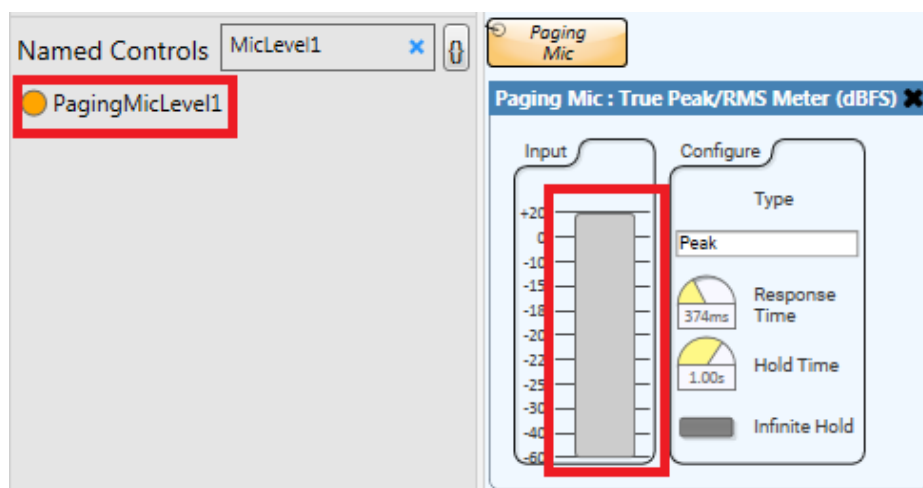
Gain and Mute functions like the ones shown below can be controlled using the Gain and Mute commands with their corresponding Control ID. For example, configure Gain(Control ID: Gain) for control and status of the Gain function below.



*** For Gain blocks from audio expansion units with range -100 to 83 dB use the **Gain Expansion** command.

Level Meter

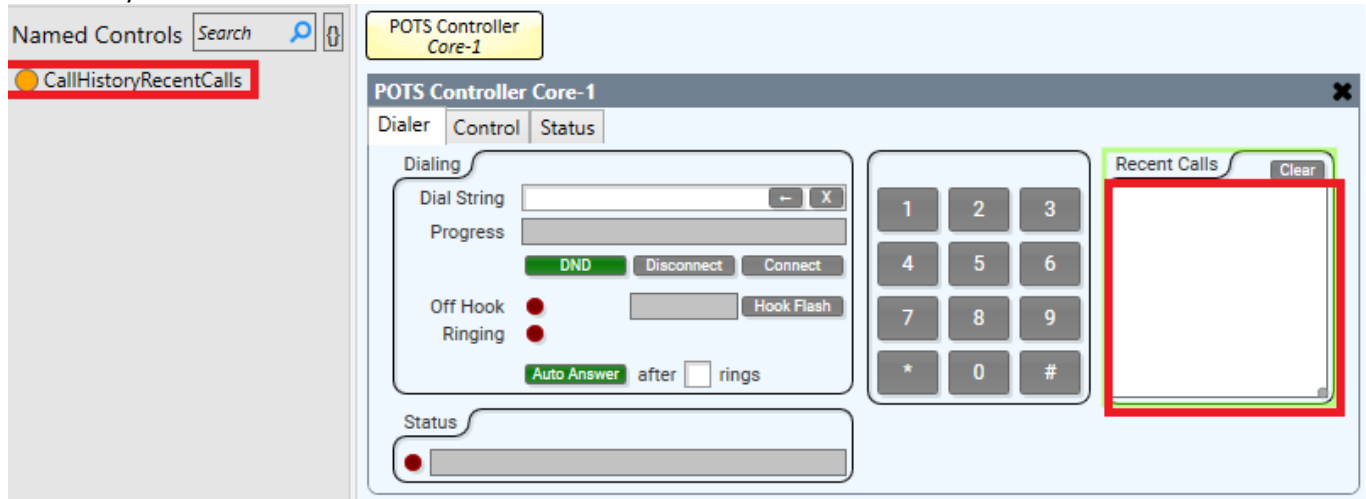
Meter functions from the Level Meter component like the one shown below can be configured using the Level Meter command with their corresponding Control ID. For example, configure Level Meter(Control ID: PagingMicLevel1) to display the meter status.



Configuring Commands (continued)

Call History Commands

Call History commands apply to functions like Recent Calls shown below that corresponds to Control ID CallHistoryRecentCalls.

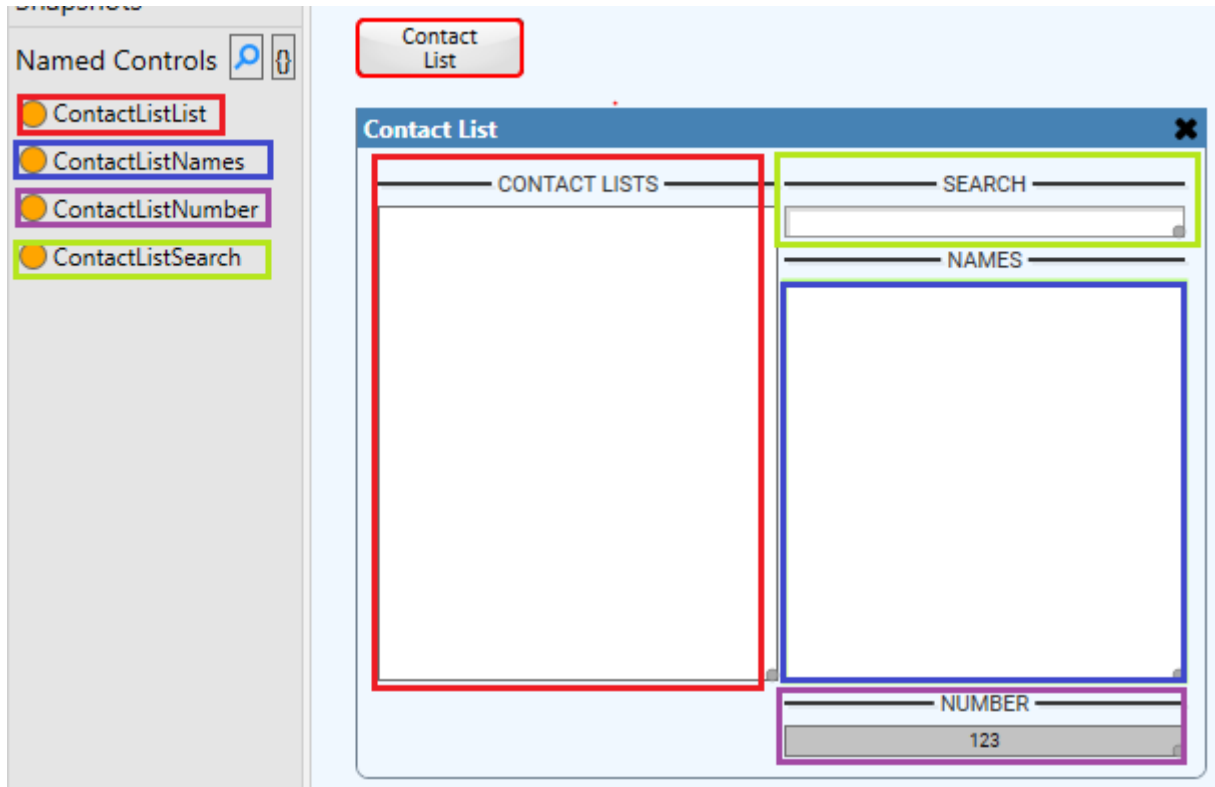


- **Call History Update**
This command queries for the call history to update and populate Call History Results.
- **Call History Results**
This displays the entries of the call history returned from Call History Update. The number of entries to display is set by the module parameter Number of Call History Results which is set to 5 by default.
- **Call History Result Set**
This command sends the corresponding Call History Results dial string to the Dial String function. The Dial String ID must be set to the Dial String function's Control ID.
- **Call History Navigation**
This command is used for navigating through the call history entries.

Configuring Commands (continued)

Phonebook Commands

Phonebook commands apply to functions like Contact Lists, Names, Number, and Search shown below.



- **Phonebook Control**
This command will send the dial string from the Number function specified by Phonebook ID to a string function specified by Control ID. For example, Phonebook Control(Control ID: DialString, Phonebook ID: ContactListNumber) will send the selected number to a Dial String field. Note, this requires Get Status String (Control ID: ContactListNumber) to be populated to work. To regularly update Get Status String, configure live status.
- **Phonebook List Update**
This command queries for the contact lists to update and populate Phonebook Results. Note, this queries for the list of phonebooks, not the contacts within a phonebook.
- **Phonebook Navigation**
This command is used for navigating through the phonebook entries.

-
- **Phonebook Results**
This displays the entries of the phonebook returned from Phonebook List Update and Phonebook Update. The number of entries to display is set by the module parameter Number of Phonebook Results which is set to 5 by default.
 - **Phonebook Result Set**
This command sends the corresponding Phonebook Results entry to the Phonebook Selected.
 - **Phonebook Search**
This command sets the Search function to the string specified by Phonebook Search String to filter the phonebook results returned from Phonebook Update. Note, the Search function only filters entries within a phonebook, not the list of phonebooks.
 - **Phonebook Search String**
This command supplies the search string for Phonebook Search.
 - **Phonebook Selected**
This command displays the phonebook entry selected by the Phonebook Result Set command.
 - **Phonebook Update**
This command queries for the entries within a phonebook to update and populate Phonebook Results.

Control Commands

Format with Qualifier:

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

Format without Qualifier:

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

Command CallHistoryNavigation	Value 'Up' 'Page Down'	Value 'Down'	Value 'Page Up'
Qualifier Key 'Control ID'	Qualifier Value 'String'		
# CallHistoryNavigation example InterfaceName.Set('CallHistoryNavigation', 'Up', {'Control ID': 'String'})			
Command CallHistoryResultSet	Value None		
Qualifier Key 'Dial String ID'	Qualifier Value 'String'		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
Qualifier Key 'Position'	Qualifier Value '1' – '10'		
# CallHistoryResultSet example InterfaceName.Set('CallHistoryResultSet', None, {'Dial String ID': 'String', 'Control ID': 'String', 'Position': '1'})			
Command CallHistoryUpdate	Value None		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
# CallHistoryUpdate example InterfaceName.Set('CallHistoryUpdate', None, {'Control ID': 'String'})			
Command ControlSetString	Value None		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
Qualifier Key 'Control String'	Qualifier Value 'String'		
# ControlSetString example InterfaceName.Set('ControlSetString', None, {'Control ID': 'String', 'Control String': 'String'})			
Command ControlSetValue	Value None		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
Qualifier Key 'Control Value'	Qualifier Value 0 – 100		
# ControlSetValue example InterfaceName.Set('ControlSetValue', None, {'Control ID': 'String', 'Control Value': 100})			

Command FocusControl	Value 'Enable'	Value 'Disable'
Qualifier Key 'Control ID'	Qualifier Value 'String'	
# FocusControl example InterfaceName.Set('FocusControl', 'Enable', {'Control ID': 'String'})		
Command FocusSpeed	Value 0.001 to 0 in steps of 0.001	
Qualifier Key 'Control ID'	Qualifier Value 'String'	
# FocusSpeed example InterfaceName.Set('FocusSpeed', 0, {'Control ID': 'String'})		
Command Function	Value 'Enable'	Value 'Disable'
Qualifier Key 'Control ID'	Qualifier Value 'String'	
# Function example InterfaceName.Set('Function', 'Enable', {'Control ID': 'String'})		
Command Gain	Value -100 to 20 in steps of 0.1	
Qualifier Key 'Control ID'	Qualifier Value 'String'	
# Gain example InterfaceName.Set('Gain', 20, {'Control ID': 'String'})		
Command GainExpansion	Value -100 to 83 in steps of 0.1	
Qualifier Key 'Control ID'	Qualifier Value 'String'	
# GainExpansion example InterfaceName.Set('GainExpansion', 83, {'Control ID': 'String'})		
Command Mute	Value 'On'	Value 'Off'
Qualifier Key 'Control ID'	Qualifier Value 'String'	
# Mute example InterfaceName.Set('Mute', 'On', {'Control ID': 'String'})		
Command PhonebookControl	Value None	
Qualifier Key 'Control ID'	Qualifier Value 'String'	
Qualifier Key 'Phonebook ID'	Qualifier Value 'String'	
# PhonebookControl example InterfaceName.Set('PhonebookControl', None, {'Control ID': 'String', 'Phonebook ID': 'String'})		
Command PhonebookListUpdate	Value None	
Qualifier Key 'Control ID'	Qualifier Value 'String'	
# PhonebookListUpdate example InterfaceName.Set('PhonebookListUpdate', None, {'Control ID': 'String'})		

Global Scripter Module
Communication Sheet

Command PhonebookNavigation	Value 'Up' 'Page Down'	Value 'Down'	Value 'Page Up'
Qualifier Key 'Control ID'	Qualifier Value 'String'		
# PhonebookNavigation example InterfaceName.Set('PhonebookNavigation', 'Up', {'Control ID': 'String'})			
Command PhonebookResultSet	Value None		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
Qualifier Key 'Position'	Qualifier Value '1' – '10'		
# PhonebookResultSet example InterfaceName.Set('PhonebookResultSet', None, {'Control ID': 'String', 'Position': '1'})			
Command PhonebookSearch	Value None		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
Qualifier Key 'Contact'	Qualifier Value 'String'		
# PhonebookSearch example InterfaceName.Set('PhonebookSearch', None, {'Control ID': 'String', 'Contact': 'String'})			
Command PhonebookSelected	Value 'String'		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
# PhonebookSelected example InterfaceName.Set('PhonebookSelected', 'String', {'Control ID': 'String'})			
Command PhonebookUpdate	Value None		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
# PhonebookUpdate example InterfaceName.Set('PhonebookUpdate', None, {'Control ID': 'String'})			
Command PTZControl	Value 'Enable'	Value 'Disable'	
Qualifier Key 'Control ID'	Qualifier Value 'String'		
# PTZControl example InterfaceName.Set('PTZControl', 'Enable', {'Control ID': 'String'})			
Command PTZSpeed	Value 0.001 to 1 in steps of 0.001		
Qualifier Key 'Control ID'	Qualifier Value 'String'		
# PTZSpeed example InterfaceName.Set('PTZSpeed', 1, {'Control ID': 'String'})			
Command Router	Value 'Enable'	Value 'Disable'	
Qualifier Key 'Control ID'	Qualifier Value 'String'		
# Router example InterfaceName.Set('Router', 'Enable', {'Control ID': 'String'})			

Global Scripter Module Communication Sheet

Command SnapshotLoad	Value '1' – '24'
Qualifier Key 'Load Time'	Qualifier Value 0 to 60 in steps of 1
Qualifier Key 'Bank'	Qualifier Value 'String'
# SnapshotLoad example InterfaceName.Set('SnapshotLoad', '1', {'Load Time': 60, 'Bank': 'String'})	
Command SnapshotSave	Value '1' – '24'
Qualifier Key 'Bank'	Qualifier Value 'String'
# SnapshotSave example InterfaceName.Set('SnapshotSave', '1', {'Bank': 'String'})	

Status Available

For all commands, call Update to receive the latest status. ConnectionStatus, CallHistoryResults, and PhonebookResults 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 CallHistoryResults	Value 'String'
Qualifier Key 'Control ID'	Qualifier Value 'String'
Qualifier Key 'Position'	Qualifier Value '1' – '10'
# CallHistoryResults example Value = InterfaceName.ReadStatus('CallHistoryResults', {'Control ID': 'String', 'Position': '1'}) InterfaceName.SubscribeStatus('CallHistoryResults', None, FeedbackHandler)	
Command CallStatus	Value 'String'
Qualifier Key 'Control ID'	Qualifier Value 'String'
# CallStatus example InterfaceName.Update('CallStatus', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('CallStatus', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('CallStatus', None, FeedbackHandler)	
Command CallerID	Value 'String'
Qualifier Key 'Control ID'	Qualifier Value 'String'
# CallerID example InterfaceName.Update('CallerID', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('CallerID', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('CallerID', None, FeedbackHandler)	
Command CallerName	Value 'String'
Qualifier Key 'Control ID'	Qualifier Value 'String'
# CallerName example InterfaceName.Update('CallerName', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('CallerName', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('CallerName', None, FeedbackHandler)	

Command	Value	Value
ConnectionStatus	'Connected'	'Disconnected'
# ConnectionStatus example Value = InterfaceName.ReadStatus('ConnectionStatus') InterfaceName.SubscribeStatus('ConnectionStatus', None, FeedbackHandler)		
Command	Value	
DesignName	'String'	
# DesignName example InterfaceName.Update('DesignName') Value = InterfaceName.ReadStatus('DesignName') InterfaceName.SubscribeStatus('DesignName', None, FeedbackHandler)		
Command	Value	Value
FocusMode	'Auto'	'Manual'
Qualifier Key	Qualifier Value	
'Control ID'	'String'	
# FocusMode example InterfaceName.Update('FocusMode', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('FocusMode', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('FocusMode', None, FeedbackHandler)		
Command	Value	
FocusSpeed	0.001 to 0 in steps of 0.001	
Qualifier Key	Qualifier Value	
'Control ID'	'String'	
# FocusSpeed example InterfaceName.Update('FocusSpeed', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('FocusSpeed', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('FocusSpeed', None, FeedbackHandler)		
Command	Value	Value
Function	'Enable'	'Disable'
Qualifier Key	Qualifier Value	
'Control ID'	'String'	
# Function example InterfaceName.Update('Function', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('Function', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('Function', None, FeedbackHandler)		
Command	Value	
Gain	-100 to 20 in steps of 0.1	
Qualifier Key	Qualifier Value	
'Control ID'	'String'	
# Gain example InterfaceName.Update('Gain', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('Gain', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('Gain', None, FeedbackHandler)		
Command	Value	
GainExpansion	-100 to 83 in steps of 0.1	
Qualifier Key	Qualifier Value	
'Control ID'	'String'	
# GainExpansion example InterfaceName.Update('GainExpansion', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('GainExpansion', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('GainExpansion', None, FeedbackHandler)		

Global Scripter Module
Communication Sheet

Command GetStatusString	Value 'String'
Qualifier Key 'Control ID'	Qualifier Value 'String'
# GetStatusString example InterfaceName.Update('GetStatusString', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('GetStatusString', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('GetStatusString', None, FeedbackHandler)	
Command GetStatusValue	Value 'On' Value 'Off'
Qualifier Key 'Control ID'	Qualifier Value 'String'
# GetStatusValue example InterfaceName.Update('GetStatusValue', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('GetStatusValue', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('GetStatusValue', None, FeedbackHandler)	
Command LevelMeter	Value -120 – 20
Qualifier Key 'Control ID'	Qualifier Value 'String'
# LevelMeter example InterfaceName.Update('LevelMeter', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('LevelMeter', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('LevelMeter', None, FeedbackHandler)	
Command Mute	Value 'On' Value 'Off'
Qualifier Key 'Control ID'	Qualifier Value 'String'
# Mute example InterfaceName.Update('Mute', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('Mute', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('Mute', None, FeedbackHandler)	
Command PhonebookResults	Value 'String'
Qualifier Key 'Control ID'	Qualifier Value 'String'
Qualifier Key 'Position'	Qualifier Value '1' – '10'
# PhonebookResults example Value = InterfaceName.ReadStatus('PhonebookResults', {'Control ID': 'String', 'Position': '1'}) InterfaceName.SubscribeStatus('PhonebookResults', None, FeedbackHandler)	
Command PTZSpeed	Value 0.001 to 1 in steps of 0.001
Qualifier Key 'Control ID'	Qualifier Value 'String'
# PTZSpeed example InterfaceName.Update('PTZSpeed', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('PTZSpeed', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('PTZSpeed', None, FeedbackHandler)	

Global Scripter Module Communication Sheet

Command	Value	Value
Router	'Enable'	'Disable'
Qualifier Key	Qualifier Value	
'Control ID'	'String'	
<pre># Router example InterfaceName.Update('Router', {'Control ID': 'String'}) Value = InterfaceName.ReadStatus('Router', {'Control ID': 'String'}) InterfaceName.SubscribeStatus('Router', None, FeedbackHandler)</pre>		

Cable and Adapter Requirements

(Q-Sys Core 110f and Cinema Core 110c) Captive Screw to 3-Pin 5mm Euro Connector RS-232 Serial Cable

(Q-Sys Core 8 Flex and Q-Sys Core Nano) Captive Screw to Captive Screw RS-232 Serial Cable

(Other models) Captive Screw to Female DB9 RS-232 Serial Cable

Notes for the Device

Configure RS-232 settings on the device using the Q-SYS Designer software.

Serial communication

Port Type: RS-232

Baud Rate: 115200

Data Bits: 8

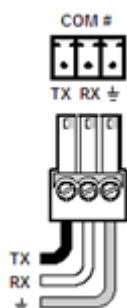
Parity: None

Stop Bits: One

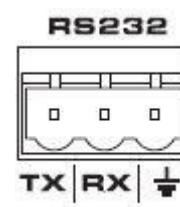
Flow Control: None

Pin Assignments Diagram

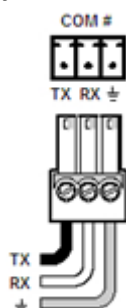
Q-Sys Core 110f and Cinema Core 110c:



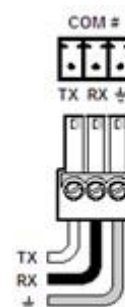
Signal	Main Cable	Signal
TxD		TxD
RxD		RxD
GND		GND



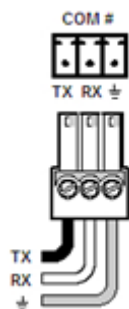
Q-Sys Core 8 Flex:



Signal	Main Cable	Signal
TxD		TxD
RxD		RxD
GND		GND



Other models:



Signal	Main Cable	Pin	Signal
TxD	→	2	RxD
RxD	←	3	TxD
GND	→	5	GND



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: 1702

Logon Credentials Yes

Supported:

Multi-Connection Yes

Capabilities:

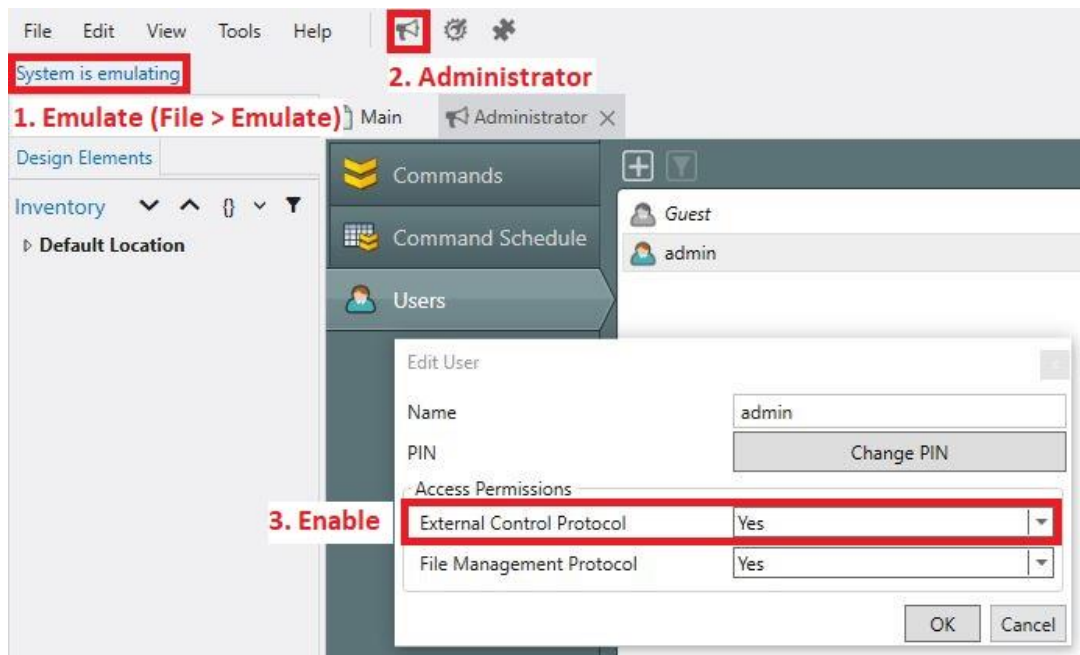
Port Changeability: No

Ethernet Module Configuration Description

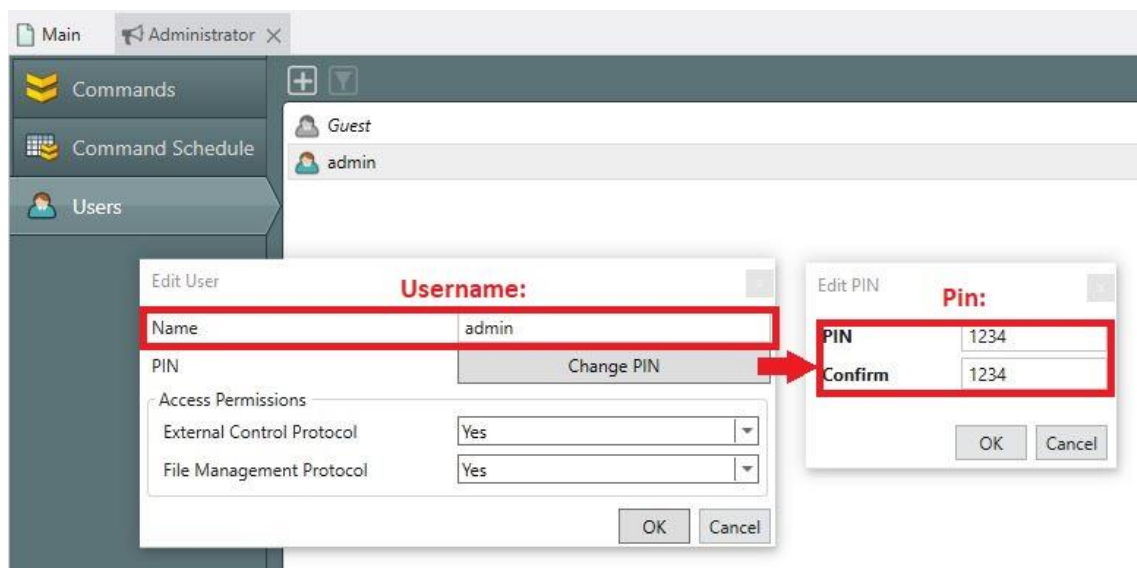
Please refer to user manual for settings and changes to the network communication

Notes for the Device

To ensure the functionality of the module, within Q-SYS Designer, **External Control Protocol** must be enabled for the user that the module will authenticate with the steps shown in the image:



Note: If authentication is enabled, please supply the username and its respective pin configured in Q-SYS Designer to the module in Global Scripter. If authentication is disabled, please ensure the username and password for the module are blank (empty string). Failure to do so will cause the module to not authenticate, resulting in no control commands and status queries from sending to the device. In this case, please verify the authentication requirements configured for the device via Q-SYS Designer matches the module in Global Scripter and rebuild your Global Scripter project.



Appendix A. Set Commands

Call History Update None Control ID testString	cgm "testString"\x0A
Control Set String None Control ID testString Control String testString	css "testString" "testString"\x0A
Control Set Value None Control ID testString Control Value 0	csv "testString" 0\x0A
Control Set Value None Control ID testString Control Value 100	csv "testString" 100\x0A
Focus Control Disable Control ID testString	csv "testString" 0\x0A
Focus Control Enable Control ID testString	csv "testString" 1\x0A
Focus Speed 0 Control ID testString	csv "testString" 0\x0A
Function Disable Control ID testString	csv "testString" 0\x0A
Function Enable Control ID testString	csv "testString" 1\x0A
Gain -100 Control ID testString	csv "testString" -100\x0A
Gain 20 Control ID testString	csv "testString" 20\x0A
Gain Expansion -100 Control ID testString	csv "testString" -100\x0A
Gain Expansion 83 Control ID testString	csv "testString" 83\x0A
Mute Off Control ID testString	csv "testString" 0\x0A
Mute On Control ID testString	csv "testString" 1\x0A
Phonebook Control None Control ID testString Phonebook ID testString	css "testString" "Dial String"\x0A
Phonebook List Update None Control ID testString	cgm "testString"\x0A
Phonebook Search None Control ID testString	css "testString" "Contact String"\x0A
PTZ Control Disable Control ID testString	csv "testString" 0\x0A
PTZ Control Enable Control ID testString	csv "testString" 1\x0A
PTZ Speed 0 Control ID testString	csv "testString" 0\x0A
PTZ Speed 1 Control ID testString	csv "testString" 1\x0A
Router Disable Control ID testString	csv "testString" 0\x0A
Router Enable Control ID testString	csv "testString" 1\x0A
Snapshot Load 1 Load Time 0 Bank testString	ssl "testString" 1 0\x0A
Snapshot Load 1 Load Time 60 Bank testString	ssl "testString" 1 60\x0A
Snapshot Load 24 Load Time 0 Bank testString	ssl "testString" 24 0\x0A
Snapshot Load 24 Load Time 60 Bank testString	ssl "testString" 24 60\x0A
Snapshot Save 1 Bank testString	sss "testString" 1\x0A
Snapshot Save 24 Bank testString	sss "testString" 24\x0A

Appendix B. Update Commands

Call Status Control ID testString	cg "testString"\x0A
Caller ID Control ID testString	cg "testString"\x0A
Caller Name Control ID testString	cg "testString"\x0A
Design Name	sg\x0A
Focus Mode Control ID testString	cg "testString"\x0A
Focus Speed Control ID testString	cg "testString"\x0A
Function Control ID testString	cg "testString"\x0A
Gain Control ID testString	cg "testString"\x0A
Gain Expansion Control ID testString	cg "testString"\x0A
Get Status String Control ID testString	cg "testString"\x0A
Get Status Value Control ID testString	cg "testString"\x0A
Level Meter Control ID testString	cg "testString"\x0A
Mute Control ID testString	cg "testString"\x0A
PTZ Speed Control ID testString	cg "testString"\x0A
Router Control ID testString	cg "testString"\x0A