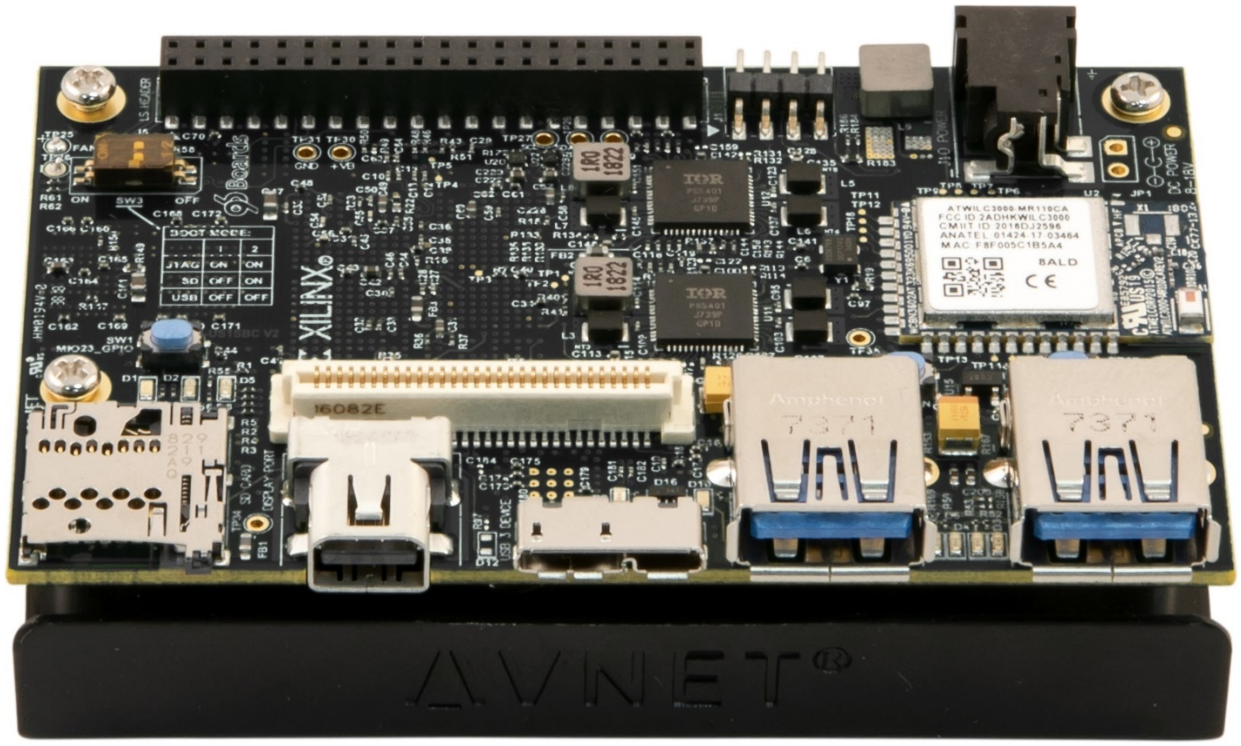


Ultra96-V2 Getting Started Guide

Version 2.1

LIT# Ultra96-V2-GSG-v2-1

Document Control

Document Version: 2.1

Document Date: 03 Mar 2021

Prior Version History

Version	Date	Comment
1.0	25 Jun 2019	Initial Ultra96-V2 Getting Started Guide (30 May 2019 image)
1.1	01 Oct 2019	Updated based on 17 Sep 2019 image
2.0	18 Dec 2020	Update to the 2020.1 out-of-box image (23 Nov 2020 image); Added PMIC read and program instructions.
2.1	03 Mar 2021	Updated 2020.1 out-of-box image (03 Mar 2021 image) that fixes Wi-Fi and App issues. Instructions for iperf apps added.

Contents

1 Getting Started with Ultra96-V2 7
2 What's Inside the Box? 8
2.1 Optional add-on items: 8
3 What's on the Web? 9
3.1 Official Documentation: 9
3.2 Tutorials and Reference Designs: 9
3.3 Trainings and Videos: 9
4 Ultra96-V2 Key Features 10
5 Ultra96-V2 Basic Setup and Operation 12
6 Example Design 13
7 Run the Out of Box Design 13
8 Connect to Webserver 17
9 Ultra96-V2 GPIO LEDs Example Project 19
10 Tutorials 21
11 Access Ultra96-V2 Linux Terminal over SSH 22
12 Connect to External Wi-Fi 25
13 Example Projects 28
13.1 Ultra96-V2 GPIO LEDs 28
13.2 WiFi iperf3 Server Performance Test 28
13.3 WiFi iperf3 Client Performance Test 30
13.4 OpenAMP Echo Test Application 32
13.5 OpenAMP Matrix Multiplication 33
13.6 OpenAMP Proxy Application 35
14 Reading On-board PMIC and MPSoC Telemetry 37
15 PMIC Version Check and Update 41
15.1 PMIC Version Checking 41
15.2 PMIC Reprogramming 46
16 Ethernet Gadget 47
17 Power Off 48
18 Getting Help and Support 49
18.1 Avnet Support 49
18.2 Xilinx Support 50
19 Installing and Licensing Xilinx Software 51
19.1 Install Vivado Design Suite, Design Edition 51

20 Certification Disclaimer 55
21 Safety Warnings 55
22 RF Certification... 55

Figures

Figure 1 – Ultra96-V2 7
Figure 2 – Ultra96-V2 Block Diagram 11
Figure 3 – Ultra96-V2 Topology 12
Figure 4 – Ultra96-V2 SW3 Boot Mode Switch Location 13
Figure 5 – Ultra96-V2 with older 3-pin JTAG/UART Pod 14
Figure 6 – Ultra96-V2 with newer 4-pin JTAG/UART Pod 15
Figure 7 – Press the SW4 Power On Button 16
Figure 8 – Ultra96-V2 As An Available Network 17
Figure 9 – Connected to Ultra96-V2 Webserver 18
Figure 10 – Ultra96-V2 User LEDs 19
Figure 11 – Ultra96-V2 GPIO LEDs 19
Figure 12 – LED0/D3 Changed to Steady-state ON 20
Figure 13 – Ultra96-V2 Tutorials 21
Figure 14 – TeraTerm New Connection 22
Figure 15 – SSH Terminal Settings 23
Figure 16 – Click Continue 23
Figure 17 – SSH Authentication 24
Figure 18 – Ultra96-V2 Terminal 24
Figure 19 – WiFi Setup 25
Figure 20 – Discovering Networks 25
Figure 21 – Successfully Connected 26
Figure 22 – Determine IP Address Assigned to Ultra96-V2 26
Figure 23 – Connected to Ultra96-V2 Via External Wi-Fi 27
Figure 24 – WiFi iperf3 Server Performance Test 28
Figure 25 – iperf3 results on Host from Server Test 29
Figure 26 – iperf3 results on Ultra96-V2 from Server Test 29
Figure 27 – WiFi iperf3 Client Performance Test 30
Figure 28 – iperf3 Results on Host from Client Test 31
Figure 29 – iperf3 Results on Ultra96-V2 from Client Test 31
Figure 30 – OpenAMP Echo Description 32
Figure 31 – Echo Test Starting 32
Figure 32 – Echo Test Complete 33

Figure 33 – OpenAMP Matrix Multiplication Description 33
Figure 34 – Matrix Multiply Results 34
Figure 35 – OpenAMP Proxy Description 35
Figure 36 – Enter Proxy Info 35
Figure 37 – OpenAMP Proxy Results 36
Figure 38 – I2Cdetect 37
Figure 39 – PMIC on I2C Mux Channel 4 37
Figure 40 – PMICs on I2C-6 38
Figure 41 – Telemetry from PMIC 0x15/45 38
Figure 42 – Telemetry from PMIC 0x13/43 39
Figure 43 – Telemetry from MPSoC Sysmon 39
Figure 44 – Telemetry from PMIC 0x14/44 40
Figure 45 – PMIC Config Files 41
Figure 46 – PMICs Detected On I2C Bus 6 43
Figure 47 – Example datecode October 30 for Device 0x13 44
Figure 48 – Compare PMIC Config to a File, 0x13 or 0x14 45
Figure 49 – Compare PMIC Config to a File, 0x15 45
Figure 50 – Carefully Read, then Confirm to Proceed 46
Figure 51 – Programming 0x13 Succeeded 46
Figure 52 – Power Down Initiated Through Short Press of SW4 48
Figure 53 – Voucher Confirmation 52
Figure 54 – Generate Node-Locked 52
Figure 55 – Select Host Information 53

1 Getting Started with Ultra96-V2

The Avnet Ultra96-V2 enables hardware and software developers to explore the capabilities of the Zynq® UltraScale+™ MPSoC. Designers can create or evaluate designs for both the Zynq Processor Subsystem (PS) and the Programmable Logic (PL) fabric.

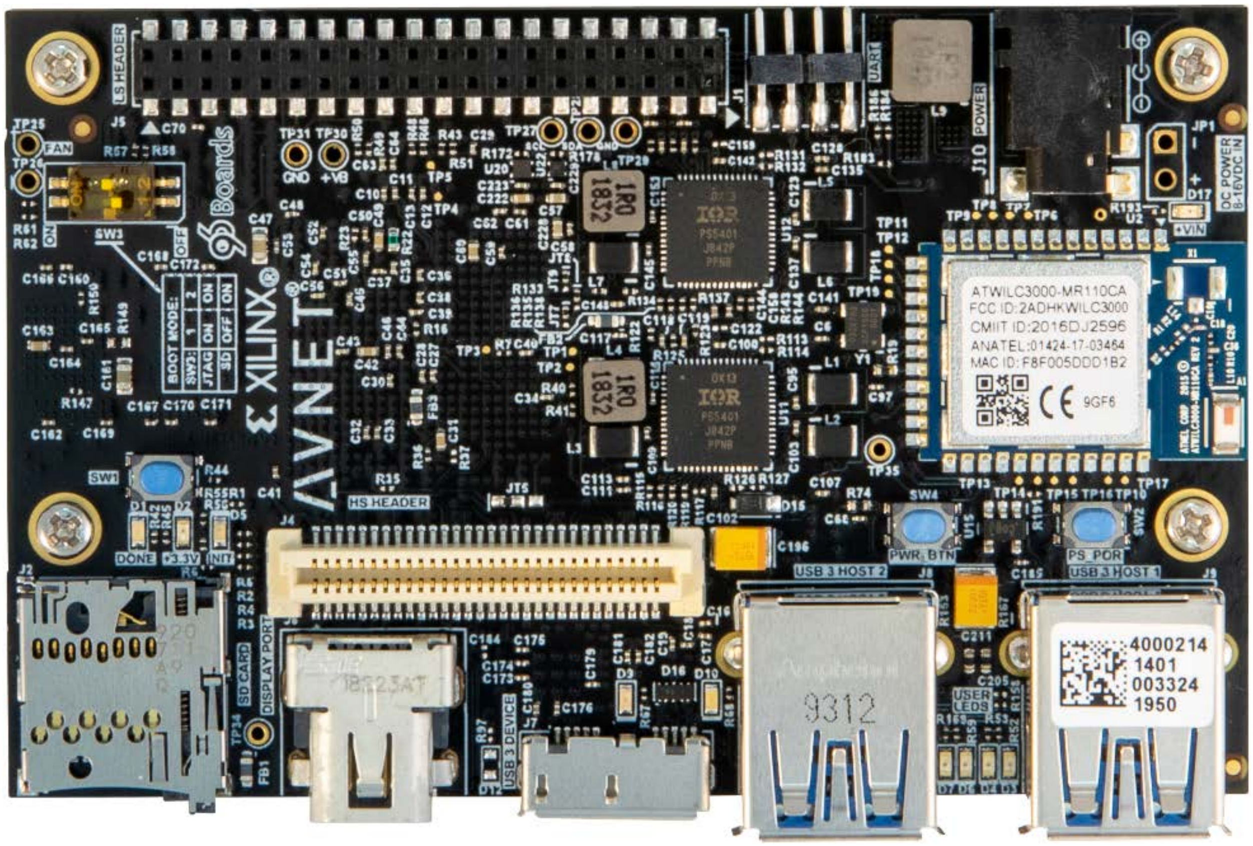


Figure 1 – Ultra96-V2

This ***Getting Started Guide*** will outline the steps to setup the Ultra96-V2 hardware. It documents the procedure to run a Linux design running on the Quad-core ARM Cortex-A53 MPCore Processing System (PS).

2 What's Inside the Box?

- Ultra96-V2 development board
- 16GB microSD card with SD adapter and jewel case
- Voucher for SDSoC license from Xilinx
  - Not required when using Vitis but will be included through 2021
- Quick Start Instruction card

2.1 Optional add-on items:

- External 96Boards compliant power supply kit
  - ***A power supply is required to operate the board***
  - AES-ACC-U96-4APWR (12V, 4A, International plugs)
  - http://avnet.me/96boardpower
- USB-to-JTAG/UART pod for Ultra96-V2
  - The JTAG/UART pod is not required to operate the board, but the pod does provide terminal communications, JTAG programming, and JTAG debug, so the pod is highly recommended.
  - AES-ACC-U96-JTAG
  - http://avnet.me/Ultra96JTAG
- 96Boards Click Mezzanine for adding Click boards to the Ultra96-V2
  - Mezzanine only -- AES-ACC-U96-ME-MEZ
  - Starter kit including 3 Click boards -- AES-ACC-U96-ME-SK
  - http://avnet.me/ClickMezzanine
  - Additional Click boards available at http://avnet.me/Click
- 96Boards Dual Camera Mezzanine for adding image sensors to the Ultra96-V2
  - AES-ACC-U96-ONCAM-MEZ
  - http://avnet.me/DualCameraMezz
- 96Boards Quad Ethernet Mezzanine from Opsero
  - https://opsero.com/product/96b-quad-ethernet-mezzanine/
- DisplayPort Monitor
  - miniDP-to-DP cable required
- HDMI Monitor
  - Requires **Active** miniDP-to-HDMI adapter or cable
- USB-to-Ethernet adapter
  - Ultra96-V2 does NOT have native wired Ethernet, but you can easily add wired Ethernet with inexpensive and readily available adapters

See http://avnet.me/Ultra96_Accessories for specific part numbers and other suggestions.

3 What's on the Web?

Ultra96-V2 is a community-oriented kit, with all materials being made available through the http://avnet.me/ultra96-v2 community website.

3.1 Official Documentation:

- Getting Started Guide
- Hardware User Guide
- Schematics
- Bill of Materials
- Mechanical drawing
- 3D Model
- Programmable logic (PL) master user constraints
- Board definition files for Vivado integration available at https://github.com/Avnet/bdf.

3.2 Tutorials and Reference Designs:

- https://www.hackster.io/avnet/products/ultra96-v2

3.3 Trainings and Videos:

- Live and On-Demand Technical Training Courses at http://avnet.me/TTC

4 Ultra96-V2 Key Features

- Linaro 96Boards Consumer Edition compatible
- Xilinx Zynq UltraScale+ MPSoC ZU3EG SBVA484
    - AES-ULTRA96-V2-G: -E
    - AES-ULTRA96-V2-I-G : -I
- Memory
    - Micron 2 GB (512M x32) LPDDR4 Memory
    - microSD Socket with microSD card
        - AES-ULTRA96-V2-G: Ships with Delkin Utility MLC 16GB card
        - AES-ULTRA96-V2-I-G: Ships with Delkin Utility+ MLC 16GB card
- Microchip Wi-Fi / Bluetooth
- Renesas (IDT) VersaClock 6E clock generator
- Infineon and ON Semiconductor high efficiency power management
- Mini DisplayPort (MiniDP or mDP)
- 1x USB 3.0 Type Micro-B upstream port
- 2x USB 3.0 Type A downstream ports
- 1x USB 2.0 Type A downstream port (on High-speed expansion header)
- 40-pin Low-speed expansion header
- 60-pin High-speed expansion header
- Thermally dissipative device
    - Rev 1, datecodes 1935 and earlier: Mounted on thermal bracket with fan
    - Rev 1, datecodes 1936 and later: Custom Aavid passive heatsink
    - Rev 2: Custom Aavid passive heatsink

Note that there is no on-board, wired Ethernet interface, although this can be added through the expansion connectors or USB. All communications without add-ons must be done via USB, Wi-Fi, JTAG, or expansion interface

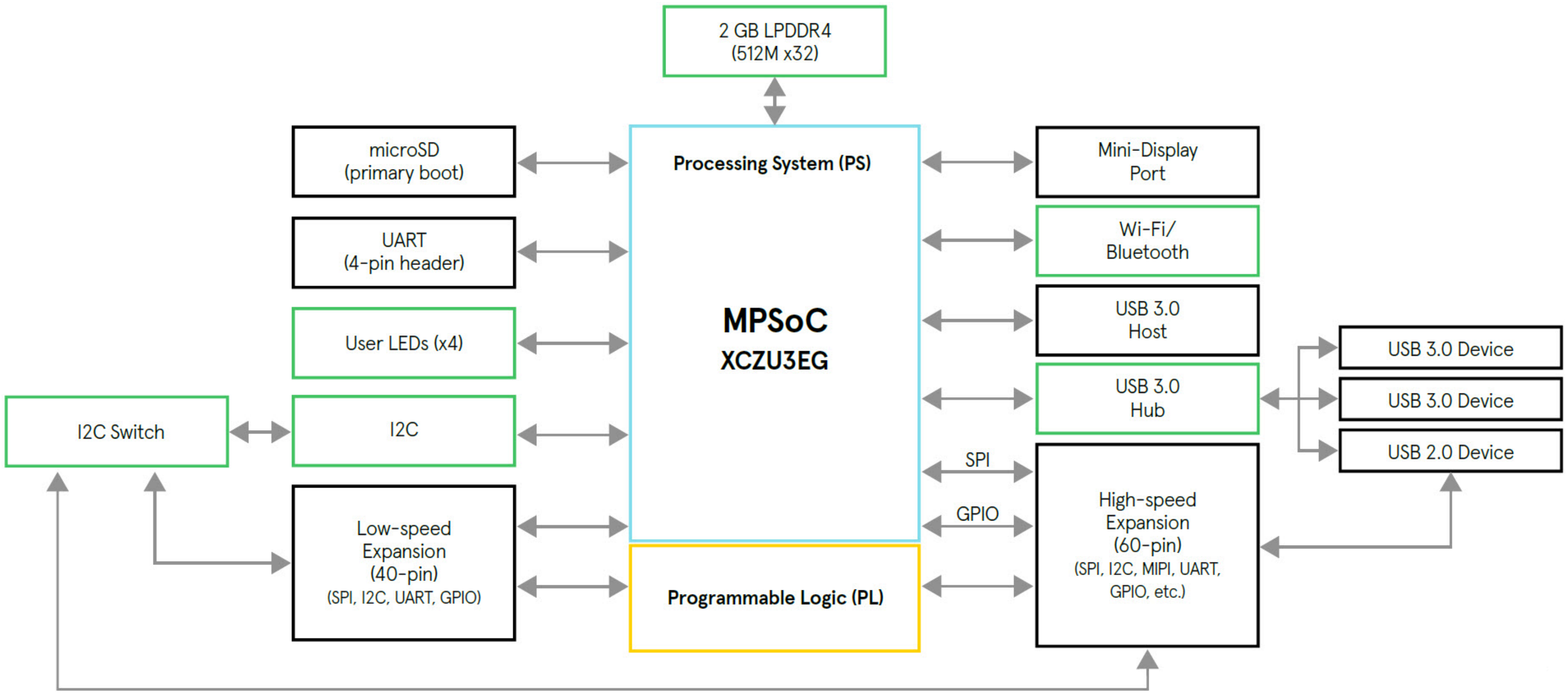


Figure 2 – Ultra96-V2 Block Diagram

5 Ultra96-V2 Basic Setup and Operation

The functionality of the Ultra96-V2 is determined by the application booted from the non-volatile memory – by default the microSD Card. This *Getting Started Guide* allows system developers to exercise and demonstrate multiple circuits through PetaLinux, including:

- SSH Terminal Access
- GPIO LEDs
- Wi-Fi

In addition to the items included in the kit, you will also need the following to complete the exercises in this tutorial.

- Ultra96 USB-to-JTAG/UART Pod (required for terminal access)

or

- Monitor (requires connection to miniDP port), keyboard, mouse

An Ultra96-V2 image in its expected out-of-box configuration is shown below along with various topology components highlighted.

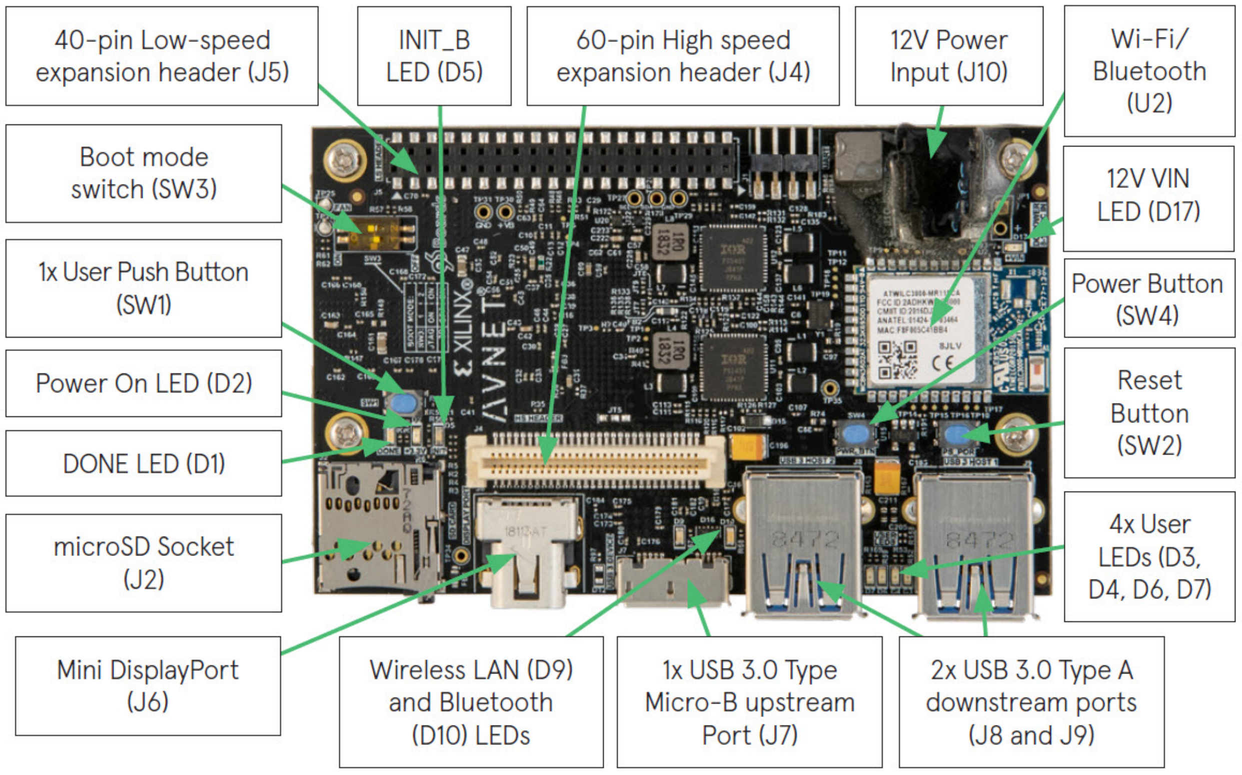

40-pin Low-speed expansion header (J5)
INIT_B LED (D5)
60-pin High speed expansion header (J4)
12V Power Input (J10)
Wi-Fi/ Bluetooth (U2)
Boot mode switch (SW3)
12V VIN LED (D17)
1x User Push Button (SW1)
Power Button (SW4)
Power On LED (D2)
Reset Button (SW2)
DONE LED (D1)
microSD Socket (J2)
4x User LEDs (D3, D4, D6, D7)
Mini DisplayPort (J6)
Wireless LAN (D9) and Bluetooth (D10) LEDs
1x USB 3.0 Type Micro-B upstream Port (J7)
2x USB 3.0 Type A downstream ports (J8 and J9)

Figure 3 – Ultra96-V2 Topology

6 Example Design

The Ultra96-V2 example design must first be written to the 16GB microSD Card, which ships blank.

1. Please download the image and instructions using these links:
   Instructions: http://avnet.me/Ultra96-flash-write
   Image: http://avnet.me/ultra96-v2-oob
   MD5 Checksum for Image: 7c2735d611de6c89c118c1bae1e745ae

2. Complete the process to write the image to your 16GB card.

3. Insert the microSD card into the Ultra96-V2 card cage J2.

7 Run the Out of Box Design

1. A terminal program is recommended if you want to view terminal output. TeraTerm was used in this example which can be downloaded from the TeraTerm project on the SourceForge Japan page: ttssh2.sourceforge.jp Install TeraTerm or another terminal program of your choice.

2. Set the Ultra96-V2 boot mode switch SW3 to SD Card boot mode as shown below with Switch 1 in the OFF position and Switch 2 in the ON position.

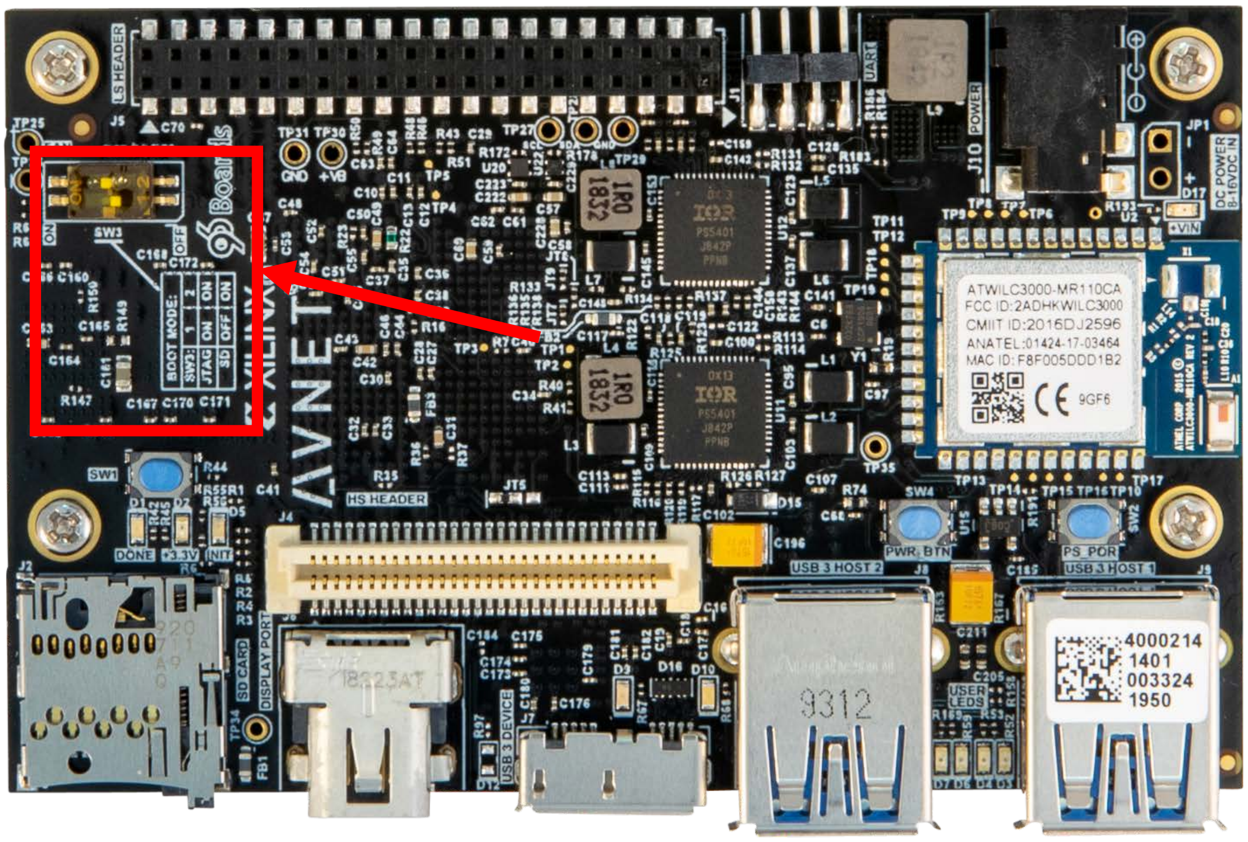


Figure 4 – Ultra96-V2 SW3 Boot Mode Switch Location

3. If you will be using a USB-to-JTAG/UART Pod for terminal access, plug that into J1 and J3 before plugging in 12V power. Note that the Ultra96-V2 is compatible with Pods that have 3-pin and 7-pin receptacles (align as seen in Figure 5 below), as well as Pods with 4-pin and 8-pin receptacles (as seen in Figure 6 below).

   You can find more information about the JTAG/UART pod here: http://avnet.me/JTAG-UART-pod-vref

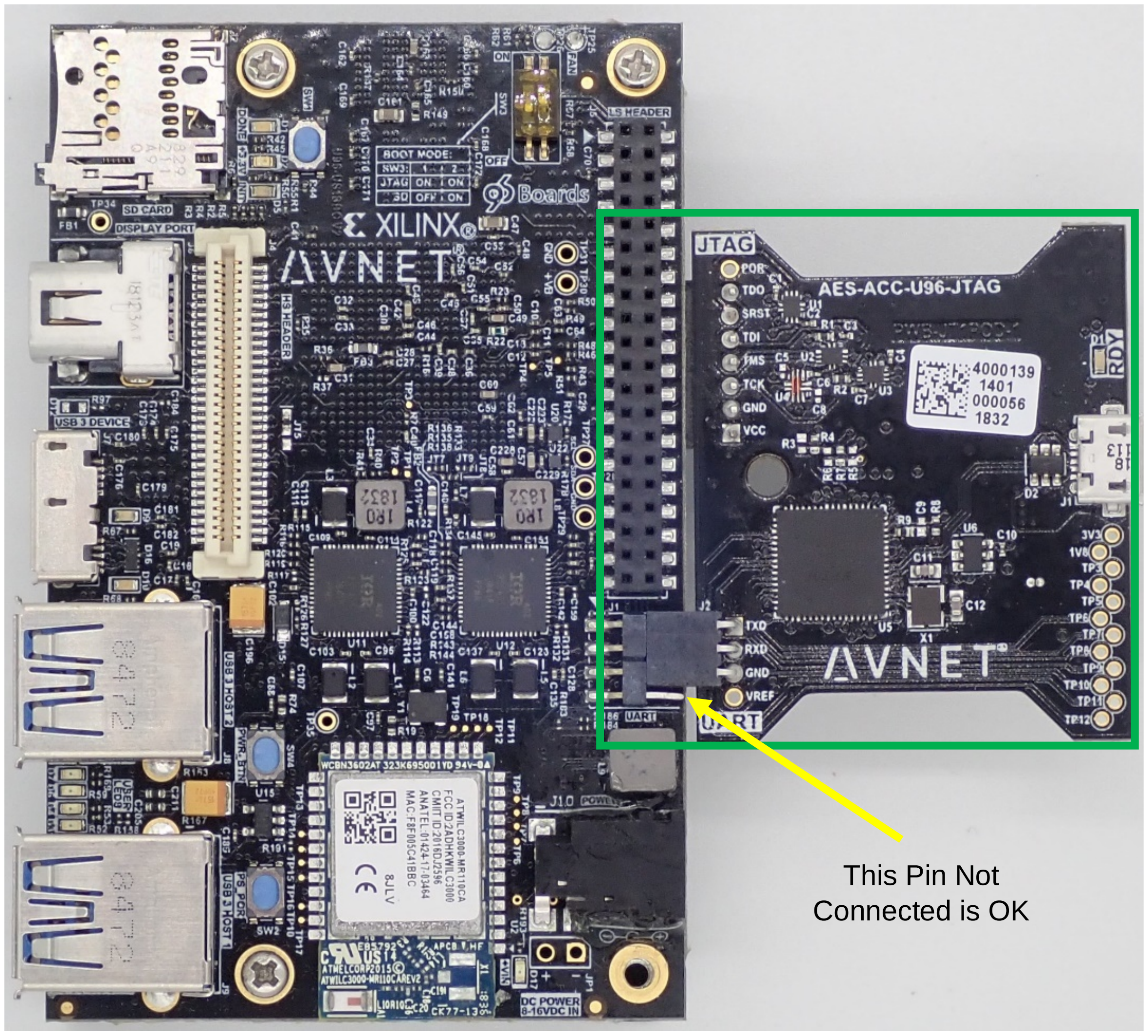


Figure 5 – Ultra96-V2 with older 3-pin JTAG/UART Pod

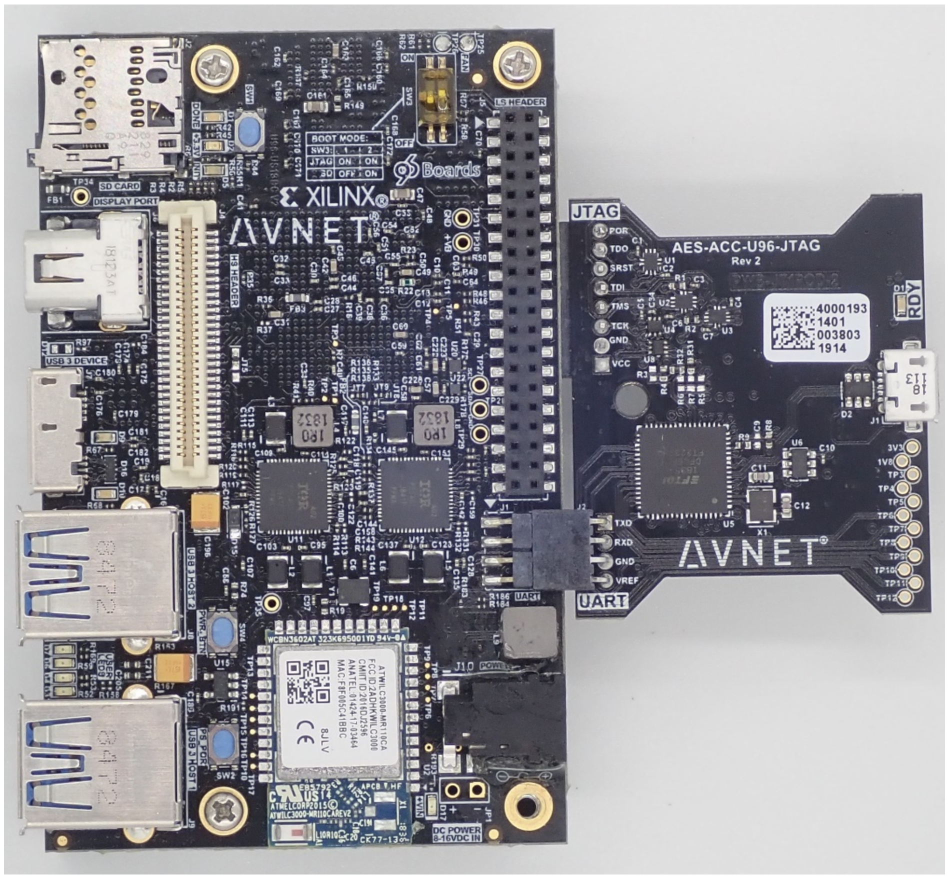


Figure 6 – Ultra96-V2 with newer 4-pin JTAG/UART Pod

4. If you are viewing the display on a monitor, plug in the appropriate cable to the monitor. Also, plug in a keyboard/mouse.

    - mini-DP to DisplayPort cable with DisplayPort Monitor
    - ACTIVE mini-DP to HDMI adapter/cable with HDMI Monitor

5. Plug in your 12V Barrel Jack power supply into a wall outlet and then connect the barrel jack to J10 on your Ultra96-V2. Green Vin status LED D17 will light, but the board is not yet powered on.

    Note: DC power supply is not included in the Ultra96-V2 kit but can be purchased separately – *http://avnet.me/96boardpower*

The Ultra96 USB-to-JTAG/UART Pod ships with pre-programmed firmware that allows the JTAG interface to be recognized by Xilinx Vivado software. Additionally, most host machines will also automatically install the driver for the Serial Terminal interface.

6. If using the JTAG/Pod, do the following:
   - Plug a microUSB cable between the Pod's microUSB Port (J1) and a host computer.
   - If the serial terminal drivers do not automatically install, you can manually install the driver for the FT2232H device. Visit www.ftdichip.com/Drivers/VCP.htm then download and install the appropriate driver for your operating system.
   - Launch your Serial Terminal with settings of 115200-8-N-1.

7. Press and release the power button (SW4).

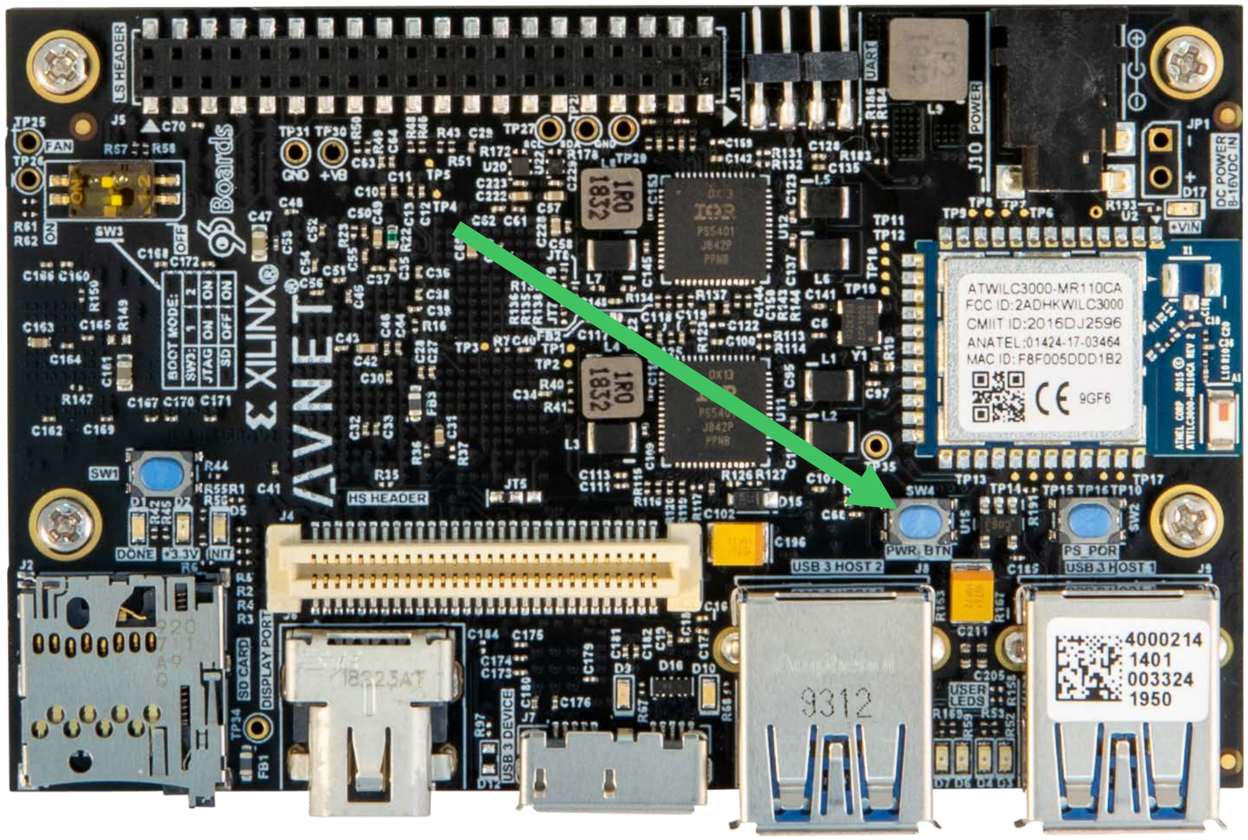


Figure 7 – Press the SW4 Power On Button

8. The Green Power On LED (D2), Red INIT_B LED (D5) and the Green User LEDs should illuminate. After a few seconds, the INIT_B LED will turn off and the Blue DONE LED (D1) will illuminate. You will immediately see output to the terminal screen as Linux boots. At ~30 seconds, the Green User LED D7 will blink in a heartbeat fashion. The miniDP output shows a Chromium browser page.

8 Connect to Webserver

1. From your Wi-Fi capable host machine, show the available Wi-Fi networks. You should see an Open network called Ultra96-V2_<MAC_ADDRESS> as shown below. Note that the MAC Address shown will be different than the image below as it will match your specific board's MAC Address.

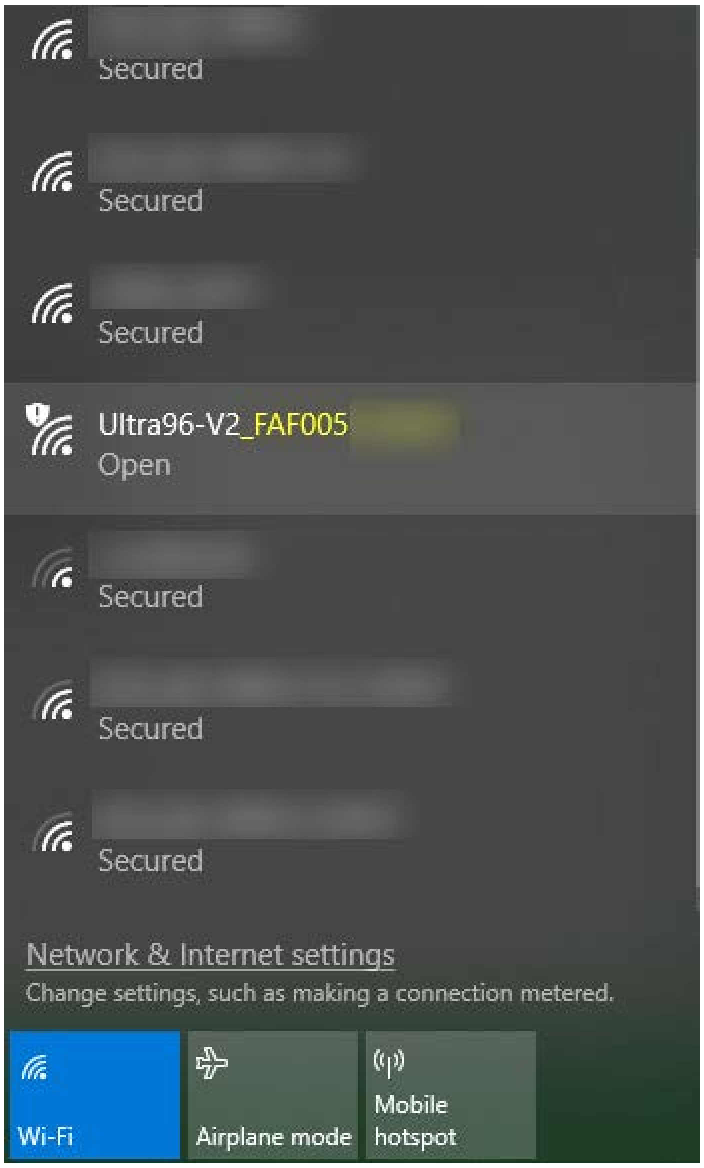
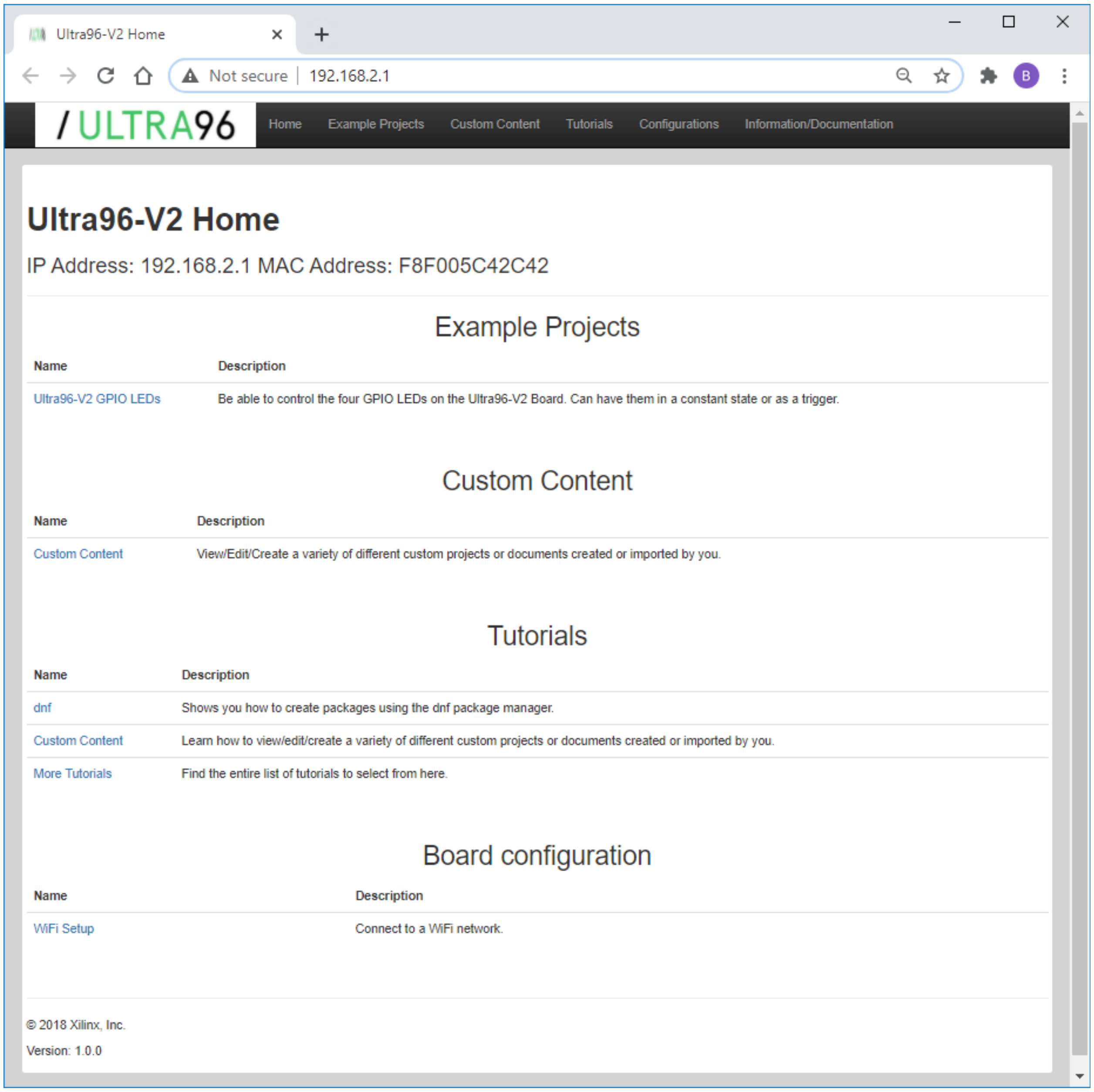


Figure 8 – Ultra96-V2 As An Available Network

2. Select the Ultra96-V2 Network and Connect. Note that if you previously connected to Ultra96-V2 to your network, that connection is persistent and you can skip this step. You will still see the Access Point, but you will not be able to connect to a different network in the presence of your old network until you delete the Wi-Fi credentials and reboot (`/var/lib/connman/ultra96.config`).

3. Once connected, open a browser on the connected machine, and browse to the IP address of the board, which is http://192.168.2.1. The browser page will show like below.



Ultra96-V2 Home

IP Address: 192.168.2.1 MAC Address: F8F005C42C42

Example Projects

Name	Description
Ultra96-V2 GPIO LEDs	Be able to control the four GPIO LEDs on the Ultra96-V2 Board. Can have them in a constant state or as a trigger.

Custom Content

Name	Description
Custom Content	View/Edit/Create a variety of different custom projects or documents created or imported by you.

Tutorials

Name	Description
dnf	Shows you how to create packages using the dnf package manager.
Custom Content	Learn how to view/edit/create a variety of different custom projects or documents created or imported by you.
More Tutorials	Find the entire list of tutorials to select from here.

Board configuration

Name	Description
WiFi Setup	Connect to a WiFi network.

© 2018 Xilinx, Inc.

Version: 1.0.0

Figure 9 – Connected to Ultra96-V2 Webserver

9 Ultra96-V2 GPIO LEDs Example Project

The Ultra96-V2 LEDs are D3, D4, D6, and D7, shown below:

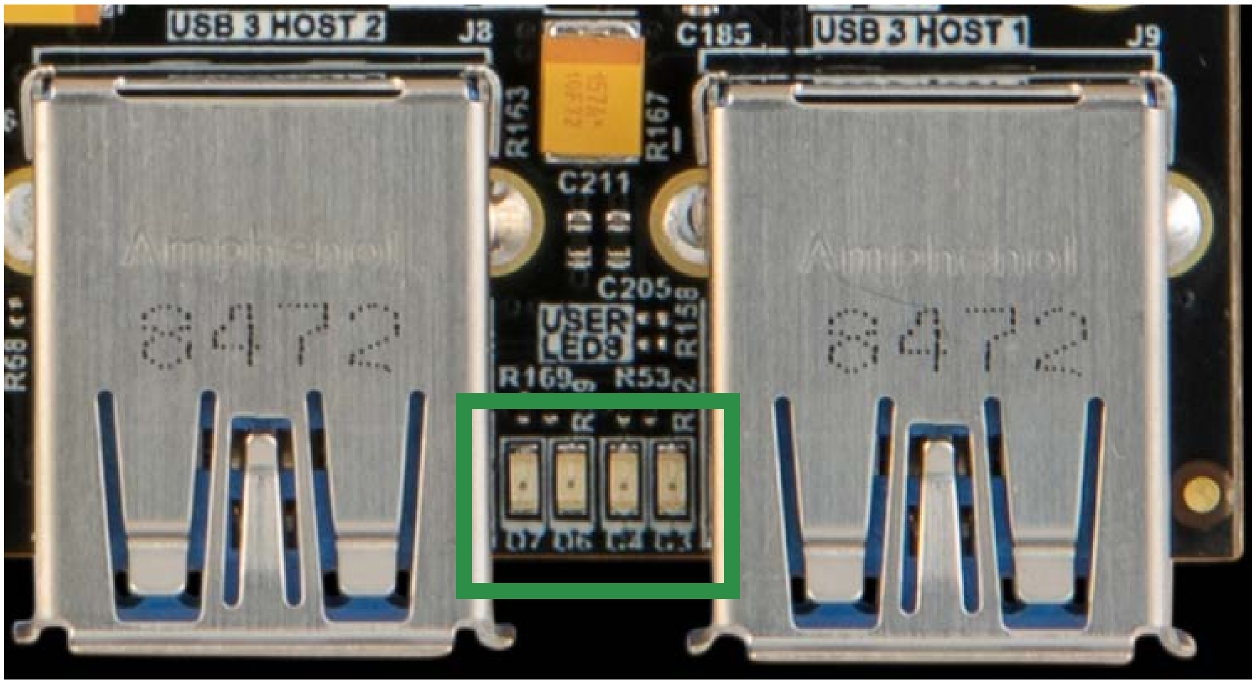
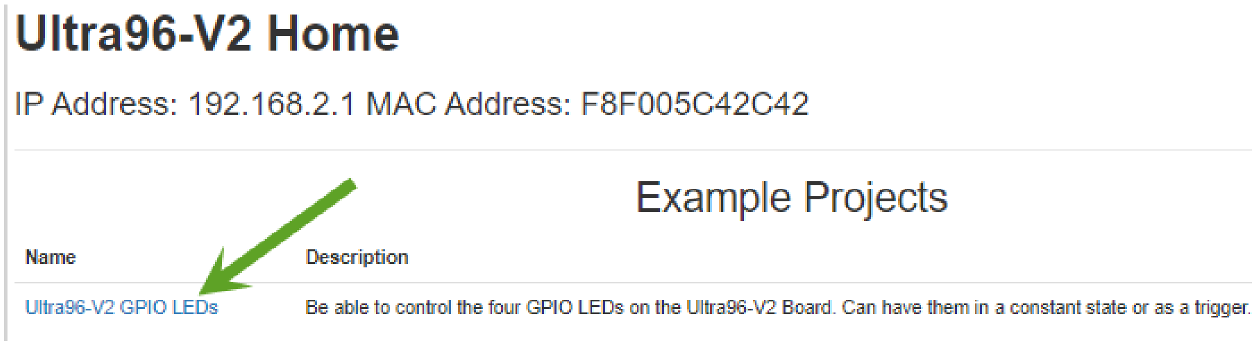


Figure 10 – Ultra96-V2 User LEDs

1. Next we want to access the Ultra96-V2 GPIO LEDs example project. From the Ultra96-V2 home page select **Ultra96-V2 GPIO LEDs** example project



Ultra96-V2 Home

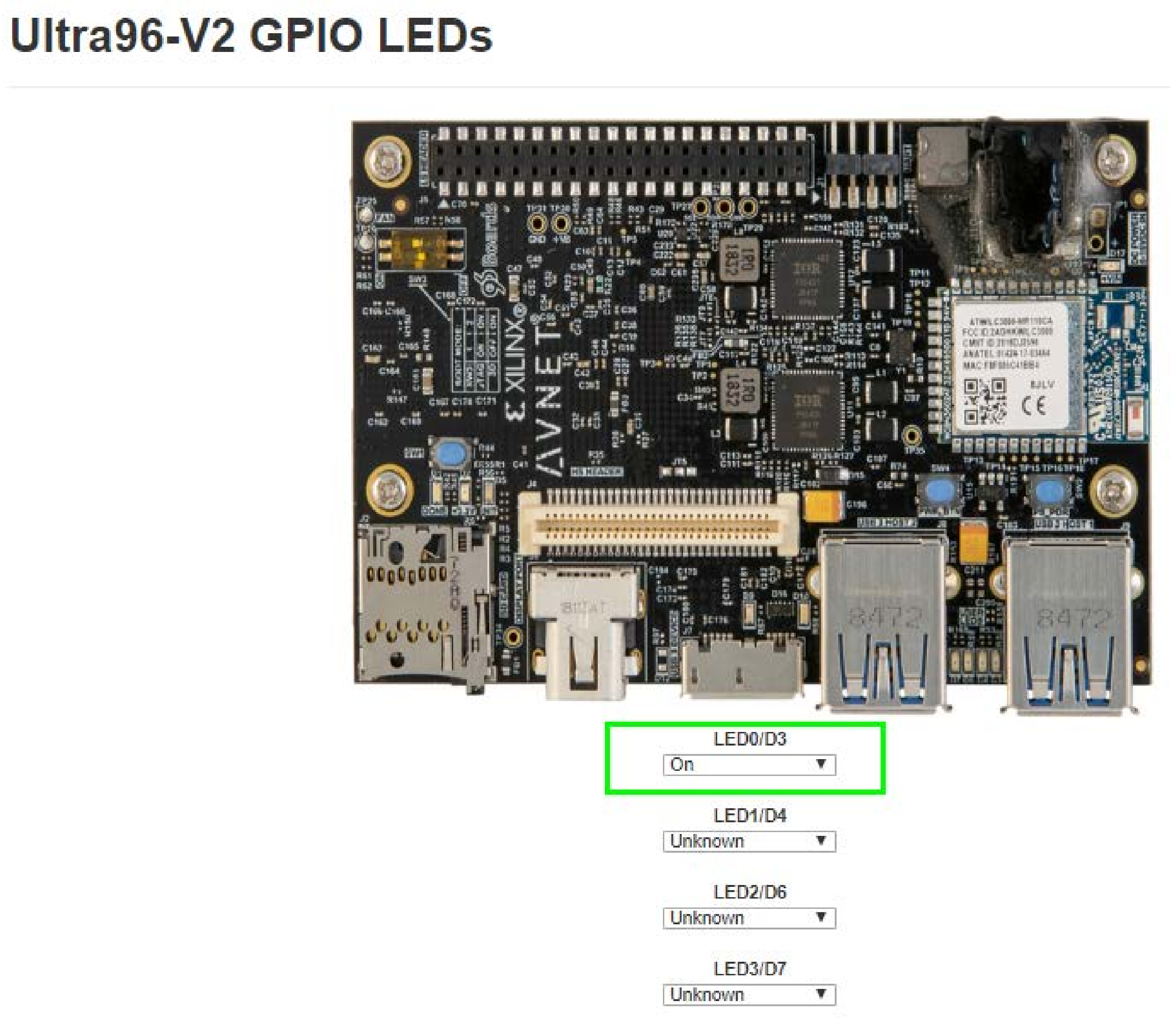
IP Address: 192.168.2.1 MAC Address: F8F005C42C42

Example Projects

Name	Description
Ultra96-V2 GPIO LEDs	Be able to control the four GPIO LEDs on the Ultra96-V2 Board. Can have them in a constant state or as a trigger.

Figure 11 – Ultra96-V2 GPIO LEDs

2. All LEDs will be at an unknown state to begin with. Select the drop down menus and begin changing the status of the GPIO LEDs. The LED blinking upon boot-up is D3, so consider changing that one first to “On.”


Ultra96-V2 GPIO LEDs

LED0/D3: On

LED1/D4: Unknown

LED2/D6: Unknown

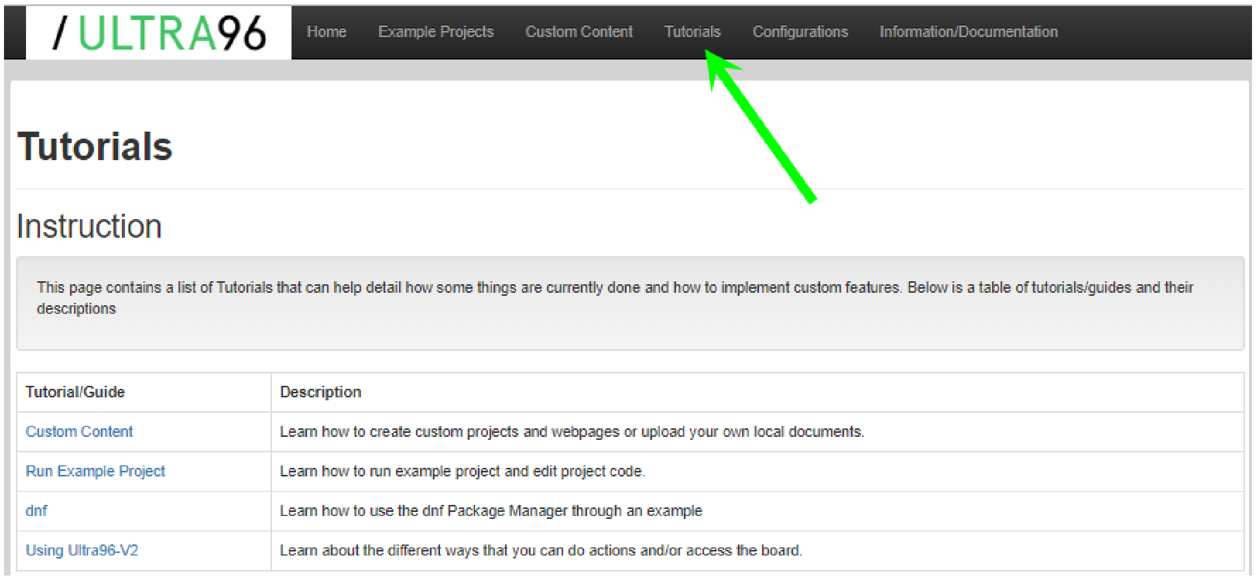
LED3/D7: Unknown

Figure 12 – LED0/D3 Changed to Steady-state ON

3. Scroll to the bottom of the webpage and you will see a definition table for various LED selection options.

10 Tutorials

1. Select the **Tutorials** tab at the top of the page. You will be redirected to the Tutorials page.



Tutorials

Instruction

This page contains a list of Tutorials that can help detail how some things are currently done and how to implement custom features. Below is a table of tutorials/guides and their descriptions

Tutorial/Guide	Description
Custom Content	Learn how to create custom projects and webpages or upload your own local documents.
Run Example Project	Learn how to run example project and edit project code.
dnf	Learn how to use the dnf Package Manager through an example
Using Ultra96-V2	Learn about the different ways that you can do actions and/or access the board.

Figure 13 – Ultra96-V2 Tutorials

2. Click on **Custom Content**. This explains three ways to add custom content. Click **Tutorials** or back in your browser to go back.

3. Click **Using Ultra96-V2**.

4. This tutorial goes over the various ways you can interact with the Ultra96-V2. As of now we have interacted using the Webserver and UART on the Pod.

5. To explore your Ultra96-V2 over miniDP, you will need a compatible cable and monitor. For example, the following combinations work:

   - miniDP-to-DP cable with DisplayPort Monitor

   - Active miniDP-to-HDMI cable with HDMI Monitor

6. Read through the SSH section, it states we can access the Ultra96-V2 terminal using TeraTerm or a PuTTY terminal application.

7. Since we have already downloaded and installed TeraTerm at the beginning of this guide let's access the Ultra96-V2's Linux terminal over SSH using TeraTerm

11 Access Ultra96-V2 Linux Terminal over SSH

1. If connected via UART, click **File → Disconnect**
2. Open TeraTerm and then select **File → New connection…** as seen in the image below.

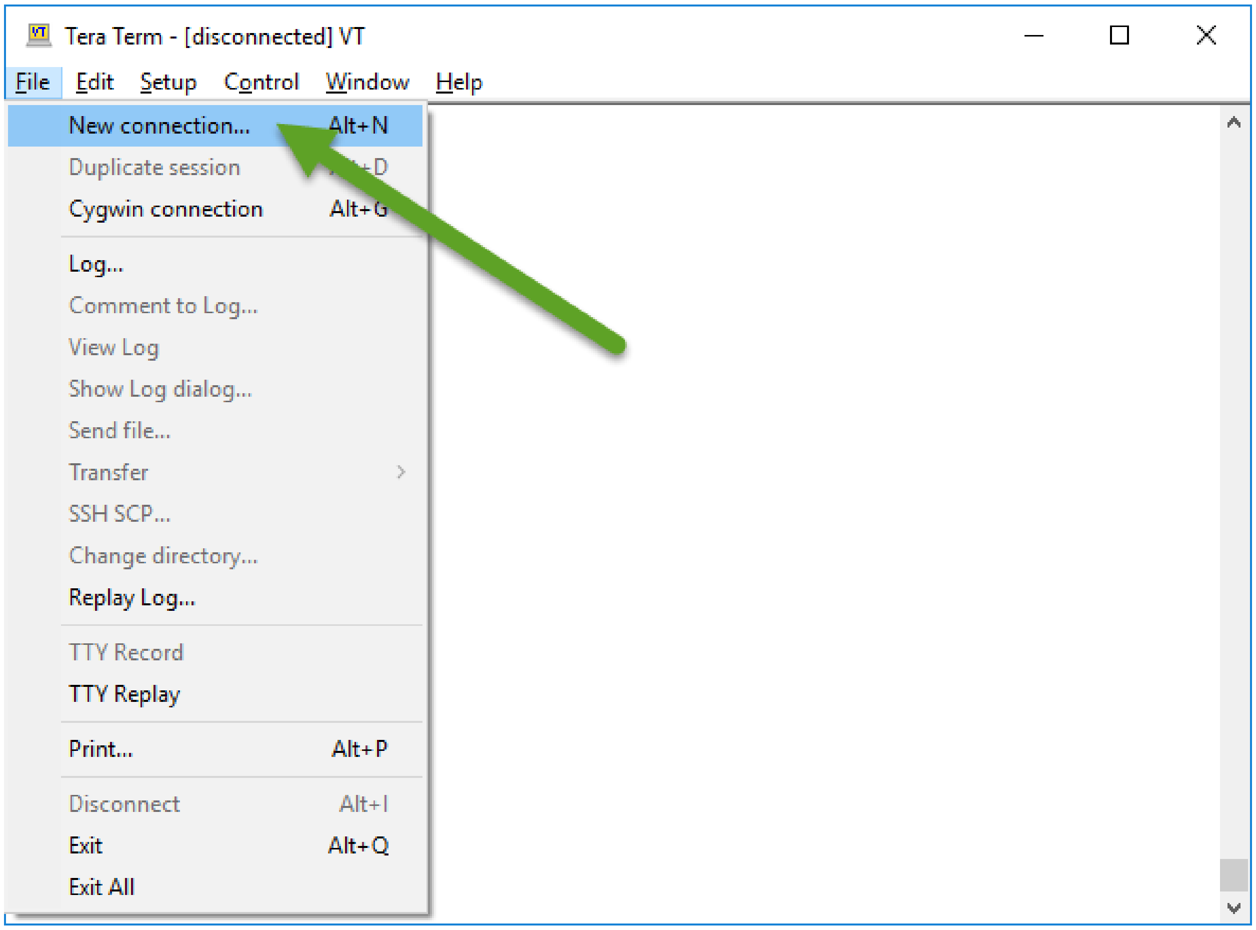
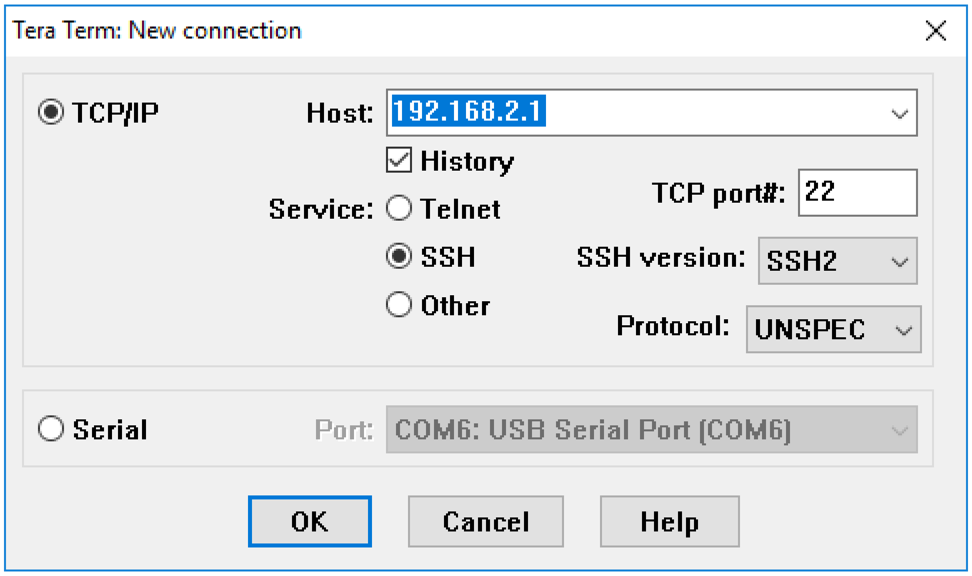


Figure 14 – TeraTerm New Connection

3. A new **TeraTerm: New connection** window will open. We now want to connect to Ultra96-V2 over SSH, select TCP/IP and then configure your Terminal settings to use the IP address that you discovered previously, similar to the below figure.



Tera Term: New connection

[✓ TCP/IP] Host: 192.168.2.1

[✓ History]

Service: Telnet [✓ SSH] Other

TCP port#: 22

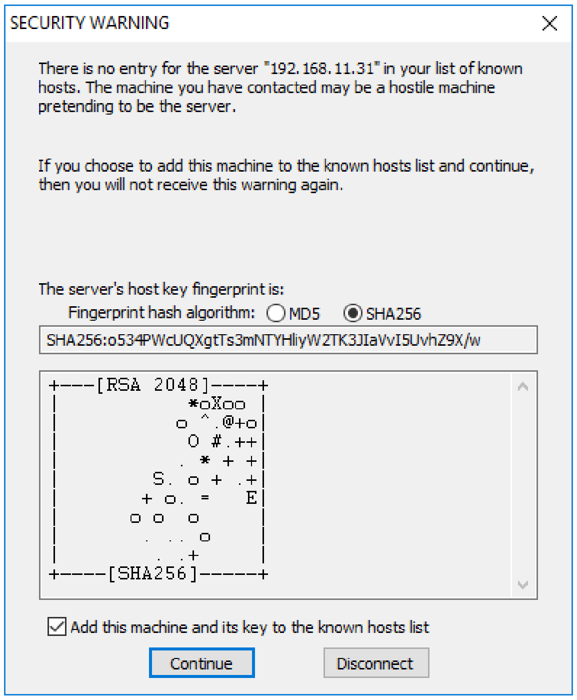
SSH version: SSH2

Protocol: UNSPEC

Serial Port: COM6: USB Serial Port (COM6)

Figure 15 – SSH Terminal Settings

4. Select **OK.** If you get a SECURITY WARNING, click **Continue** to add this machine to the known hosts list.



SECURITY WARNING

There is no entry for the server "192.168.11.31" in your list of known hosts. The machine you have contacted may be a hostile machine pretending to be the server.

If you choose to add this machine to the known hosts list and continue, then you will not receive this warning again.

The server's host key fingerprint is:

Fingerprint hash algorithm: MD5 [✓ SHA256]

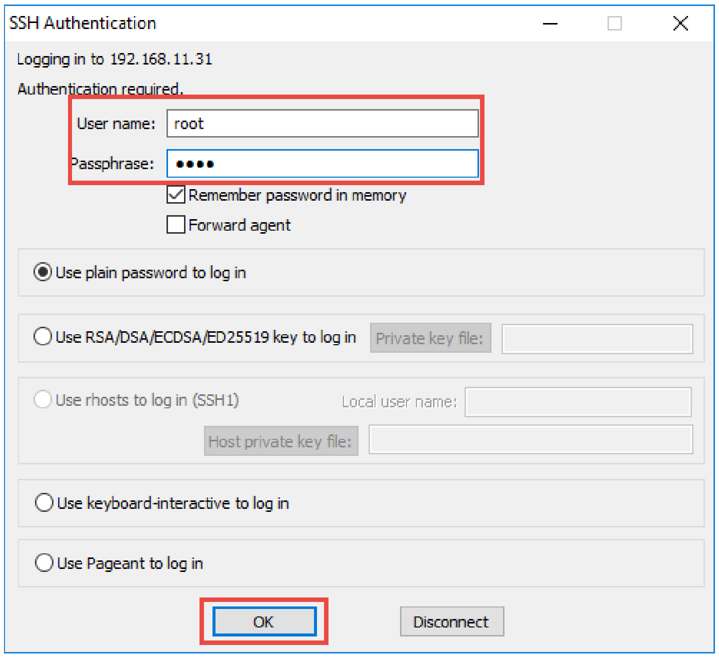
SHA256:o534PWcUQXgtTs3mNTYHliyW2TK3JIaVvI5UvhZ9X/w

```
+---[RSA 2048]----+
|           *oXoo |
|          o ^.@+o|
|           O #.++|
|          . * + +|
|        S. o + .+|
|       + o. =   E|
|      o o  o     |
|       . .. o    |
|        . .+     |
+----[SHA256]-----+
```

[✓] Add this machine and its key to the known hosts list

Figure 16 – Click Continue

5. You will then be prompted to enter *SSH Authentication* information. In our case it is looking for the Linux terminal's user name and passphrase which are **root** and **root**. Please type in **root** for the *User name* and then type in **root** for the *Passphrase* as well. Then select **OK**.


SSH Authentication

Logging in to 192.168.11.31

Authentication required.

User name: root

Passphrase: ●●●●

[✓] Remember password in memory

[ ] Forward agent

(●) Use plain password to log in

( ) Use RSA/DSA/ECDSA/ED25519 key to log in — Private key file:

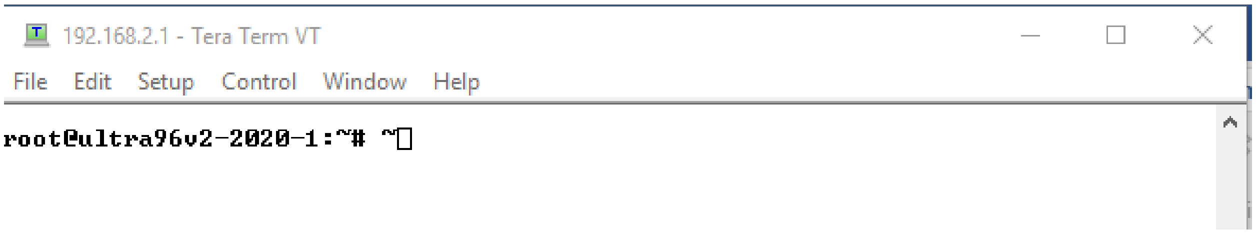
( ) Use rhosts to log in (SSH1) — Local user name: — Host private key file:

( ) Use keyboard-interactive to log in

( ) Use Pageant to log in

Figure 17 – SSH Authentication

6. You now have access to the Ultra96-V2 Terminal!


192.168.2.1 - Tera Term VT
File Edit Setup Control Window Help

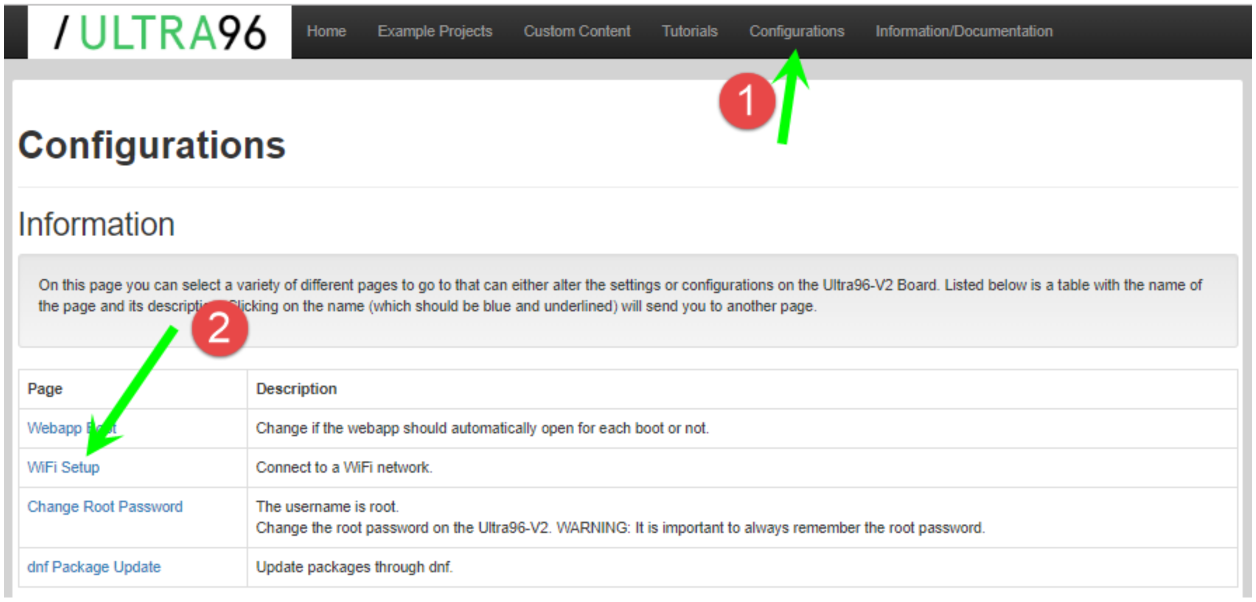
```
root@ultra96v2-2020-1:~# ~
```

Figure 18 – Ultra96-V2 Terminal

12 Connect to External Wi-Fi

To perform package updates or access anything from the Internet, the board must be connected to an external internet source.

1. Click **Configurations** and then **WiFi Setup**.



Configurations

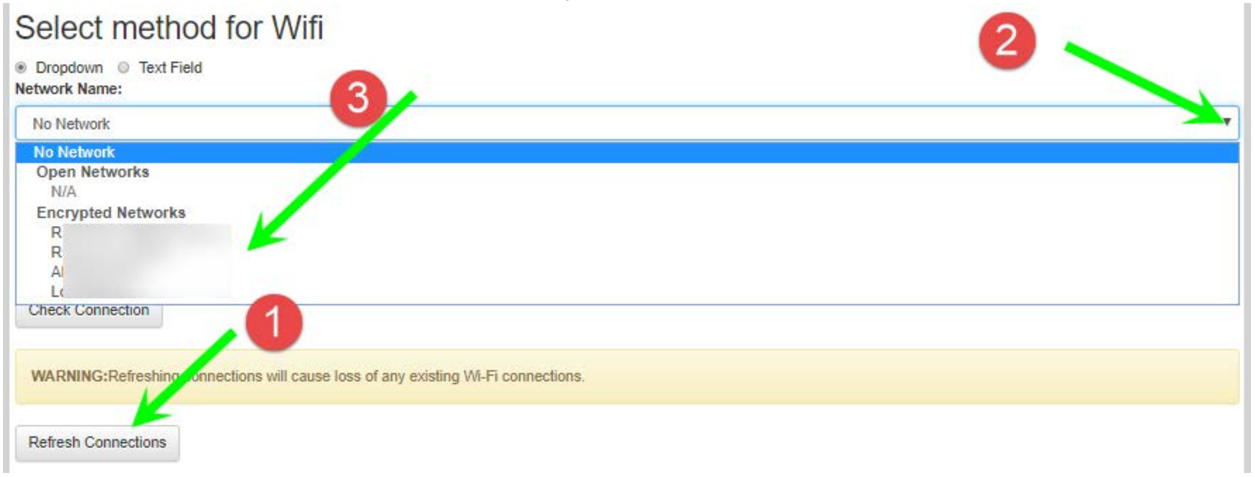
Information

On this page you can select a variety of different pages to go to that can either alter the settings or configurations on the Ultra96-V2 Board. Listed below is a table with the name of the page and its description. Clicking on the name (which should be blue and underlined) will send you to another page.

Page	Description
Webapp Boot	Change if the webapp should automatically open for each boot or not.
WiFi Setup	Connect to a WiFi network.
Change Root Password	The username is root. Change the root password on the Ultra96-V2. WARNING: It is important to always remember the root password.
dnf Package Update	Update packages through dnf.

Figure 19 – WiFi Setup

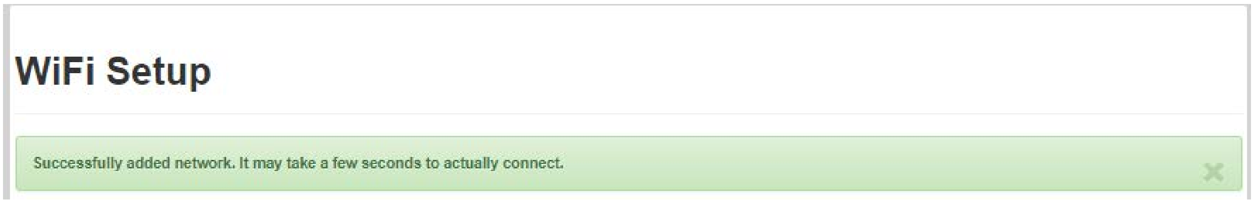
2. Click the **Refresh Connections** button then click the pull-down for Network Name. Next, select a network available to you.



WARNING:Refreshing connections will cause loss of any existing Wi-Fi connections.

Figure 20 – Discovering Networks

3. If a passphrase is required, enter the Network Passphrase into the *Network Passphrase* box, then click **Connect**.
4. Be patient as it can take several seconds for the connection to work. When successful, you should see the following message:



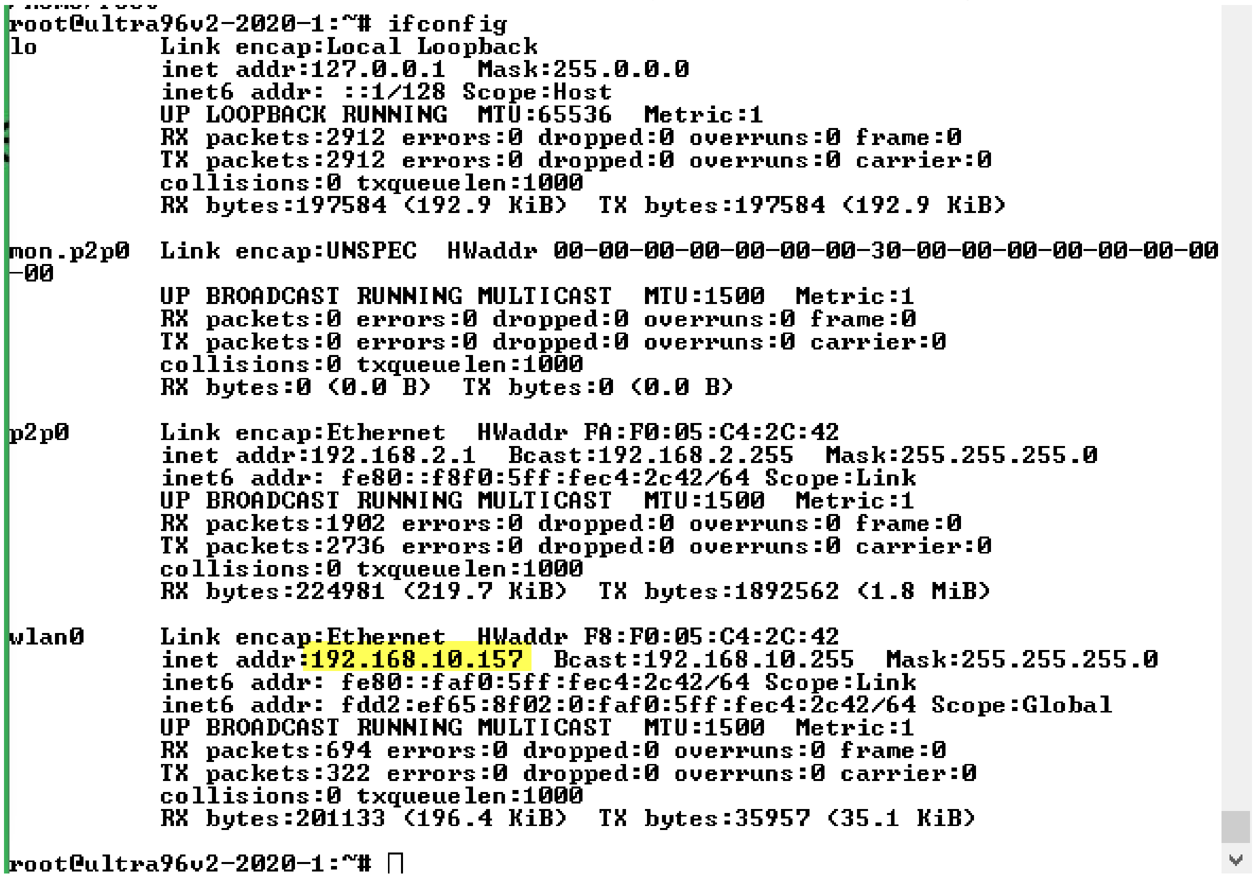
WiFi Setup

Successfully added network. It may take a few seconds to actually connect.

Figure 21 – Successfully Connected

At this point, the Access Point may no longer work. To connect back to the board, you must now connect to the IP addressed assigned by the external Wi-Fi router. You can do this by logging into the router, or you can use the terminal from the JTAG/UART Pod as shown below.

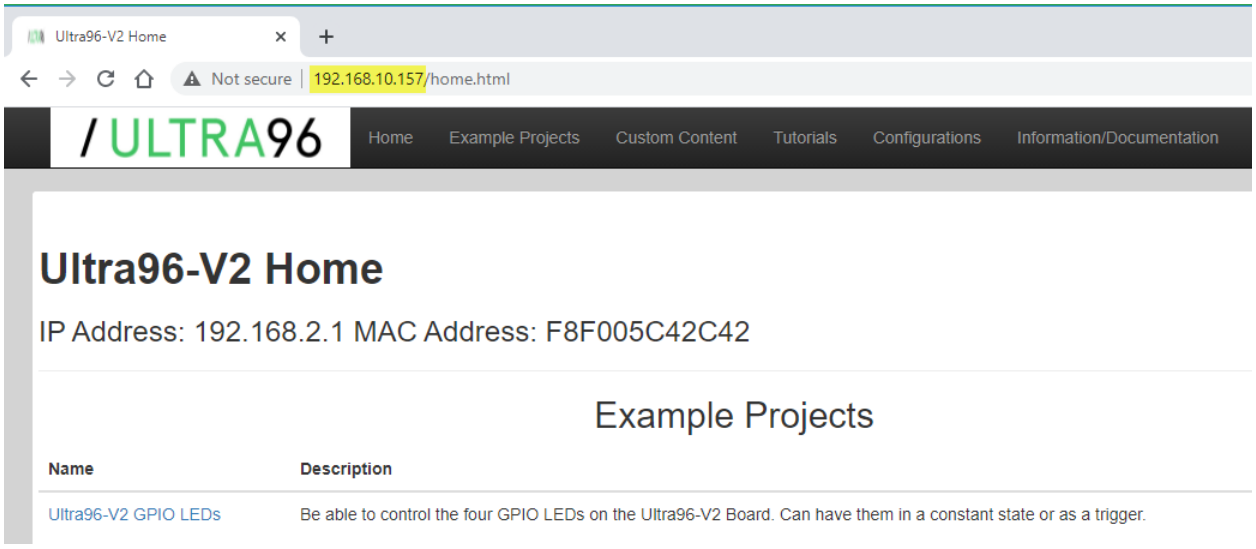
4. In the terminal, enter command ‘`ifconfig`’ to determine the assigned IP address.



```
root@ultra96v2-2020-1:~# ifconfig
lo         Link encap:Local Loopback
           inet addr:127.0.0.1  Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
           UP LOOPBACK RUNNING  MTU:65536  Metric:1
           RX packets:2912 errors:0 dropped:0 overruns:0 frame:0
           TX packets:2912 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:197584 (192.9 KiB)  TX bytes:197584 (192.9 KiB)

mon.p2p0   Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-30-00-00-00-00-00-00-00-00
-00
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

p2p0       Link encap:Ethernet  HWaddr FA:F0:05:C4:2C:42
           inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
           inet6 addr: fe80::f8f0:5ff:fec4:2c42/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:1902 errors:0 dropped:0 overruns:0 frame:0
           TX packets:2736 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:224981 (219.7 KiB)  TX bytes:1892562 (1.8 MiB)

wlan0      Link encap:Ethernet  HWaddr F8:F0:05:C4:2C:42
           inet addr:192.168.10.157  Bcast:192.168.10.255  Mask:255.255.255.0
           inet6 addr: fe80::faf0:5ff:fec4:2c42/64 Scope:Link
           inet6 addr: fdd2:ef65:8f02:0:faf0:5ff:fec4:2c42/64 Scope:Global
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:694 errors:0 dropped:0 overruns:0 frame:0
           TX packets:322 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:201133 (196.4 KiB)  TX bytes:35957 (35.1 KiB)

root@ultra96v2-2020-1:~# 
```

Figure 22 – Determine IP Address Assigned to Ultra96-V2

5. Connect your host to the same Wi-Fi source. With both your host and Ultra96-V2 board connected to the same Wi-Fi source, point your browser to this new IP address, and you will be back to the same interactive browser page for Ultra96-V2.



Ultra96-V2 Home

IP Address: 192.168.2.1 MAC Address: F8F005C42C42

Example Projects

Name	Description
Ultra96-V2 GPIO LEDs	Be able to control the four GPIO LEDs on the Ultra96-V2 Board. Can have them in a constant state or as a trigger.

Figure 23 – Connected to Ultra96-V2 Via External Wi-Fi

13 Example Projects

1. Select **Example Projects** up at the top of the page. You will see a list of projects along with descriptions of each.

13.1 Ultra96-V2 GPIO LEDs

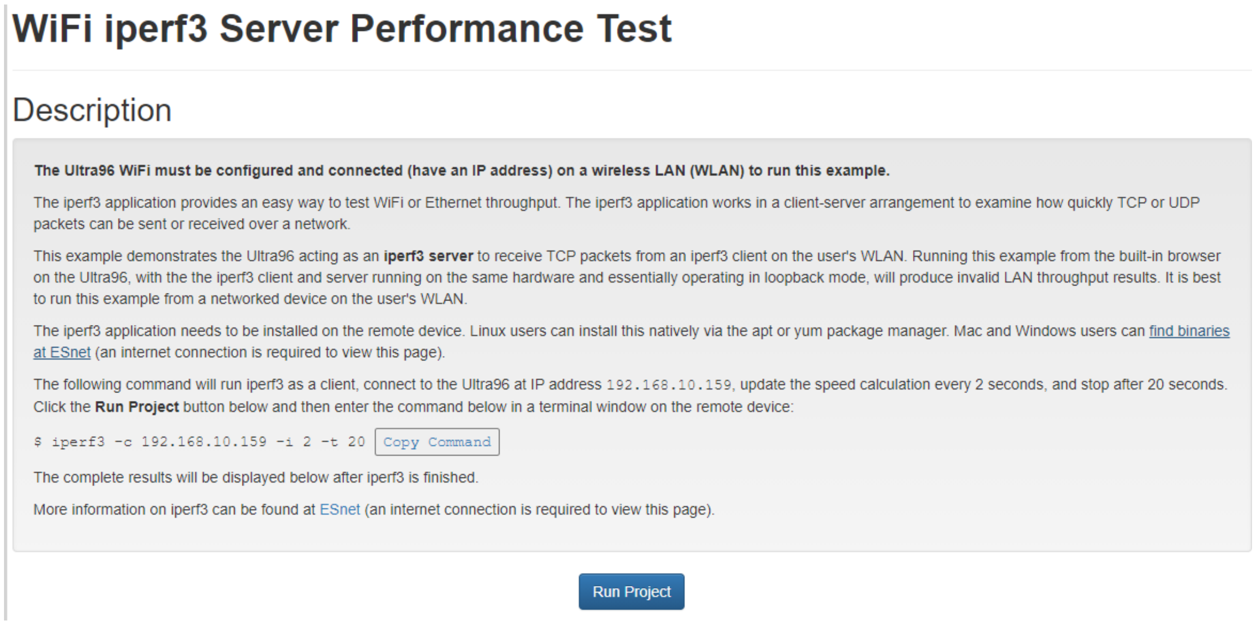
This is the same example that you previously executed.

13.2 WiFi iperf3 Server Performance Test

Note that this test works best if you are using a web browser connected to the Ultra96-V2 with the machine on the same network.

1. Click on **WiFi iperf3 Server Performance Test**

2. Read through the instructions.



WiFi iperf3 Server Performance Test

Description

The Ultra96 WiFi must be configured and connected (have an IP address) on a wireless LAN (WLAN) to run this example.

The iperf3 application provides an easy way to test WiFi or Ethernet throughput. The iperf3 application works in a client-server arrangement to examine how quickly TCP or UDP packets can be sent or received over a network.

This example demonstrates the Ultra96 acting as an **iperf3 server** to receive TCP packets from an iperf3 client on the user's WLAN. Running this example from the built-in browser on the Ultra96, with the the iperf3 client and server running on the same hardware and essentially operating in loopback mode, will produce invalid LAN throughput results. It is best to run this example from a networked device on the user's WLAN.

The iperf3 application needs to be installed on the remote device. Linux users can install this natively via the apt or yum package manager. Mac and Windows users can find binaries at ESnet (an internet connection is required to view this page).

The following command will run iperf3 as a client, connect to the Ultra96 at IP address `192.168.10.159`, update the speed calculation every 2 seconds, and stop after 20 seconds. Click the **Run Project** button below and then enter the command below in a terminal window on the remote device:

```
$ iperf3 -c 192.168.10.159 -i 2 -t 20
```

The complete results will be displayed below after iperf3 is finished.

More information on iperf3 can be found at ESnet (an internet connection is required to view this page).

Figure 24 – WiFi iperf3 Server Performance Test

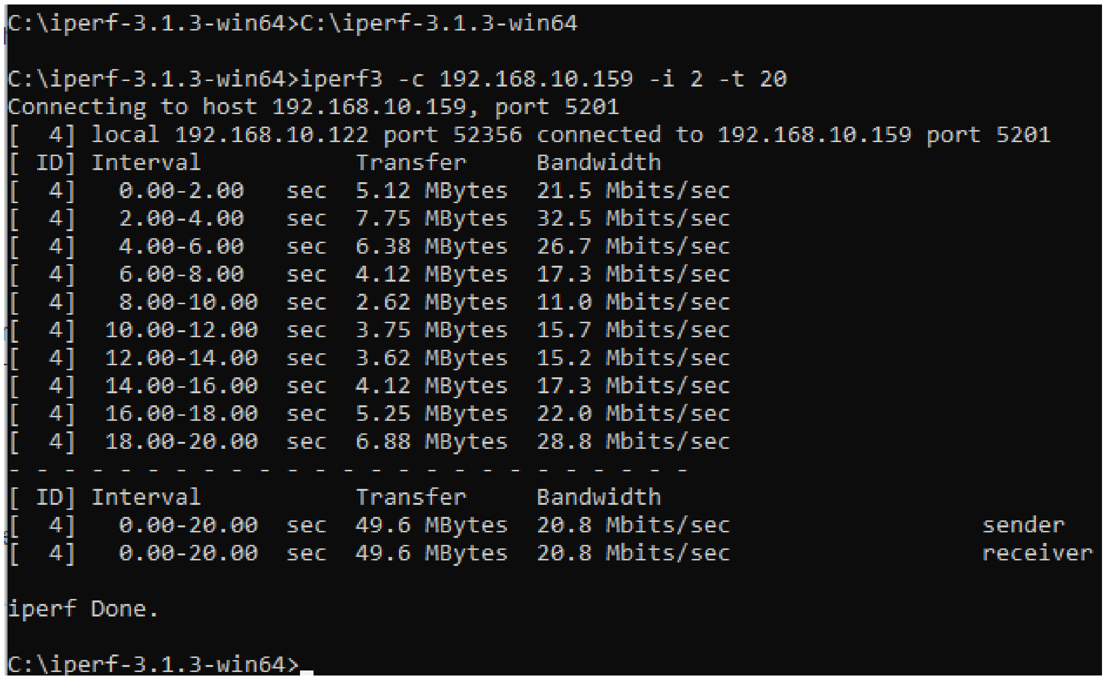
3. If you haven't installed iperf3 on your host machine, go to https://software.es.net/iperf/obtaining.html#binary-distributions, download, and install iperf3.

4. Open a command terminal on your host machine. Change directories to where iperf3 executable is located. Click the **Copy Command** button, but do not run yet.

5. Click the **Run Project** button.

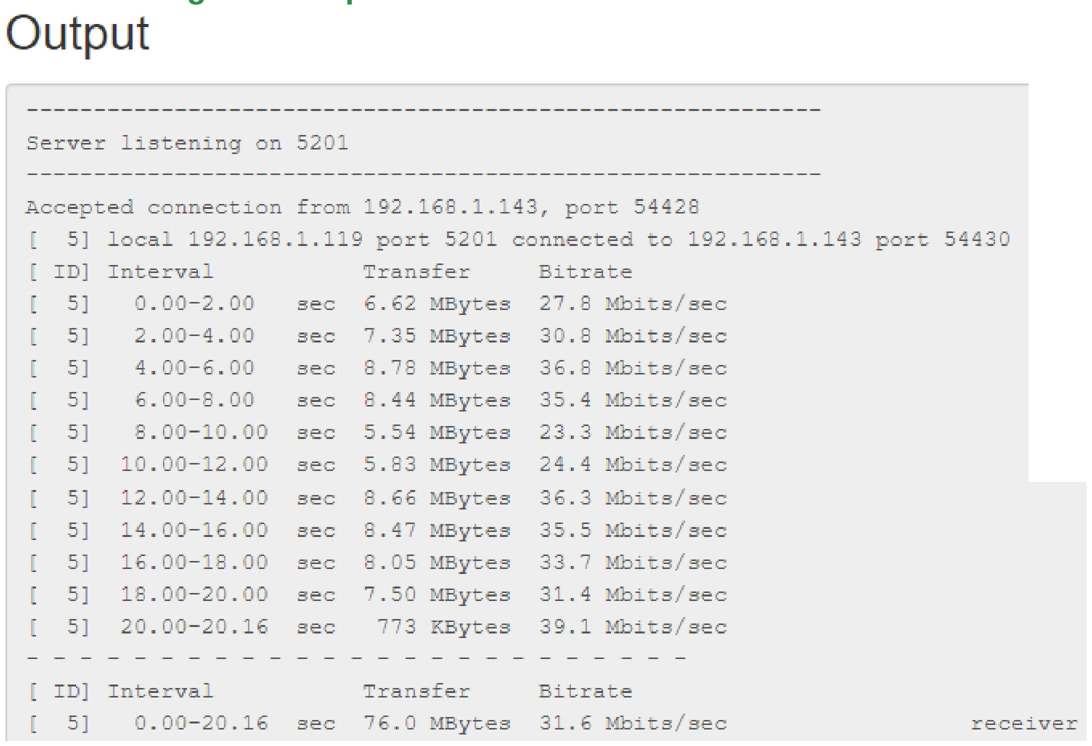
6. Now paste the copied command into the command window and execute. You should see results scroll immediately to the command window. This illustrates the performance seen by the host. Once the 10 iterations are complete, results will

also be displayed in the browser window, which illustrate the performance seen by the Ultra96-V2. Note that differences in architecture and iperf3 version will result in differences depending on which device is measuring the results.



```
C:\iperf-3.1.3-win64>C:\iperf-3.1.3-win64

C:\iperf-3.1.3-win64>iperf3 -c 192.168.10.159 -i 2 -t 20
Connecting to host 192.168.10.159, port 5201
[  4] local 192.168.10.122 port 52356 connected to 192.168.10.159 port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-2.00   sec  5.12 MBytes  21.5 Mbits/sec
[  4]   2.00-4.00   sec  7.75 MBytes  32.5 Mbits/sec
[  4]   4.00-6.00   sec  6.38 MBytes  26.7 Mbits/sec
[  4]   6.00-8.00   sec  4.12 MBytes  17.3 Mbits/sec
[  4]   8.00-10.00  sec  2.62 MBytes  11.0 Mbits/sec
[  4]  10.00-12.00  sec  3.75 MBytes  15.7 Mbits/sec
[  4]  12.00-14.00  sec  3.62 MBytes  15.2 Mbits/sec
[  4]  14.00-16.00  sec  4.12 MBytes  17.3 Mbits/sec
[  4]  16.00-18.00  sec  5.25 MBytes  22.0 Mbits/sec
[  4]  18.00-20.00  sec  6.88 MBytes  28.8 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-20.00  sec  49.6 MBytes  20.8 Mbits/sec                  sender
[  4]   0.00-20.00  sec  49.6 MBytes  20.8 Mbits/sec                  receiver

iperf Done.

C:\iperf-3.1.3-win64>
```

Figure 25 – iperf3 results on Host from Server Test



Output

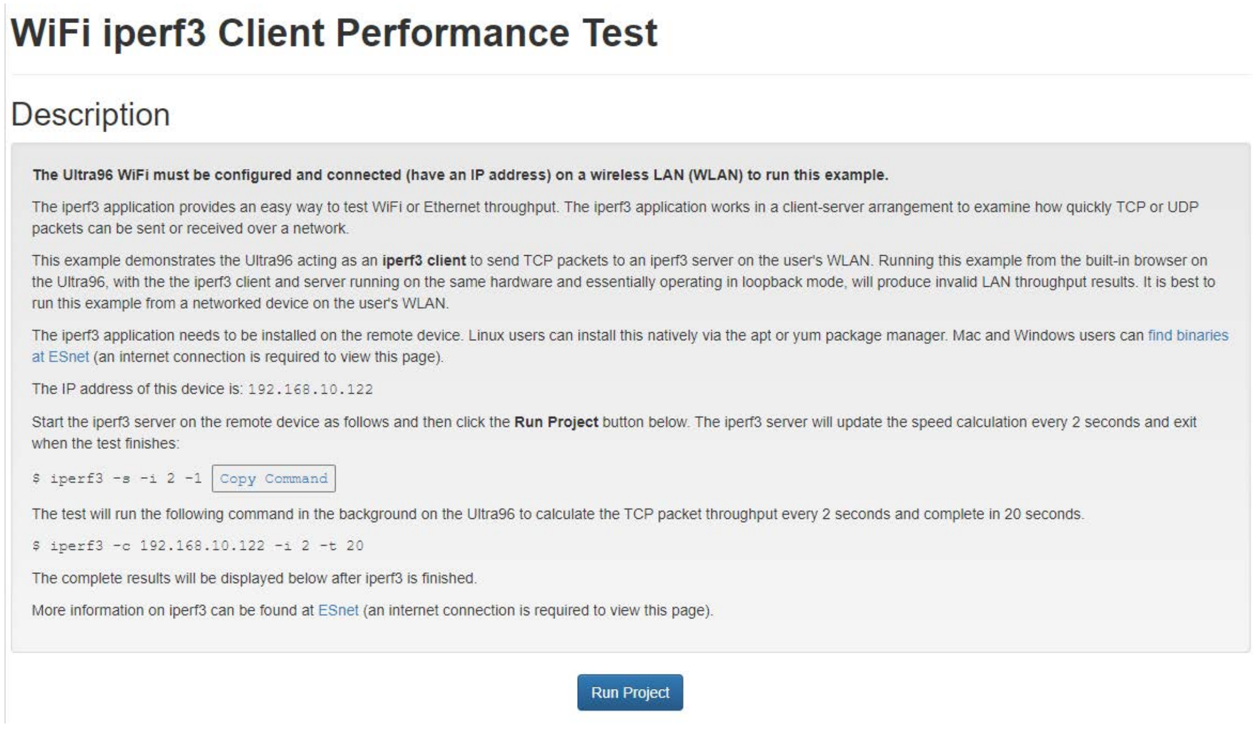
```
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.1.143, port 54428
[  5] local 192.168.1.119 port 5201 connected to 192.168.1.143 port 54430
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-2.00   sec  6.62 MBytes  27.8 Mbits/sec
[  5]   2.00-4.00   sec  7.35 MBytes  30.8 Mbits/sec
[  5]   4.00-6.00   sec  8.78 MBytes  36.8 Mbits/sec
[  5]   6.00-8.00   sec  8.44 MBytes  35.4 Mbits/sec
[  5]   8.00-10.00  sec  5.54 MBytes  23.3 Mbits/sec
[  5]  10.00-12.00  sec  5.83 MBytes  24.4 Mbits/sec
[  5]  12.00-14.00  sec  8.66 MBytes  36.3 Mbits/sec
[  5]  14.00-16.00  sec  8.47 MBytes  35.5 Mbits/sec
[  5]  16.00-18.00  sec  8.05 MBytes  33.7 Mbits/sec
[  5]  18.00-20.00  sec  7.50 MBytes  31.4 Mbits/sec
[  5]  20.00-20.16  sec   773 KBytes  39.1 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-20.16  sec  76.0 MBytes  31.6 Mbits/sec                  receiver
```

Figure 26 – iperf3 results on Ultra96-V2 from Server Test

13.3 WiFi iperf3 Client Performance Test

1. Select **Example Projects** again at the top of the page and then click **WiFi iperf3 Client Performance Test**.

2. Read through the instructions.



WiFi iperf3 Client Performance Test

Description

The Ultra96 WiFi must be configured and connected (have an IP address) on a wireless LAN (WLAN) to run this example.

The iperf3 application provides an easy way to test WiFi or Ethernet throughput. The iperf3 application works in a client-server arrangement to examine how quickly TCP or UDP packets can be sent or received over a network.

This example demonstrates the Ultra96 acting as an **iperf3 client** to send TCP packets to an iperf3 server on the user's WLAN. Running this example from the built-in browser on the Ultra96, with the the iperf3 client and server running on the same hardware and essentially operating in loopback mode, will produce invalid LAN throughput results. It is best to run this example from a networked device on the user's WLAN.

The iperf3 application needs to be installed on the remote device. Linux users can install this natively via the apt or yum package manager. Mac and Windows users can find binaries at ESnet (an internet connection is required to view this page).

The IP address of this device is: `192.168.10.122`

Start the iperf3 server on the remote device as follows and then click the **Run Project** button below. The iperf3 server will update the speed calculation every 2 seconds and exit when the test finishes:

```
$ iperf3 -s -i 2 -1
```

Copy Command

The test will run the following command in the background on the Ultra96 to calculate the TCP packet throughput every 2 seconds and complete in 20 seconds.

```
$ iperf3 -c 192.168.10.122 -i 2 -t 20
```

The complete results will be displayed below after iperf3 is finished.

More information on iperf3 can be found at ESnet (an internet connection is required to view this page).

Run Project

Figure 27 – WiFi iperf3 Client Performance Test

3. Click the **Copy Command** button. Paste into your command window and execute.

4. Click the **Run Project** button.

5. You should see results scroll immediately to the command window. This illustrates the performance seen by the host. Once the 10 iterations are complete, results will also be displayed in the browser window, which illustrate the performance seen by the Ultra96-V2. Note that differences in architecture and iperf3 version will result in differences depending on which device is measuring the results.

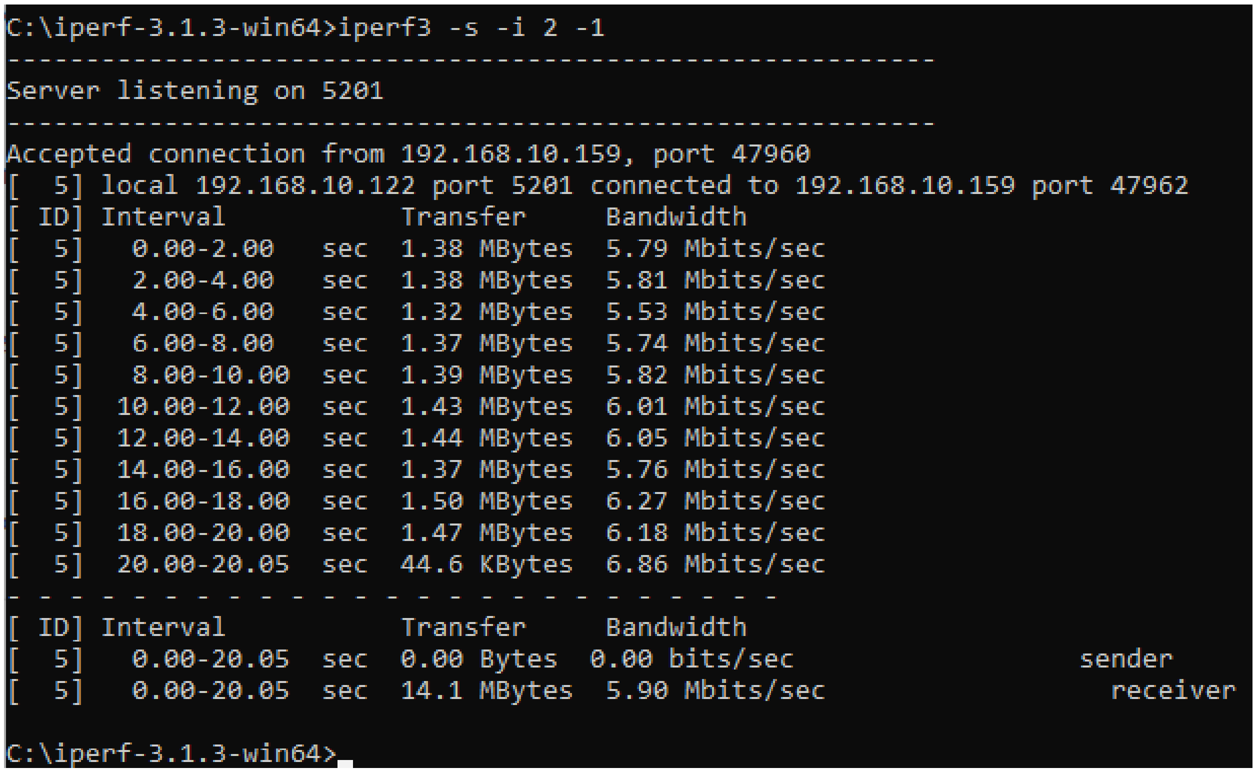

```
C:\iperf-3.1.3-win64>iperf3 -s -i 2 -1
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.10.159, port 47960
[  5] local 192.168.10.122 port 5201 connected to 192.168.10.159 port 47962
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-2.00   sec  1.38 MBytes  5.79 Mbits/sec
[  5]   2.00-4.00   sec  1.38 MBytes  5.81 Mbits/sec
[  5]   4.00-6.00   sec  1.32 MBytes  5.53 Mbits/sec
[  5]   6.00-8.00   sec  1.37 MBytes  5.74 Mbits/sec
[  5]   8.00-10.00  sec  1.39 MBytes  5.82 Mbits/sec
[  5]  10.00-12.00  sec  1.43 MBytes  6.01 Mbits/sec
[  5]  12.00-14.00  sec  1.44 MBytes  6.05 Mbits/sec
[  5]  14.00-16.00  sec  1.37 MBytes  5.76 Mbits/sec
[  5]  16.00-18.00  sec  1.50 MBytes  6.27 Mbits/sec
[  5]  18.00-20.00  sec  1.47 MBytes  6.18 Mbits/sec
[  5]  20.00-20.05  sec  44.6 KBytes  6.86 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-20.05  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-20.05  sec  14.1 MBytes  5.90 Mbits/sec                  receiver

C:\iperf-3.1.3-win64>
```

Figure 28 – iperf3 Results on Host from Client Test

Output

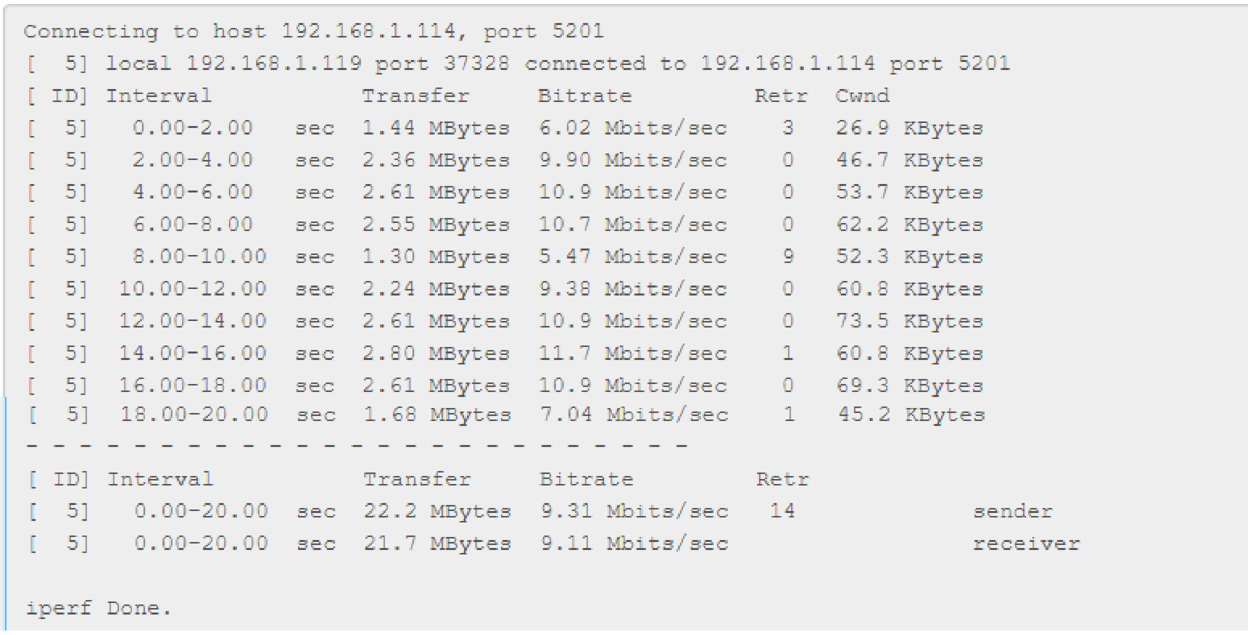
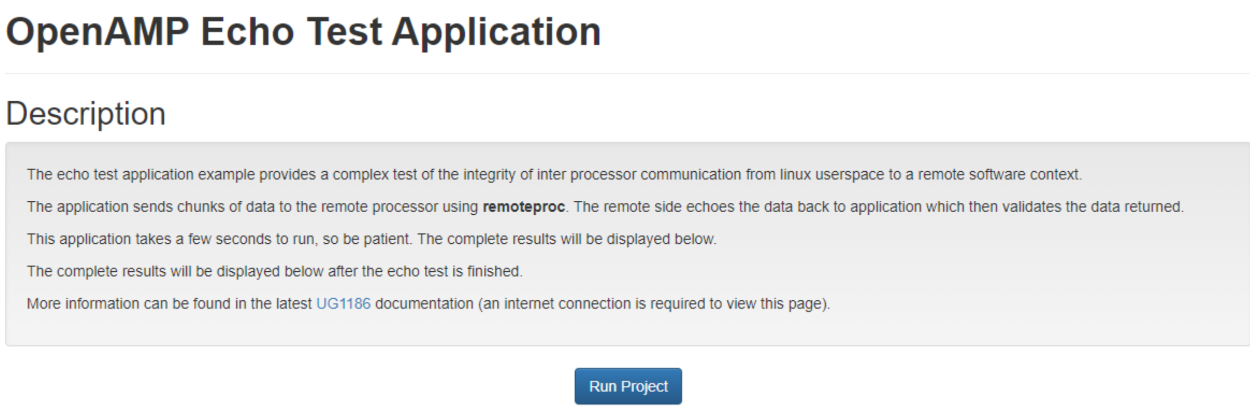

```
Connecting to host 192.168.1.114, port 5201
[  5] local 192.168.1.119 port 37328 connected to 192.168.1.114 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-2.00   sec  1.44 MBytes  6.02 Mbits/sec    3   26.9 KBytes
[  5]   2.00-4.00   sec  2.36 MBytes  9.90 Mbits/sec    0   46.7 KBytes
[  5]   4.00-6.00   sec  2.61 MBytes  10.9 Mbits/sec    0   53.7 KBytes
[  5]   6.00-8.00   sec  2.55 MBytes  10.7 Mbits/sec    0   62.2 KBytes
[  5]   8.00-10.00  sec  1.30 MBytes  5.47 Mbits/sec    9   52.3 KBytes
[  5]  10.00-12.00  sec  2.24 MBytes  9.38 Mbits/sec    0   60.8 KBytes
[  5]  12.00-14.00  sec  2.61 MBytes  10.9 Mbits/sec    0   73.5 KBytes
[  5]  14.00-16.00  sec  2.80 MBytes  11.7 Mbits/sec    1   60.8 KBytes
[  5]  16.00-18.00  sec  2.61 MBytes  10.9 Mbits/sec    0   69.3 KBytes
[  5]  18.00-20.00  sec  1.68 MBytes  7.04 Mbits/sec    1   45.2 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-20.00  sec  22.2 MBytes  9.31 Mbits/sec   14             sender
[  5]   0.00-20.00  sec  21.7 MBytes  9.11 Mbits/sec                  receiver

iperf Done.
```

Figure 29 – iperf3 Results on Ultra96-V2 from Client Test

13.4 OpenAMP Echo Test Application

1. Select **Example Projects** again at the top of the page and then click **OpenAMP Echo Test Application.**

2. Read through the description.



OpenAMP Echo Test Application

Description

The echo test application example provides a complex test of the integrity of inter processor communication from linux userspace to a remote software context.

The application sends chunks of data to the remote processor using **remoteproc**. The remote side echoes the data back to application which then validates the data returned.

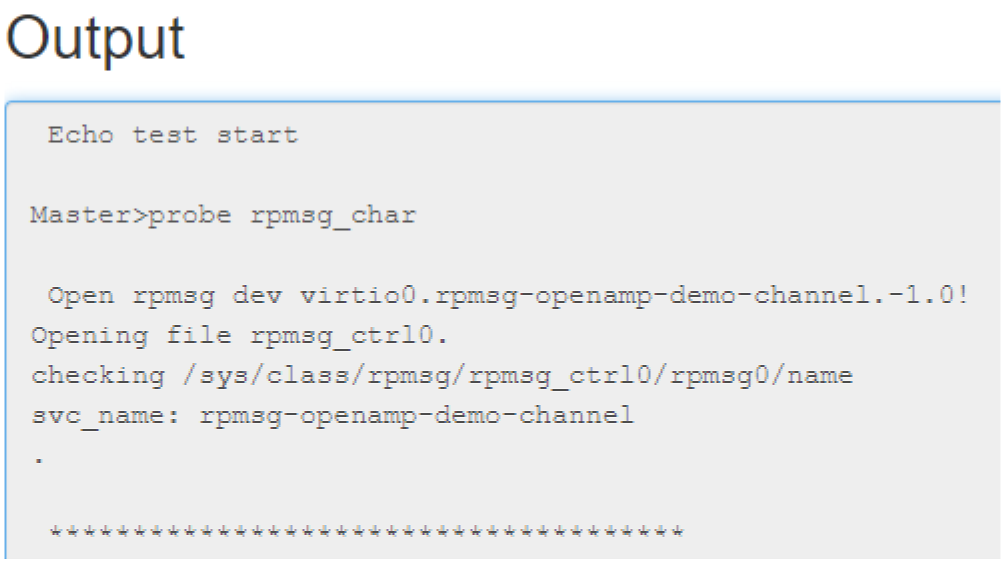
This application takes a few seconds to run, so be patient. The complete results will be displayed below.

The complete results will be displayed below after the echo test is finished.

More information can be found in the latest UG1186 documentation (an internet connection is required to view this page).

Figure 30 – OpenAMP Echo Description

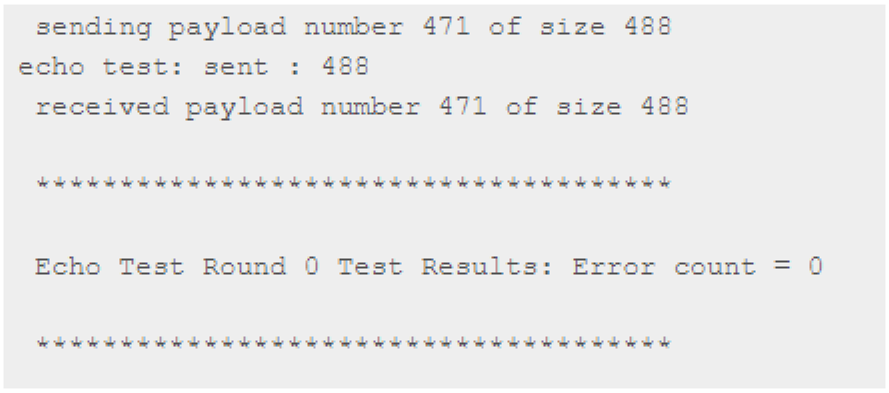
3. Click the **Run Project** button.



Output

```
 Echo test start

Master>probe rpmsg_char

 Open rpmsg dev virtio0.rpmsg-openamp-demo-channel.-1.0!
Opening file rpmsg_ctrl0.
checking /sys/class/rpmsg/rpmsg_ctrl0/rpmsg0/name
svc_name: rpmsg-openamp-demo-channel
.

 **************************************
```

Figure 31 – Echo Test Starting



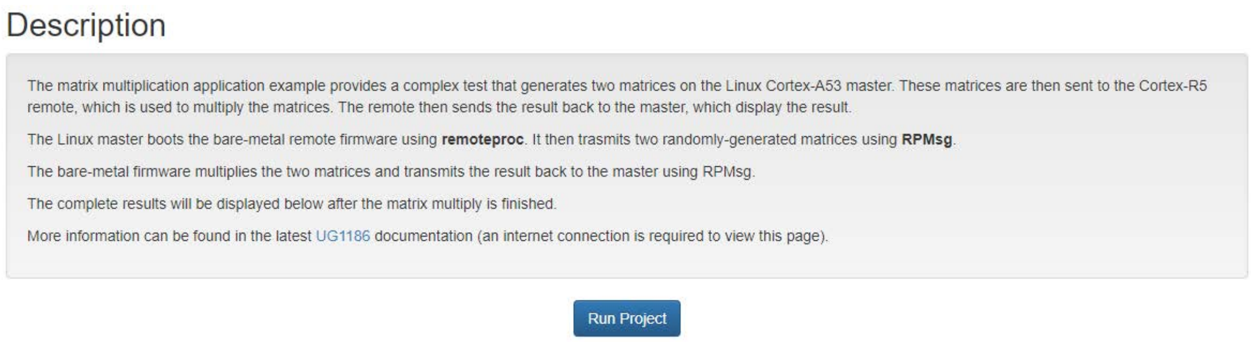
```
 sending payload number 471 of size 488
echo test: sent : 488
 received payload number 471 of size 488

 **************************************

 Echo Test Round 0 Test Results: Error count = 0

 **************************************
```

Figure 32 – Echo Test Complete

13.5 OpenAMP Matrix Multiplication

1. Select **Example Projects** again at the top of the page and then click **OpenAMP Matrix Multiplication**.

2. Read through the description which goes over what is going to happen in the OpenAMP Matrix Multiplication Design



Description

The matrix multiplication application example provides a complex test that generates two matrices on the Linux Cortex-A53 master. These matrices are then sent to the Cortex-R5 remote, which is used to multiply the matrices. The remote then sends the result back to the master, which display the result.

The Linux master boots the bare-metal remote firmware using **remoteproc**. It then trasmits two randomly-generated matrices using **RPMsg**.

The bare-metal firmware multiplies the two matrices and transmits the result back to the master using RPMsg.

The complete results will be displayed below after the matrix multiply is finished.

More information can be found in the latest UG1186 documentation (an internet connection is required to view this page).

Figure 33 – OpenAMP Matrix Multiplication Description

3. Select **Run Project.**

4. In the Output section you will see the two input matrices and then the matrix multiplication results.

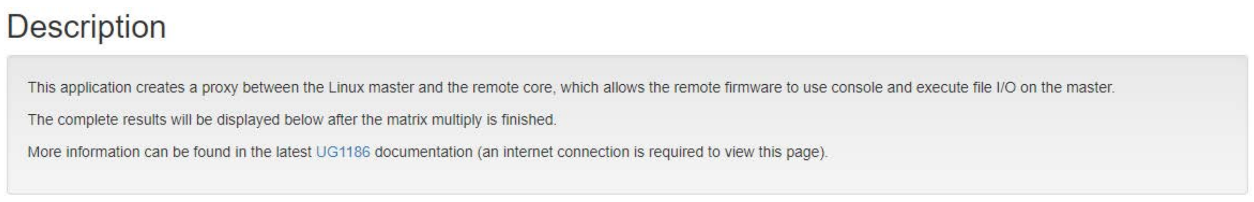
```
Matrix multiplication demo start

Master>probe rpmsg_char

 Open rpmsg dev virtio0.rpmsg-openamp-demo-channel.-1.0!
Opening file rpmsg_ctrl0.
checking /sys/class/rpmsg/rpmsg_ctrl0/rpmsg0/name
svc_name: rpmsg-openamp-demo-channel
.

 Creating ui_thread and compute_thread ...

 **************************************

  Matrix multiplication demo Round 0

 **************************************

 Compute thread unblocked ..
 The compute thread is now blocking ona read() from rpmsg device

 Generating random matrices now ...

 Master : Linux : Input matrix 0

 0  5  4  6  8  9
 1  8  1  8  9  7
 7  7  1  1  9  8
 5  7  3  3  0  7
 5  4  7  6  0  1
 2  2  6  7  0  7

 Master : Linux : Input matrix 1

 8  1  7  9  0  8
 6  9  5  9  1  6
 8  8  4  1  1  6
 1  8  3  8  6  3
 9  8  7  8  5  7
 7  3  0  4  4  2

 Writing generated matrices to rpmsg rpmsg device, 296 bytes ..

 Received results! - 148 bytes from rpmsg device (transmitted from remote context)

 Master : Linux : Printing results
 203  216  115  197  121  146
 202  238  138  246  130  163
 244  182  154  239  91  186
 158  137  91  163  56  123
 133  148  101  140  51  126
 132  145  69  126  78  99

End of Matrix multiplication demo Round 0

 Quitting application ..
 Matrix multiplication demo end

 Quitting application ..
```

Figure 34 – Matrix Multiply Results

13.6 OpenAMP Proxy Application

1. Select **Example Projects** again at the top of the page and then click **OpenAMP Proxy Application**.

2. Read through the description. Note that the Remote Processor is one of the R5 processors within the ZU+ and the Master is the A53 CPU0. There is a typo in this description that refers to a matrix multiply, so please ignore that reference.



Description

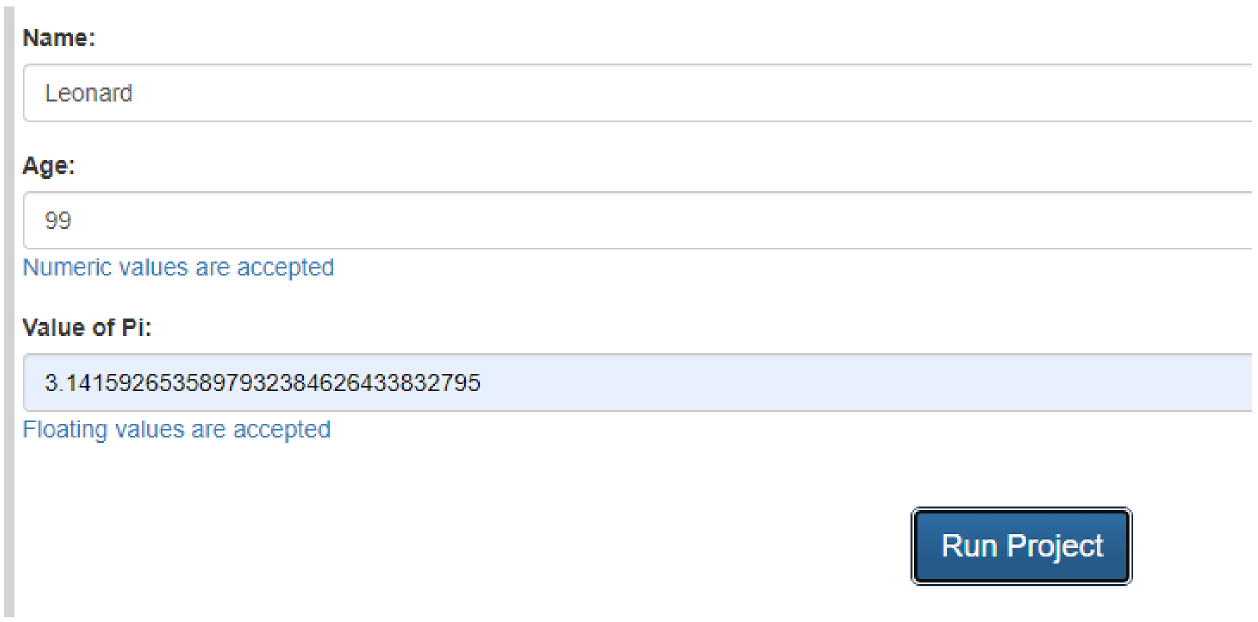
This application creates a proxy between the Linux master and the remote core, which allows the remote firmware to use console and execute file I/O on the master.

The complete results will be displayed below after the matrix multiply is finished.

More information can be found in the latest UG1186 documentation (an internet connection is required to view this page).

Figure 35 – OpenAMP Proxy Description

3. Enter a Name, Age, and Value of Pi. Click **Run Project**.



Name:

Leonard

Age:

99

Numeric values are accepted

Value of Pi:

3.14159265358979323846264338327950

Floating values are accepted

Figure 36 – Enter Proxy Info

```
Remote>Baremetal Remote Procedure Call (RPC) Demonstration

Remote>***************************************************

Remote>Rpmsg based retargetting to proxy initialized..

Remote>FileIO demo ..

Remote>Creating a file on master and writing to it..

Remote>Opened file 'remote.file' with fd = 7

Remote>Wrote to fd = 7, size = 45, content = This is a test string being written to file..

Remote>Closed fd = 7

Remote>Reading a file on master and displaying its contents..

Remote>Opened file 'remote.file' with fd = 7

Remote>Read from fd = 7, size = 90, printing contents below .. This is a test string being written
to file..This is a test string being written to file..

Remote>Closed fd = 7

Remote>Remote firmware using scanf and printf ..

Remote>Scanning user input from master..

Remote>Enter name

Remote>Enter age

Remote>Enter value for pi

Remote>User name = 'Leonard'

Remote>User age = '99'

Remote>User entered value of pi = '3.141593'

Remote>Repeat demo ? (enter yes or no)

Remote>RPC retargetting quitting ...

Remote> Firmware's rpmsg-rpc-channel going down!

Master>Loading remote firmware

Master>probe rpmsg_char
Opening file rpmsg_ctrl0.
checking /sys/class/rpmsg/rpmsg_ctrl0/rpmsg0/name
svc_name: rpmsg-openamp-demo-channel
.

Master>RPC service started !!
handle_read: 4, 90
handle_read: 4, 48

Master>RPC service exiting !!
```

Figure 37 – OpenAMP Proxy Results

14 Reading On-board PMIC and MPSoC Telemetry

The Infineon PMICs on the Ultra96-V2 connect to the ZU+ MPSoC device via PMBus. This allows us to read important telemetry from the power regulators on board. To learn more about this feature, read this blog and view this recorded webinar:

https://www.element14.com/community/groups/power-management/blog/2017/10/11/benefits-of-power-system-telemetry-using-pmbus

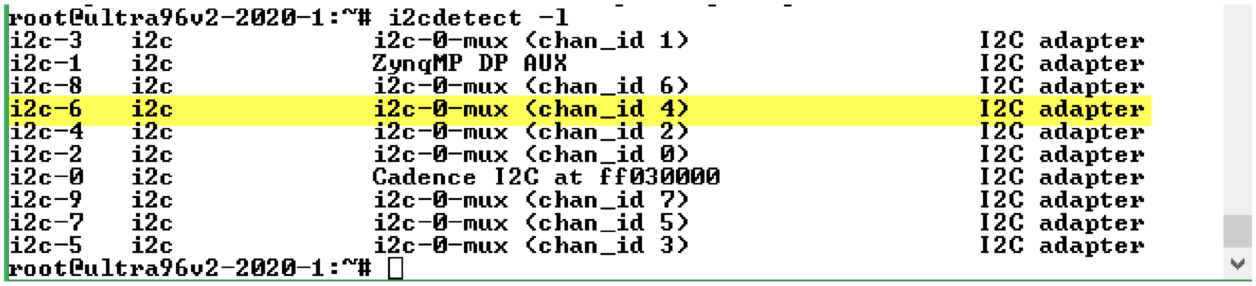
https://www.element14.com/community/events/5588/l/pmbus-and-power-supply-telemetry

You can also view a more advanced project related to this on Hackster:

https://www.hackster.io/AlbertaBeef/monitoring-power-on-the-avnet-platforms-e52a23

For now, we will show the simple method of reading back the telemetry.

1. Connect to the Linux Terminal either through SSH or the UART. To see the available I2C busses available, type in your console **`i2cdetect -l`**



```
root@ultra96v2-2020-1:~# i2cdetect -l
i2c-3   i2c             i2c-0-mux (chan_id 1)                   I2C adapter
i2c-1   i2c             ZynqMP DP AUX                           I2C adapter
i2c-8   i2c             i2c-0-mux (chan_id 6)                   I2C adapter
i2c-6   i2c             i2c-0-mux (chan_id 4)                   I2C adapter
i2c-4   i2c             i2c-0-mux (chan_id 2)                   I2C adapter
i2c-2   i2c             i2c-0-mux (chan_id 0)                   I2C adapter
i2c-0   i2c             Cadence I2C at ff030000                 I2C adapter
i2c-9   i2c             i2c-0-mux (chan_id 7)                   I2C adapter
i2c-7   i2c             i2c-0-mux (chan_id 5)                   I2C adapter
i2c-5   i2c             i2c-0-mux (chan_id 3)                   I2C adapter
root@ultra96v2-2020-1:~# 
```

Figure 38 – I2Cdetect

This shows all the available I2C busses on the Ultra96-V2. The one corresponding to the PMICs is found based on the hardware connections in the schematic. The signals are called 'PMIC_SCL' and 'PMIC_SDA' and are connected to Channel 4 (SC/SD4) of the I2C Expander U7, which is i2c-6 in Linux.

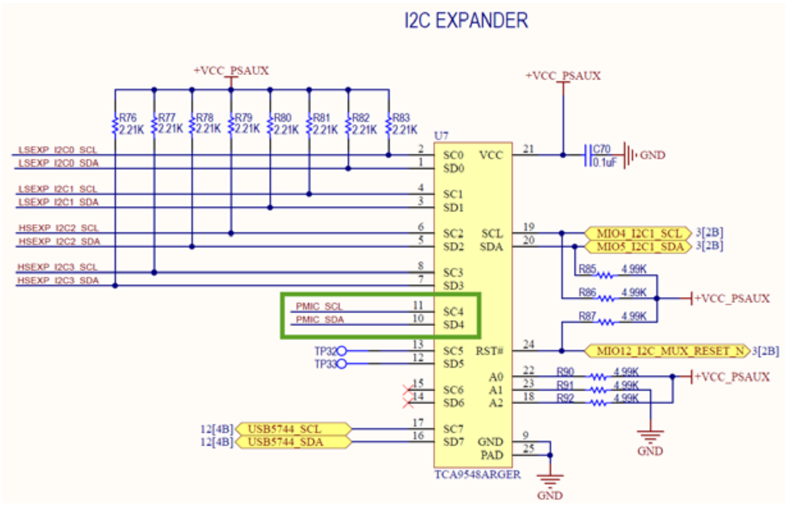
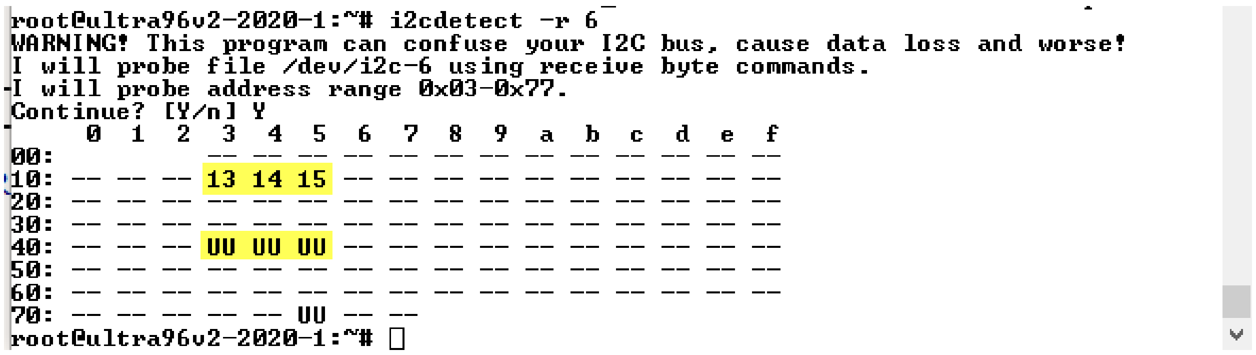


Figure 39 – PMIC on I2C Mux Channel 4

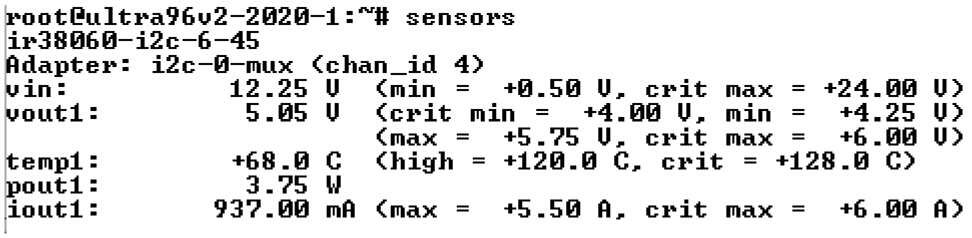
2. To see what is specifically on i2c-6, enter **`i2cdetect -r 6`**, and then enter ‘**Y**’ to confirm. The addresses associated with the PMICs are shown below.



```
root@ultra96v2-2020-1:~# i2cdetect -r 6
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-6 using receive byte commands.
I will probe address range 0x03-0x77.
Continue? [Y/n] Y
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- 13 14 15 -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- UU UU UU -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- UU -- --
root@ultra96v2-2020-1:~# 
```

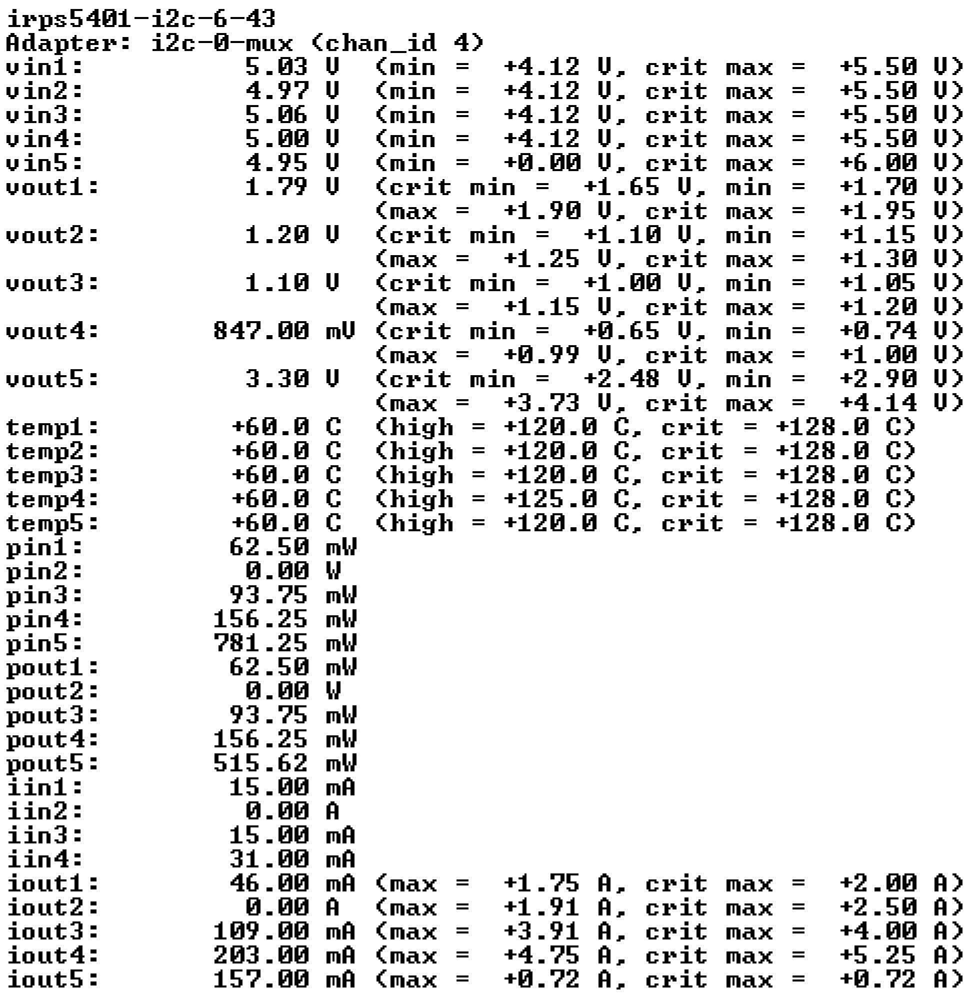
Figure 40 – PMICs on I2C-6

3. Read the telemetry from the PMICs by entering **`sensors`** in your terminal. The output will scroll to the screen, which is copied below in sections.



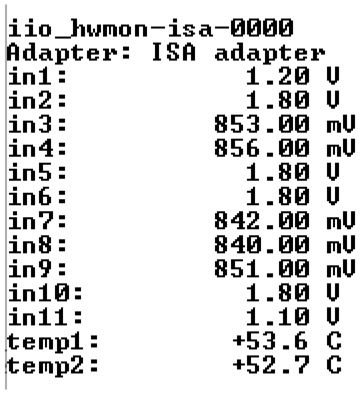
```
root@ultra96v2-2020-1:~# sensors
ir38060-i2c-6-45
Adapter: i2c-0-mux (chan_id 4)
vin:          12.25 V  (min =  +0.50 V, crit max = +24.00 V)
vout1:         5.05 V  (crit min =  +4.00 V, min =  +4.25 V)
                       (max =  +5.75 V, crit max =  +6.00 V)
temp1:        +68.0 C  (high = +120.0 C, crit = +128.0 C)
pout1:         3.75 W
iout1:       937.00 mA (max =  +5.50 A, crit max =  +6.00 A)
```

Figure 41 – Telemetry from PMIC 0x15/45



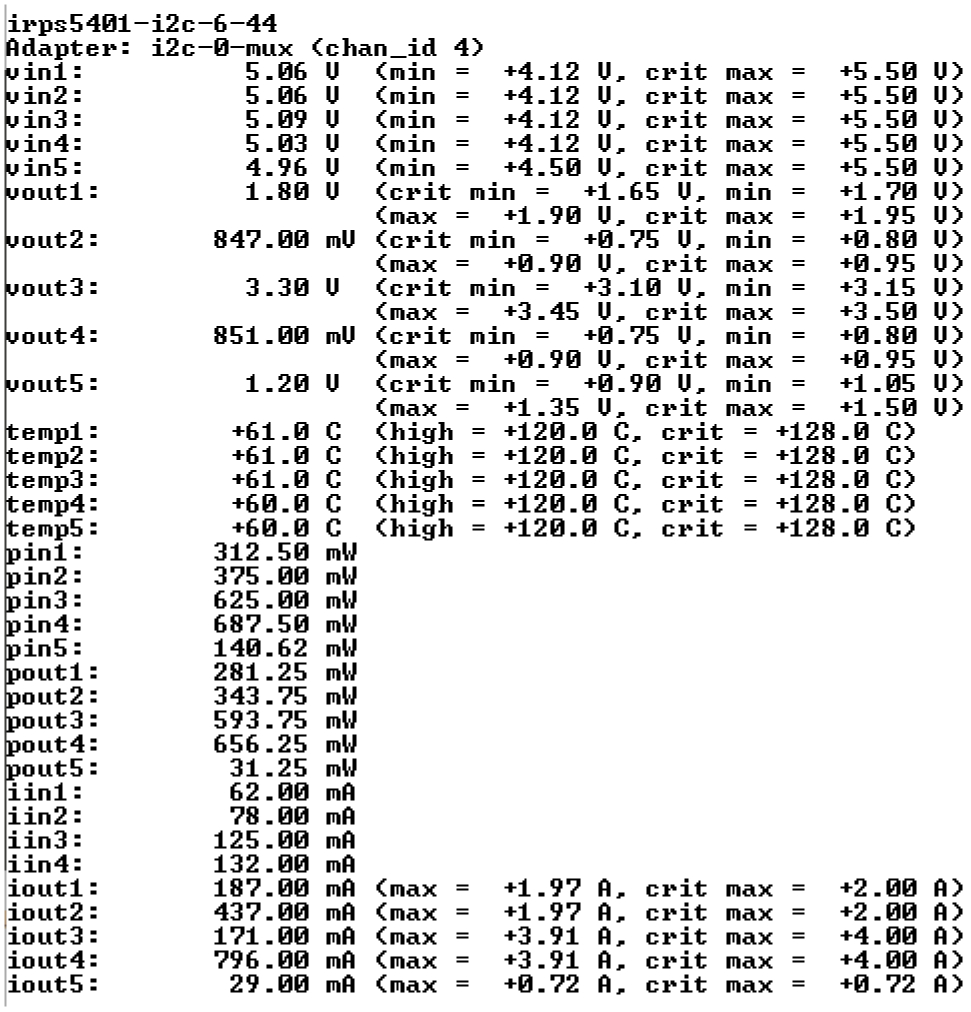
```
irps5401-i2c-6-43
Adapter: i2c-0-mux (chan_id 4)
vin1:          5.03 V  (min =  +4.12 V, crit max =  +5.50 V)
vin2:          4.97 V  (min =  +4.12 V, crit max =  +5.50 V)
vin3:          5.06 V  (min =  +4.12 V, crit max =  +5.50 V)
vin4:          5.00 V  (min =  +4.12 V, crit max =  +5.50 V)
vin5:          4.95 V  (min =  +0.00 V, crit max =  +6.00 V)
vout1:         1.79 V  (crit min =  +1.65 V, min =  +1.70 V)
                       (max =  +1.90 V, crit max =  +1.95 V)
vout2:         1.20 V  (crit min =  +1.10 V, min =  +1.15 V)
                       (max =  +1.25 V, crit max =  +1.30 V)
vout3:         1.10 V  (crit min =  +1.00 V, min =  +1.05 V)
                       (max =  +1.15 V, crit max =  +1.20 V)
vout4:       847.00 mV (crit min =  +0.65 V, min =  +0.74 V)
                       (max =  +0.99 V, crit max =  +1.00 V)
vout5:         3.30 V  (crit min =  +2.48 V, min =  +2.90 V)
                       (max =  +3.73 V, crit max =  +4.14 V)
temp1:        +60.0 C  (high = +120.0 C, crit = +128.0 C)
temp2:        +60.0 C  (high = +120.0 C, crit = +128.0 C)
temp3:        +60.0 C  (high = +120.0 C, crit = +128.0 C)
temp4:        +60.0 C  (high = +125.0 C, crit = +128.0 C)
temp5:        +60.0 C  (high = +120.0 C, crit = +128.0 C)
pin1:         62.50 mW
pin2:          0.00 W
pin3:         93.75 mW
pin4:        156.25 mW
pin5:        781.25 mW
pout1:        62.50 mW
pout2:         0.00 W
pout3:        93.75 mW
pout4:       156.25 mW
pout5:       515.62 mW
iin1:         15.00 mA
iin2:          0.00 A
iin3:         15.00 mA
iin4:         31.00 mA
iout1:        46.00 mA (max =  +1.75 A, crit max =  +2.00 A)
iout2:         0.00 A  (max =  +1.91 A, crit max =  +2.50 A)
iout3:       109.00 mA (max =  +3.91 A, crit max =  +4.00 A)
iout4:       203.00 mA (max =  +4.75 A, crit max =  +5.25 A)
iout5:       157.00 mA (max =  +0.72 A, crit max =  +0.72 A)
```

Figure 42 – Telemetry from PMIC 0x13/43



```
iio_hwmon-isa-0000
Adapter: ISA adapter
in1:           1.20 V
in2:           1.80 V
in3:         853.00 mV
in4:         856.00 mV
in5:           1.80 V
in6:           1.80 V
in7:         842.00 mV
in8:         840.00 mV
in9:         851.00 mV
in10:          1.80 V
in11:          1.10 V
temp1:        +53.6 C
temp2:        +52.7 C
```

Figure 43 – Telemetry from MPSoC Sysmon



```
irps5401-i2c-6-44
Adapter: i2c-0-mux (chan_id 4)
vin1:          5.06 V  (min =  +4.12 V, crit max =  +5.50 V)
vin2:          5.06 V  (min =  +4.12 V, crit max =  +5.50 V)
vin3:          5.09 V  (min =  +4.12 V, crit max =  +5.50 V)
vin4:          5.03 V  (min =  +4.12 V, crit max =  +5.50 V)
vin5:          4.96 V  (min =  +4.50 V, crit max =  +5.50 V)
vout1:         1.80 V  (crit min =  +1.65 V, min =  +1.70 V)
                       (max =  +1.90 V, crit max =  +1.95 V)
vout2:       847.00 mV (crit min =  +0.75 V, min =  +0.80 V)
                       (max =  +0.90 V, crit max =  +0.95 V)
vout3:         3.30 V  (crit min =  +3.10 V, min =  +3.15 V)
                       (max =  +3.45 V, crit max =  +3.50 V)
vout4:       851.00 mV (crit min =  +0.75 V, min =  +0.80 V)
                       (max =  +0.90 V, crit max =  +0.95 V)
vout5:         1.20 V  (crit min =  +0.90 V, min =  +1.05 V)
                       (max =  +1.35 V, crit max =  +1.50 V)
temp1:        +61.0 C  (high = +120.0 C, crit = +128.0 C)
temp2:        +61.0 C  (high = +120.0 C, crit = +128.0 C)
temp3:        +61.0 C  (high = +120.0 C, crit = +128.0 C)
temp4:        +60.0 C  (high = +120.0 C, crit = +128.0 C)
temp5:        +60.0 C  (high = +120.0 C, crit = +128.0 C)
pin1:        312.50 mW
pin2:        375.00 mW
pin3:        625.00 mW
pin4:        687.50 mW
pin5:        140.62 mW
pout1:       281.25 mW
pout2:       343.75 mW
pout3:       593.75 mW
pout4:       656.25 mW
pout5:        31.25 mW
iin1:         62.00 mA
iin2:         78.00 mA
iin3:        125.00 mA
iin4:        132.00 mA
iout1:       187.00 mA (max =  +1.97 A, crit max =  +2.00 A)
iout2:       437.00 mA (max =  +1.97 A, crit max =  +2.00 A)
iout3:       171.00 mA (max =  +3.91 A, crit max =  +4.00 A)
iout4:       796.00 mA (max =  +3.91 A, crit max =  +4.00 A)
iout5:        29.00 mA (max =  +0.72 A, crit max =  +0.72 A)
```

Figure 44 – Telemetry from PMIC 0x14/44

As you can see the current, voltage, and temperature measurements are reported back..

15 PMIC Version Check and Update

Another capability that we have is to check the version of the programming inside the PMICs. As described in the Ultra96-V2 PCN19003, there have been multiple releases of the PMIC programming files. This is also described in this blog.

https://www.element14.com/community/groups/power-management/blog/2020/01/23/infineon-pmic-updates-to-avnet-products

You can now check the version of your PMIC programming and update to the latest version if you want.

WARNINGS!

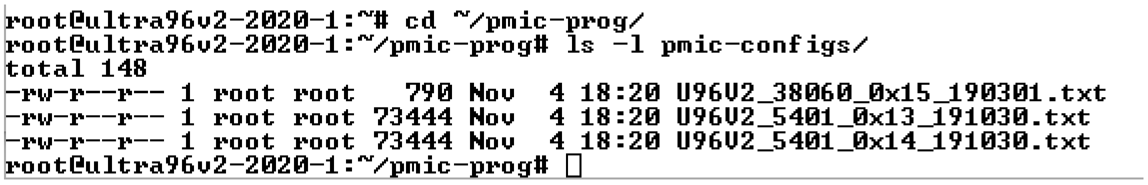
The Infineon PMICs only have 26 programming locations. Each time the devices are reprogrammed, another location is consumed. Use with caution!

If you accidentally program a device with the wrong file (like the 0x14 device with the 0x13 file or vice versa), you WILL damage your board, and it will NOT be covered under warranty. You must program the **correct file** to the **correct device**.
Permanent damage will occur otherwise.

15.1 PMIC Version Checking

1. Change directories to the location of the PMIC programmer and list the config files.

```
# cd ~/pmic-prog

# ls -l pmic-configs/
```



```
root@ultra96v2-2020-1:~# cd ~/pmic-prog/
root@ultra96v2-2020-1:~/pmic-prog# ls -l pmic-configs/
total 148
-rw-r--r-- 1 root root   790 Nov  4 18:20 U96V2_38060_0x15_190301.txt
-rw-r--r-- 1 root root 73444 Nov  4 18:20 U96V2_5401_0x13_191030.txt
-rw-r--r-- 1 root root 73444 Nov  4 18:20 U96V2_5401_0x14_191030.txt
root@ultra96v2-2020-1:~/pmic-prog# 
```

Figure 45 – PMIC Config Files

The naming convention for the files is as follows.

```
<board>_<device>_<channel>_<datecode in YYMMDD>
```

`<board>`	`U96V2`
`<device>`	`5401 or 38060`
`<channel>`	`0x13, 0x14, or 0x15`
`<datecode>`	`191030 = 2019, October 30`

7. To see the capabilities of the pmic_prog utility, access the help menu

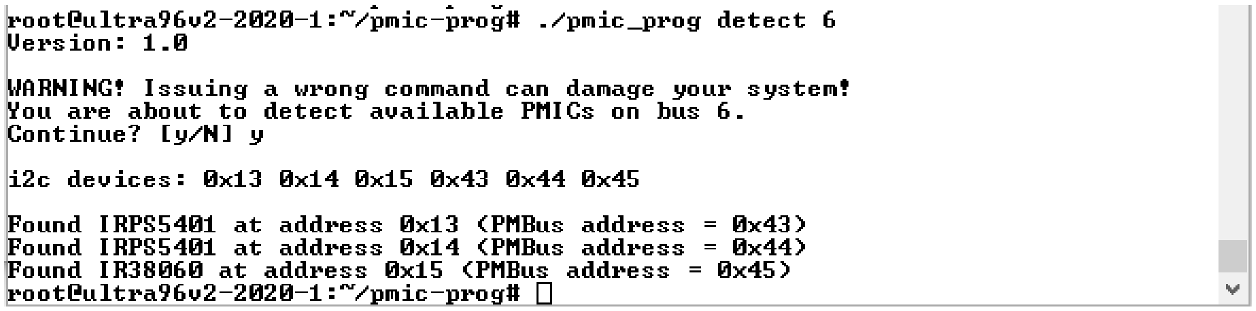
```
# ./pmic_prog -help
```

```
Version: 1.0

Usage: pmic_prog [-h] COMMANDS
  -h = display this help and exit

COMMANDS:

detect I2C_BUS to detect all the PMICs on an i2c bus
    with:
      I2C_BUS the i2c bus number (decimal: 0-255)
    Example:
    " pmic_prog detect 6 " to detect all the PMICs on an i2c bus 6

read I2C_BUS CHIP_ADDR REG_ADDR to read a byte at a specific address
    with:
      I2C_BUS the i2c bus number (decimal: 0-255)
      CHIP_ADDR the chip address (hex: 0x00 - 0xFF)
      REG_ADDR the address of the register to read  (hex: 0x0000 - 0xFFFF)

    Example:
    " pmic_prog read 6 0x13 0x069e " to read the byte at address 0x069e (loop A ot_fault_limit for
IRPS5401) of chip 0x13 on i2c bus 6

write I2C_BUS CHIP_ADDR REG_ADDR VALUE to write a byte at a specific address
    with:
      I2C_BUS the i2c bus number (decimal: 0-255)
      CHIP_ADDR the chip address (hex: 0x00 - 0xFF)
      REG_ADDR the address of the register to write  (hex: 0x0000 - 0xFFFF)
      VALUE the byte to write (hex: 0x00 - 0xFF)

    Example:
    " pmic_prog write 6 0x13 0x069e 0x7F " to write 0x7F at address 0x069e (loop A ot_fault_limit
for IRPS5401) of chip 0x13 on i2c bus 6

read-registers I2C_BUS CHIP_ADDR [-o OUTPUT_FILE] [-i INPUT_FILE] to read all the registers
    with:
      I2C_BUS the i2c bus number (decimal: 0-255)
      CHIP_ADDR the chip address (hex: 0x00 - 0xFF)
      -o OUTPUT_FILE: Save the register values in the 'OUTPUT_FILE' file (optional)
      -i INPUT_FILE: Check the register values from the given 'INPUT_FILE' file, and check the
differences (optional)

    Example:
    " pmic_prog read-registers 6 0x13 -o new_file.txt -i 0x13_Oct_01.txt " to save the register
            values in the file 'new_file.txt' and check the differences with those in
'0x13_Oct_01.txt'.

program I2C_BUS CHIP_ADDR [-d] [-i INPUT_FILE] to write the registers and program the PMIC on the
first available USER slot
    with:
      I2C_BUS the i2c bus number (decimal: 0-255)
      CHIP_ADDR the chip address (hex: 0x00 - 0xFF)
      -d: dry-run The command will only check the number of USER slots available and write registers
if input file given (optional)
      -i INPUT_FILE: write the registers from this config file (optional)

    Example:
    " pmic_prog program 6 0x13 -i 0x13_Oct_01.txt " to program the PMIC with register values from
'0x13_Oct_01.txt'.
```

Next use a basic ‘detect’ command to confirm we see the correct PMICs.

8. Enter the ‘detect’ command on I2C bus 6

```
# ./pmic_prog detect 6
```

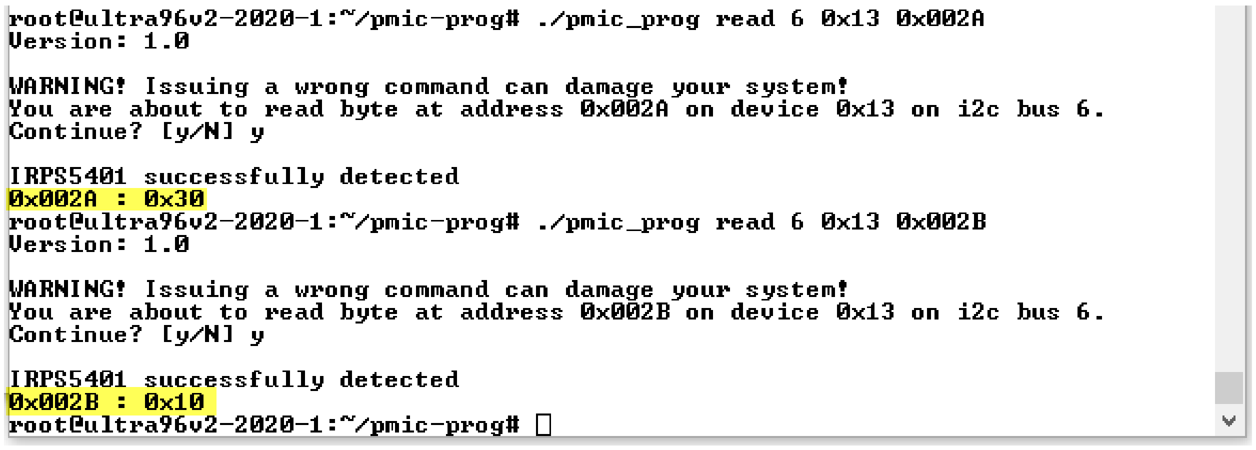


```
root@ultra96v2-2020-1:~/pmic-prog# ./pmic_prog detect 6
Version: 1.0

WARNING! Issuing a wrong command can damage your system!
You are about to detect available PMICs on bus 6.
Continue? [y/N] y

i2c devices: 0x13 0x14 0x15 0x43 0x44 0x45

Found IRPS5401 at address 0x13 (PMBus address = 0x43)
Found IRPS5401 at address 0x14 (PMBus address = 0x44)
Found IR38060 at address 0x15 (PMBus address = 0x45)
root@ultra96v2-2020-1:~/pmic-prog#
```

Figure 46 – PMICs Detected On I2C Bus 6

9. For the Rocky/5401 devices (0x13 and 0x14) a date code is stored at locations 0x2A and 0x2B for each PMIC. You can see which revision you have by reading these two registers. 0x2A has the day and 0x2B has the month of release.

```
# ./pmic_prog read 6 0x13 0x002A

# ./pmic_prog read 6 0x13 0x002B

# ./pmic_prog read 6 0x14 0x002A

# ./pmic_prog read 6 0x14 0x002B
```

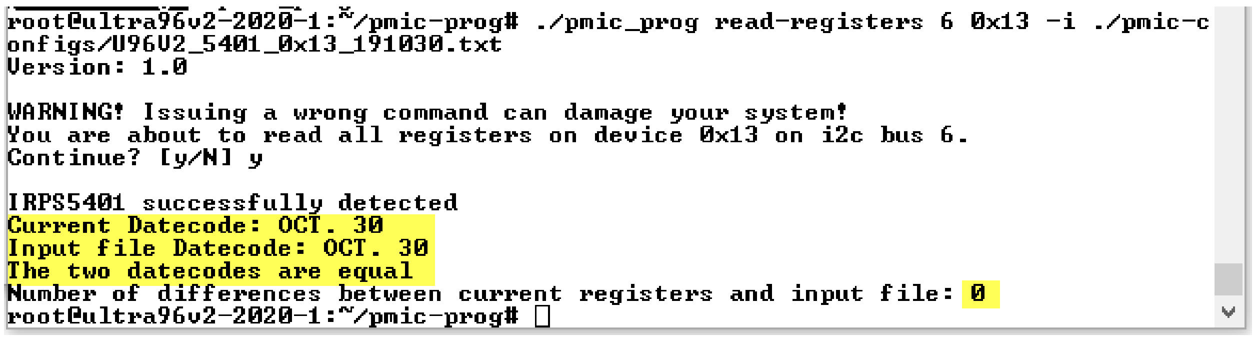


```
root@ultra96v2-2020-1:~/pmic-prog# ./pmic_prog read 6 0x13 0x002A
Version: 1.0

WARNING! Issuing a wrong command can damage your system!
You are about to read byte at address 0x002A on device 0x13 on i2c bus 6.
Continue? [y/N] y

IRPS5401 successfully detected
0x002A : 0x30
root@ultra96v2-2020-1:~/pmic-prog# ./pmic_prog read 6 0x13 0x002B
Version: 1.0

WARNING! Issuing a wrong command can damage your system!
You are about to read byte at address 0x002B on device 0x13 on i2c bus 6.
Continue? [y/N] y

IRPS5401 successfully detected
0x002B : 0x10
root@ultra96v2-2020-1:~/pmic-prog#
```

Figure 47 – Example datecode October 30 for Device 0x13

10. To perform a comparison against an existing configuration file, use the read-registers command with an input file, as follows.

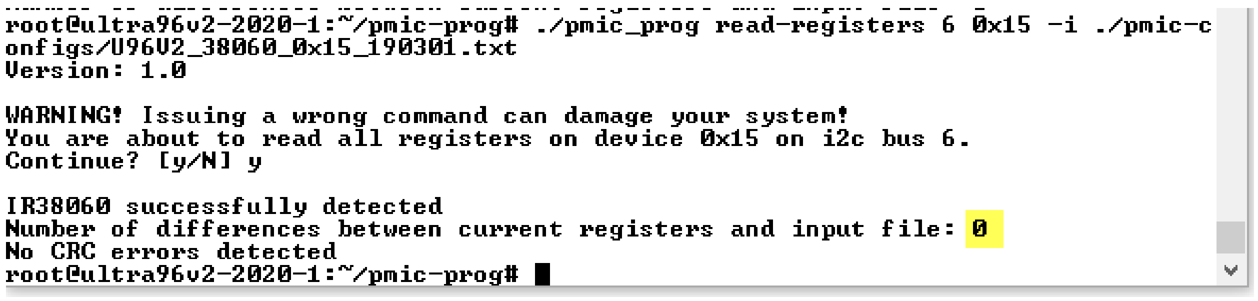
```
# ./pmic_prog read-registers 6 0x13 -i ./pmic-configs/U96V2_5401_0x13_191030.txt

# ./pmic_prog read-registers 6 0x14 -i ./pmic-configs/U96V2_5401_0x14_191030.txt

# ./pmic_prog read-registers 6 0x15 -i ./pmic-configs/U96V2_38060_0x15_190301.txt
```



```
root@ultra96v2-2020-1:~/pmic-prog# ./pmic_prog read-registers 6 0x13 -i ./pmic-c
onfigs/U96V2_5401_0x13_191030.txt
Version: 1.0

WARNING! Issuing a wrong command can damage your system!
You are about to read all registers on device 0x13 on i2c bus 6.
Continue? [y/N] y

IRPS5401 successfully detected
Current Datecode: OCT. 30
Input file Datecode: OCT. 30
The two datecodes are equal
Number of differences between current registers and input file: 0
root@ultra96v2-2020-1:~/pmic-prog#
```

Figure 48 – Compare PMIC Config to a File, 0x13 or 0x14

Since the Manhattan/38060 does not store a date code in the config, look specifically at the number of differences to see if your programming matches.



```
root@ultra96v2-2020-1:~/pmic-prog# ./pmic_prog read-registers 6 0x15 -i ./pmic-c
onfigs/U96V2_38060_0x15_190301.txt
Version: 1.0

WARNING! Issuing a wrong command can damage your system!
You are about to read all registers on device 0x15 on i2c bus 6.
Continue? [y/N] y

IR38060 successfully detected
Number of differences between current registers and input file: 0
No CRC errors detected
root@ultra96v2-2020-1:~/pmic-prog#
```

Figure 49 – Compare PMIC Config to a File, 0x15

15.2 PMIC Reprogramming

If you are not using the latest PMIC programming files, then you may find your board does not perform to its maximum potential. The included pmic_prog utility can also reprogram your PMICs.

WARNINGS!

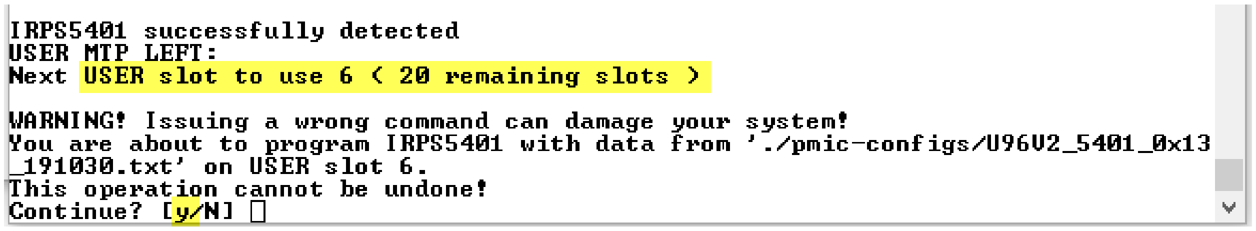
The Infineon PMICs only have 26 programming locations. Each time the devices are reprogrammed, another location is consumed. Use with caution!

If you accidentally program a device with the wrong file (like the 0x14 device with the 0x13 file or vice versa), you WILL damage your board, and it will NOT be covered under warranty. You must program the **correct file** to the **correct device**.
Permanent damage will occur otherwise.

1. To reprogram the 0x13 device, enter the following command, **making sure that the two highlighted numbers match**!

```
# ./pmic_prog program 6 0x13 -i ./pmic-configs/U96V2_5401_0x13_191030.txt
```

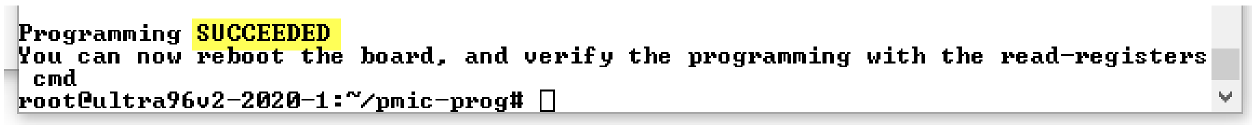
2. Enter ‘y’ to confirm.
3. A second warning requires you to confirm again. This also tells you how many of the USER memory slots you have consumed on your board. Read carefully, then enter ‘y’ to proceed.



```
IRPS5401 successfully detected
USER MTP LEFT:
Next USER slot to use 6 ( 20 remaining slots )

WARNING! Issuing a wrong command can damage your system!
You are about to program IRPS5401 with data from './pmic-configs/U96V2_5401_0x13
_191030.txt' on USER slot 6.
This operation cannot be undone!
Continue? [y/N]
```

Figure 50 – Carefully Read, then Confirm to Proceed

4. The programming only takes a few seconds, after which you should receive the following confirmation.



```
Programming SUCCEEDED
You can now reboot the board, and verify the programming with the read-registers
 cmd
root@ultra96v2-2020-1:~/pmic-prog#
```

Figure 51 – Programming 0x13 Succeeded

If you will program another device, do not reboot yet.

5. To reprogram the 0x14 device, enter the following command, **making sure that the two highlighted numbers match**!

```
# ./pmic_prog program 6 0x14 -i ./pmic-configs/U96V2_5401_0x14_191030.txt
```

6. Follow the procedure as before.

The 0x15 / Manhattan / 38060 device programming has not been updated since the original release of the board. Therefore, this one does not need to be reprogrammed.

7. You should now power cycle the board. Enter the '`poweroff`' command. After the board has fully shut down (no LEDs other than Input Power), then turn the board back on by pressing the ON switch (SW4).

16 Ethernet Gadget

If you would like to enable the USB Gadget Ethernet on the Ultra96-V2, please follow the instructions found in this blog.

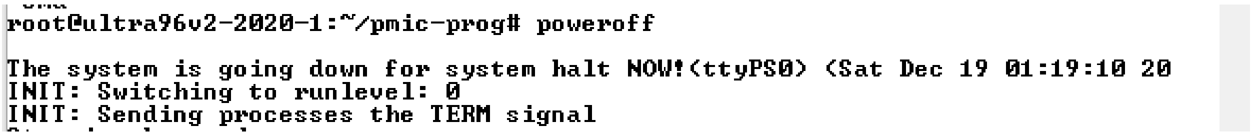
http://avnet.me/ultra96-ethernet-gadget

17 Power Off

When you are done experimenting with your Ultra96-V2 and wish to power off the board, there are several ways to power off the board. You can do it from the command line with a ‘shutdown -h now’ command. However, we will have you take advantage of the on-board On/Off Controller that interacts with the MPSoC Power Management Unit to initiate a controlled shutdown.

1. Press and release the Power button (SW4) located on the top side of your Ultra96-V2 next to USB port J8.

2. You will notice your board does not power down immediately. It will take roughly 10-20 seconds for your board to completely power down. The reason behind this is it is adhering to the various power down sequencing requirements. See the message in the terminal as shown.



```
root@ultra96v2-2020-1:~/pmic-prog# poweroff

The system is going down for system halt NOW!(ttyPS0) (Sat Dec 19 01:19:10 20
INIT: Switching to runlevel: 0
INIT: Sending processes the TERM signal
```

Figure 52 – Power Down Initiated Through Short Press of SW4

3. Please note, if you do not let your Ultra96-V2 power off as per the power down sequencing requirements (such as unplugging the barrel jack), your microSD Card may get corrupted or damaged.

4. To force poweroff of the Ultra96-V2, you can also press and hold SW4 for 10 seconds. This is useful when the soft power-off doesn’t work.

18 Getting Help and Support

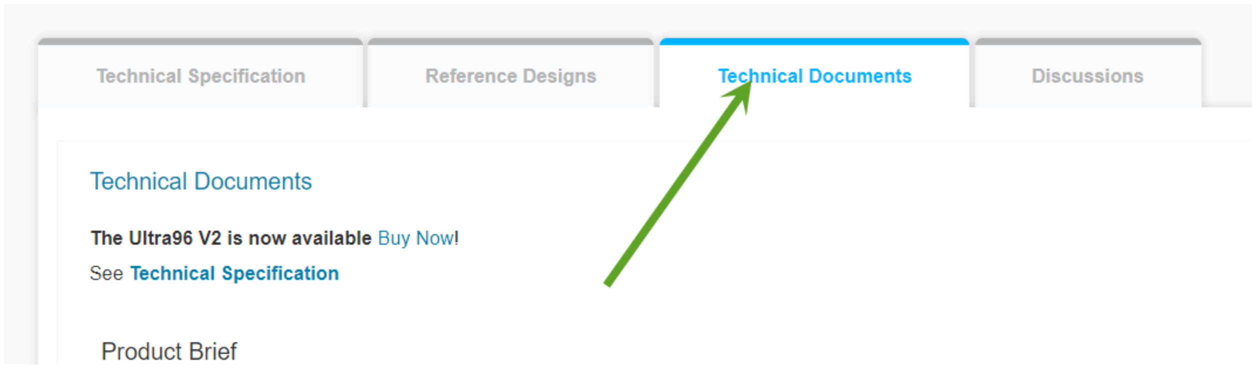
18.1 Avnet Support

The Ultra96-V2 is a versatile development kit that allows evaluation of the Zynq MPSoC, which can help you adopt Zynq into your next design. All technical support is offered through http://avnet.me/Ultra96_Forum. Ultra96-V2 users are encouraged to participate in the forums and offer help to others when possible.

To access Ultra96-V2 collateral, please visit the community support page at:

http://avnet.me/ultra96-v2

To access the latest Ultra96-V2 documentation, click on the ***Technical Documents*** tab:



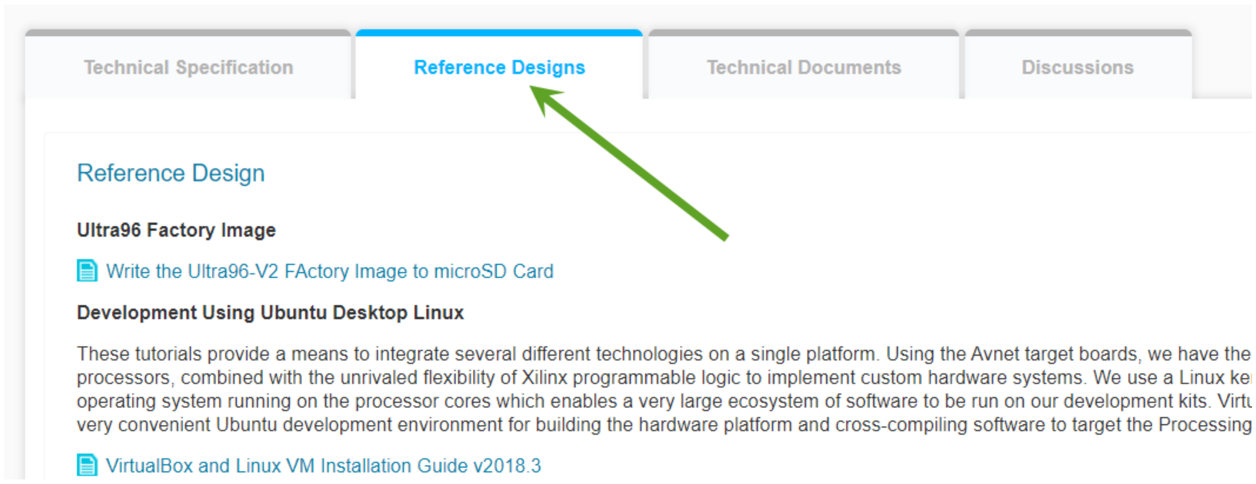
Technical Documents

The Ultra96 V2 is now available Buy Now!

See **Technical Specification**

Product Brief

To access the latest reference designs for Ultra96-V2, click on the Reference Designs tab:



Reference Design

Ultra96 Factory Image

Write the Ultra96-V2 FActory Image to microSD Card

Development Using Ubuntu Desktop Linux

These tutorials provide a means to integrate several different technologies on a single platform. Using the Avnet target boards, we have the processors, combined with the unrivaled flexibility of Xilinx programmable logic to implement custom hardware systems. We use a Linux ke operating system running on the processor cores which enables a very large ecosystem of software to be run on our development kits. Virtu very convenient Ubuntu development environment for building the hardware platform and cross-compiling software to target the Processing

VirtualBox and Linux VM Installation Guide v2018.3

To access the Ultra96-V2 technical forums, go to http://avnet.me/Ultra96_Forum .

To view training and videos, go to http://avnet.me/TTC or http://avnet.me/TTC_on_Demand.

18.2 Xilinx Support

For questions regarding products within the Product Entitlement Account, visit the Contact Support site for Xilinx:

https://www.xilinx.com/support/service-portal/contact-support.html

For technical support including the installation and use of the product license file, contact Xilinx Online Technical Support at www.xilinx.com/support. The following assistance resources are also available on the website:

- Software, IP and documentation updates
- Access to technical support web tools
- Searchable answer database with over 4,000 solutions
- User forums

19 Installing and Licensing Xilinx Software

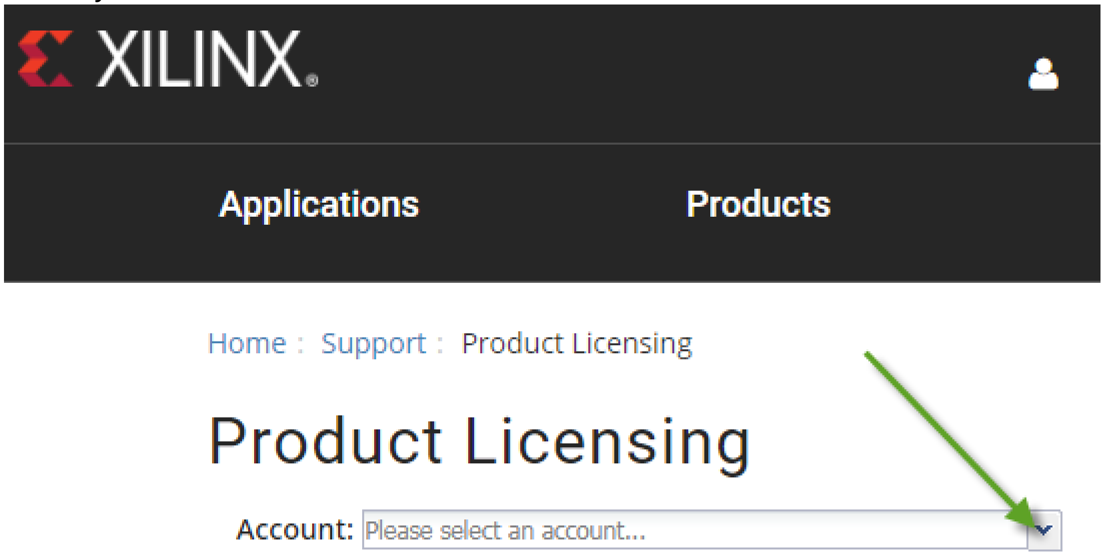
19.1 Install Vivado Design Suite, Design Edition

The Zynq device on the Ultra96-V2 is supported in Vivado Design Suite, Design Edition. Version 2018.1 or later is required to use the board definition file provided on the Avnet GitHub.

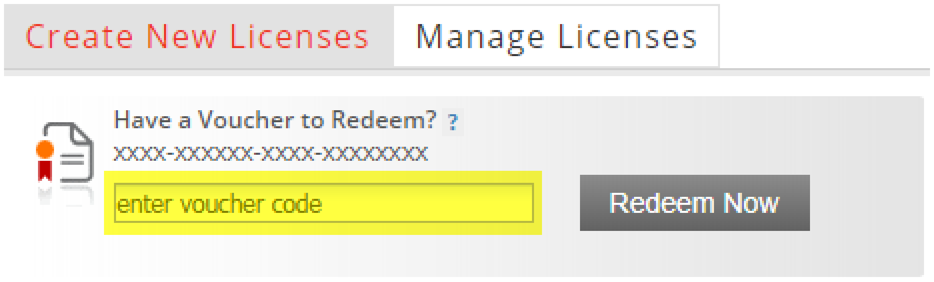
You must license your Vivado Design Suite, Design Edition with the license that came with your Ultra96-V2. To obtain your free license, visit the following website and insert the voucher code from the certificate included in your kit:

http://www.xilinx.com/getlicense

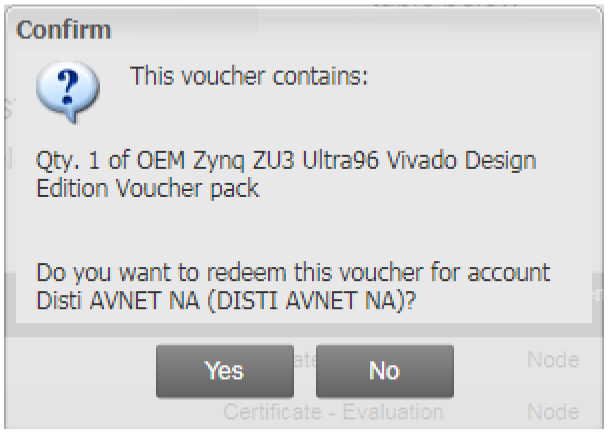
1. Log in
2. Fill out information at Product Licensing - Name and Address Verification, then click Next
3. Select your Account


XILINX
Applications
Products
Home : Support : Product Licensing
Product Licensing
Account:
Please select an account...

4. Enter your voucher code here, then click Redeem Now.



5. At the confirmation screen, click Yes.


Confirm

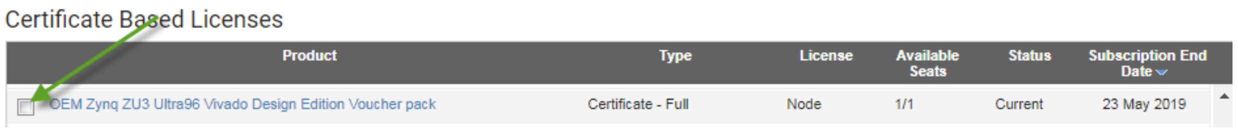
This voucher contains:

Qty. 1 of OEM Zynq ZU3 Ultra96 Vivado Design Edition Voucher pack

Do you want to redeem this voucher for account Disti AVNET NA (DISTI AVNET NA)?

Figure 53 – Voucher Confirmation

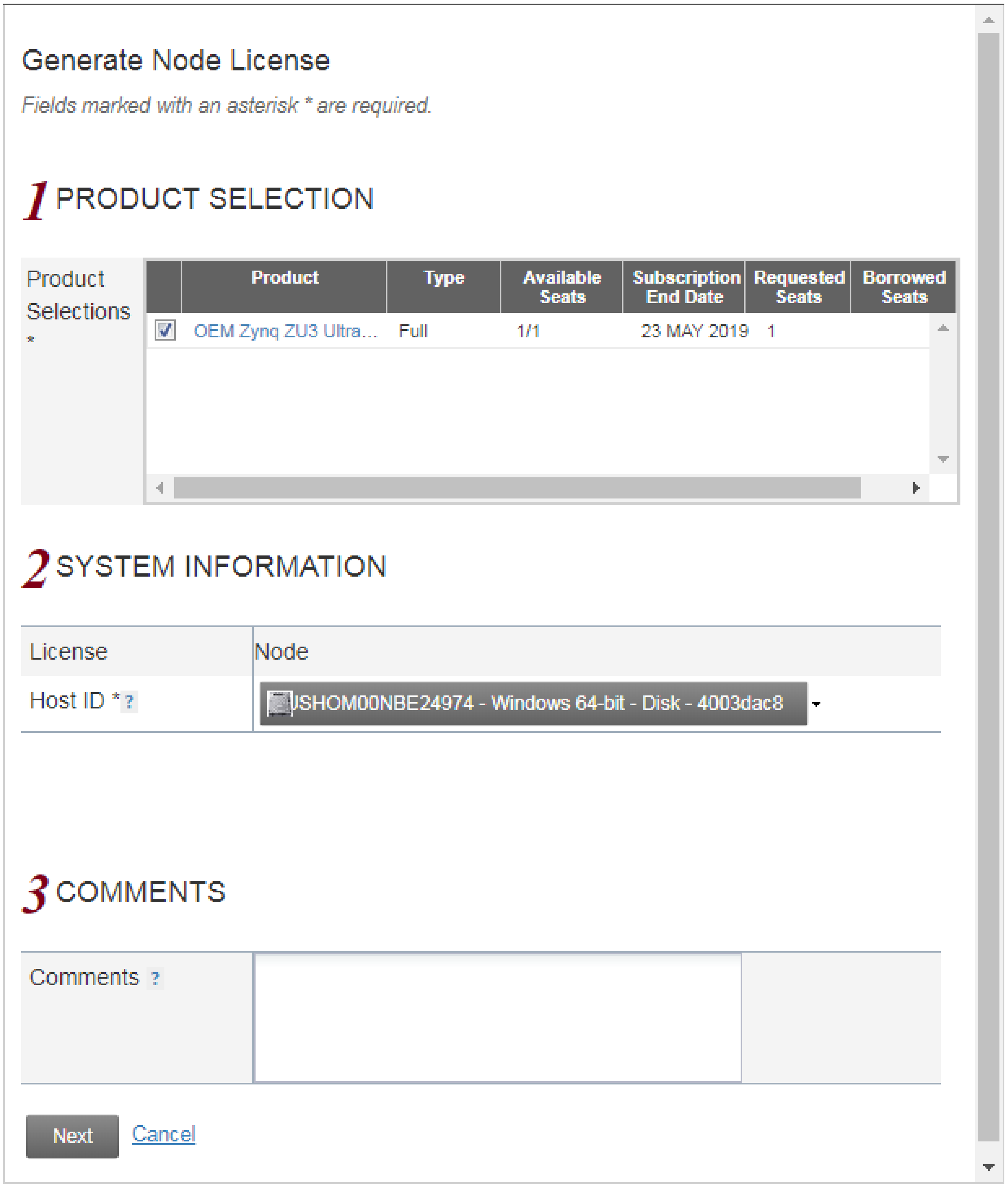
6. Under Certificate Based Licenses, find OEM Zynq ZU3 Ultra96 Vivado Design Edition Voucher pack and check the box. Now click Generate Node-Locked License.


Certificate Based Licenses

Product	Type	License	Available Seats	Status	Subscription End Date
OEM Zynq ZU3 Ultra96 Vivado Design Edition Voucher pack	Certificate - Full	Node	1/1	Current	23 May 2019

Figure 54 – Generate Node-Locked

7. Create or select your Host ID. Click Next.



Generate Node License

*Fields marked with an asterisk * are required.*

1 PRODUCT SELECTION

Product Selections *

	Product	Type	Available Seats	Subscription End Date	Requested Seats	Borrowed Seats
✓	OEM Zynq ZU3 Ultra...	Full	1/1	23 MAY 2019	1	

2 SYSTEM INFORMATION

License	Node
Host ID * ?	USHOM00NBE24974 - Windows 64-bit - Disk - 4003dac8

3 COMMENTS

Comments ?

Figure 55 – Select Host Information

8. Review the license request, then click **Next** again.

If a full seat of Vivado System or Design Edition has already been installed, then no further software will be needed. Please check online for any updates at:

www.xilinx.com/support/download/index.htm

For detailed instructions on installing and licensing the Xilinx tools, please refer to the latest version of **Vivado Design Suite User Guide *Release Notes, Installation, and Licensing* (UG973)**.

20 Certification Disclaimer

Both CE and FCC certifications are necessary for system level products in those countries governed by these regulatory bodies.

Because Avnet boards are intended for evaluation kits only and destined for professionals (you) to be used solely at research and development facilities for such purposes, they are considered exempt from the EU product directives and normally are not tested for CE or FCC compliance.

If you choose to use your board to transmit using an antenna, it is your responsibility to make sure that you are in compliance with all laws for the country, frequency, and power levels in which the device is used. Additionally, some countries regulate reception in certain frequency bands. Again, it is the responsibility of the user to maintain compliance with all local laws and regulations.

This board should be used in a controlled lab environment by professional developers for prototype and development purposes only. The board included in the kit is not intended for production use unless additional end product testing and certification is performed.

21 Safety Warnings

This product shall only be connected to an external power supply that is 96boards compliant.

Only compatible plug-in modules shall be connected to Ultra96-V2. The connection of incompatible devices may affect compliance or result in damage to the unit and void the warranty.

This product shall be operated in a well-ventilated environment. If a case is used, it shall have adequate ventilation.

22 RF Certification

The frequency range is 2.412GHz ~ 2.472GHz (2.4GHz ISM Band).

The radio is IEEE 802.11 b/g/n (1x1) compliant for up to 72 Mbps PHY rate

The ATWILC3000-MR110CA has regulatory approval in more than 75 countries around the world. More information on RF certification for the Microchip ATWILC3000 module is available here:

http://ww1.microchip.com/downloads/en/DeviceDoc/ATWILC3000-MR110CA%20Worldwide%20Regulatory%20Information.pdf