CONNECTIVE PERIPHERALS

Simple Troubleshooting Steps for Connective Peripherals Serial Converters

The USB-Serial converters from Connective Peripherals have been designed to be easy to install and use. In most cases, it is simply a case of setting jumpers/switches, installing the driver (or letting Windows do this automatically), connecting the converter between the USB port and the serial device, and beginning to use your serial device as if it was connected via a real serial port on your PC.

However, if you encounter issues, the steps shown in this article will help you to troubleshoot the issue and to confirm if your Connective Peripherals converter is working correctly. Having the results from these tests available will also help us to resolve your issue should you need to contact Connective Peripherals Support.

USB Cabling and Power

The USB cabling and power are important in ensuring that the converter connects to your computer reliably:

- Ensure a good quality USB cable is used,
- Try a different USB cable,
- Try the cable supplied with the converter (if using your own USB cable)

Ensure the USB port on the computer is working correctly:

- Try connecting the converter to a different USB port on the same computer,
- Try a different USB device (such as a USB flash drive) on the USB port on the computer to check if the port itself is faulty.

Make sure the converter has enough power:

- If using a USB hub, use a self-powered hub (with its own power supply). Some converters need more than 100mA. You may see a Windows message pop up if the USB port is not able to provide enough power,
- Always use the correct power adapter supplied (if your model of converter requires external power),
- Some converters have a power output on the DB9/Terminal block for the serial device; if you use this, try disconnecting this line in case the external device is drawing too much current.

Status LEDs

If your model of converter has an LED to indicate power (normally an amber colour) then check if this is illuminated. This indicates that the converter is being powered.

For converters which are powered by the USB cable (no external PSU), the power LED may not fully illuminate until the driver has completed installing. For converters which have an external power supply, check that the power supply is connected and is switched on at the mains plug.

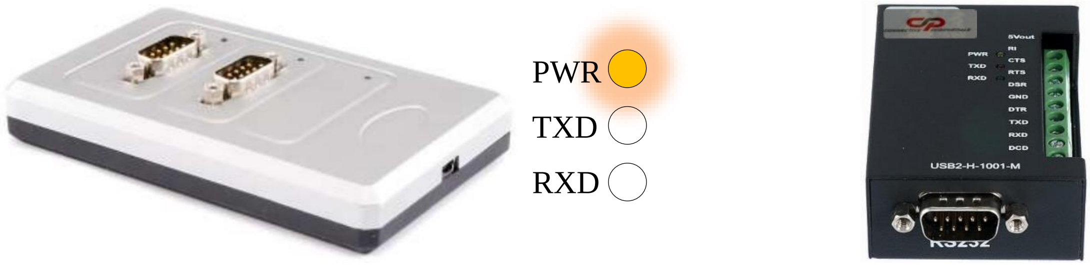


Figure 1 Power indication

Note that some models only have Tx/Rx LEDs or may have no LEDs. Check the datasheet for your model of converter to confirm which LED indicators it has.

The Tx and Rx LEDs (usually red or green) may blink just after the converter is connected to the computer (or as the computer is starting up if the converter is connected whilst the computer is off). They will then remain off whilst the converter is idle, only blinking when the converter is transferring data in the associated direction.

Device Manager

In Device Manager, check that your converter shows up correctly.
You can access the Device Manager in Windows through the Settings menu or by searching for “Device Manager” as shown in **Figure 2**.

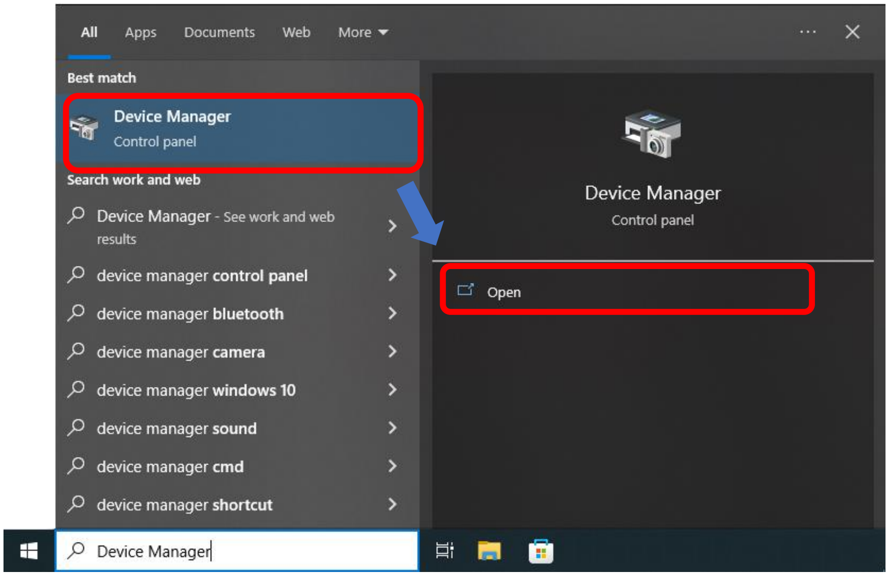
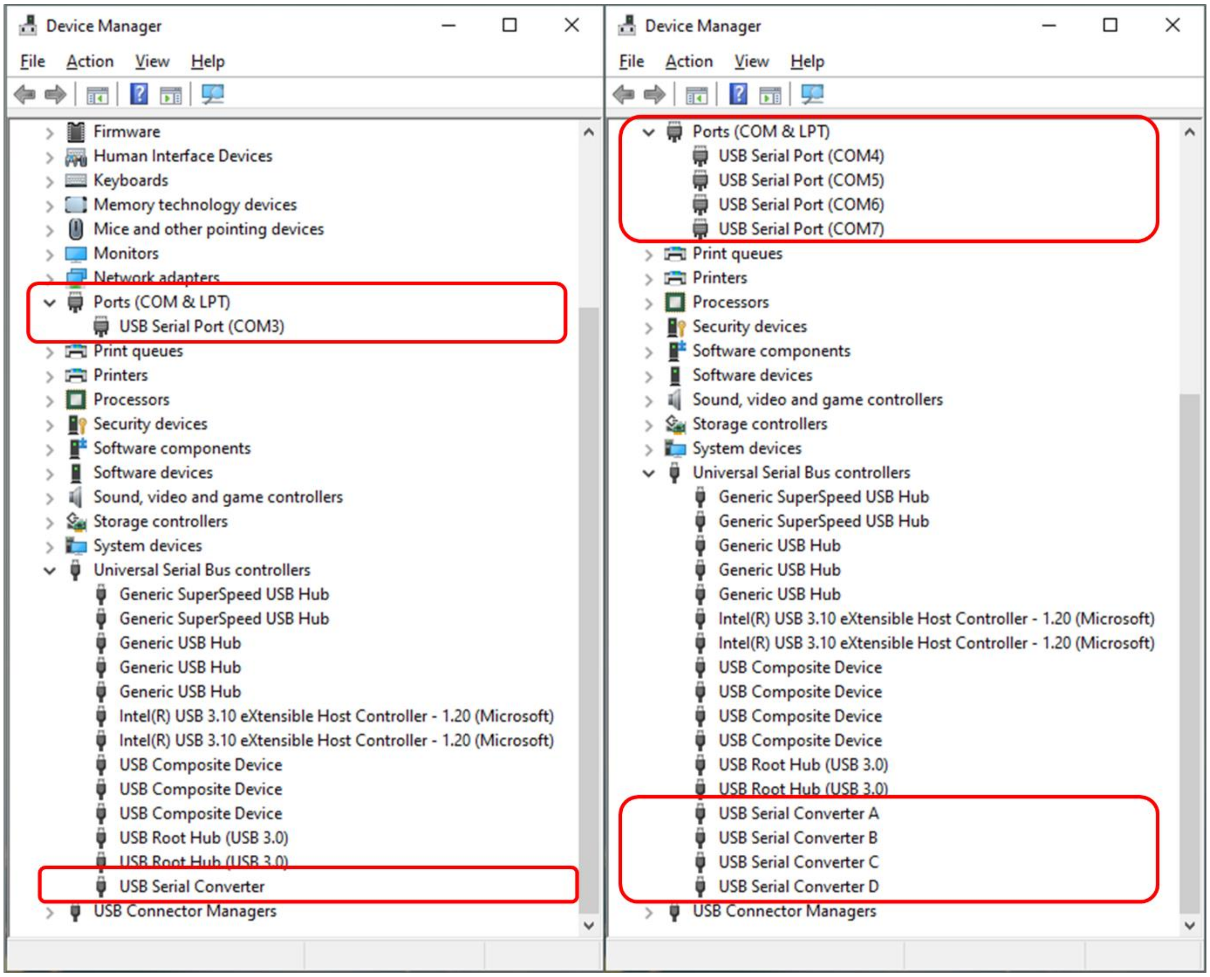


Figure 2 Opening Device Manager

Once the Device Manager is open, there should be an entry under Universal Serial Bus Controllers and an entry under Ports (COM & LPT) for each port on the converter.

For example, a single port converter will have one entry under each section (see the left-hand side of **Figure 3**) whilst a 4-port converter will have four entries under each section (see the right-hand side of **Figure 3**).



- Firmware
- Human Interface Devices
- Keyboards
- Memory technology devices
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - USB Serial Port (COM3)
- Print queues
- Printers
- Processors
- Security devices
- Software components
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers
  - Generic SuperSpeed USB Hub
  - Generic SuperSpeed USB Hub
  - Generic USB Hub
  - Generic USB Hub
  - Generic USB Hub
  - Intel(R) USB 3.10 eXtensible Host Controller - 1.20 (Microsoft)
  - Intel(R) USB 3.10 eXtensible Host Controller - 1.20 (Microsoft)
  - USB Composite Device
  - USB Composite Device
  - USB Composite Device
  - USB Root Hub (USB 3.0)
  - USB Root Hub (USB 3.0)
  - USB Serial Converter
- USB Connector Managers

- Ports (COM & LPT)
  - USB Serial Port (COM4)
  - USB Serial Port (COM5)
  - USB Serial Port (COM6)
  - USB Serial Port (COM7)
- Print queues
- Printers
- Processors
- Security devices
- Software components
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers
  - Generic SuperSpeed USB Hub
  - Generic SuperSpeed USB Hub
  - Generic USB Hub
  - Generic USB Hub
  - Generic USB Hub
  - Intel(R) USB 3.10 eXtensible Host Controller - 1.20 (Microsoft)
  - Intel(R) USB 3.10 eXtensible Host Controller - 1.20 (Microsoft)
  - USB Composite Device
  - USB Composite Device
  - USB Composite Device
  - USB Composite Device
  - USB Root Hub (USB 3.0)
  - USB Root Hub (USB 3.0)
  - USB Serial Converter A
  - USB Serial Converter B
  - USB Serial Converter C
  - USB Serial Converter D
- USB Connector Managers

Figure 3 Device Manager showing 1-port and 4-port converters

If you have connected a large number of FTDI-based converters in the past, then you may find your COM port numbers assigned are very high (e.g. COM 100). One example is in an application where you are servicing a large number of devices with FTDI chips inside. If you no longer use some of the converters, you can remove the driver and un-allocate these COM ports so that newly connected converters can use these lower COM numbers. You can use the re-install procedure in the Connective Peripherals Driver Guide to do a clean install of the driver and remove these older COM assignments.

USBview

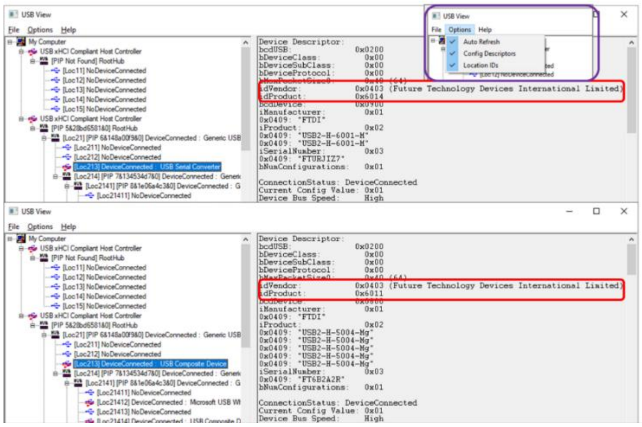
If your converter does not show up in Device Manager as detailed above, or if it shows up in a different section such as 'Other Devices' then you can use the USBview program on a Windows computer to check that the converter is being detected correctly by the computer. USBview is useful as it shows the device details even if the driver is not installed, and so can help to verify that the hardware (converter, USB port, cabling, power) are all working correctly.

- Download the utility from this link: USBView
- Unzip the downloaded file and locate the usbview.exe file in the unzipped folder
- Run the USBView utility
- Connect your converter to the USB port of the PC
- Open the Options menu and tick all of the options (see top-right of
- **Figure 4**)
- Press F5 to refresh the screen
- Find your device in the device tree (e.g. USB Serial Converter) and select it
- Check the details in the right-hand pane as shown in
- **Figure 4**

Connective Peripherals converters use the FTDI chipset and so should have the following values.

- idVendor of 0x0403 and
- idProduct of 0x6001 or 0x6014 or 0x6015 or 0x6010 or 0x6011 depending on the model of converter

If the converter does not appear at all in the left-hand device tree, then it is not being detected by the computer at all. If the converter does appear but the values in the right-hand pane are all 0, the converter is detected but is unable to report its details. In both cases this may indicate an issue with the USB cabling, USB port, or power, or the converter may be faulty.



```
My Computer
  USB xHCI Compliant Host Controller
    [PIP Not Found] RootHub
      [Loc11] NoDeviceConnected
      [Loc12] NoDeviceConnected
      [Loc13] NoDeviceConnected
      [Loc14] NoDeviceConnected
      [Loc15] NoDeviceConnected
  USB xHCI Compliant Host Controller
    [PIP 5&28bd6581&0] RootHub
      [Loc21] [PIP 6&148a00f9&0] DeviceConnected : Generic USB
        [Loc211] NoDeviceConnected
        [Loc212] NoDeviceConnected
        [Loc213] DeviceConnected : USB Serial Converter
        [Loc214] [PIP 7&134534d7&0] DeviceConnected : Generi
          [Loc2141] [PIP 8&1e06a4c3&0] DeviceConnected : G
            [Loc21411] NoDeviceConnected
```

```
Device Descriptor:
bcdUSB:             0x0200
bDeviceClass:         0x00
bDeviceSubClass:      0x00
bDeviceProtocol:      0x00
bMaxPacketSize0:      0x40 (64)
idVendor:           0x0403 (Future Technology Devices International Limited)
idProduct:          0x6014
bcdDevice:          0x0900
iManufacturer:        0x01
0x0409: "FTDI"
iProduct:             0x02
0x0409: "USB2-H-6001-M"
0x0409: "USB2-H-6001-M"
iSerialNumber:        0x03
0x0409: "FTURJIZ7"
bNumConfigurations:   0x01

ConnectionStatus: DeviceConnected
Current Config Value: 0x01
Device Bus Speed:     High
```

```
My Computer
  USB xHCI Compliant Host Controller
    [PIP Not Found] RootHub
      [Loc11] NoDeviceConnected
      [Loc12] NoDeviceConnected
      [Loc13] NoDeviceConnected
      [Loc14] NoDeviceConnected
      [Loc15] NoDeviceConnected
  USB xHCI Compliant Host Controller
    [PIP 5&28bd6581&0] RootHub
      [Loc21] [PIP 6&148a00f9&0] DeviceConnected : Generic USB
        [Loc211] NoDeviceConnected
        [Loc212] NoDeviceConnected
        [Loc213] DeviceConnected : USB Composite Device
        [Loc214] [PIP 7&134534d7&0] DeviceConnected : Generi
          [Loc2141] [PIP 8&1e06a4c3&0] DeviceConnected : G
            [Loc21411] NoDeviceConnected
            [Loc21412] DeviceConnected : Microsoft USB Wi
            [Loc21413] NoDeviceConnected
            [Loc21414] DeviceConnected : USB Composite D
```

```
Device Descriptor:
bcdUSB:             0x0200
bDeviceClass:         0x00
bDeviceSubClass:      0x00
bDeviceProtocol:      0x00
bMaxPacketSize0:      0x40 (64)
idVendor:           0x0403 (Future Technology Devices International Limited)
idProduct:          0x6011
bcdDevice:          0x0800
iManufacturer:        0x01
0x0409: "FTDI"
iProduct:             0x02
0x0409: "USB2-H-5004-Mg"
0x0409: "USB2-H-5004-Mg"
0x0409: "USB2-H-5004-Mg"
0x0409: "USB2-H-5004-Mg"
0x0409: "USB2-H-5004-Mg"
iSerialNumber:        0x03
0x0409: "FT6B2A2R"
bNumConfigurations:   0x01

ConnectionStatus: DeviceConnected
Current Config Value: 0x01
Device Bus Speed:     High
```

Figure 4 USBView showing a 1-port and a 4-port converter

Driver Re-Install

If the previous steps have all shown the converter to be connected correctly, check that the latest driver is installed. The procedure for removing the current driver and installing the latest version can be found in the Connective Peripherals Driver Guide.

Loopback Test

If the steps above were successful, the USB side of the converter appears to be operating correctly. To cover the serial side, a simple loopback test can be used. This test makes sure that both the USB-Serial converter chip and the line driver/receiver chip in the converter are able to send and receive data.
In this section, we show two ways to do the loopback test. This depends on whether you have only one port available or whether you have two ports available (either a second converter of the same type or a converter with multiple ports). You can use a terminal program such as PuTTY, HTerm, Coolterm, or many others. *Connective Peripherals have no affiliation to any terminal programs or the organisations who provide them and accept no liability whatsoever for the use of third-party websites and software.*

To avoid the risk of damage, it is important to check the user guide from the CP website for your model of converter to ensure that the suggested loopback connections match the pinout. Check the signal assigned to each pin. Make sure that any optional 5V output on any of these pins is disabled. If in doubt check with Connective Peripherals support before doing the loopback test.

When doing a loopback test, there are a few things to note about the terminal:

1. Some terminals (including PuTTY) send whilst you type and others may require you to press Enter to send what you have typed.
2. Some terminals have a 'Local Echo' setting
    - When ON, you will see the characters as you type them and then a second copy which is returned through the loopback.
    - When OFF, the terminal does not display characters as you type them. You will only see characters in the terminal window which have been received via the loopback.
3. In RS485, some devices have an RS485 Echo on/off option on the converter itself (via a DIP switch for example).
    - When ON, when you type data into the terminal, both the sending and the receiving terminal will receive the same characters.
    - When OFF (and assuming the terminal's own echo is also off as per note 2 above) then you will not see an echo of what you type on the sending terminal. You will only see the data received on the terminal (as shown in **Figure 12**)

One useful way to check if the characters observed are from the echo or from the loopback is to temporarily unplug your loopback plug from the DB9 of the converter and repeat the test of typing some characters. If the characters typed are still displayed by the sending terminal, the terminal itself or the converter appear to have echo enabled.

Loopback using 1 Port

If you have a single-port converter, you can create a small adapter to loop data back to the same port. This could be constructed using a female DB-9 connector shell with solder bucket terminals, to connect directly to the male DB-9 on the converter. Some converters also have terminal blocks and so short wires could be used to create a loopback there.

See example loopback connections for your protocol (RS232, RS422, RS485) in **Figure 5**, **Figure 6** and **Figure 7** below.

For RS485, when using a single port, the technique differs slightly. No loopback adapter is used as the lines are bi-directional and so there is no Tx to loop back to Rx. Ideally the Loopback using 2 Ports described in the next section would be used. Or if you have only one converter port, you can enable the echo feature (this may be by a DIP switch, jumper or using a utility if using the USB2-F-5001). Ensure to disconnect any RS485 devices and leave the serial connector on the converter unconnected. Also, ensure to restore the echo setting to your required setting afterwards.

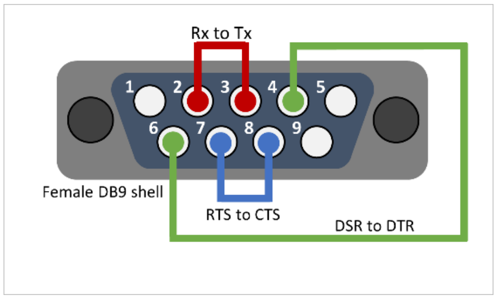


Figure 5 RS232 loopback (1-port)

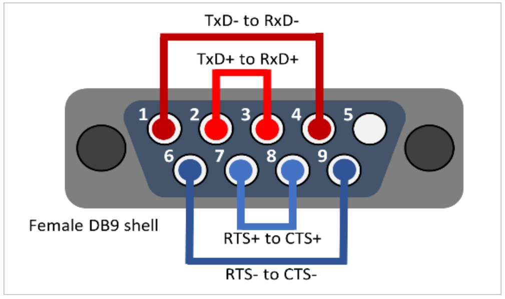


Figure 6 RS422 loopback (1-port)

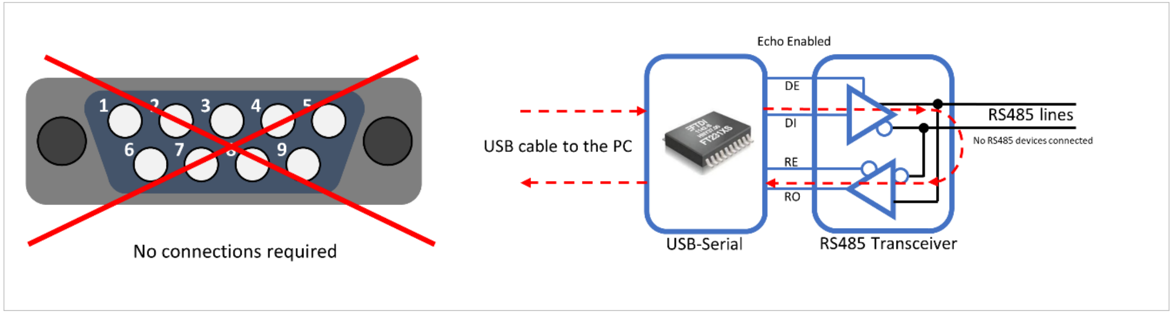
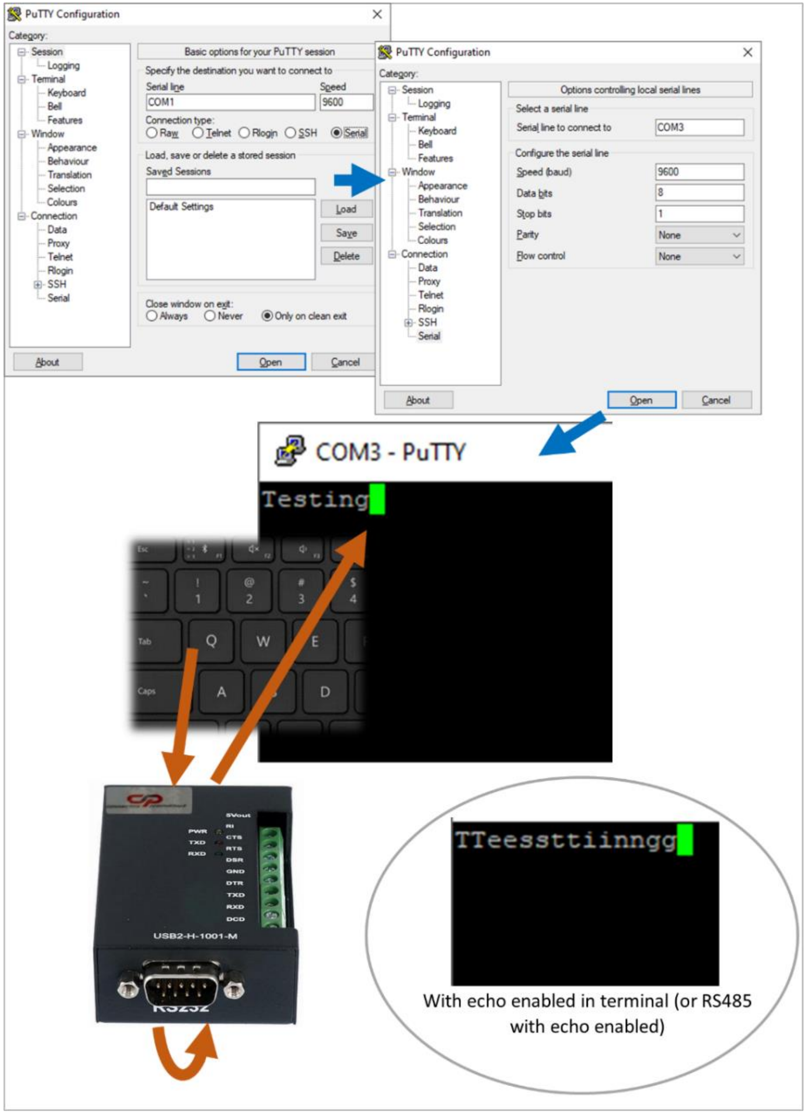


Figure 7 RS485 loopback information (1-port)

Open the terminal program. Set the baud rate to a typical value such as 9600 and set flow control to NONE. Open the port (you may need to check Device Manager to see which COM number was assigned). See **Figure 8** below.

Type some characters and you should see the Tx and Rx LEDs blink (if available on your converter). The characters typed into the terminal will travel through the loopback and appear back in the terminal.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM1

Speed: 9600

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

PuTTY Configuration

Options controlling local serial lines

Select a serial line

Serial line to connect to: COM3

Configure the serial line

Speed (baud): 9600

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

COM3 - PuTTY

```
Testing
```

```
TTeessttiinngg
```

With echo enabled in terminal (or RS485 with echo enabled)

Figure 8 Loopback using single port

You can also try higher baud rates and other handshaking options (for RS232 and RS422 provided that the model of converter has the appropriate signals on the connector) but a basic 9600 Baud test with no handshaking is recommended first.

Loopback using 2 Ports

You can create a small cable harness to loop data between two ports. See the connections for your protocol (RS232, RS422, RS485) in **Figure 9**, **Figure 10** and **Figure 11** below. The harness could be constructed using female DB-9 connector shells with solder bucket terminals, in order to connect directly to the male DB-9 on the converter. Some converters also have terminal blocks and so short wires could be used to connect the two ports to create a loopback there.

Connect one end of the harness to the port which you want to check and the other end to another converter of the same type or a working port on a multi-port converter.

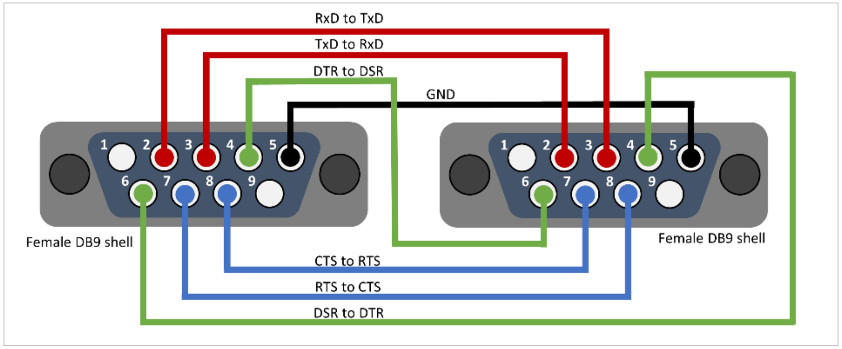


Figure 9 RS232 loopback harness using 2 ports

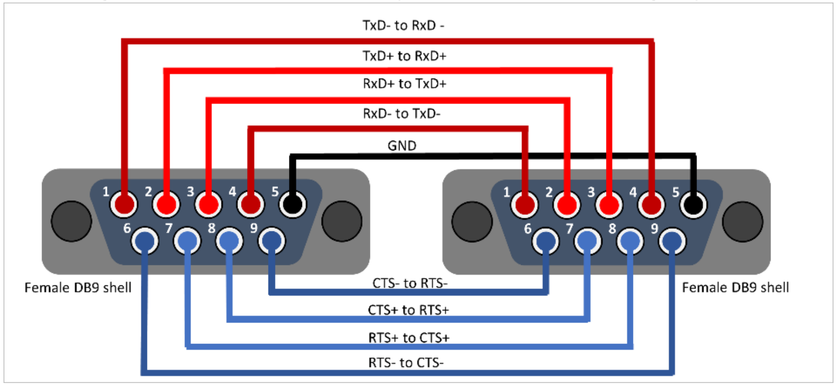


Figure 10 RS422 loopback harness using 2 ports

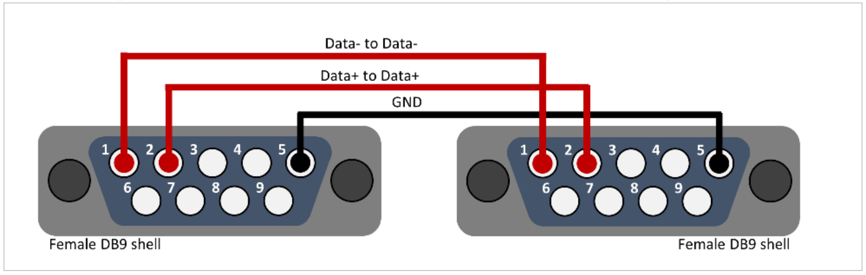
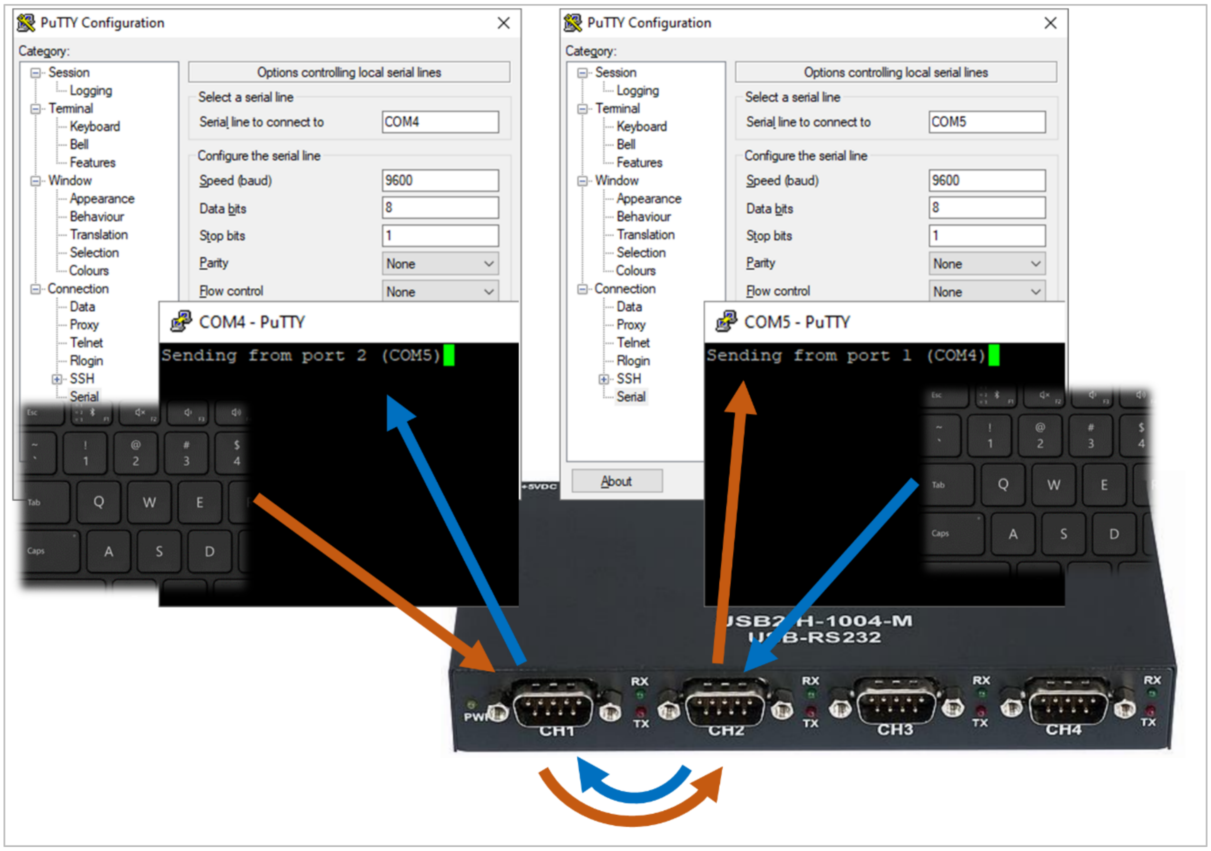


Figure 11 RS485 loopback harness using 2 ports

Open two instances of the terminal program. Set the baud rate in both instances to a typical value such as 9600 and set flow control to NONE. The settings must be the same for both instances. Open the Port Under Test in the first instance of the terminal and open the other port in the second instance. You may need to check Device Manager to see which COM number was assigned.

Type some characters into the Port Under Test terminal (for example COM4 in **Figure 12**). You should see the Tx LED blink on the associated port of the converter and the Rx LED will blink on the other port which has the harness connected (for example COM5 in **Figure 12**). The characters typed into the first terminal (COM4) will travel through the test harness and appear in the second terminal (COM5).

Then repeat the same in the reverse direction. Type some characters into the second terminal (COM5). You should see the Tx LED blink on the associated port and the Rx LED blink on the port to which it is connected by the test harness (COM4). The characters typed into the second terminal (COM5) will travel through the test harness and appear in the Port under Test terminal (COM4).



PuTTY Configuration

Options controlling local serial lines

Select a serial line

Serial line to connect to: COM4

Configure the serial line

Speed (baud): 9600

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

COM4 - PuTTY

```
Sending from port 2 (COM5)
```

PuTTY Configuration

Options controlling local serial lines

Select a serial line

Serial line to connect to: COM5

Configure the serial line

Speed (baud): 9600

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

COM5 - PuTTY

```
Sending from port 1 (COM4)
```

Figure 12 **Loopback example using 2 ports**

You can also try higher baud rates and other handshaking options (for RS232 and RS422 provided that the model of converter has the appropriate signals on the connector) but a basic 9600 Baud test with no handshaking is recommended first.

Application-Related Issues

If the earlier steps including the loopback test are successful, the converter itself is working correctly. If the converter still does not work when used in the intended application, it may be a mismatch in the serial cabling or the applications itself.

It is important to consult the documentation of your serial device to check what kind of communication it uses as well as checking details such as the pinout and serial settings. When detecting and using a connection with a serial device, some applications open the port and listen for serial data from the device whilst others send commands and await replies. Knowing what kind of communication messages are used when opening/configuring the device and during ongoing communication is very helpful in understanding and narrowing down the problem.

Serial Cabling

Does the type of serial protocol match between the Connective Peripherals converter and the attached serial device? There are many different types of serial communication, each of which has different voltage levels and signalling. Each of these protocols requires a different type of line driver/receiver in the converter and uses different signal polarities, and so it is important that the converter type matches the type of serial connection for correct operation and to avoid damage:

- RS232 is full-duplex, with lines operating over a positive and negative voltage range: Logic 1 is a negative voltage e.g. -15V to -3V and Logic 0 is a positive voltage e.g. 3V to 15V
- RS485 is half-duplex with Data+ and Data- lines which operate differentially allowing RS485 to operate over longer distances. Many devices may share the same data lines but only one should transmit at a time.
- RS422 is full-duplex (like RS232) allowing data to be sent and received simultaneously, but uses differential lines (like RS485)
- UART is full-duplex and uses logic level signals: Logic 1 is 5V or 3.3V whilst Logic 0 is 0V. As it uses low voltages and is not differential, it is suitable for shorter distances. This is often used for connecting the UART of a Microcontroller to a computer over USB.

Is the cable of the correct type? This could include Impedance, Length, and Shielding / Twisted Pairs
Is the pinout of the cabling correct? Check the manual of the Connective Peripherals converter and of the attached serial device to ensure you have linked the correct signals. For example, on RS232, the TxD output of the Connective Peripherals converter should go to the RxD input of the serial device etc.

- Consult the manual of the serial device to confirm which signals are inputs and which are outputs Some manufacturers use non-standard terminology for signal names. For example, check that RxD on your serial device is the input.
- For RS485, check which signals (A or B) are Data + and Data- as the naming conventions can vary.

On RS485, does the RS485 line need bias resistors added? Seeing the expected messages being transferred but with extra 0x00 bytes at the beginning or end can be a sign that biasing is needed. Some serial devices do not have a true failsafe feature which ensures that they see the data lines as idle when they are floating (when no device is driving the bus). These devices without true failsafe may see spurious data when the lines are floating unless biasing is provided to pull the lines to a known state when the bus is idle. However, check the user guide for your RS485 devices to ensure that biasing is not already present before adding resistors to avoid the risk of overloading the bus. Many converters in the Connective Peripherals range have a jumper or switch to enable biasing resistors.

Is the termination used in the correct places in RS485/RS422? Termination is normally used in a maximum of two places, at each end of the bus. Having too many termination resistors can overload the line drivers, causing damage or causing the voltage levels of the lines to be below spec.

Is the attached serial device powered correctly? Some serial devices (such as in-line opto-isolator adapters and RS232-RS485 converters) use power from the signal lines instead of having their own power supply. The modern low-power transceivers used in most serial equipment including the Connective Peripherals converters is not designed to provide this power. If you are connecting a device to the Connective Peripherals converter which tries to take its power from the signal lines, it may not get sufficient current, and it may also cause the signal line voltages to droop.

- Try using the external power supply for your device instead to avoid this issue.
- Connective Peripherals have converters with built-in optical isolation and RS232-RS485 converters which have their own power supply. Visit our website at https://connectiveperipherals.com or contact our Sales team at sales@connectiveperipherals.com if your isolator or RS232-RS485/422 unit which you are attaching to the USB-Serial converter is unsuitable.

Do you have issues with grounding and ground loops? If you use your device in an application where the controlling computer and the serial peripheral have different ground potentials or are exposed to spikes and surges, this can cause communication problems

- Consider an opto-isolated converter which provides a barrier between the PC and the serial device, within the USB-serial converter.
- Visit our website at https://connectiveperipherals.com or contact our Sales team at sales@connectiveperipherals.com for information on our range of isolated converters.

Application Settings

If the cabling is correct and all of the other steps earlier in this guide have worked correctly, but the intended application is still not working, here are some steps that you can check in the actual serial application (terminal or manufacturer-specific program) which is normally used to communicate with your device.

Does your application list the COM port of your converter? If not:

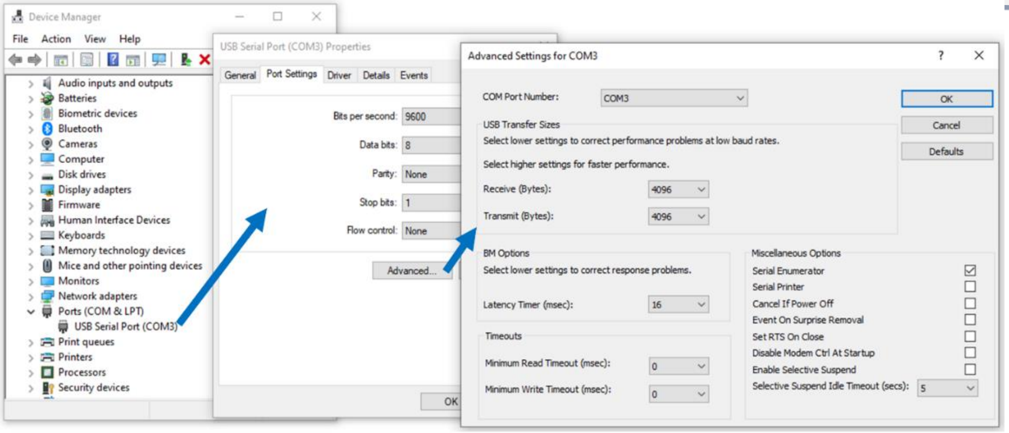
- Some old applications can only use certain COM numbers (e.g. COM 1 to COM9). If this is the case and if your converter was assigned a higher number, try changing the assigned port using the COM Port Number drop-down in the Advanced Driver Options (see **Figure 13**)
- Find out from the manufacturer of your serial device and serial application software about how the software detects the serial device and what the criteria are for it to be detected and listed as available in the application.

Is the port showing as being already open?

- Check that another program does not already have the port open. This technical note from FTDI has some guidance on how to check whether a port is open already.
- Check if you are able to open the port, using the terminal program used in the loopback test earlier in this article, to confirm that it can be opened. Remember to close it after checking this as having the port open in the terminal will prevent your other application from opening it.

Is your peripheral detected by Windows as a mouse? If your serial device sends data into the receive pin of the converter whilst the converter is being connected to the USB port, Windows may mistake it for a serial mouse. You may see the mouse pointer on the screen moving erratically. The driver supports a Windows feature which was used for auto detecting serial mice, whereby Windows looks for devices streaming short packets of data similar to the behaviour of a mouse. If you see this happen:

- You can turn it off by un-ticking the Serial Enumerator option in the Advanced Driver Options (see **Figure 13**). Disconnect your serial device from the converter and un-plug/re-plug the USB cable so that it is no longer detected as a mouse, and then disable the serial enumerator.
- Alternatively, set your serial device so that it only sends data once the converter is connected and ready.



Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

COM Port Number: COM3

USB Transfer Sizes
Select lower settings to correct performance problems at low baud rates.
Select higher settings for faster performance.
Receive (Bytes): 4096
Transmit (Bytes): 4096

BM Options
Select lower settings to correct response problems.
Latency Timer (msec): 16

Timeouts
Minimum Read Timeout (msec): 0
Minimum Write Timeout (msec): 0

Miscellaneous Options
Serial Enumerator ☑
Serial Printer ☐
Cancel If Power Off ☐
Event On Surprise Removal ☐
Set RTS On Close ☐
Disable Modem Ctrl At Startup ☐
Enable Selective Suspend ☐
Selective Suspend Idle Timeout (secs): 5

Figure 13 Advanced Driver Options

If your serial application can open the COM port associated with your converter and tries to communicate with your serial device but fails to communicate:

- Check the baud rate and parity setting that the attached serial device uses
- Check the handshaking mode that the attached serial device uses. RTS/CTS mode is beneficial to ensure no data loss where the serial device supports it. But setting RTS/CTS mode in the application program when the RTS and CTS pins are not connected or used by the attached serial device or cabling will result in data not being transmitted.
- Try setting the serial settings within the application itself. You can set them via Device Manager but the application may override these and so setting them in the actual application will help to avoid this.
- If using RS485, ensure that only one node on the bus communicates at any time to avoid data collisions and conflicts. If designing an RS485-based system where the remote nodes along the bus send data back to the computer, one way is to create a messaging structure so that each node only replies when requested to by the host.
- If your application has a debug window showing the data sent and received, check to see if the messages sent and received are expected. If you use RS485 and see extra unexpected 0x00 bytes at the start or end of a message, you may require biasing to be enabled.

Timing and Protocol factors

Due to the way that the USB protocol works, USB-Serial converters buffer up a number of bytes and transfer them in packets for efficiency rather than as individual bytes. In the FTDI-based Connective Peripherals converters, a timer makes sure that any smaller number of bytes left in the buffer are sent back after a set period (default 16ms). If an application transfers only a few bytes of data at a time, the default settings can result in the data having a slightly higher latency than legacy RS232 ports on older computers. Some older serial software designed for legacy RS232 ports assumes very low latency for individual bytes and so may time out if a reply to a command or request is not received immediately.

- If your serial device does reply (as indicated on the Rx LEDs) but the application program does not see the reply or times out, try setting the Latency Timer (msec) to 2 instead of 16 in the Advanced Driver Options (see **Figure 13**). This will send any replies back more quickly.

Check which protocol, your serial device uses. Some protocols (such as some types of MODBUS) require specific timing between bytes in order to indicate and detect separations between messages. The USB-Serial converters provide accurate bit timing within the data bytes themselves but are not designed to guarantee an accurate timing between bytes (as this is not required for the vast majority of applications). However, there are a few specific protocols which require this.

- If your serial device supports different messaging formats, try a format which uses a different delimiting method (such as ASCII).

Serial applications use a variety of ways to detect that the associated serial peripheral is connected. This can vary from simply being able to open the COM port specified by the user, to sending commands and expecting a reply back from the device. Some applications will open all of the available COM ports in turn, sending a message and awaiting a reply, to auto detect which COM port number the device uses.

- Check with your manufacturer about which technique their software uses as this may help to narrow down where the communication is failing.

Conclusion

In this article we have presented a series of simple tests and checks that you can use if you have problems getting your Connective Peripherals Serial Converters to work. In most cases these will help to confirm the functionality of the converter and to resolve the issue, but if you still see problems please contact Connective Peripherals Technical Support at support@connectiveperipherals.com.

About Connective Peripherals

Connective Peripherals Pte Ltd is a supplier of communications and instrumentation products, with expertise in serial connectivity solutions that are based on USB, CAN and RS232/RS422/RS485 interfaces. The company’s products encompass adapter boards and cables delivering connectivity between USB and serial interfaces, plus award-winning USB based oscilloscopes, data loggers and logic analysers. OEM and ODM design services are also offered.

Headquartered in Singapore, in the heart of the Asia Pacific region, the company’s dedication to quality of service and exceptional technical support has been appreciated by both corporate and individual customers. As a result, its sales have extensively expanded into the global market.

For more information go to https://www.connectiveperipherals.com