
Linux[®] Quick Start Guide

1.0 INTRODUCTION

The PCI11414 Quick Start Guide provides a visual view of step-by-step procedures used to exercise each peripheral system available in EVB-PCI11414. The PCI11414 is a PCIe[®] bridge that supports SPI, I²C, GPIO, Ethernet, USB (2.0 and 3.1 Gen2), and UART. The versatile nature of this product makes it ideal for embedded systems and industrial applications.

Detailed information is included in the following sections:

- [Section 2.0, General Considerations](#)
- [Section 3.0, Board Setup](#)
- [Section 4.0, Programming the Board](#)
- [Section 5.0, Verifying Hardware](#)

2.0 GENERAL CONSIDERATIONS

2.1 Required References

The PCI11414 implementor should have the following documents on hand:

- *EVB-PCI11414 Evaluation Board User's Guide*
- *AN5104 MPLab Connect PCI1XXXX Options*

2.2 Hardware Requirements and Setup

All the hardware devices mentioned in this quick start guide can be acquired using their part numbers from www.microchiptdirect.com. Below is a required device:

- EVB-PCI11414 Evaluation Board

2.3 Software Requirements

The following software programs are required for carrying out the procedures in this guide:

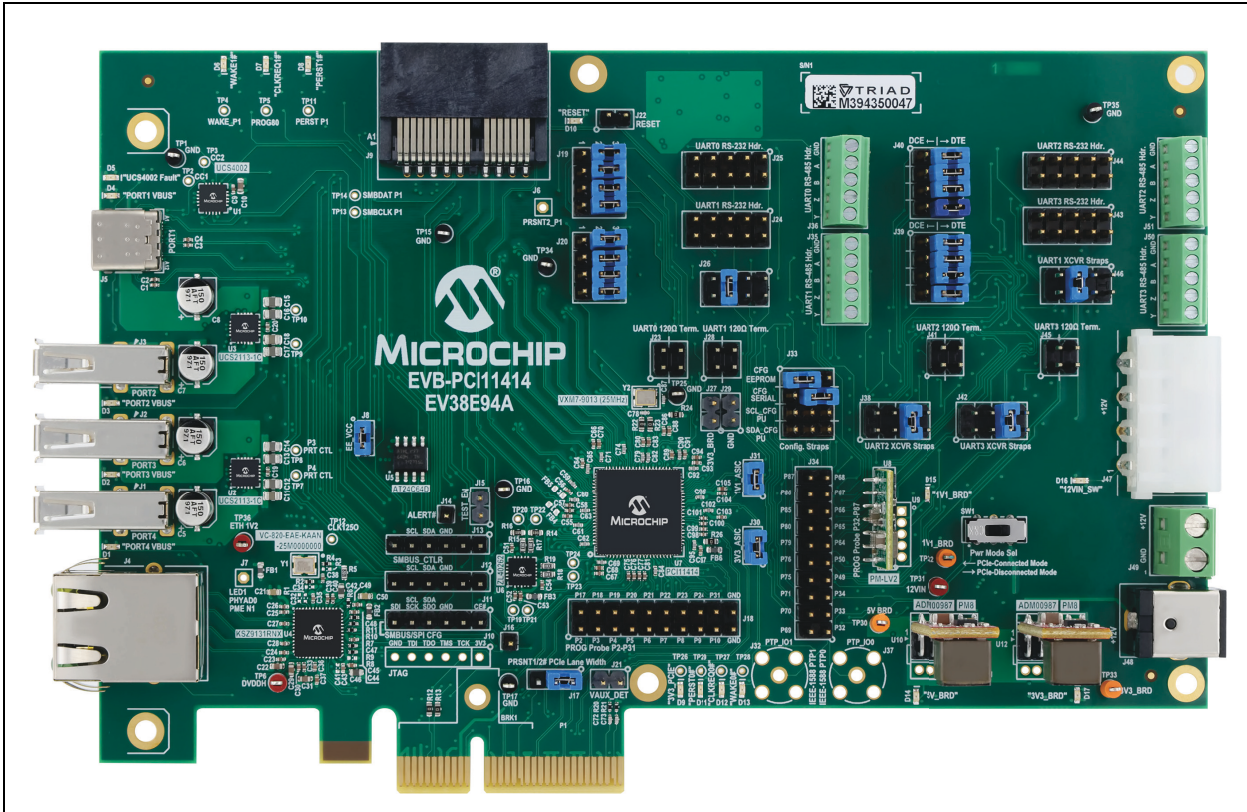
- Drivers. Linux[®] Kernel 6.1.0 is used in conjunction with the drivers below to verify the functionality of the PCI11414. The latest drivers are mainlined into the Linux 6.1.x kernel and can additionally be found on the PCI11414 product page.
 - GP_Driver V1.0.0 or newer
 - I²C_Driver V1.0.0 or newer
 - SPI_Driver V1.0.0 or newer
 - UART_Driver V1.0.0 or newer
- SW Packages
 - Spi_driver_tools (must be compiled):
 - I²C tools
 - Minicom (built into Linux)
 - Pymodslave (can be installed with `pip install pyModSlave`, which only works through Python 3.9)
 - Gmodmaster (can be downloaded from <https://sourceforge.net/projects/qmodmaster/>)

PCI11414

3.0 BOARD SETUP

Figure 3-1 shows the EVB-PCI11414 Evaluation Board.

FIGURE 3-1: EVB-PCI11414 EVALUATION BOARD



4.0 PROGRAMMING THE BOARD

This chapter demonstrates steps for programming the EVB-PCI11414 Evaluation Board.

4.1 Generating a Configuration File

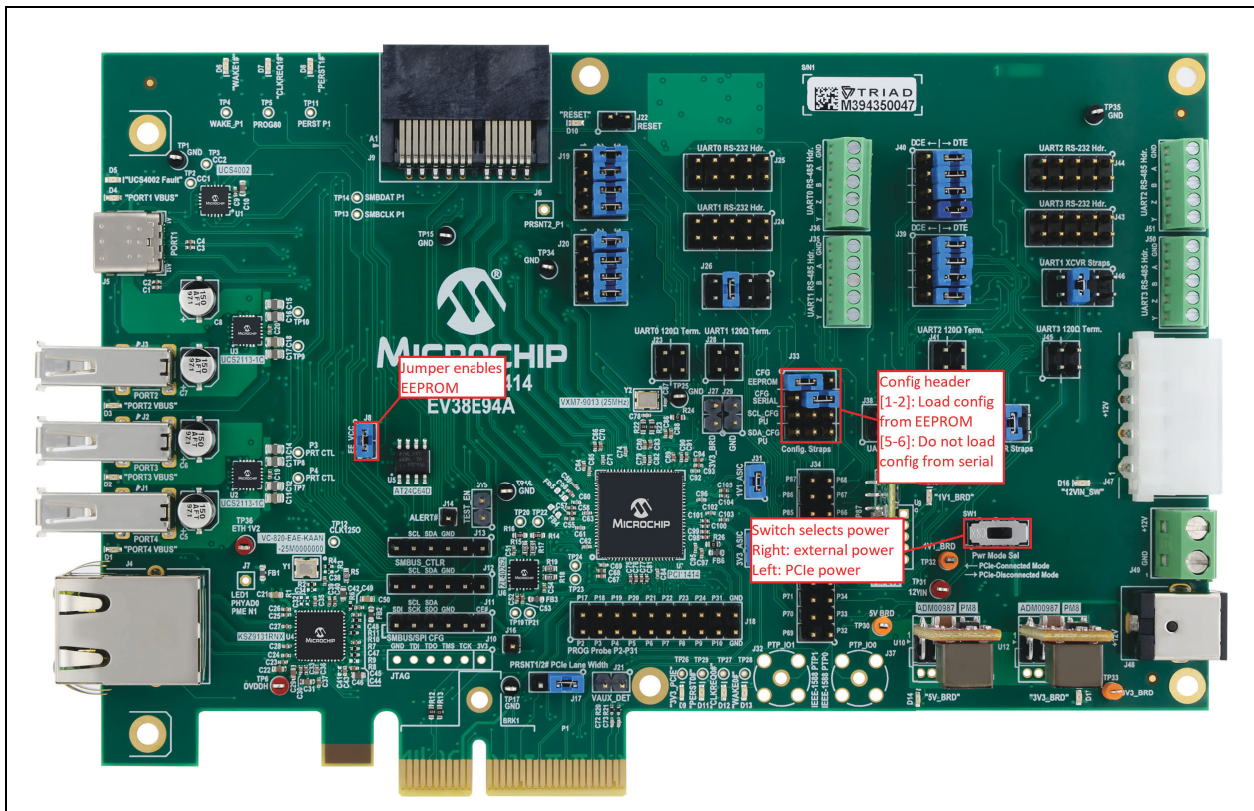
Refer to the *AN5104 MPLab Connect PCI1XXXX Options* for information on how to generate a configuration file for the PCI11414.

4.2 Creating a Hardware Connection

Perform the following procedure to enable EEPROM to be programmed:

1. Verify if there is a jumper on J8. This jumper enables the EEPROM. This allows the EEPROM to be programmed and enables the EVB-PCI11414 to load the configuration from the EEPROM. (See [Figure 4-1](#).)
2. Connect the card either to a PC via PCIe or to a 12V power source. Verify if the position of SW1 is set to match how the board is powered. Switch on the board.

FIGURE 4-1: EVB-PCI11414 JUMPER SETTINGS

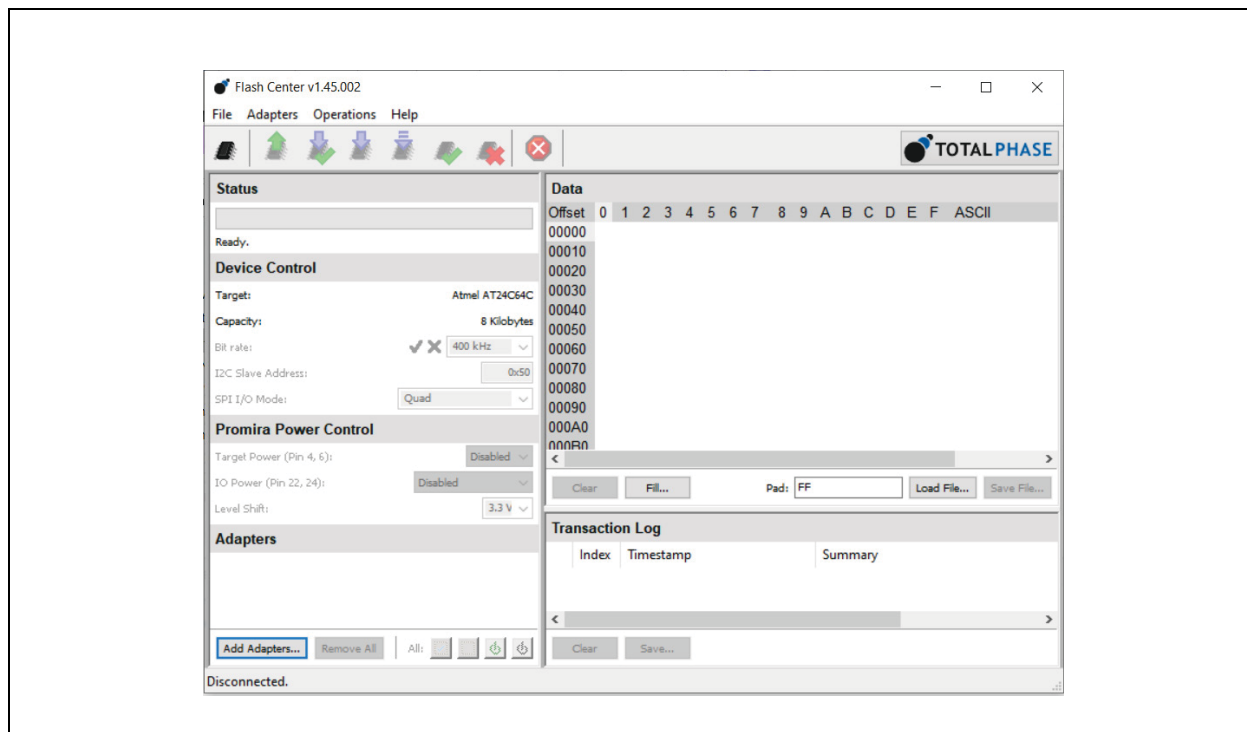


PCI11414

4.3 Programming with Windows®

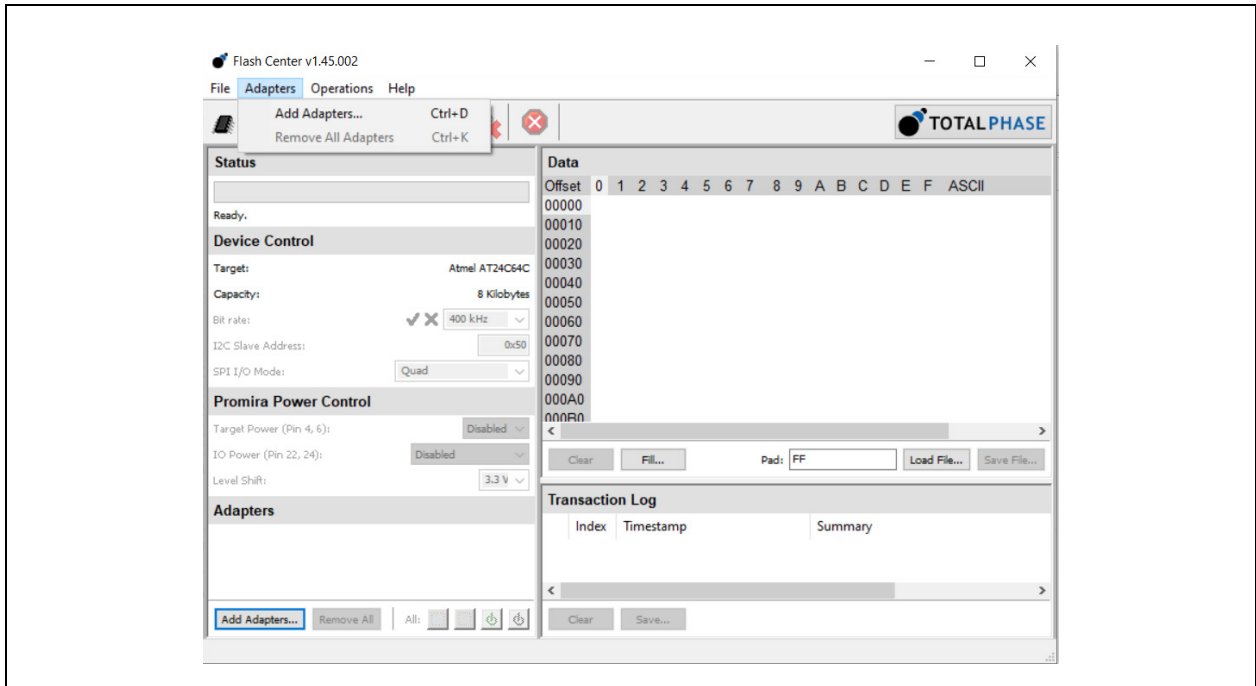
1. Plug in an I²C programmer (an Aardvark is used in this guide) to header J13.
2. Open Flash Center for Windows®. The PCI11414 configuration can be both read and written to from this application. (See [Figure 4-2.](#))

FIGURE 4-2: FLASH CENTER START SCREEN



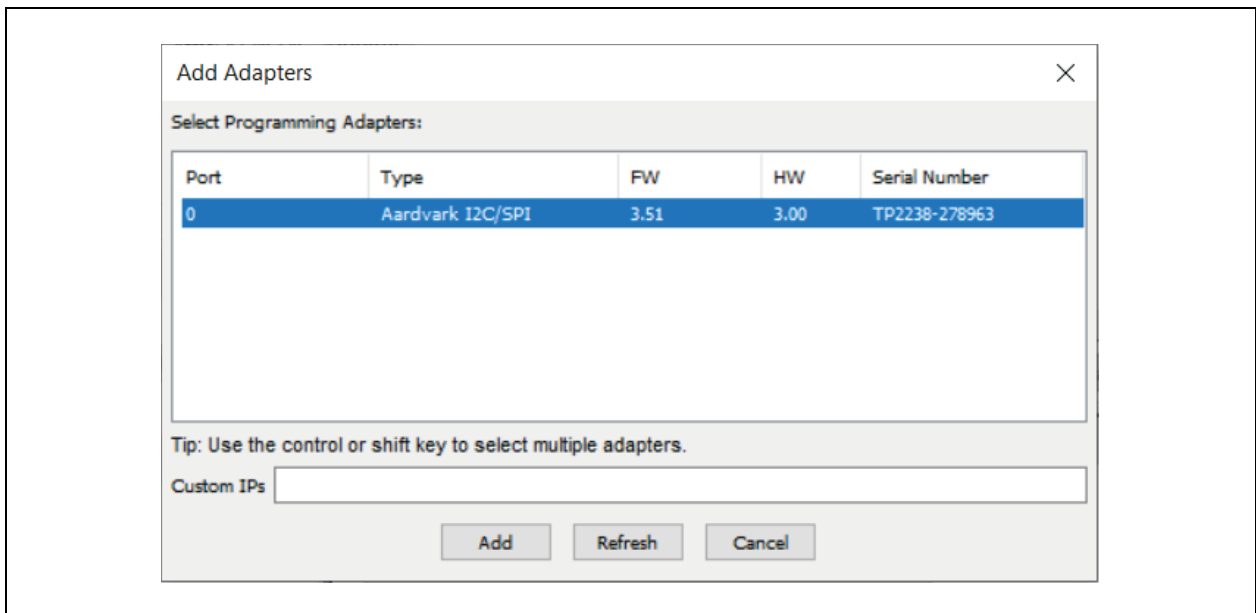
3. Select **Adapters>Add Adapters**. (See [Figure 4-3](#).)

FIGURE 4-3: ADDING ADAPTERS ON FLASH CENTER



4. Plug in the Aardvark and click on **Refresh**. When the device shows up, select it and click on **Add**. (See [Figure 4-4](#).)

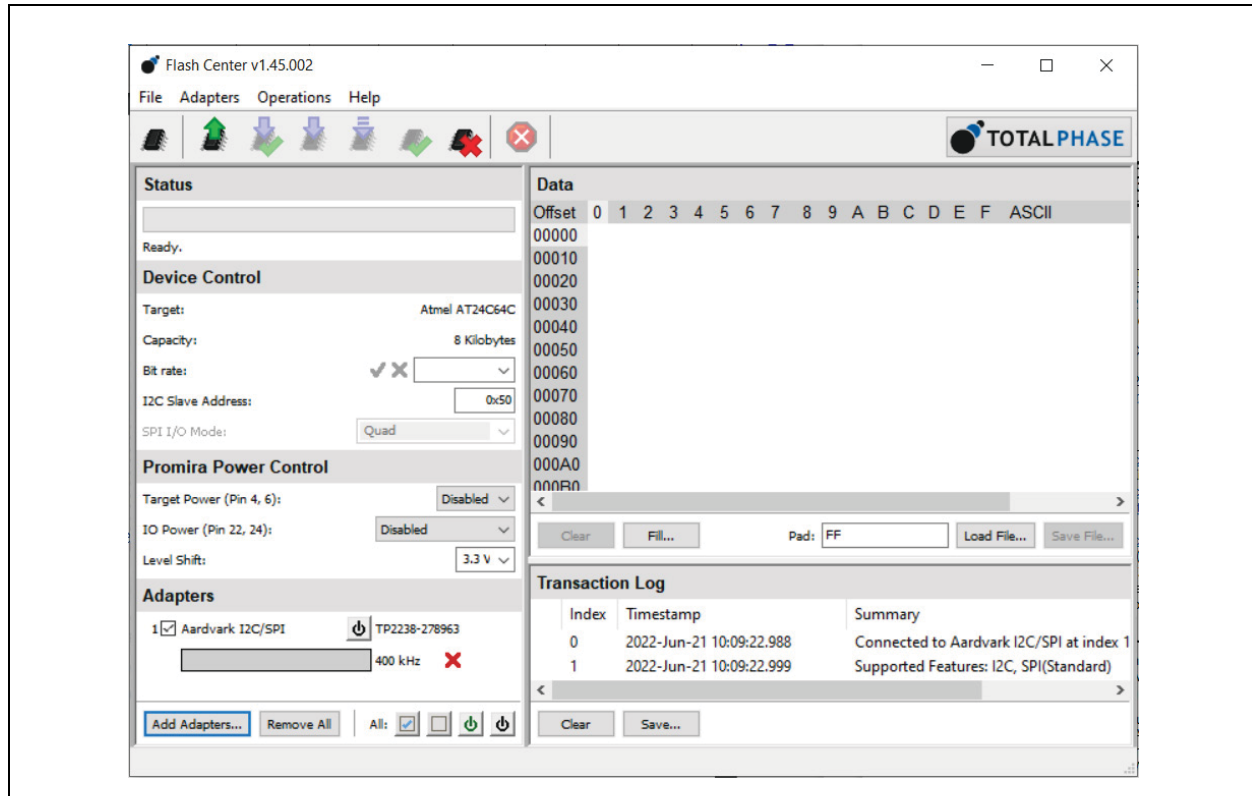
FIGURE 4-4: ADDING THE ADAPTER



PCI11414

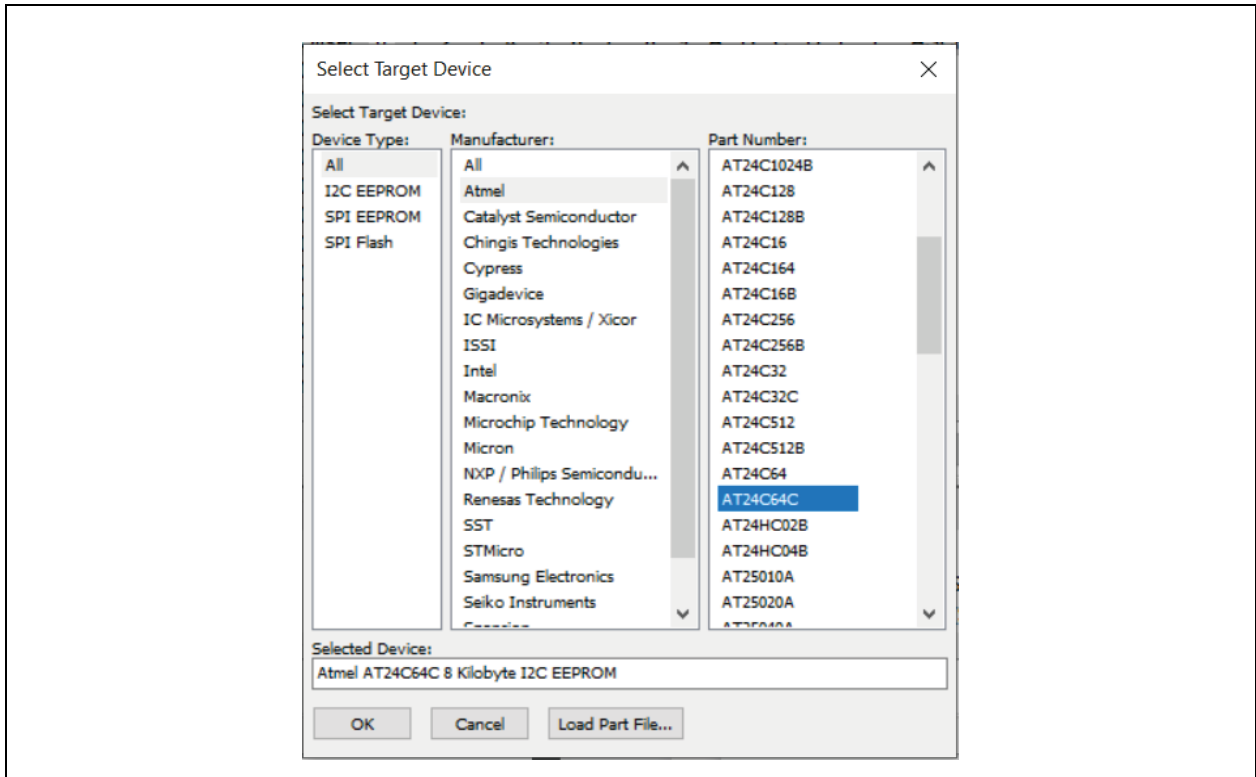
5. Select Choose Target. (See Figure 4-5.)

FIGURE 4-5: FLASH CENTER TARGET



6. From the provided list, select All>Atmel>AT24C64C. Click on **OK**. (See [Figure 4-6](#).)

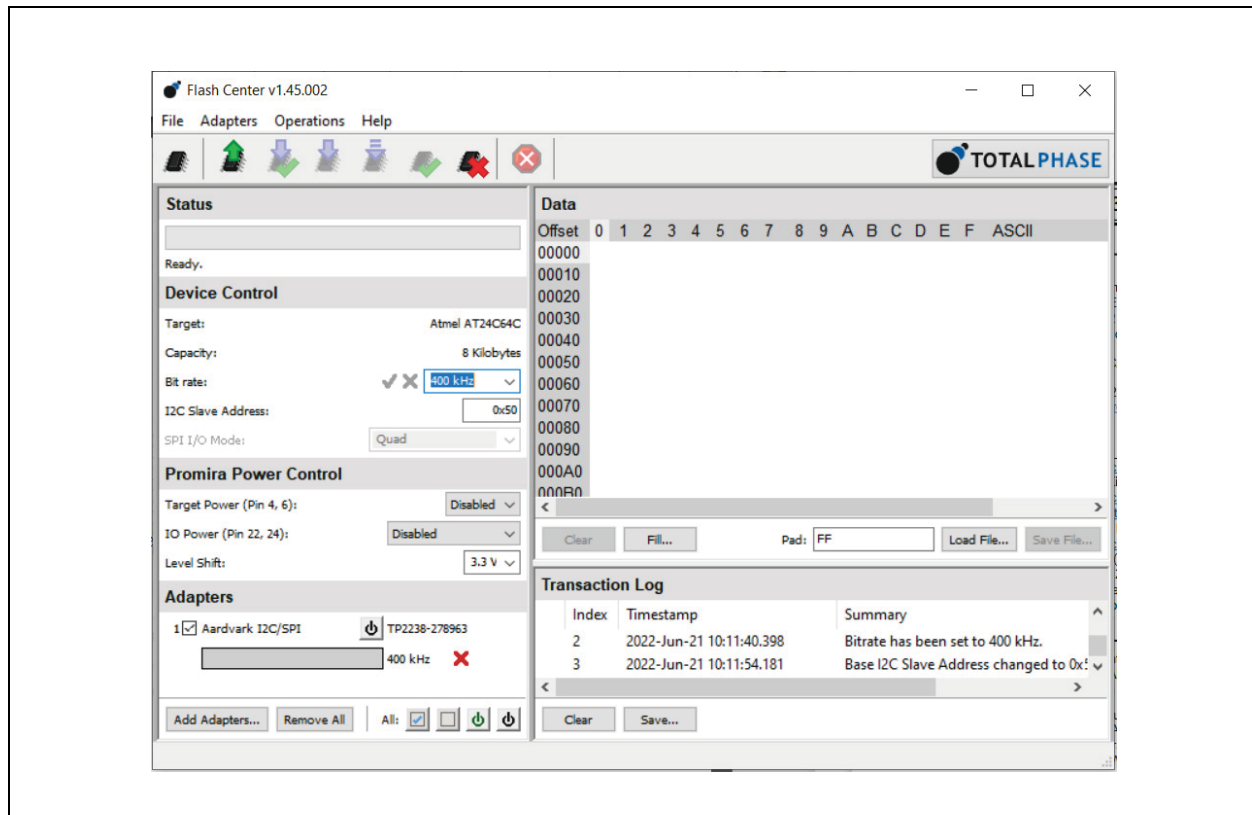
FIGURE 4-6: SELECTING TARGET DEVICE



7. Configure the following settings as shown in [Figure 4-7](#):

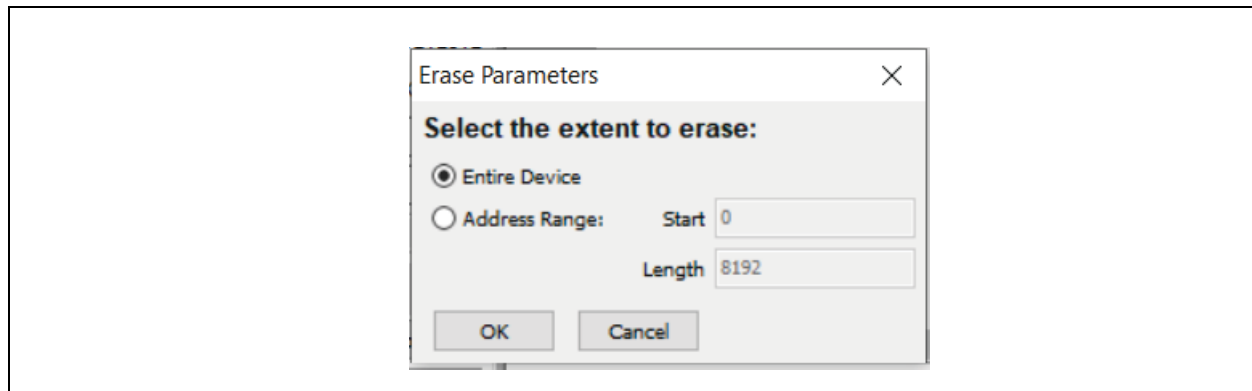
- Bit rate is 400 kHz.
- I²C Slave Address is left as default.
- Target power is disabled.
- IO power is disabled.

FIGURE 4-7: FLASH CENTER CONFIGURATION



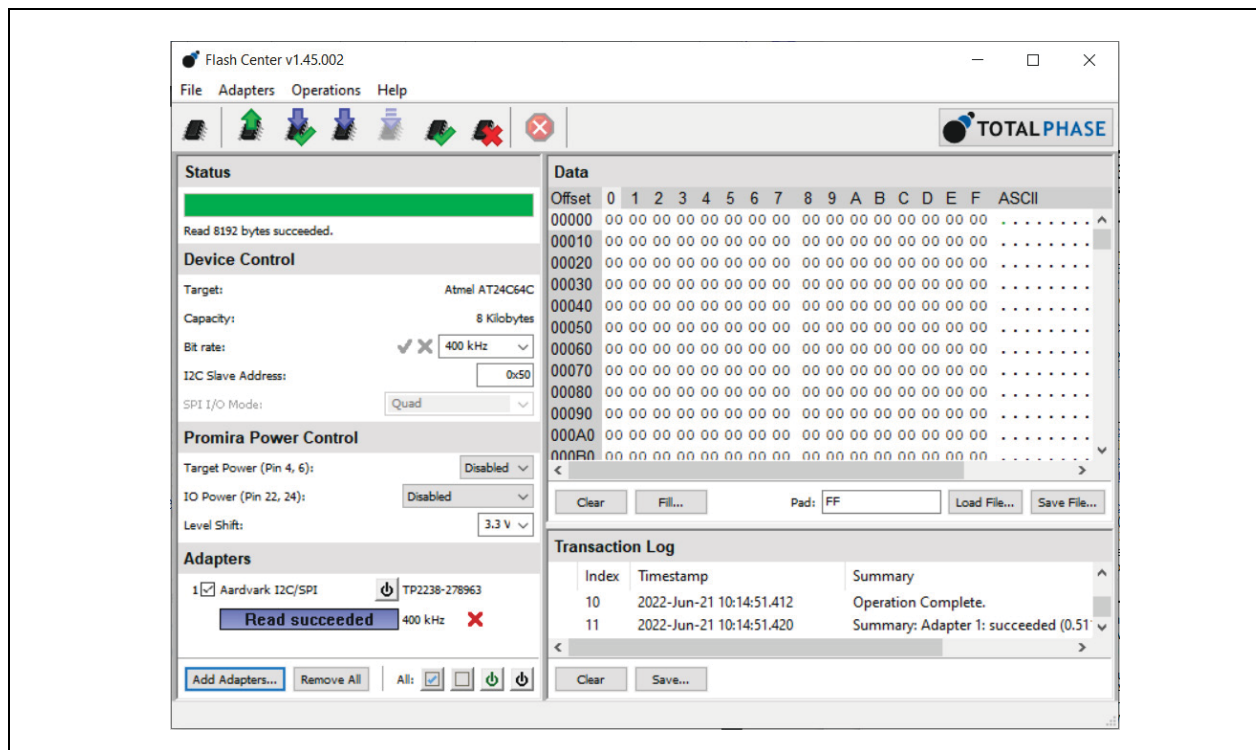
8. Click on **Erase Device**. Select Entire Device and then click on **OK**. (See Figure 4-8.)

FIGURE 4-8: ERASING DEVICE



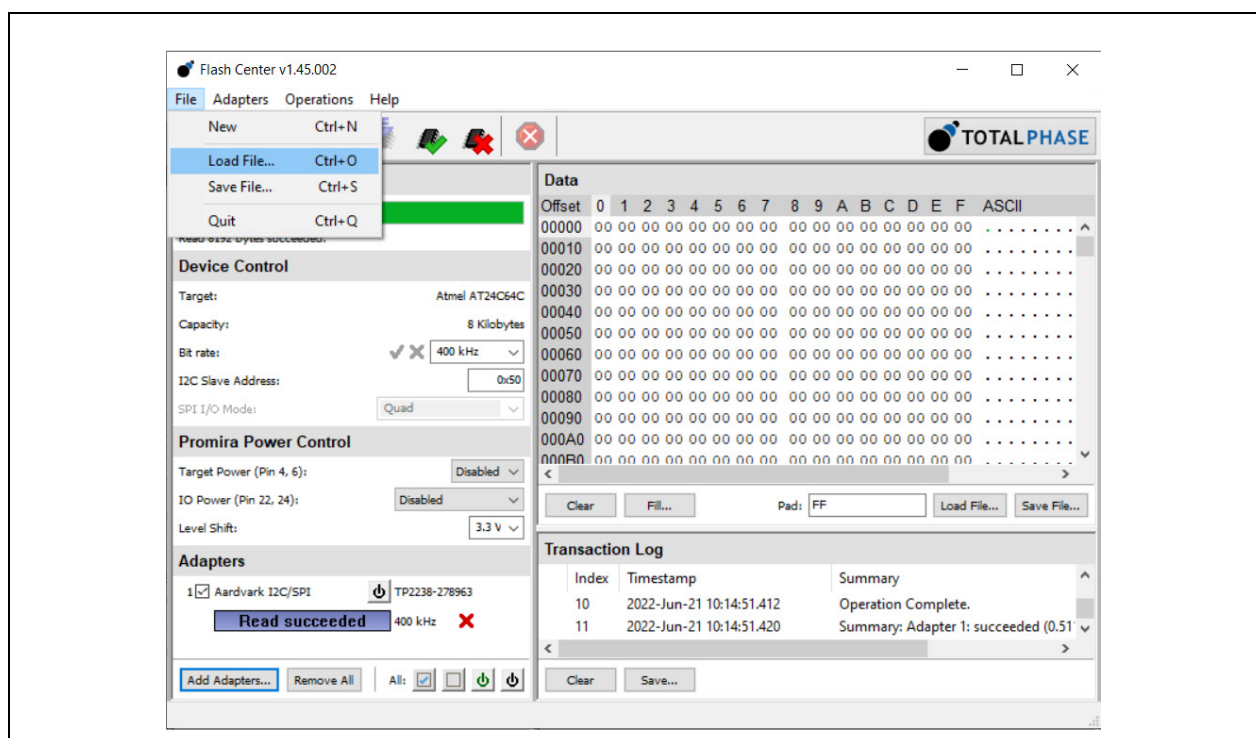
- Verify if the device was erased by reading the EEPROM back in. A successful erase will show all 0's. Refer to [Figure 4-9](#).

FIGURE 4-9: SUCCESSFUL ERASE



- Go to **File>Load File**. See [Figure 4-10](#).

FIGURE 4-10: FLASH CENTER LOAD FILE



PCI11414

11. Change the file type to Binary Files (*.*). Navigate to the configuration binary file and select it. See [Figure 4-11](#). The contents of the binary file will appear in the Data window as indicated in [Figure 4-12](#).

FIGURE 4-11: LOADING BINARY IMAGE

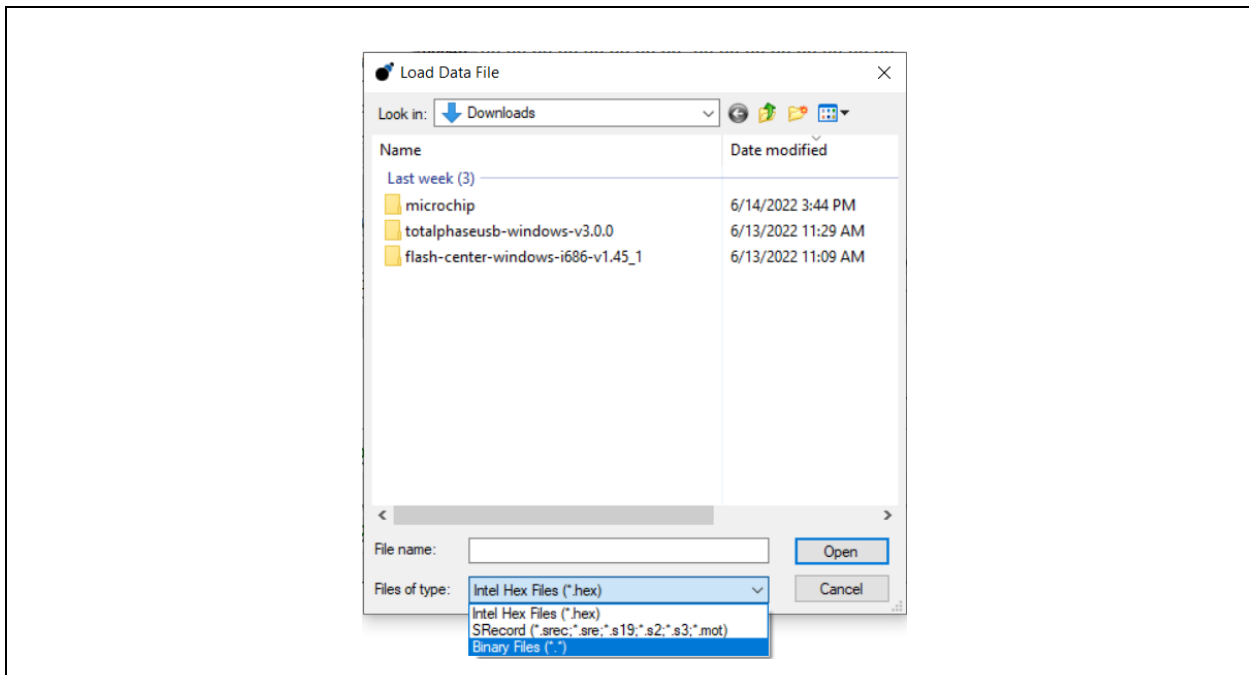
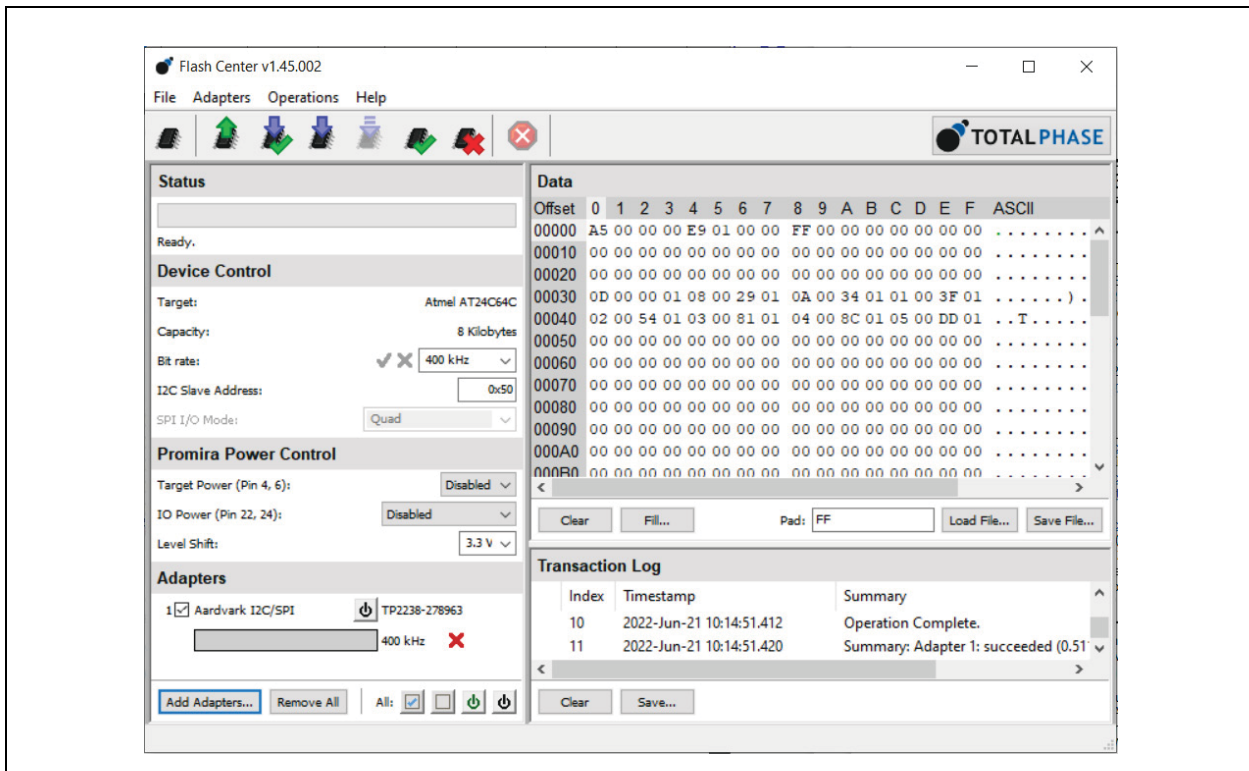
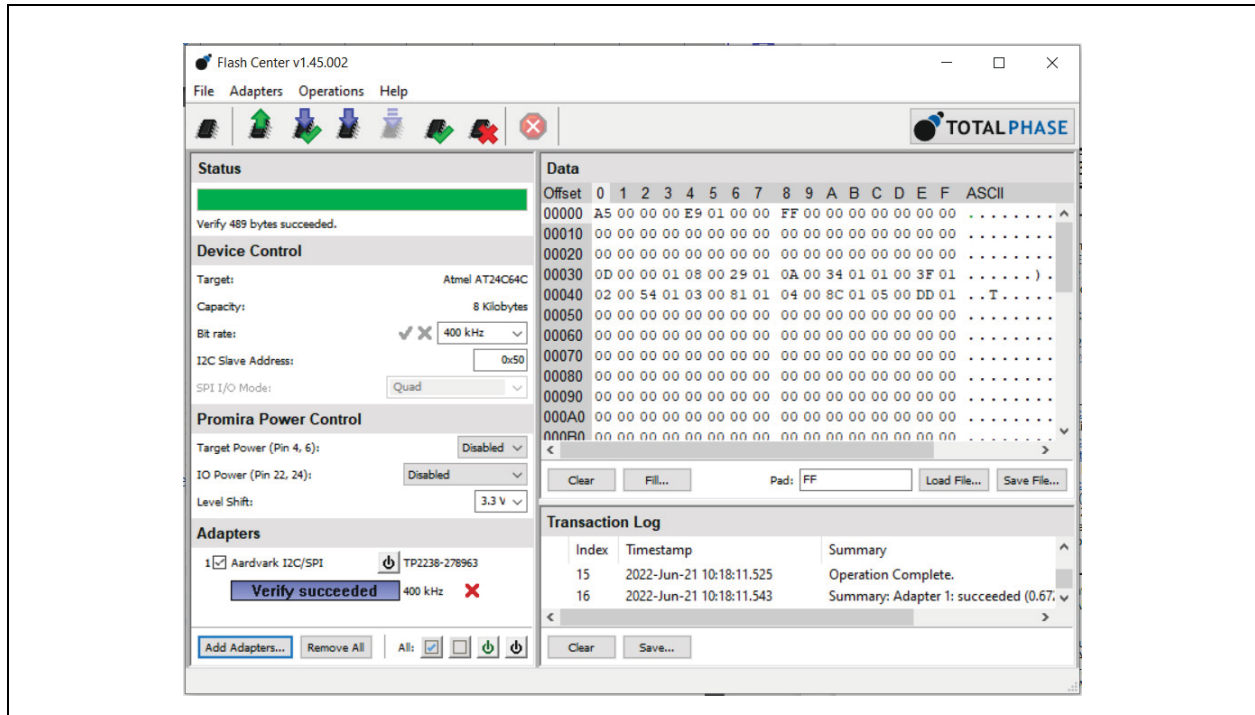


FIGURE 4-12: BINARY FILE CONTENT



12. Click on Program and verify. See [Figure 4-13](#).

FIGURE 4-13: VERIFYING BINARY FILE



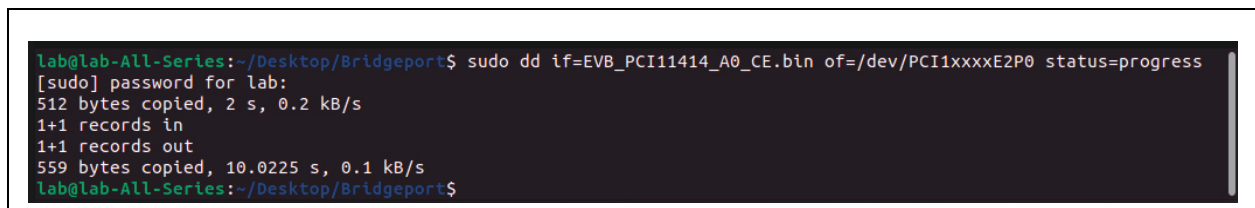
13. Read the data back from the EEPROM. There should be no change in the Data window if the EEPROM was written successfully.

14. Power cycle the board for the changes to take effect.

4.4 Programming with Linux®

1. The config on Linux can be flashed with the command, `dd if=<FILE TO BE FLASHED> of=<LOCATION OF EEPROM> status=progress`.
2. To identify the EEPROM, list the files in the `/dev/` directory with `ls /dev/PCI1xxxx`. `PCI1xxxxOTPN` is the OTP, and `PCI1xxxxE2PN` is the EEPROM. The EEPROM here is listed as `PCI1xxxE2P0`.
3. First, erase the EEPROM by running `dd if=/dev/zero of=/dev/PCI1xxxE2P0 status=progress`. Note that elevated privileges may be required to run the command.
4. Running the command, `dd if=<BINARY FILE> of=/dev/PCI1xxxE2P0 status=progress`, flashes the EEPROM. The output will update intermittently to show the progress. See [Figure 4-14](#).

FIGURE 4-14: DD COMMAND WITH PROGRESS



5. To verify that the EEPROM has been flashed, run `hexdump -C /dev/PCI1xxxE2P0`. The output should match the file that was just flashed to the EEPROM. Note that elevated privileges may be required to run the command.
6. Power cycle the PC for the changes to take effect.

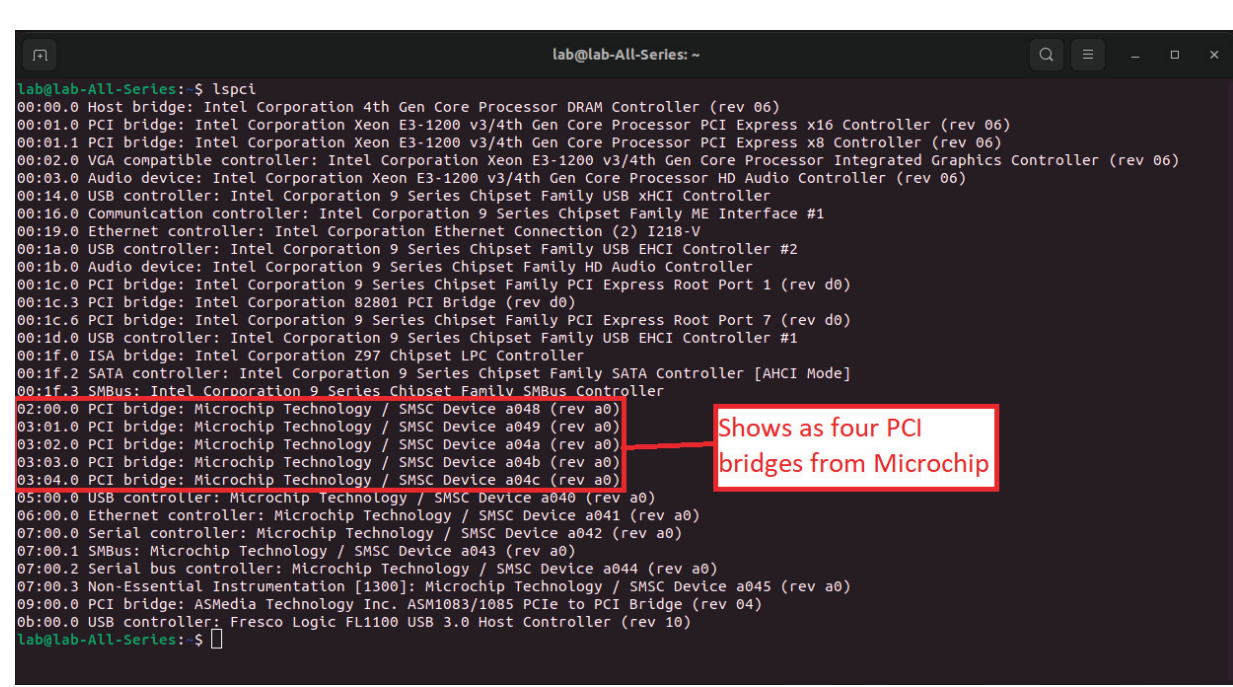
PCI11414

5.0 VERIFYING HARDWARE

Follow the succeeding procedure to verify the hardware:

1. With the PC turned off, insert the EVB-PCI11414 into one of the PCIe® slots.
2. Power on the PC. Make sure to boot into the correct kernel.
3. Open a terminal window (CTRL+ALT+T) and run the command `lspci`. Verify if the output shows five PCI bridges from Microchip. If there is only one PCI bridge, the board has not been programmed correctly. (See [Figure 5-1.](#))

FIGURE 5-1: VERIFYING THE PROGRAMMED BOARD



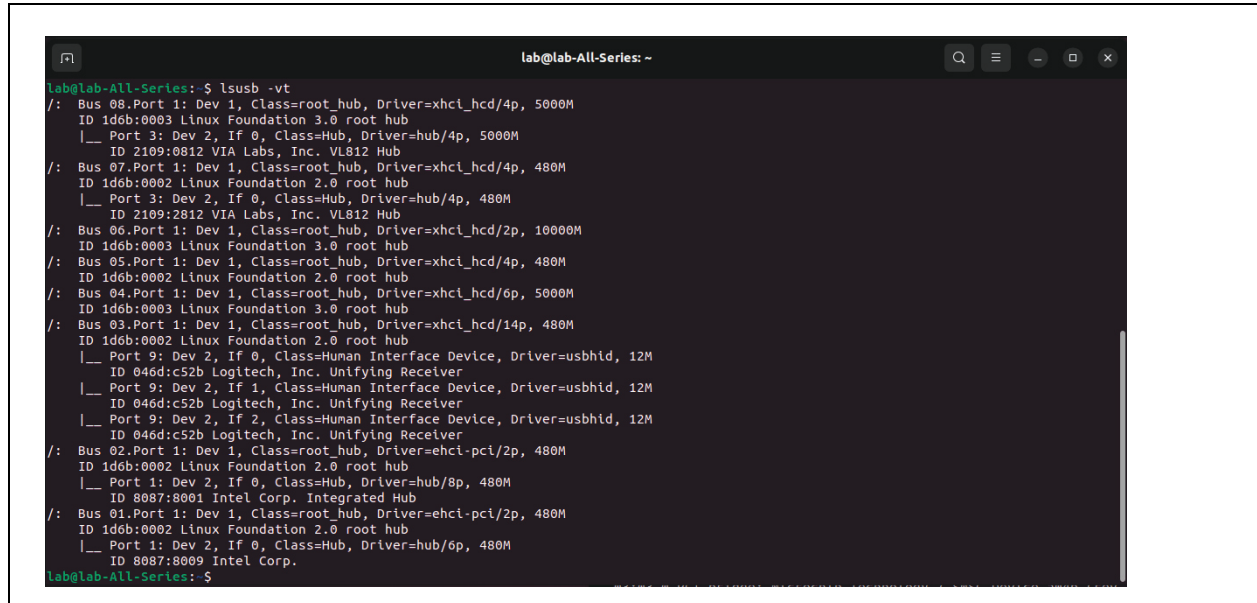
```
lab@lab-All-Series: ~  
lab@lab-All-Series:~$ lspci  
00:00.0 Host bridge: Intel Corporation 4th Gen Core Processor DRAM Controller (rev 06)  
00:01.0 PCI bridge: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor PCI Express x16 Controller (rev 06)  
00:01.1 PCI bridge: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor PCI Express x8 Controller (rev 06)  
00:02.0 VGA compatible controller: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor Integrated Graphics Controller (rev 06)  
00:03.0 Audio device: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor HD Audio Controller (rev 06)  
00:14.0 USB controller: Intel Corporation 9 Series Chipset Family USB xHCI Controller  
00:16.0 Communication controller: Intel Corporation 9 Series Chipset Family ME Interface #1  
00:19.0 Ethernet controller: Intel Corporation Ethernet Connection (2) I218-V  
00:1a.0 USB controller: Intel Corporation 9 Series Chipset Family USB EHCI Controller #2  
00:1b.0 Audio device: Intel Corporation 9 Series Chipset Family HD Audio Controller  
00:1c.0 PCI bridge: Intel Corporation 9 Series Chipset Family PCI Express Root Port 1 (rev d0)  
00:1c.3 PCI bridge: Intel Corporation 82801 PCI Bridge (rev d0)  
00:1c.6 PCI bridge: Intel Corporation 9 Series Chipset Family PCI Express Root Port 7 (rev d0)  
00:1d.0 USB controller: Intel Corporation 9 Series Chipset Family USB EHCI Controller #1  
00:1f.0 ISA bridge: Intel Corporation Z97 Chipset LPC Controller  
00:1f.2 SATA controller: Intel Corporation 9 Series Chipset Family SATA Controller [AHCI Mode]  
00:1f.3 SMBus: Intel Corporation 9 Series Chipset Family SMBus Controller  
02:00.0 PCI bridge: Microchip Technology / SMSC Device a048 (rev a0)  
03:01.0 PCI bridge: Microchip Technology / SMSC Device a049 (rev a0)  
03:02.0 PCI bridge: Microchip Technology / SMSC Device a04a (rev a0)  
03:03.0 PCI bridge: Microchip Technology / SMSC Device a04b (rev a0)  
03:04.0 PCI bridge: Microchip Technology / SMSC Device a04c (rev a0)  
05:00.0 USB controller: Microchip Technology / SMSC Device a040 (rev a0)  
06:00.0 Ethernet controller: Microchip Technology / SMSC Device a041 (rev a0)  
07:00.0 Serial controller: Microchip Technology / SMSC Device a042 (rev a0)  
07:00.1 SMBus: Microchip Technology / SMSC Device a043 (rev a0)  
07:00.2 Serial bus controller: Microchip Technology / SMSC Device a044 (rev a0)  
07:00.3 Non-Essential Instrumentation [1300]: Microchip Technology / SMSC Device a045 (rev a0)  
09:00.0 PCI bridge: ASMedia Technology Inc. ASM1083/1085 PCIe to PCI Bridge (rev 04)  
0b:00.0 USB controller: Fresco Logic FL1100 USB 3.0 Host Controller (rev 10)  
lab@lab-All-Series:~$
```

Shows as four PCI bridges from Microchip

5.1 USB

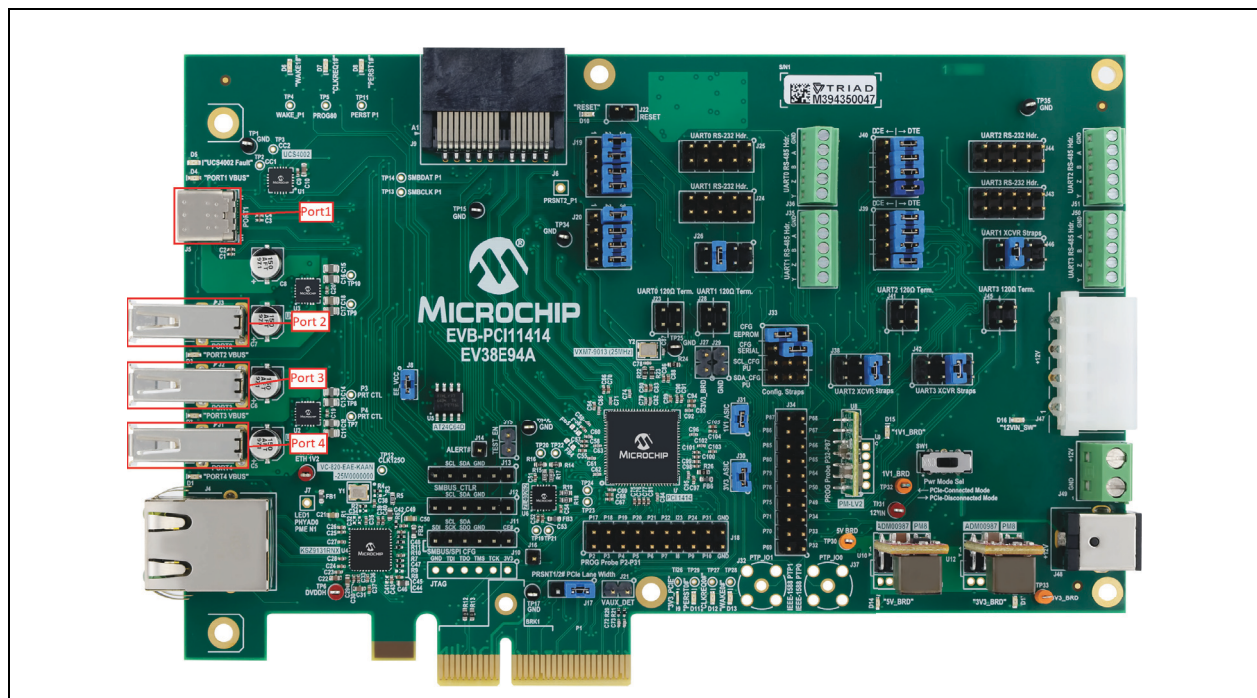
1. With no devices connected, run `lsusb -vt`. (See Figure 5-2.)

FIGURE 5-2: USB PORTS ON



2. Connect a USB device to the board and run `lsusb -vt` again.
3. Search for the device in the output. The output lists the speed that the device communicates at. The type A ports are USB 2.0 (480 Mbps), and the USB Type C® port is USB 3.1 gen 2 (10000 Mbps). The USB Type C port will show as port 1 or 2 depending on which way the device is plugged in. Refer to the image in Figure 5-3 for the position of the ports on the EVB board.

FIGURE 5-3: USB PORT LOCATIONS



PCI11414

- Remove the USB device and plug it into a different port. Turn around the USB Type C device to verify both orientations. Repeat step 3 for each port.

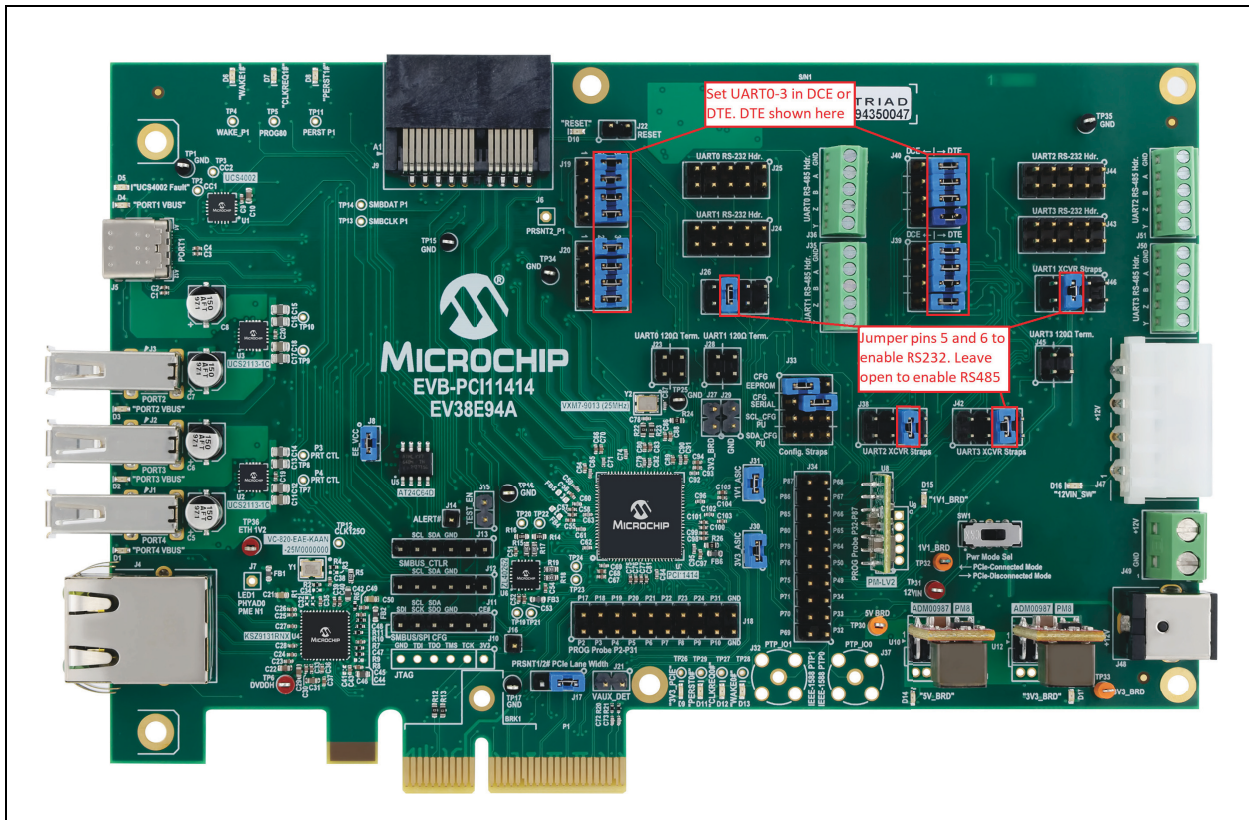
5.2 UART

5.2.1 RS-232

5.2.1.1 Preparing the board

- Headers J19, J20, J39, and J40 determine the pinout for the RS-232 UART headers. Put them all in the Data Circuit Terminating Equipment (DTE) as shown in [Figure 5-4](#). Refer to *AN4885 Interfacing PCI1xxx with RS-232, RS-422 and RS-485 Through UART Application Note* for more details.

FIGURE 5-4: UART JUMPER CONFIGURATION



- Next, wire two UARTs together. Each header will have two R pins and two T pins. To wire two headers together:
 - Connect T1 to R1.
 - Connect T2 to R2.
 - Connect R1 to T1.
 - Connect R2 to T2.

Refer to the board image in [Figure 5-5](#) and the schematics in [Figure 5-6](#).

FIGURE 5-5: WIRED UARTS

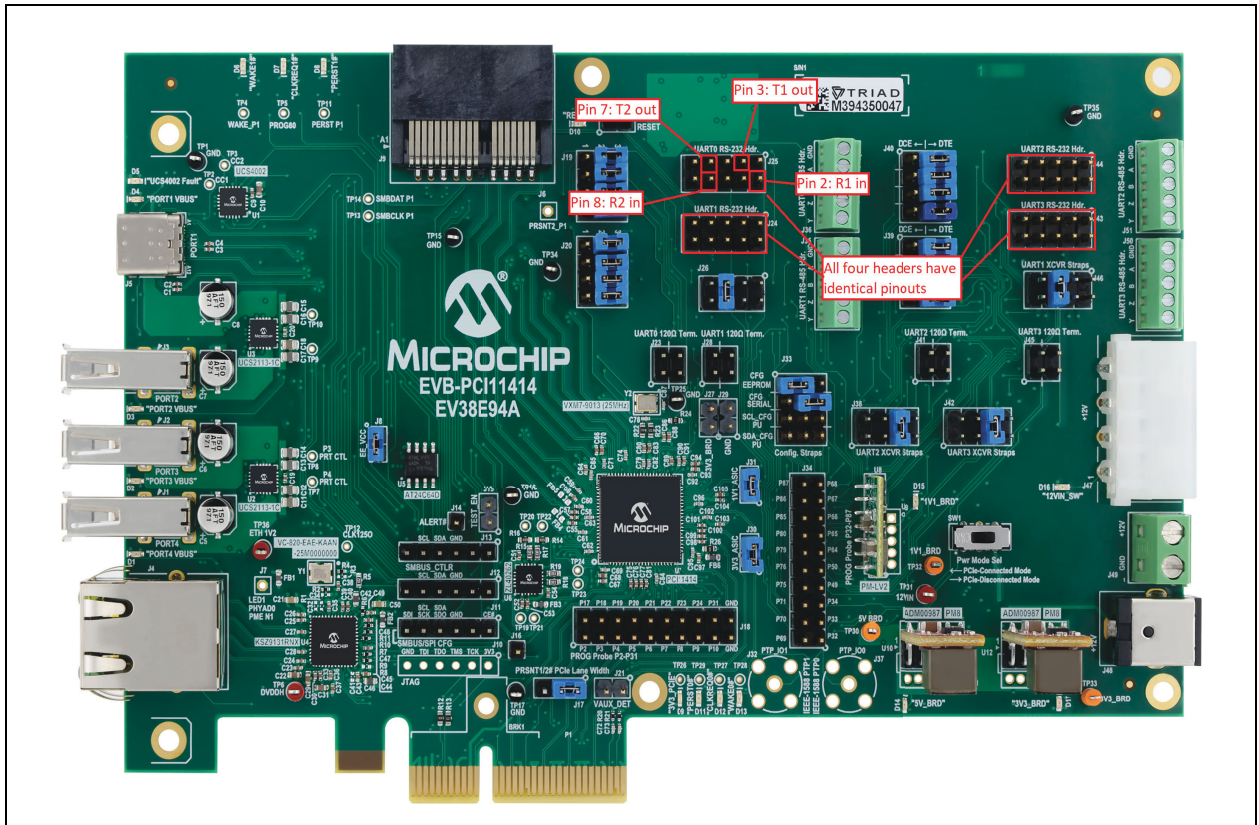
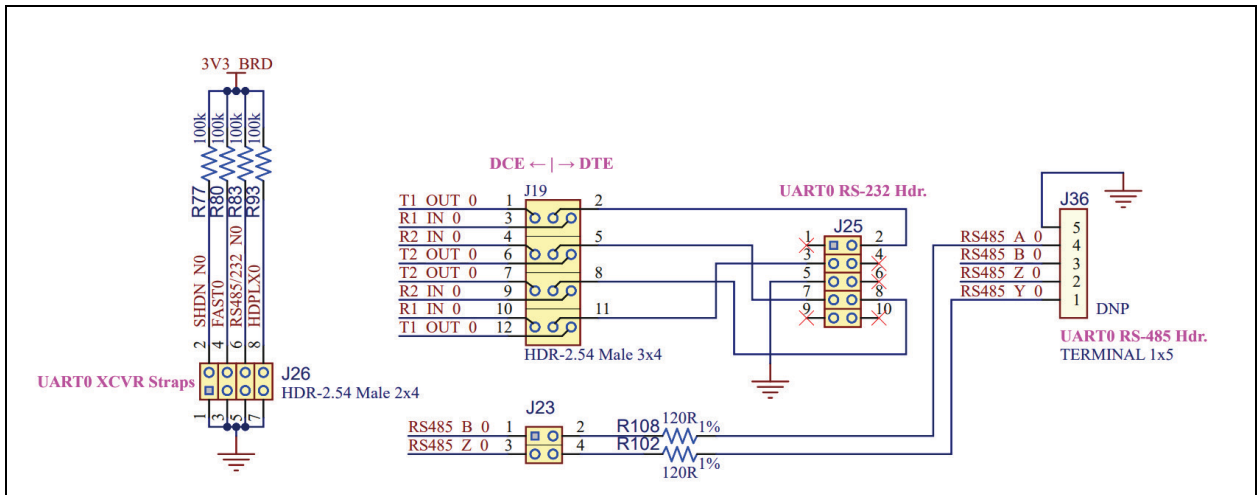


FIGURE 5-6: UART SCHEMATIC DRAWING

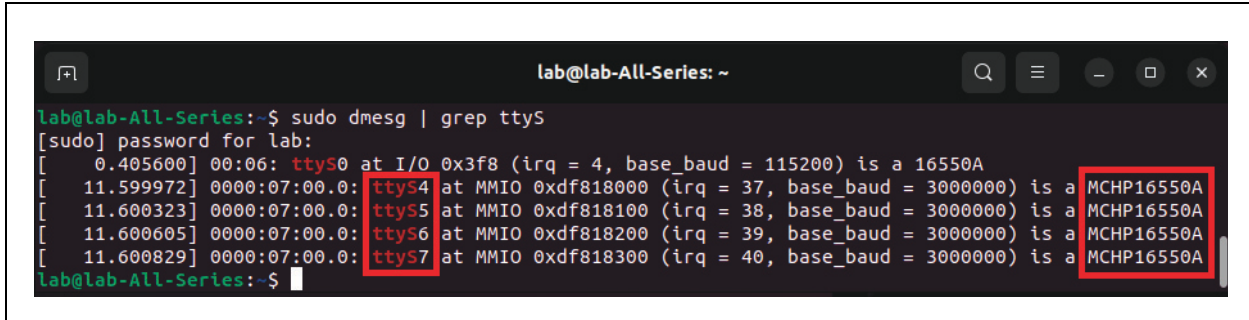


PCI11414

5.2.1.2 Verifying the UART connection

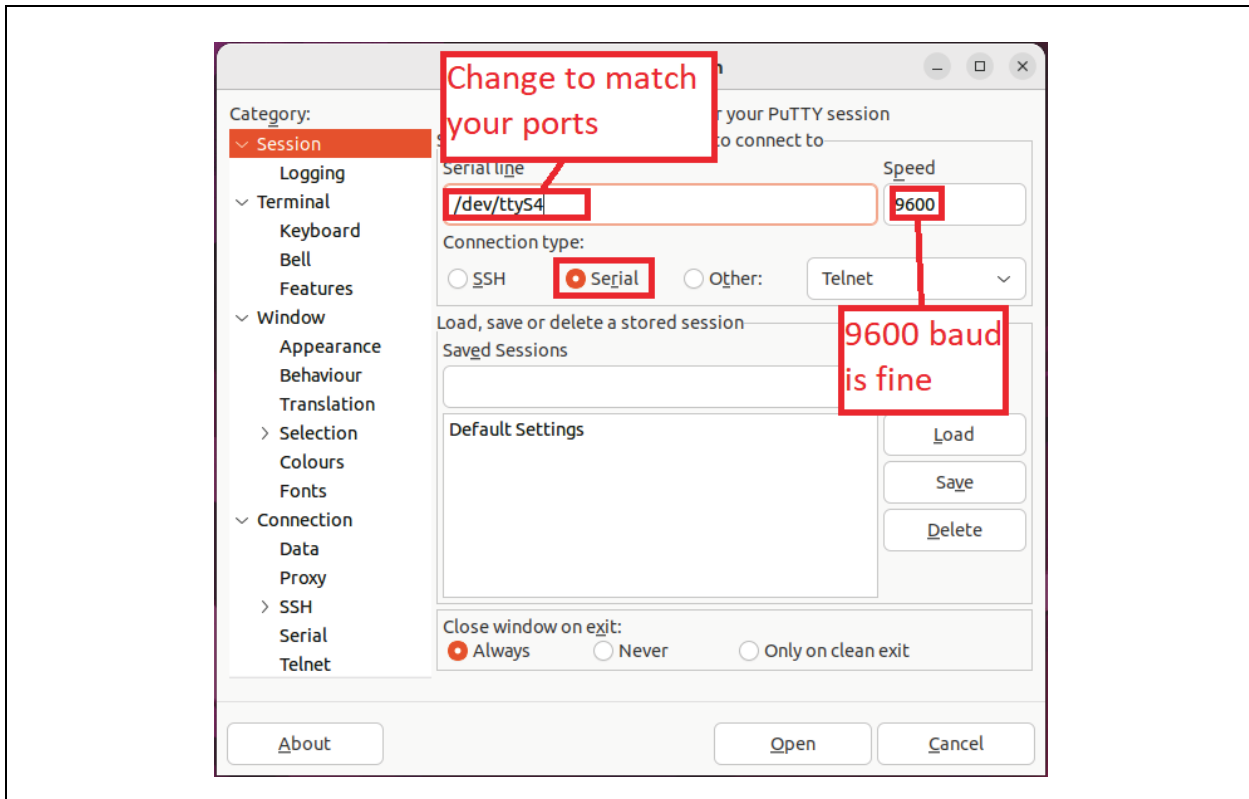
1. Start the computer with the board plugged in.
2. Run `sudo dmesg | grep ttyS`. This will list which ports the computer has mapped to the UARTs. See [Figure 5-7](#).

FIGURE 5-7: DMSG



3. Start the serial monitoring application of your choice. This verification procedure uses PuTTY.
4. Create a new session for each port listed in [Figure 5-8](#).

FIGURE 5-8: SAMPLE PUTTY SESSION



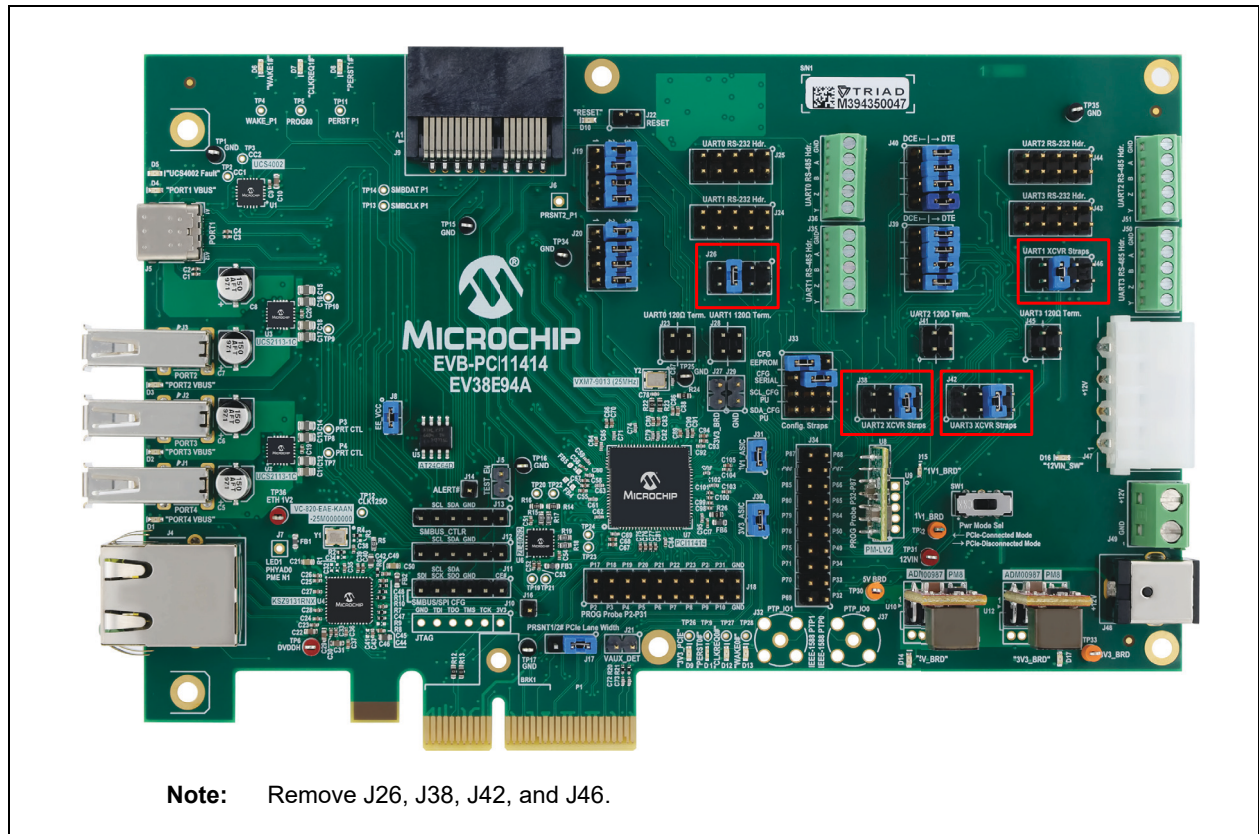
5. Start typing into each session. If UART is working correctly, you will see what you type as output in a different session.

5.2.2 RS-485

5.2.2.1 Preparing the board

1. J26, J38, J42, and J46 are the UART transceiver straps. To enable RS-485, pins 5 and 6 must remain **unbridged**. See [Figure 5-9](#).

FIGURE 5-9: UNBRIDGED PIN 5 AND PIN 7



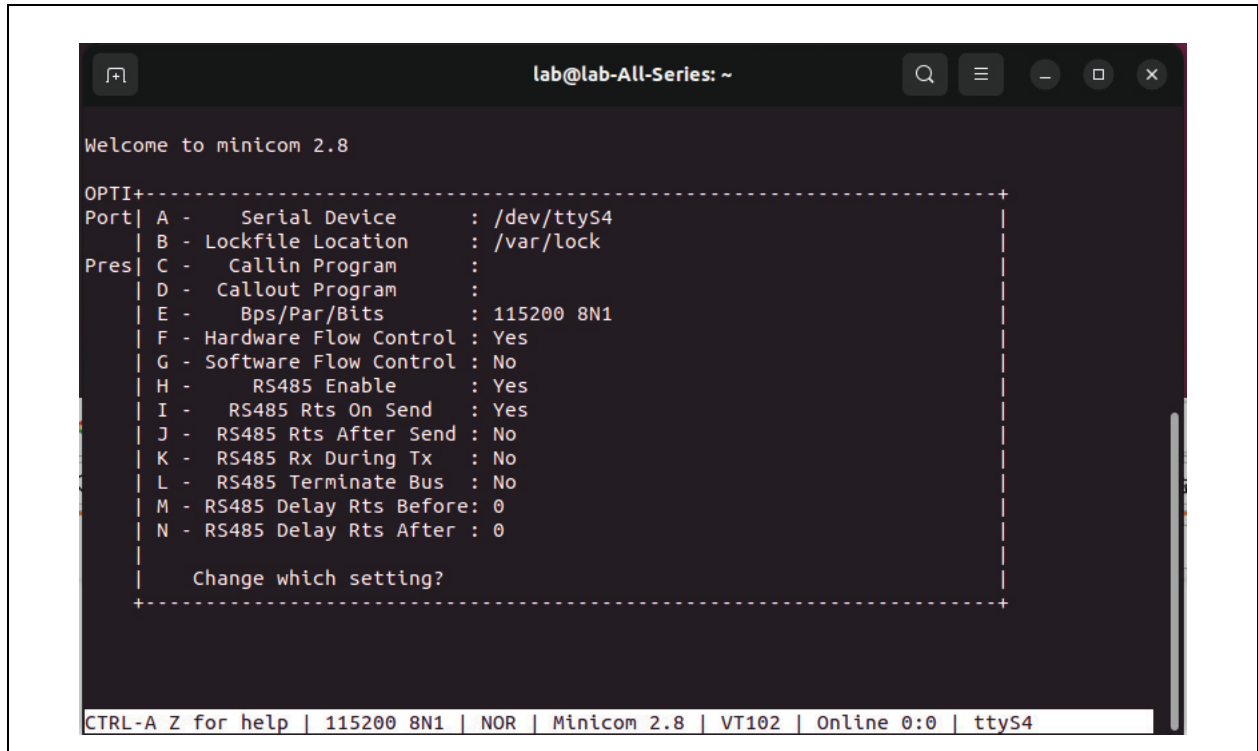
5.2.3 ESTABLISHING COMMUNICATION

5.2.3.1 Setting up the Host PC

1. To determine where the RS-485 ports are mapped, run `sudo dmesg | grep ttyS`. Note that since both RS-485 and RS-232 use the same UART pins, this command will show the same output that was shown in the RS-232 steps.
2. Next, configure the PC to use RS-485. This can be done with minicom. Run `minicom -D /dev/ttyS<PORT NUMBER>`. See [Figure 5-10](#).

- Press O to enter the configuration menu. See [Figure 5-12](#).

FIGURE 5-12: MINICOM CONFIGURATION MENU



- Select Serial port setup. See [Figure 5-13](#).

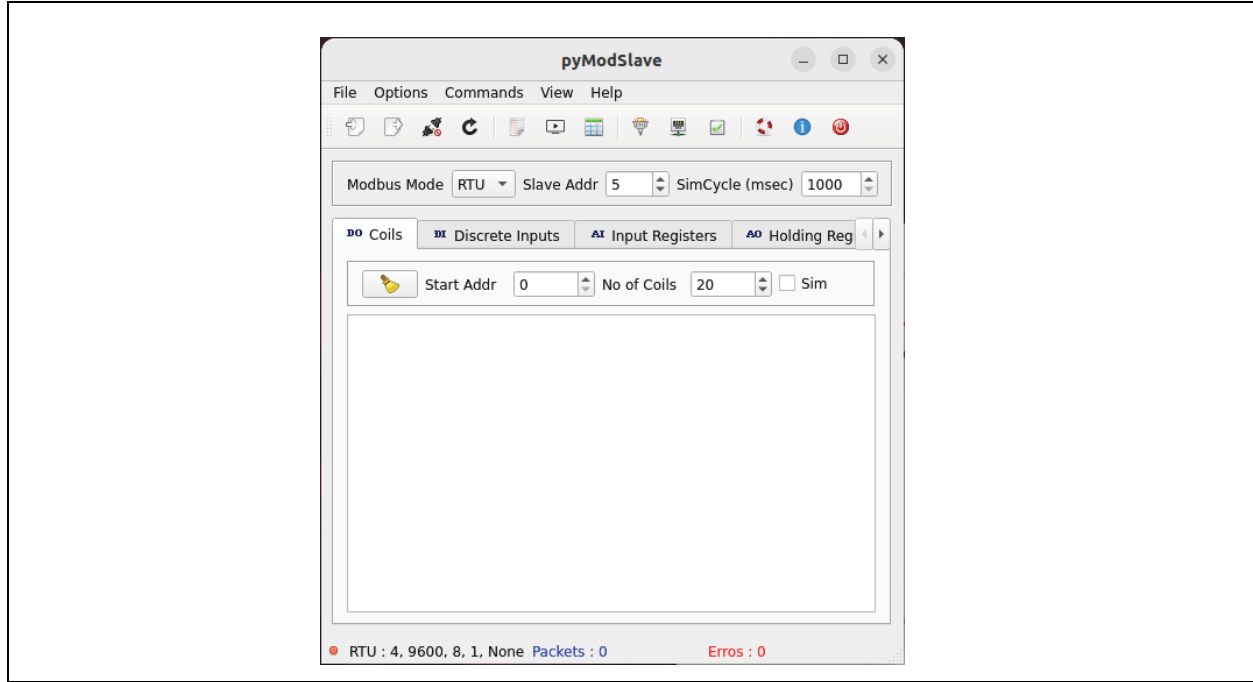
FIGURE 5-13: MINICOM SERIAL PORT SETUP



PCI11414

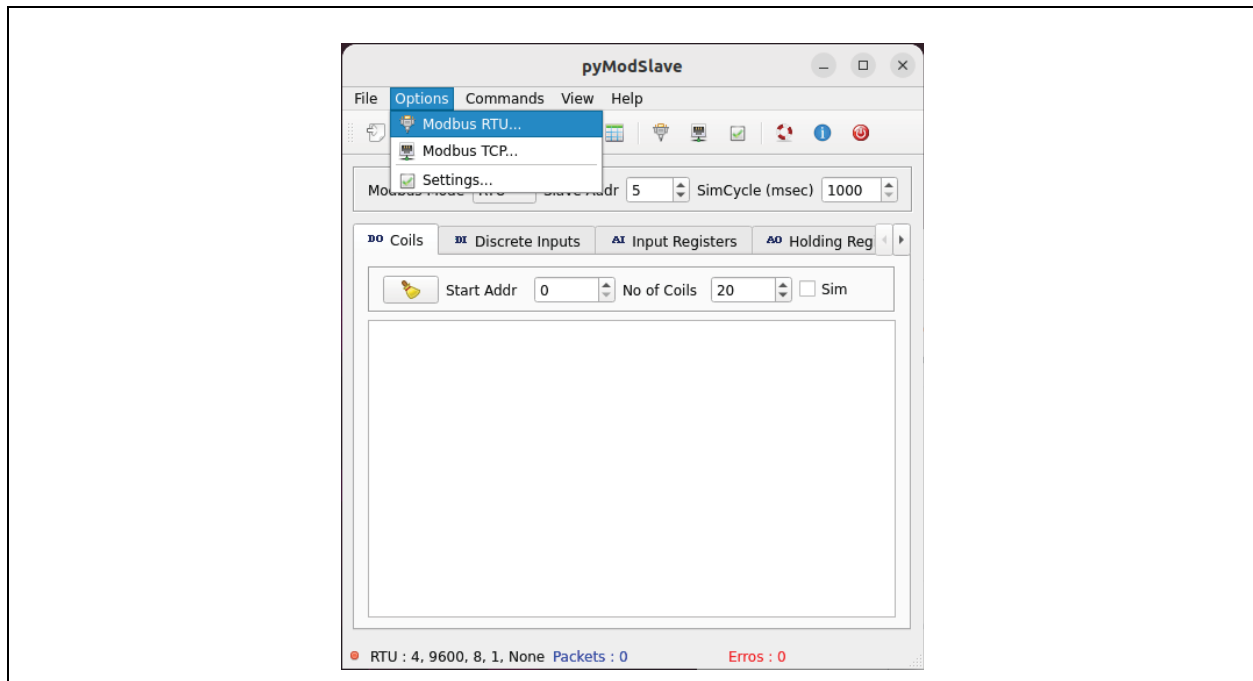
- Press in H and I to enable RS-485 and RS-485 RTS on send, respectively. Keep this terminal window open and open another terminal window.
- Navigate to where pyModSlave was installed in the computer, typically found in `~/.local/lib/python3.9/site-packages/pyModSlave`. Note that pyModSlave does not work on Python 3.10.
- Start pyModSlave with `sudo python3 pyModSlave`. This will bring up a GUI. See [Figure 5-14](#).

FIGURE 5-14: PYMODSLAVE



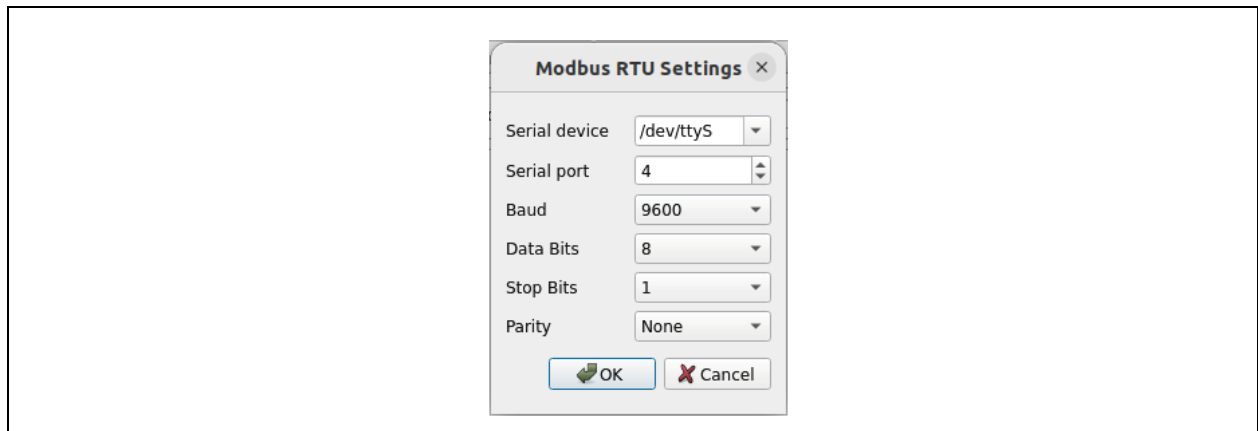
- Go to Options>Modbus RTU to configure the connection settings. See [Figure 5-15](#).

FIGURE 5-15: MODBUS RTU



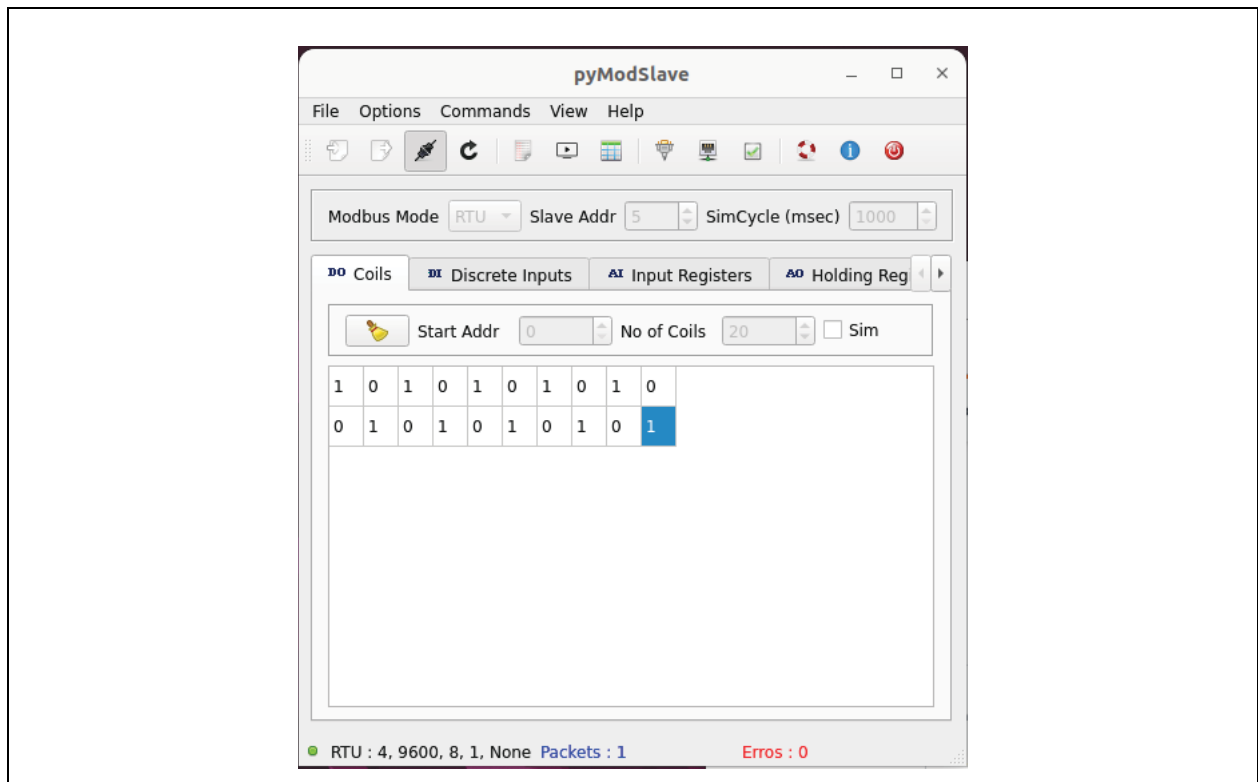
- Change the Serial port to the one set up in minicom (four in the [Figure 5-15](#) example) and leave everything else the same. Click on **OK**. (See [Figure 5-16](#).)

FIGURE 5-16: PYMODSLAVE RTU SETTINGS



- Take note of the Slave address and number of coils. These can be changed to suit the needs of the application.
- Click on Connect. The same number of coils set in the previous step will appear. These can be changed manually or randomly by selecting the sim checkbox. (See [Figure 5-17](#).)

FIGURE 5-17: PYMODSLAVE READ

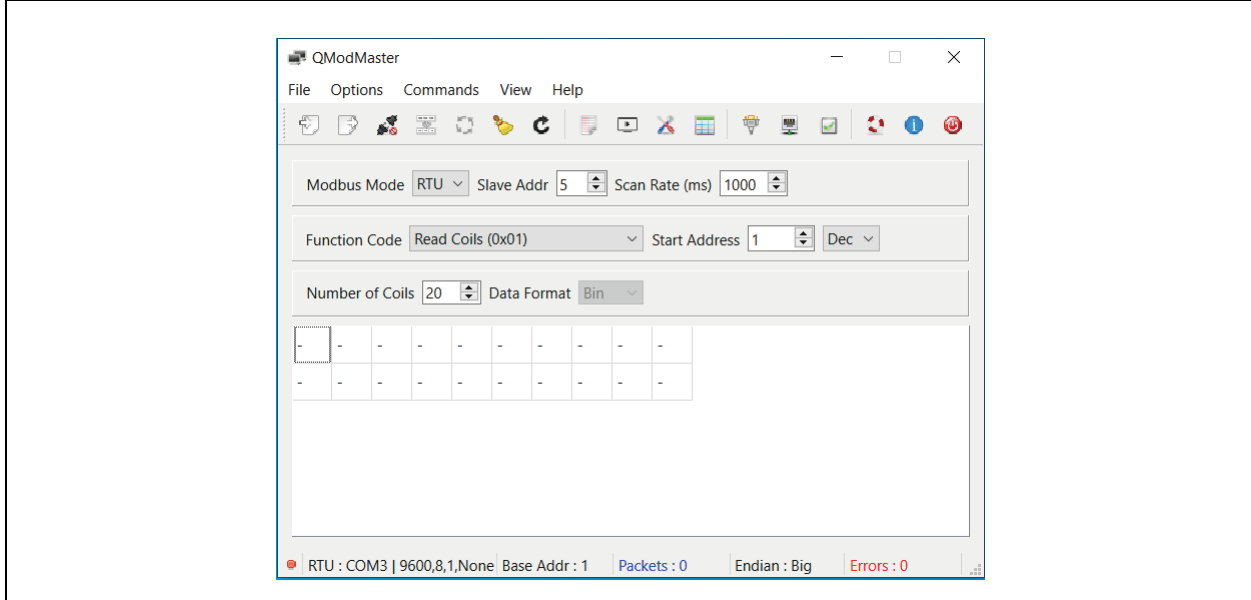


PCI11414

5.2.3.2 Setting up controlling PC

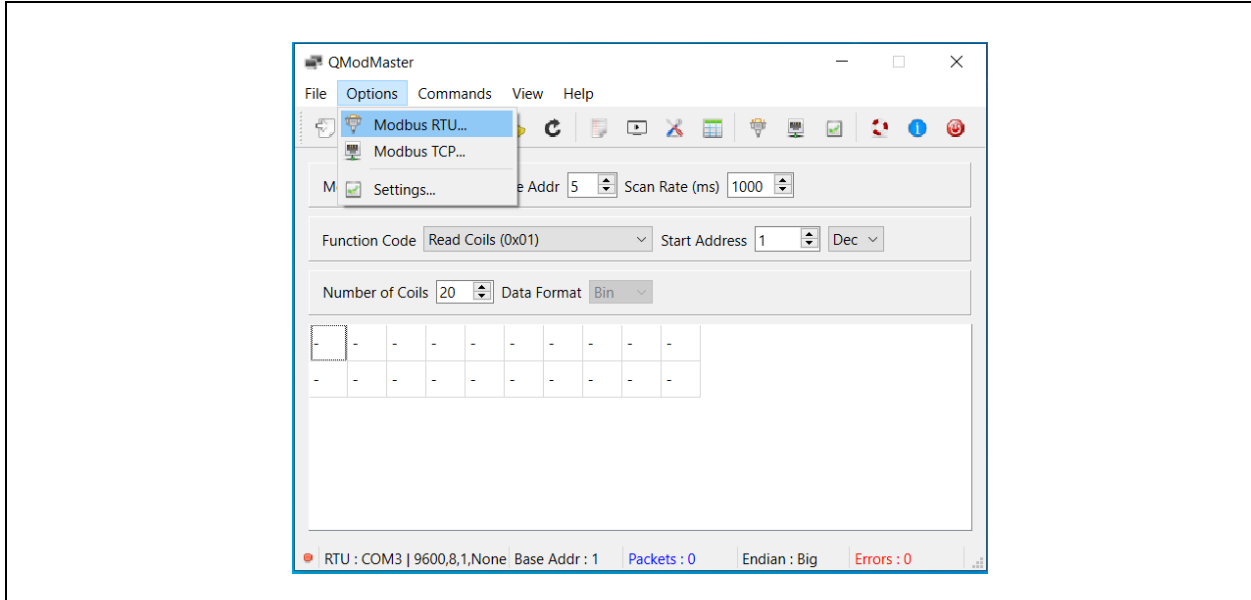
1. Connect the RS-485<->USB cable to the PC and open qModMaster. Note that drivers for the cable may need to be installed in the PC. (See [Figure 5-18](#).)

FIGURE 5-18: QMODMASTER



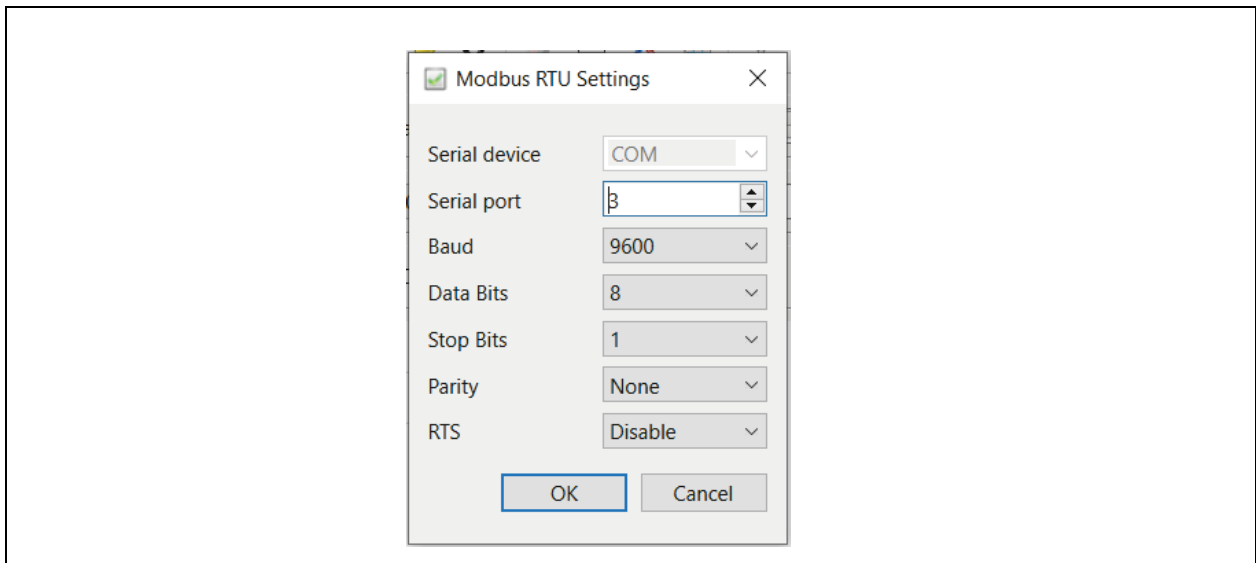
2. Go to *Options>Modbus RTU*. (See [Figure 5-19](#).)

FIGURE 5-19: QMODMASTER RTU



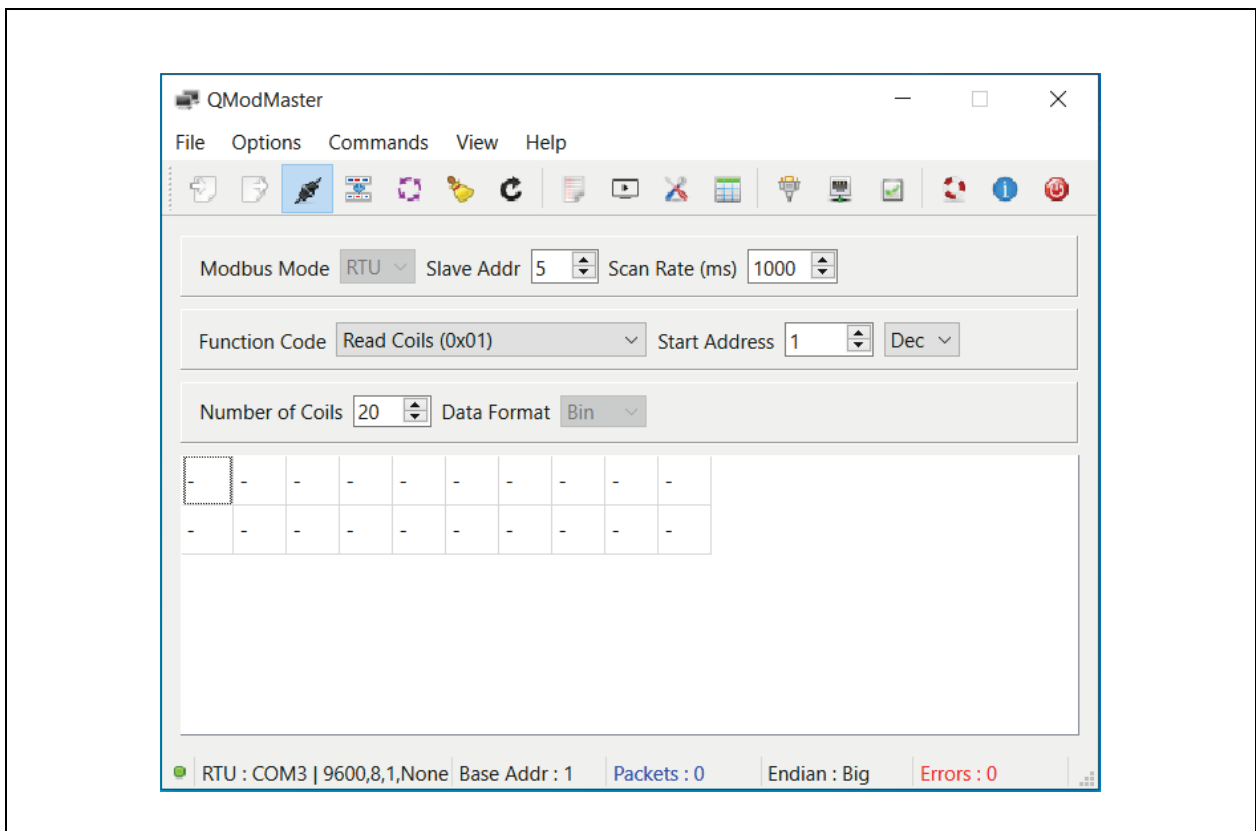
3. Change the Serial port to match the port the USB cable is plugged into and leave the other settings default. Click on **OK**. (See [Figure 5-20](#).)

FIGURE 5-20: QMODMASTER RTU SETTINGS



4. Change the slave address and number of coils to match what was set up on the host computer and click on **Connect**. (See [Figure 5-21](#).)

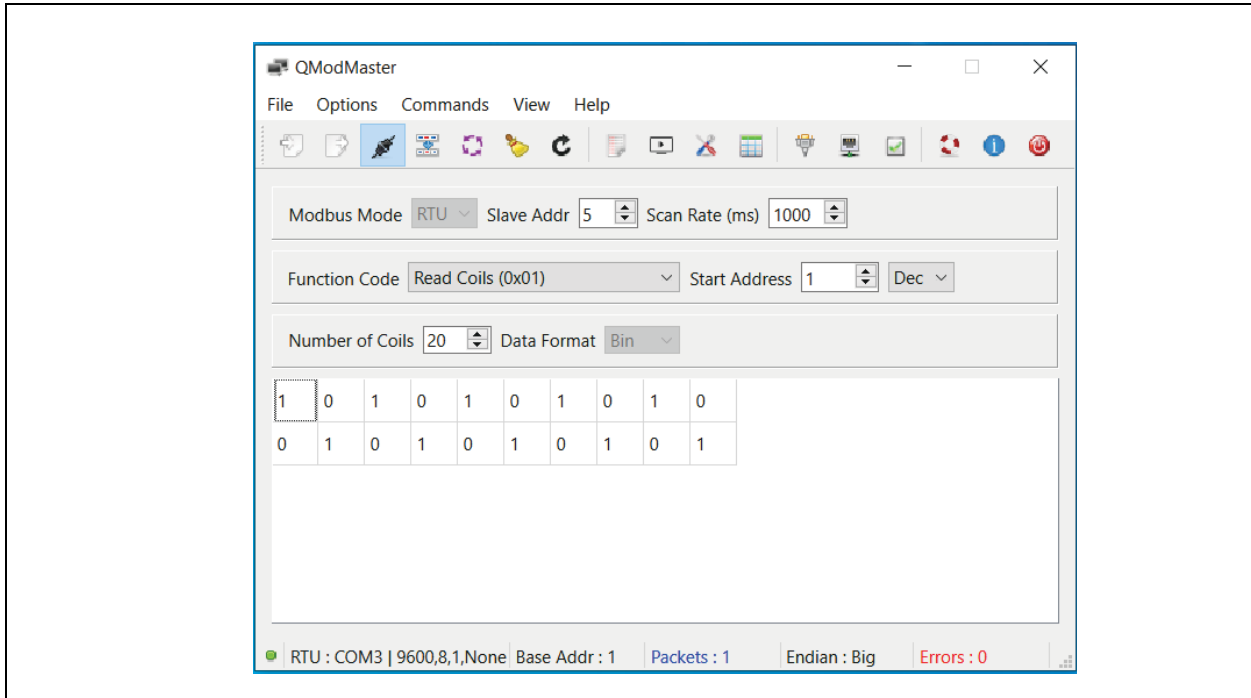
FIGURE 5-21: QMODMASTER CONNECTED



PCI11414

- Click on **Read** to get the current status. If everything is working correctly, there will be the same pattern of coils in the controlling PC that was set up on the host PC. See [Figure 5-22](#).

FIGURE 5-22: QMODMASTER READ

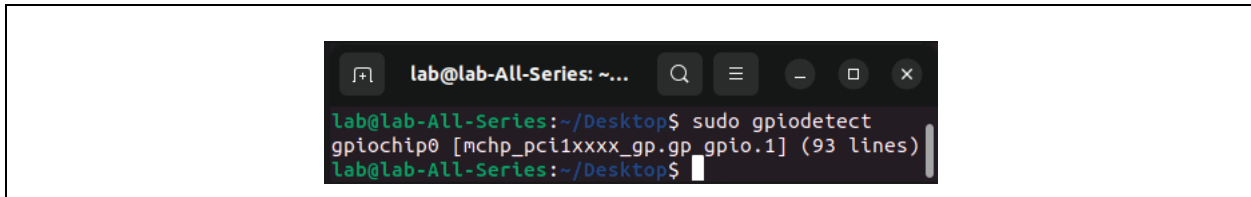


5.3 GPIO

5.3.1 VERIFYING THE GPIO BANK

- GPIO operations can be done with the `gpiod` chardev.
- First, verify that if the GPIO bank can be seen. Run `gpiodetect`. Note that elevated privileges may be required. Look for an entry labeled `mchp_pci1xxxx_gp.gp`. Note that the number after `gpiodetect` is the bank number. In this case, it is bank 0. See [Figure 5-23](#).

FIGURE 5-23: GPIODETECT



- After verifying, the GPIO bank becomes visible. Take a closer look at that GPIO by running `gpioinfo <bank>`. This tells users the direction of all the pins and any processes that may be currently using them. See [Figure 5-24](#).

FIGURE 5-24: GPIOINFO

```

lab@lab-All-Series: ~/Desktop
lab@lab-All-Series:~/Desktop$ sudo gpioinfo 0
gpiochip0 - 93 lines:
line 0:      unnamed      unused  input  active-high
line 1:      unnamed      unused  input  active-high
line 2:      unnamed      unused  input  active-high
line 3:      unnamed      unused  input  active-high
line 4:      unnamed      unused  input  active-high
line 5:      unnamed      unused  input  active-high
line 6:      unnamed      unused  input  active-high
line 7:      unnamed      unused  input  active-high
line 8:      unnamed      unused  input  active-high
line 9:      unnamed      unused  input  active-high
line 10:     unnamed      unused  input  active-high
line 11:     unnamed      unused  input  active-high
line 12:     unnamed      unused  input  active-high
line 13:     unnamed      unused  input  active-high
line 14:     unnamed      unused  input  active-high
line 15:     unnamed      unused  input  active-high
line 16:     unnamed      unused  input  active-high
line 17:     unnamed      unused  input  active-high
line 18:     unnamed      unused  input  active-high
line 19:     unnamed      unused  input  active-high
line 20:     unnamed      unused  input  active-high
line 21:     unnamed      unused  input  active-high
line 22:     unnamed      unused  input  active-high
line 23:     unnamed      unused  input  active-high
line 24:     unnamed      unused  input  active-high
line 25:     unnamed      unused  input  active-high
line 26:     unnamed      unused  input  active-high
line 27:     unnamed      unused  input  active-high
  
```

- The GPIOd automatically sets the direction of a pin with the `gpioset` and `gpioget` commands.

5.3.2 READING INPUT

- Use the command, `gpioget <bank> <pin>`, to read input with `gpioget` chardev.
- Start by tying pin 67 to ground. After running `gpioget 0 67` (pin 67 on bank 0), the output is a 0.
- The inputs have a pull-up by default. If the wire connecting pin 67 to ground is removed and the command is run again, the output becomes a 1. See [Figure 5-25](#).

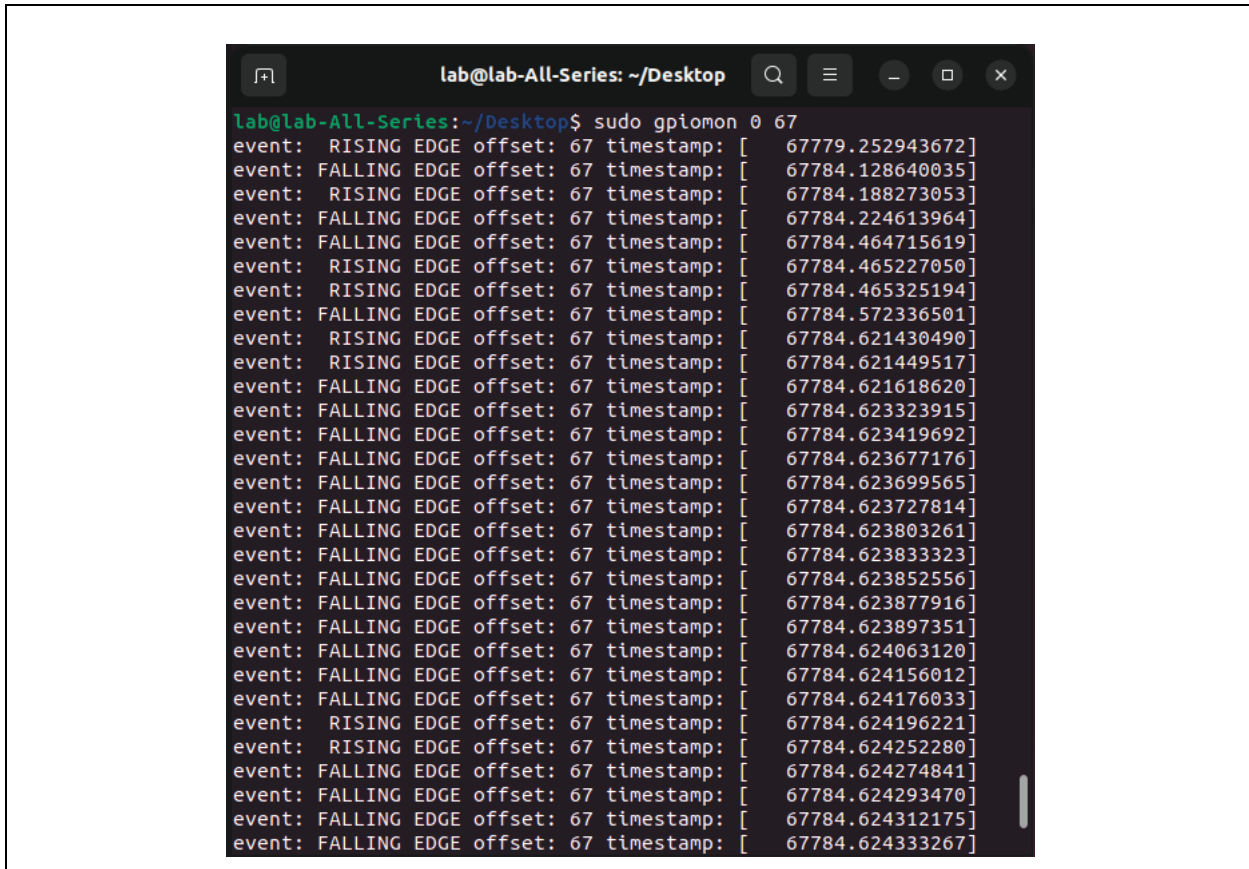
FIGURE 5-25: GPIOGET

```

lab@lab-All-Series: ~/Desktop
lab@lab-All-Series:~/Desktop$ sudo gpioget 0 67
0
lab@lab-All-Series:~/Desktop$ sudo gpioget 0 67
1
lab@lab-All-Series:~/Desktop$
  
```

- Set a pin to be automatically monitored using `gpiomon <bank> <pin>`.
- In another terminal, run `gpiomon 0 67` to set automatic monitoring of pin 67. Whenever pin 67 detects a rising or falling edge, an output can be seen in the window shown in [Figure 5-25](#).
- Connect pin 67 to ground again and note the falling edge. Disconnect the ground and note the rising edge. See [Figure 5-26](#).

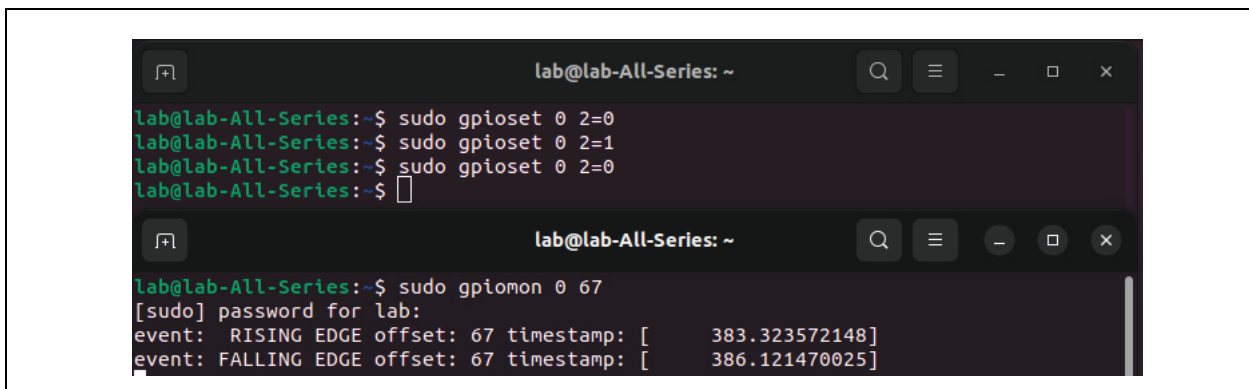
FIGURE 5-26: GPIOMON



5.3.3 SETTING OUTPUTS

1. Outputs can be set with `gpioset <bank> <pin>=<state>`.
2. Connect pin 67 to pin 2.
3. Run `gpioset 0 2=1`. Note the change in the monitoring window set up previously. If there is no change, then pin 67 was already reading a 1 prior to this command.
4. Run `gpioset 0 2=0`. Note the change in the monitoring window. See [Figure 5-27](#).

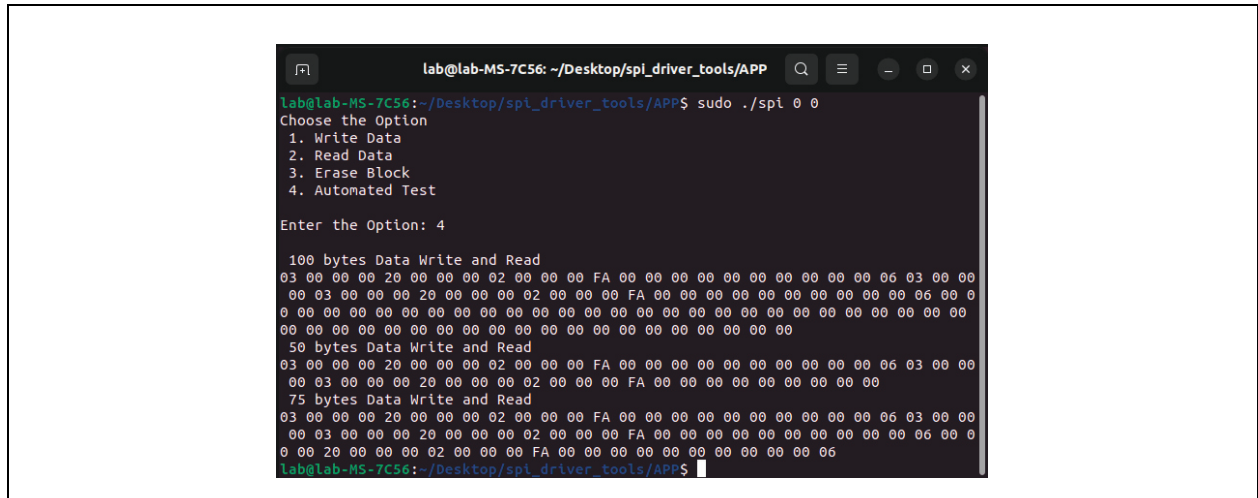
FIGURE 5-27: GPIO INPUT AND OUTPUT



5.4 SPI

1. To use SPI, the SPI configuration binary must be programmed onto the board.
2. For each SPI device, tie the Data In and Data Out pins together. For SPI controller 0, tie PROG4 and PROG5. For SPI controller 1, tie PROG8 and PROG9. Note that not all CPNs expose both or either SPI controller.
3. Compile and run the spi_driver_tools application.
4. Choosing option 4 runs an automated test that both reads and writes data. If the SPI device is working incorrectly, all the data will be 00s. See [Figure 5-28](#).

FIGURE 5-28: SPI READ AND WRITE

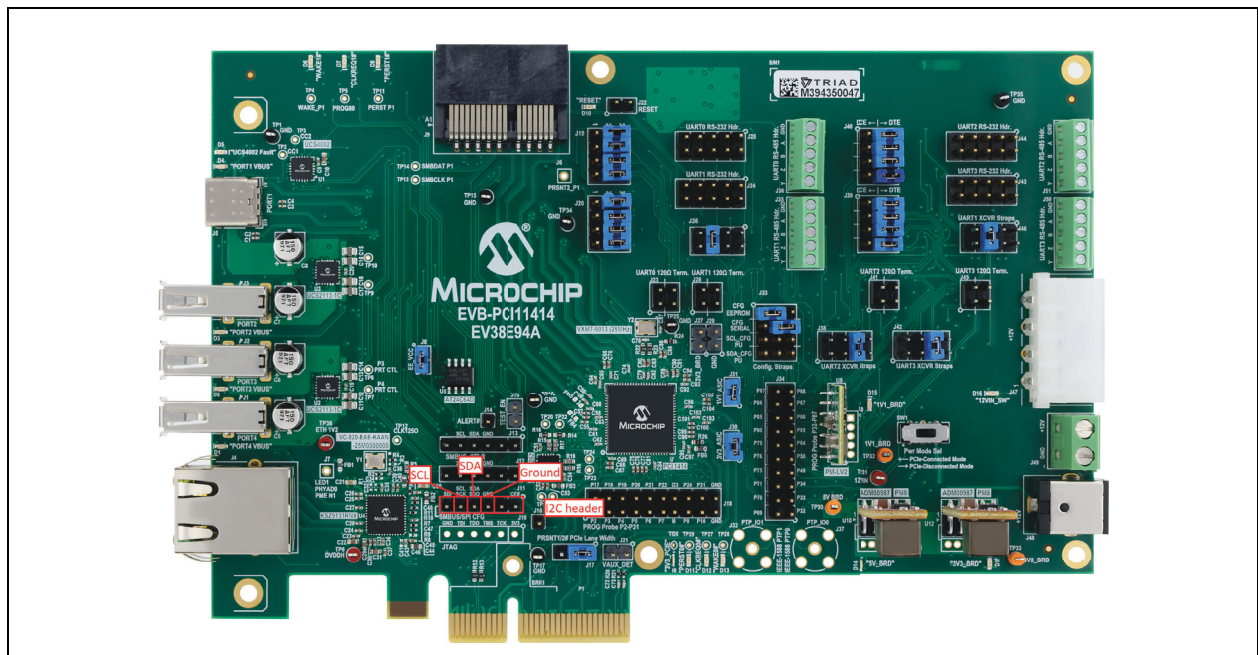


5.5 I²C

The EVB-PCI11414 has two I²C buses. One is used for programming the EEPROM, and the other is available to PCI11414.

1. Connect the SCL, SDA, and ground lines from the Aardvark to header J12. Plug the USB cable into a Windows PC. See [Figure 5-29](#).

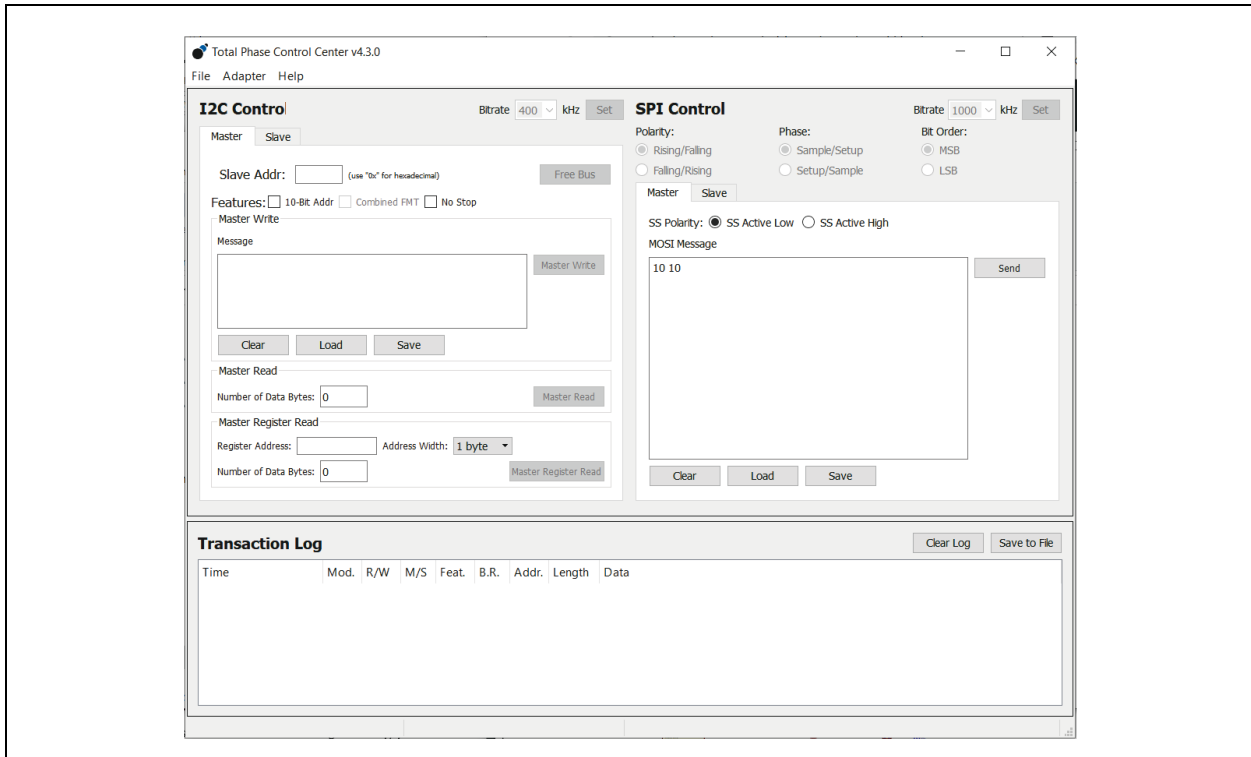
FIGURE 5-29: I²C, SCL, SDA, AND GROUND LINES



PCI11414

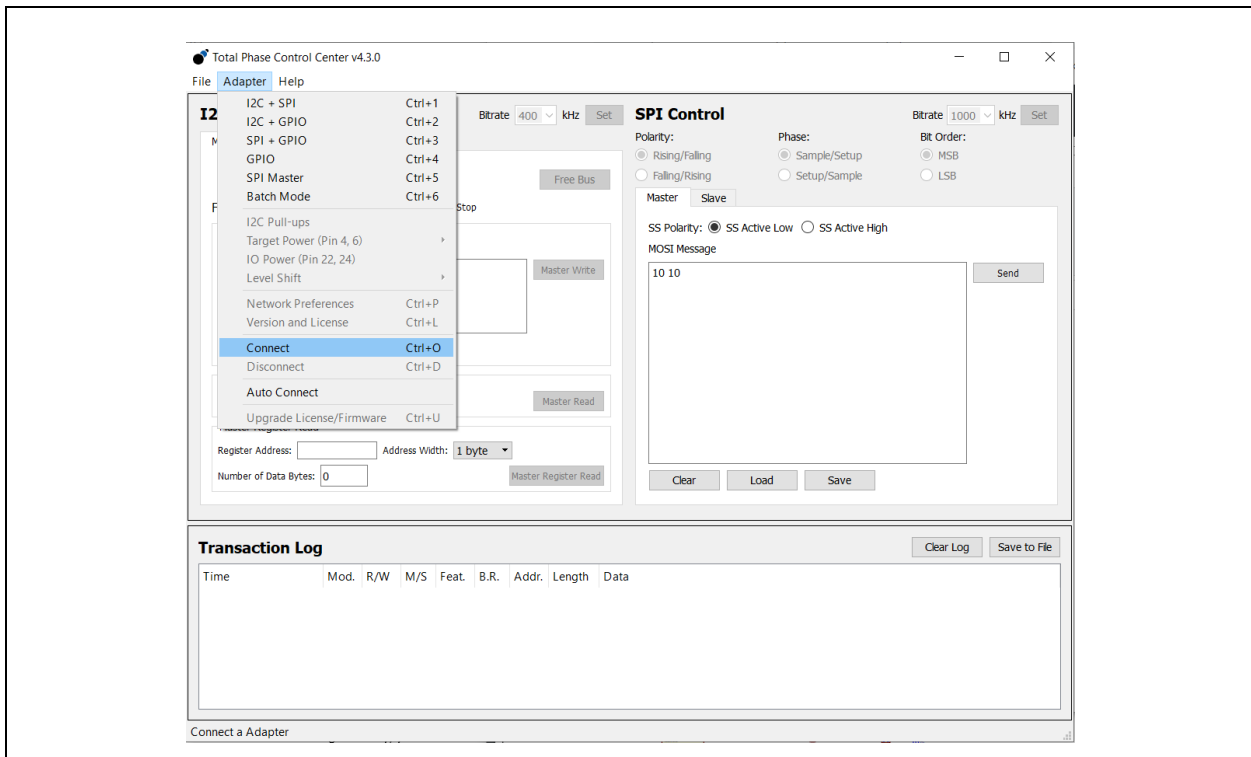
2. Open the Control Center. See [Figure 5-30](#).

FIGURE 5-30: CONTROL CENTER



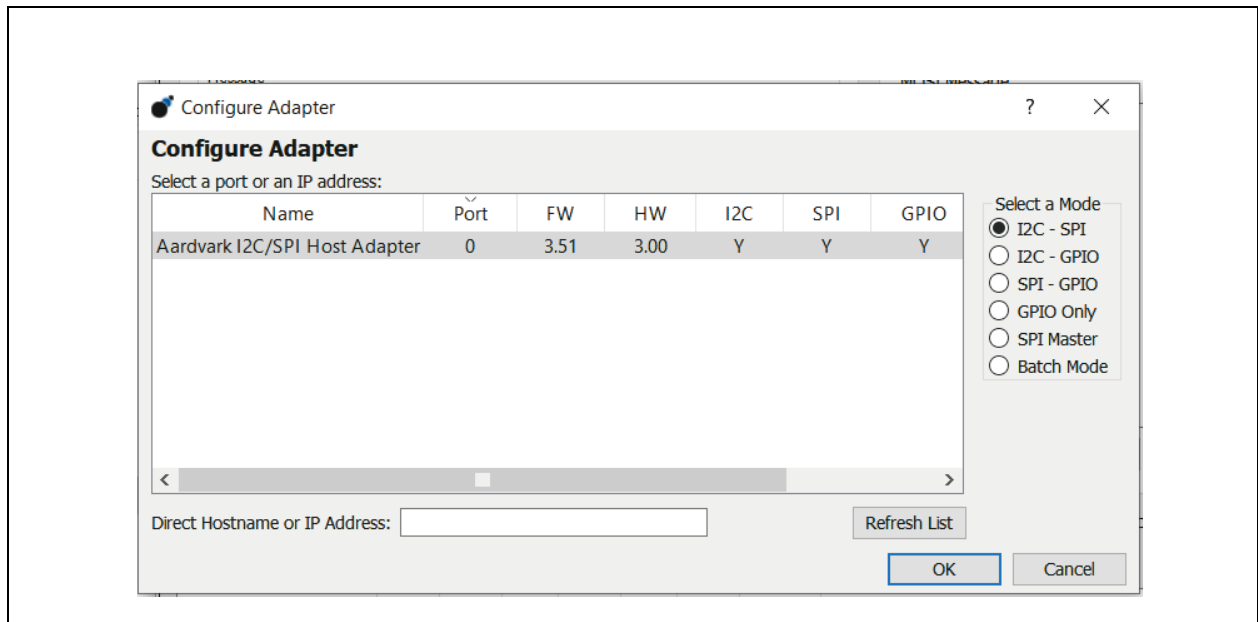
3. To bring up the list of available adapters, click on Adapter>Connect. See [Figure 5-31](#).

FIGURE 5-31: CONTROL CENTER ADAPTER LIST



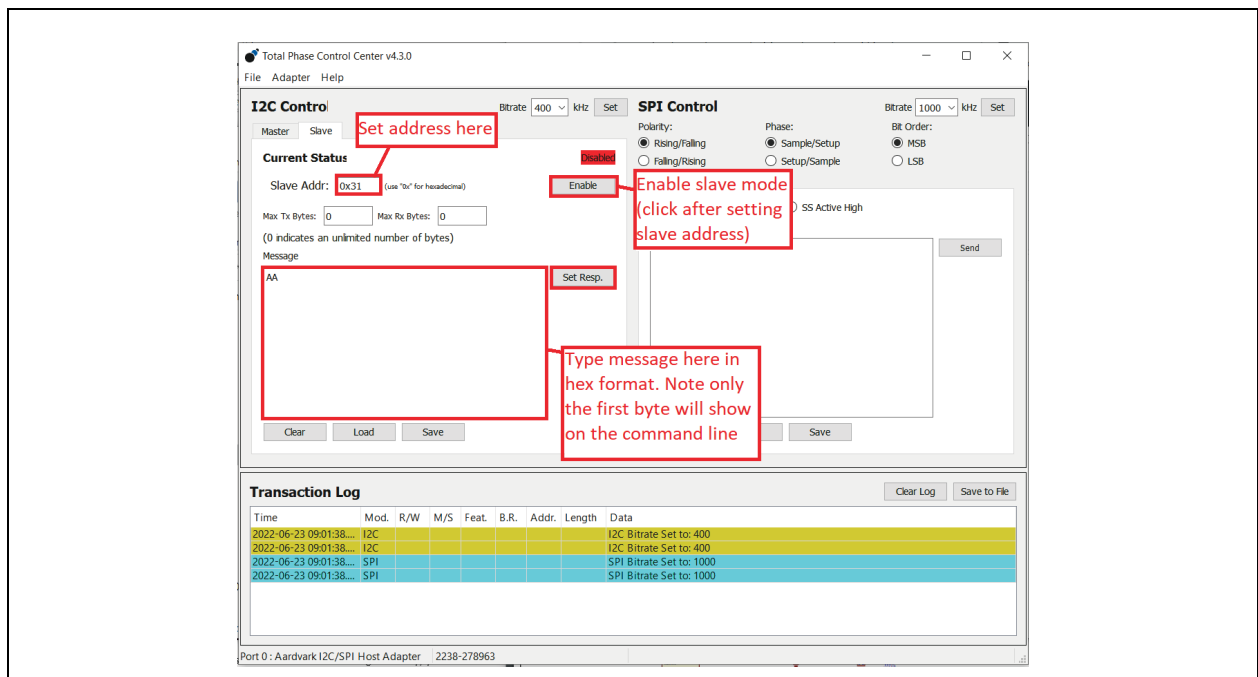
- Select the appropriate Aardvark device. Check if I²C-SPI is selected, and click on **OK**. See [Figure 5-32](#).

FIGURE 5-32: CONTROL CENTER CONFIGURE ADAPTER



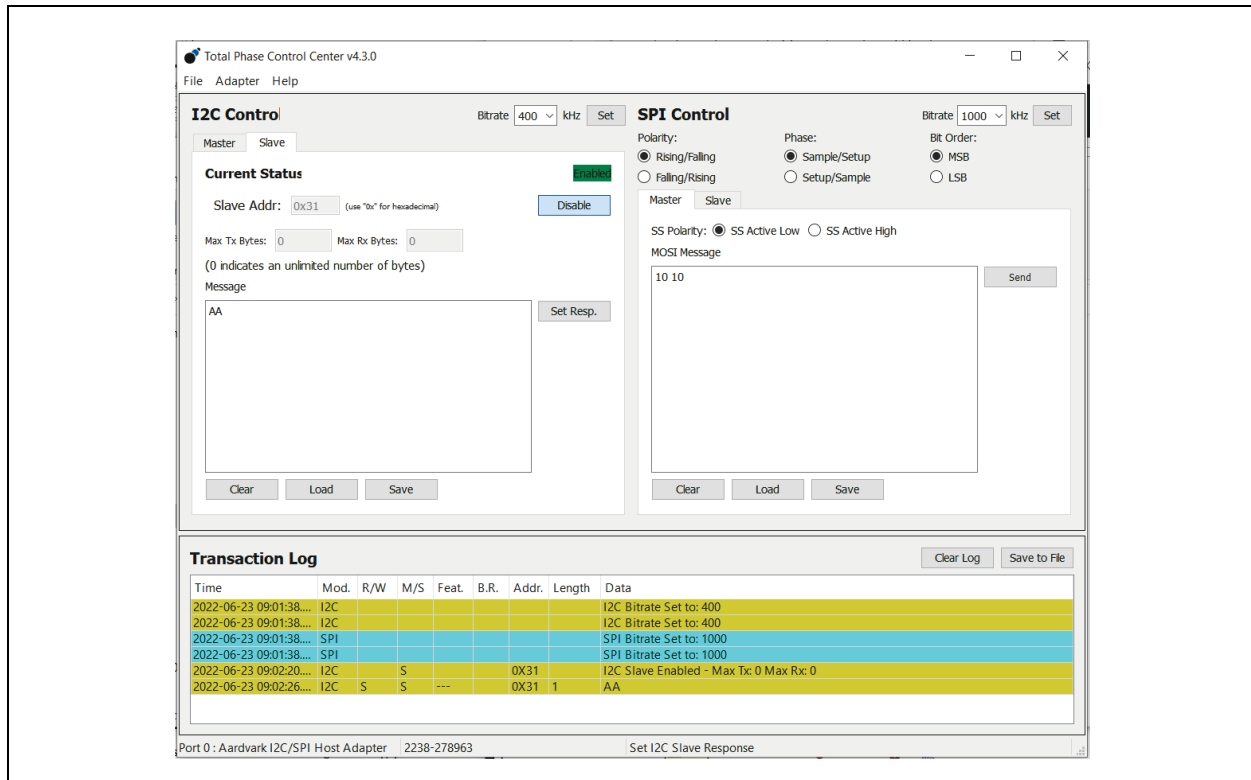
- In the I²C Control Panel, select Slave. Set address 0x31 and then click on **Enable**.
- To set the message read by PCI11414, type a message (in hex) in the message window and then click on **Set Resp.** See [Figure 5-33](#) and [Figure 5-34](#).

FIGURE 5-33: CONTROL I²C SLAVE SCREEN



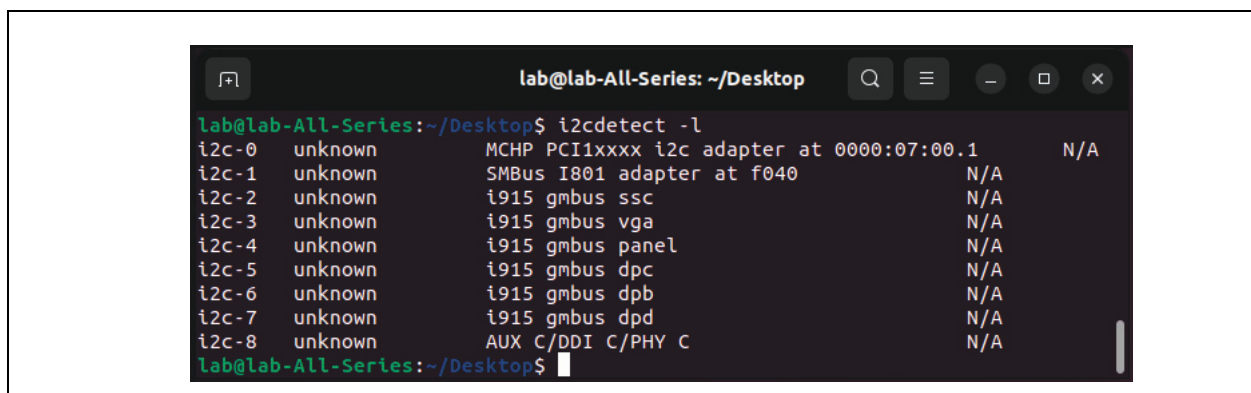
PCI11414

FIGURE 5-34: CONTROL I²C SLAVE CONFIGURATION



- To list all I²C devices connected to the computer that the EVB-PCI11414 is attached to, open a terminal and run `i2cdetect -l`. This will show the assigned device name as I²C-<DEVICE NUMBER> for each device as well as the name. PCI11414 will be listed as MCHP PCI1xxxx I²C adapter. Take note of the device number. See Figure 5-35.

FIGURE 5-35: I²CDETECT



- Run `i2cdetect <DEVICE NUMBER>`. Note that this command may need to be run with elevated privileges. When prompted to continue, type “y” and press enter.
- This will print a list of all the available addresses the command found on the specified bus number. In Figure 5-36, the command found two addresses: 0x31 and 0x57. 0x57 is a USB controller, and 0x31 is the Aardvark that was set up earlier.

FIGURE 5-36: I²CDETECT LISTED ADDRESSES

```

lab@lab-All-Series: ~/Desktop
lab@lab-All-Series:~/Desktop$ sudo i2cdetect 0
[sudo] password for lab:
Warning: Can't use SMBus Quick Write command, will skip some addresses
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-0.
I will probe address range 0x08-0x77.
Continue? [Y/n] y
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:
10:
20:
30: -- 31 --
40:
50: -- -- -- -- -- 57 --
60:
70:
lab@lab-All-Series:~/Desktop$

```

10. Run `i2cdump <BUS NUMBER> <ADDRESS>`. The address is in hex format. In this example, the address is 31. Note that this command may need to be run with elevated privileges. If prompted to continue, type “y” and press enter. See [Figure 5-37](#).

FIGURE 5-37: I²CDUMP

```

lab@lab-All-Series:~/Desktop$ sudo i2cdump 0 0x31
No size specified (using byte-data access)
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-0, address 0x31, mode byte
Continue? [Y/n] y
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  0123456789abcdef
00: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
10: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
20: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
30: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
40: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
50: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
60: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
70: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
80: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
90: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
a0: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
b0: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
c0: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
d0: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
e0: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
f0: aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa  ??????????????????
lab@lab-All-Series:~/Desktop$

```

11. This command will output the message that was set earlier in the Command Center.

APPENDIX A: DOCUMENTATION CONVENTIONS

This quick start guide uses the following conventions:

TABLE A-1: DOCUMENTATION CONVENTIONS

| Description | Represents | Examples |
|--|---|---|
| Arial Font: | | |
| Italic characters | Referenced books | <i>MPLAB[®] IDE User's Guide</i> |
| | Emphasized text | ...is the <i>only</i> compiler... |
| Initial caps | A window | the Output window |
| | A dialog | the Settings dialog |
| | A menu selection | select Enable Programmer |
| Quotes | A field name in a window or dialog | "Save project before build" |
| Underlined, italic text with right angle bracket | A menu path | <u><i>File>Save</i></u> |
| Bold characters | A dialog button | Click OK |
| | A tab | Click the Power tab |
| N'Rnnnn | A number in verilog format, where N is the total number of digits, R is the radix and n is a digit. | 4'b0010, 2'hF1 |
| Text in angle brackets < > | A key on the keyboard | Press <Enter>, <F1> |
| Courier New Font: | | |
| Plain Courier New | Sample source code | #define START |
| | Filenames | autoexec.bat |
| | File paths | c:\mcc18\h |
| | Keywords | _asm, _endasm, static |
| | Command-line options | -Opa+, -Opa- |
| | Bit values | 0, 1 |
| | Constants | 0xFF, 'A' |
| Italic Courier New | A variable argument | <i>file.o</i> , where <i>file</i> can be any valid filename |
| Square brackets [] | Optional arguments | mcc18 [options] <i>file</i> [options] |
| Curly brackets and pipe character: { } | Choice of mutually exclusive arguments; an OR selection | errorlevel {0 1} |
| Ellipses... | Replaces repeated text | var_name [, var_name...] |
| | Represents code supplied by user | void main (void) { ... } |

APPENDIX B: REVISION HISTORY

TABLE B-1: REVISION HISTORY

| Revision Level & Date | Section/Figure/Entry | Correction |
|---------------------------|----------------------|------------|
| DS50003610A (11-30-23) | Initial release | |

THE MICROCHIP WEBSITE

Microchip provides online support via our WWW site at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: <http://microchip.com/support>

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable" Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at <https://www.microchip.com/en-us/support/design-help/client-support-services>.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGL00, JukeBlox, KeeLoq, Klear, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, TimeCesium, TimeHub, TimePictra, TimeProvider, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, EyeOpen, GridTime, IdealBridge, IGaT, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, MarginLink, maxCrypto, maxView, memBrain, Mindi, MIWI, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mSiC, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICKIT, PICtail, Power MOS IV, Power MOS 7, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQL, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, Turing, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestlC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.
© 2023, Microchip Technology Incorporated and its subsidiaries.

All Rights Reserved.

ISBN: 978-1-6683-3551-2

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.



MICROCHIP

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://www.microchip.com/support>
Web Address:
www.microchip.com

Atlanta

Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX

Tel: 512-257-3370

Boston

Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago

Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas

Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Novi, MI
Tel: 248-848-4000

Houston, TX

Tel: 281-894-5983

Indianapolis

Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles

Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC

Tel: 919-844-7510

New York, NY

Tel: 631-435-6000

San Jose, CA

Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto

Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880-3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820