# How can I use the PowerDebug X51 to remotely control the target's serial terminal?

2026-08-11 - Comments (0) - Setup / update

Especially if your target is running a rich operating system like Linux, you may need to interact with its serial terminal to properly launch and debug your application. When your target board is located in a remote lab, the **PowerDebug X51** can act as a remote serial interface, allowing you to access the target's terminal from your host PC.

Note

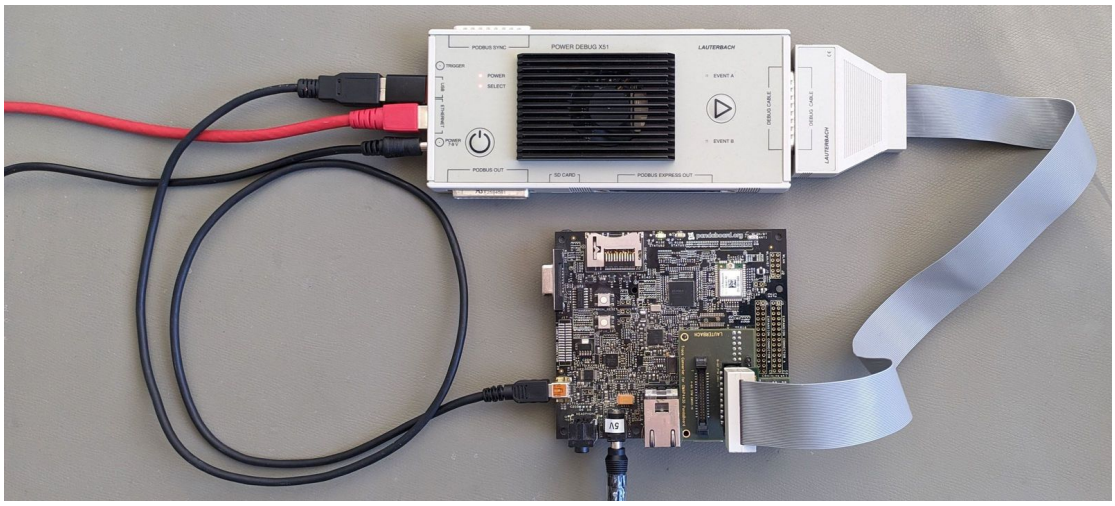
This feature requires TRACE32 release **R.2025.02 or later**.

# Step-by-Step Instructions

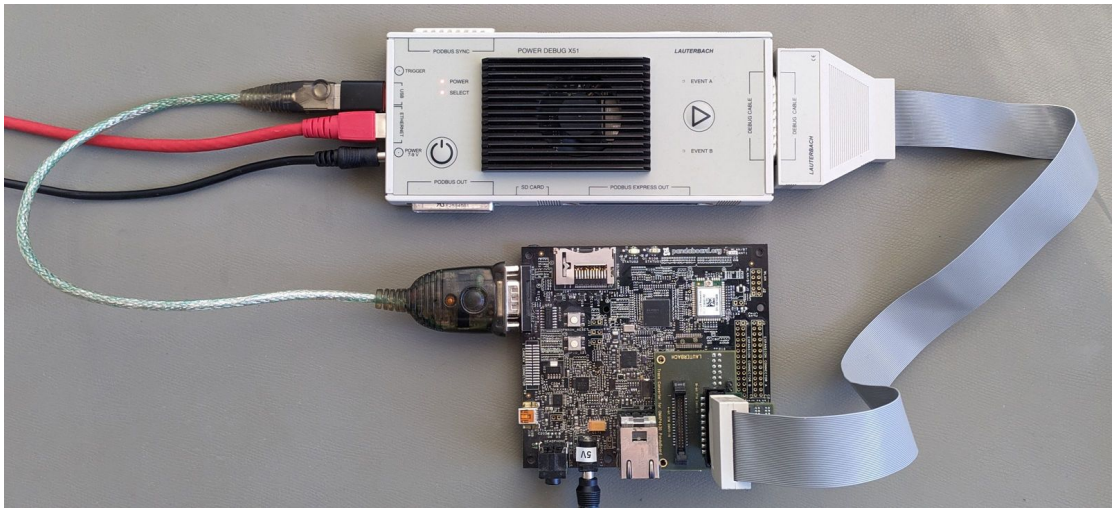
## 1) Connect the Hardware

- Connect the **PowerDebug X51** to your host PC via Ethernet.
- Connect the target’s serial interface to the USB-C port on the **PowerDebug X51**.

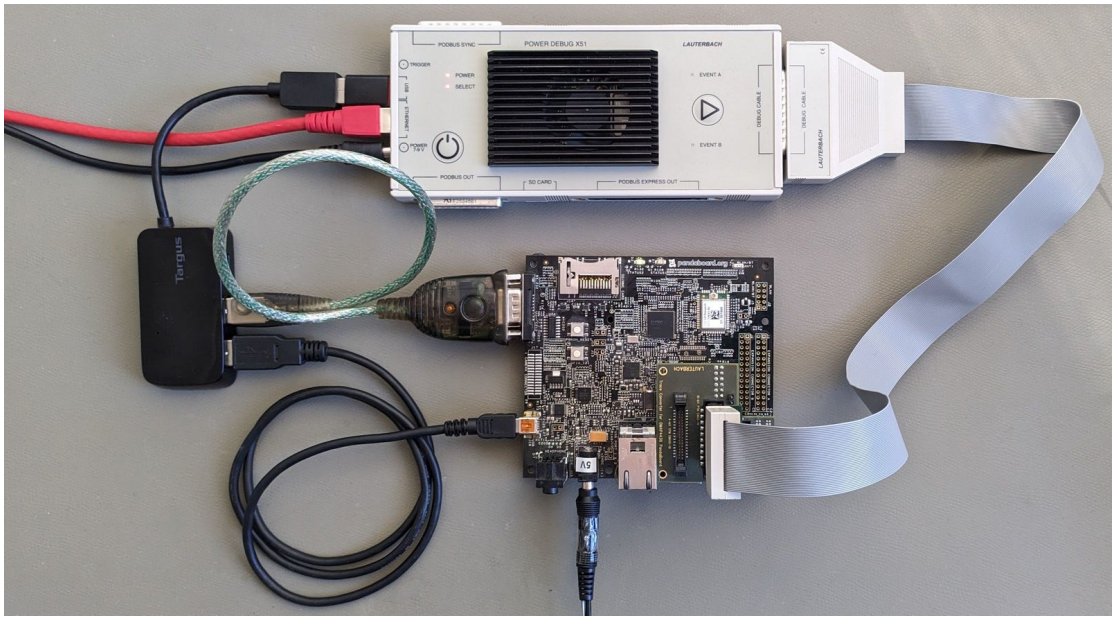
Most modern boards already integrate an RS-232 to USB converter. In that case, a standard USB cable is sufficient:




If your target board only has a classic RS-232 (9-pin) port, you will need a USB-to-Serial adapter:



A USB hub can be used if you want to connect multiple serial ports simultaneously:

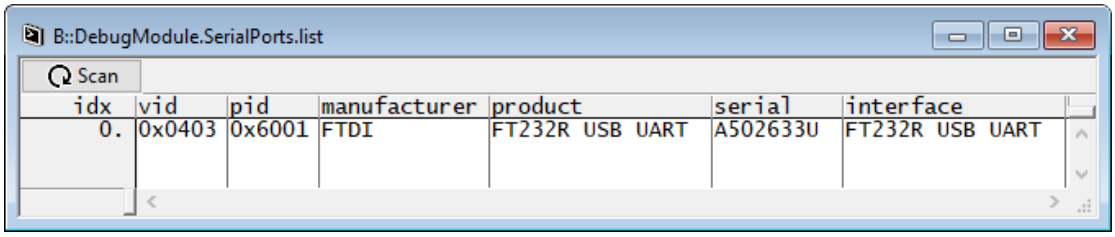


## 2) Detect Serial Ports

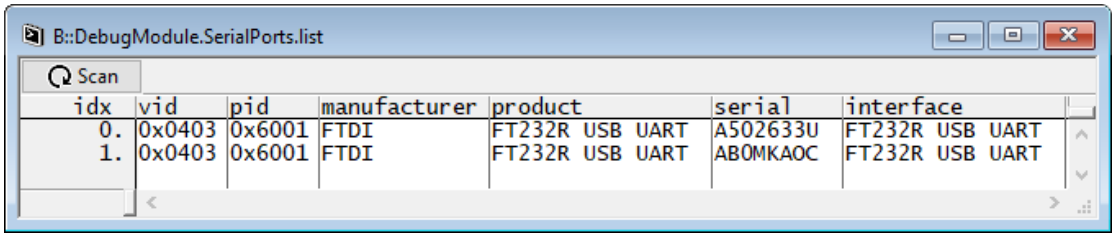
Launch TRACE32 PowerView then the following command to list the connected serial ports:

```
DebugModule.SerialPorts.list
```

This will display all serial ports detected by the **PowerDebug X51**. Note the **index** (idx) of the port you wish to use from the first column.




If you have more than one serial port connected to your PowerDebug X51 (via a USB hub), then you will see more than one device:




Remember the index of the terminal you want to control from the first column labeled “idx”.

If your serial port is not listed:

- Close PowerView and disconnect the **PowerDebug X51** from its power supply.
- Reconnect the power supply, restart PowerView, and run `DebugModule.SerialPorts.list` again.
- Verify that the serial port works when connected directly to a PC.
- Try using a different USB-to-Serial adapter.
- Contact Lauterbach Technical Support if the issue persists.

## 3) Configure the Terminal

Use the index of your serial port to configure the terminal in PowerView.
For example, if the device uses 115200 baud, 8 data bits, no parity, and 1 stop bit:

```
TERM.METHOD #1 DebugModule &i 115200 8 NONE 1STOP NONE
TERM.Mode #1 VT100
TERM.SIZE #1 80. 100. 2000.
```

Note

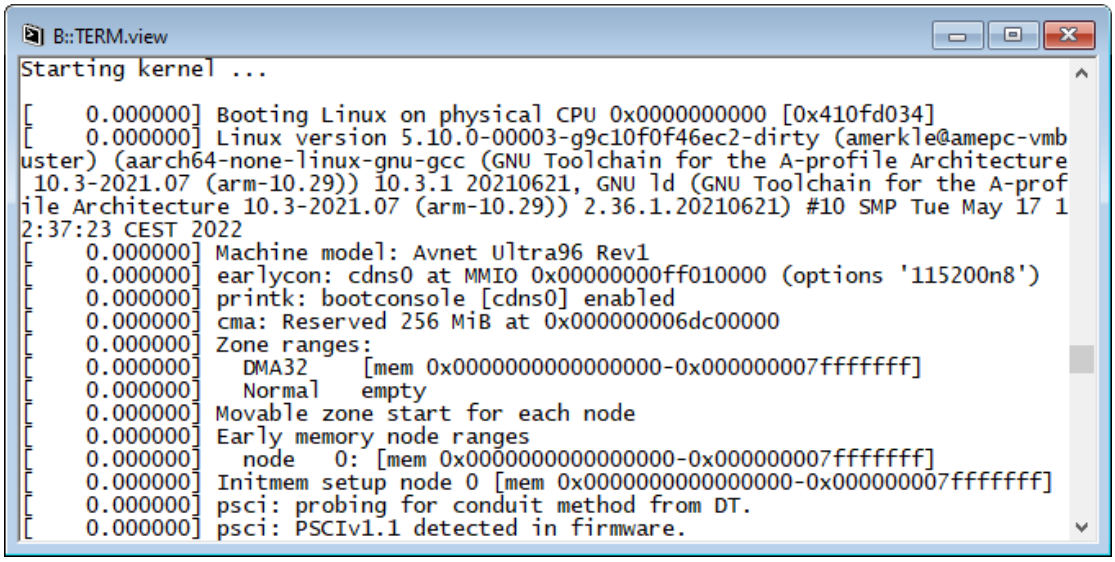
For more details on `TERM.METHOD` see the General Commands Reference Guide T.

## 4) Open the Terminal Window

To display the serial terminal use the command

`TERM.view`

If the target is running Linux, you should now see the boot log or shell prompt in the terminal window:




## Optional: Select a Port by Serial Number

If you are using multiple serial ports, you can automatically select the correct port based on its serial number using a PRACTICE script like the following:

```
DebugModule.SerialPorts.SCAN     // Detect all serial ports
PRIVATE &idx
&idx=0.
WHILE   &idx<DebugModule.SerialPorts.COUNT()
(
    IF "DebugModule.SerialPorts.INFO(&idx,"serial")"=="A502633U"
    (
        // Found port with serial number "A502633U" => show that erminal
        TERM.METHOD #1 DebugModule &idx 115200 8 NONE 1STOP NONE
        TERM.Mode #1 VT100
        TERM.SIZE #1 80. 100. 2000.
        TERM.view
    )
    &idx=&idx+1.
)
```