Control third-party peripherals via serial port

You can set up your Board, Desk, or Room series device to control third-party peripherals, such as displays, video switches, projectors, or other, via a serial connection. We refer to this as *outbound serial control*.

Outbound serial control is available on our devices, whether they are registered to an on-premises service, to the Webex cloud service, or used with Microsoft Teams Room.

There is a separate set of configurations and commands that apply to *outbound* serial control compared to the configurations and commands that can be sent **to** the device via serial port (*inbound*).

Requirements and limitations

You must connect to a USB-A port on the device using a USB-to-Serial dongle. Cisco has mainly tested this feature using dongles with the FTDI chipsets; in general, other USB-to-Serial dongles should also work.

You cannot use the built-in serial port on Codec Pro based devices or the micro-USB maintenance port on other devices for outbound control; these ports are reserved for inbound control.

These products don't support outbound serial control:

- Board 55/70 and Board 55S/70S/85S
- Room 55 Dual and Room 70
- Room 70 G2 and Room 70 Panorama

Connectionless communication

The communication between the device and peripheral is connectionless, meaning we don't establish a persistent serial port connection between them. The connection is first established when issuing the `xCommand SerialPort PeripheralControl Send` command. The connection is dropped once the command is finished.

Setting up outbound serial port control

Use the following configurations to setup outbound serial control. These configurations have no effect on the regular (inbound) serial port / maintenance port.

```
xConfiguration SerialPort Outbound Mode: <Off, On>
```

- Turn this configuration **On** to allow for serial port outbound control. We allow only one outbound serial connection. Default: Off.

```
xConfiguration SerialPort Outbound Port [1] BaudRate: <9600, 19200,
38400, 57600, 115200>
```

- Set the outbound baud rate (data transmission rate) for the serial port in (bits per second). Default: 115200 bps.

```
xConfiguration SerialPort Outbound Port [1] Description: <String: 0,
512>
```

- You can use this configuration to add text that describes what the serial port is intended to be connected to, for example "Projector". Default: an empty string.

```
xConfiguration SerialPort Outbound Port [1] Parity: <Even, None, Odd>
```

- Choose whether to add a parity bit for the serial data transmission. You can choose between adding **Even** parity, **Odd** parity, or not adding any parity (**None**). Default: None.

These serial connection parameters are not user configurable:

- Data bits: 8
- Stop bits: 1
- Flow control: None

Sending serial data from the device to the peripheral

Use this command to send data to the peripheral:

```
xCommand SerialPort PeripheralControl Send [PortId: <1>]
[ResponseTerminator: <S: 0, 128>] [ResponseTimeout: <50..5000>] Text:
<S: 0, 128>
```

This command sends the data that is specified in the *Text* parameter over the specified *PortId*.

- *Text* (required): The text to send to the peripheral.

  You can add special characters using "\" notation for special characters and "\x{ASCIIHEXCODE}" for hex characters.

  Example: To send a string ending with carriage return and new line, enter "Hello World\r\n" or "Hello World\x0D\x0A".

- *PortId* (optional): The port to send the data over. The default value, and only supported value, is 1.
- *ResponseTerminator* (optional): A character or string that indicates that the rest of the response received from the peripheral will be ignored.

  If a ResponseTerminator is specified, then any response received from the peripheral serial port after the Send command is issued will be buffered. The command will return up-until the first occurrence of the ResponseTerminator character or string.

  Example: If the ResponseTerminator is set to "\n" and the peripheral responds with "Hello to you too\nSomeMoreData\n" then the commands response, `PeripheralControlSendResult Response` will contain "Hello to you too". The rest of the received data is discarded.

  If a ResponseTerminator is not specified, either the complete response will be returned, or it will be cut when the full ResponseTimeout period expires.

The ResponseTerminator parameter is ignored if a ResponseTimeout parameter is not included in the command. The device will not wait and listen for a response at all if the ResponseTimeout is not specified.

- *ResponseTimeout* (optional): The maximum number of milliseconds (ms) to wait for a response from the peripheral. You cannot set this timeout to be more than 5000 ms.

  If a ResponseTimeout is specified, the device listens for a response from the peripheral either the full ResponseTimeout period or until it receives the first occurrence of the character string specified in the ResponseTerminator parameter.

  If a ResponseTimeout is not specified, the device terminates the command session immediately after the text-payload transmission is completed. Any response from the peripheral is discarded.

Command queue

Due to the nature of the serial port, only one command can be sent at a time. The port will be blocked during a command/response session. If you try to call `xCommand SerialPort PeripheralControl Send` while there is already an active session, the command will fail with an error saying the port is already in use. The *ResponseTimeout* parameter of the current command determines whether the session includes a period waiting for a response.

When sending consecutive serial commands from macros you should use synchronous commands (async, await) to ensure correct queuing.