

Ultra96-V2 Getting Started Guide

Version 2.0

Document Control

Document Version: 2.0

Document Date: 18 Dec 2020

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.

Contents

1	Getting Started with Ultra96-V2	6
2	What's Inside the Box?	7
2.1	Optional add-on items:	7
3	What's on the Web?.....	8
3.1	Official Documentation:	8
3.2	Tutorials and Reference Designs:	8
3.3	Trainings and Videos:.....	8
4	Ultra96-V2 Key Features	9
5	Ultra96-V2 Basic Setup and Operation	11
6	Example Design.....	12
7	Run the Out of Box Design	12
8	Connect to Webserver	16
9	Ultra96-V2 GPIO LEDs Example Project	18
10	Tutorials	20
11	Access Ultra96-V2 Linux Terminal over SSH	21
12	Connect to External Wi-Fi.....	24
13	Reading On-board Telemetry	27
14	PMIC Version Check and Update	31
14.1	PMIC Version Checking	31
14.2	PMIC Reprogramming.....	35
15	Power Off.....	37
16	Getting Help and Support	38
16.1	Avnet Support.....	38
16.2	Xilinx Support	39
17	Installing and Licensing Xilinx Software	40
17.1	Install Vivado Design Suite, Design Edition	40
18	Certification Disclaimer	44
19	Safety Warnings	44
20	RF Certification	44

Figures

Figure 1 – Ultra96-V2	6
Figure 2 – Ultra96-V2 Block Diagram.....	10
Figure 3 – Ultra96-V2 Topology	11
Figure 4 – Ultra96-V2 SW3 Boot Mode Switch Location.....	12
Figure 5 – Ultra96-V2 with older 3-pin JTAG/UART Pod.....	13
Figure 6 – Ultra96-V2 with newer 4-pin JTAG/UART Pod.....	14
Figure 7 – Press the SW4 Power On Button	15
Figure 8 – Ultra96-V2 As An Available Network.....	16
Figure 9 – Connected to Ultra96-V2 Webserver	17
Figure 10 – Ultra96-V2 User LEDs.....	18
Figure 11 – Ultra96-V2 GPIO LEDs	18
Figure 12 – LED0/D3 Changed to Steady-state ON.....	19
Figure 13 – Ultra96-V2 Tutorials	20
Figure 14 – TeraTerm New Connection	21
Figure 15 – SSH Terminal Settings.....	22
Figure 16 – Click Continue	22
Figure 17 – SSH Authentication	23
Figure 18 – Ultra96-V2 Terminal	23
Figure 19 – WiFi Setup.....	24
Figure 20 – Discovering Networks	24
Figure 21 – Successfully Connected	25
Figure 22 – Determine IP Address Assigned to Ultra96-V2	25
Figure 23 – Connected to Ultra96-V2 Via External Wi-Fi	26
Figure 24 – I2Cdetect.....	27
Figure 25 – PMIC on I2C Mux Channel 4.....	27
Figure 26 – PMICs on I2C-6.....	28
Figure 27 – Telemetry from PMIC 0x15/45	28
Figure 28 – Telemetry from PMIC 0x13/43	29
Figure 29 – Telemetry from MPSoC Sysmon.....	29
Figure 30 – Telemetry from PMIC 0x14/44	30
Figure 31 – PMIC Config Files	31
Figure 32 – PMICs Detected On I2C Bus 6.....	33

Figure 33 – Example datecode for Device 0x13, with Month = 0x10 and Day = 0x3033	
Figure 34 – Compare PMIC Config to a File, 0x13 or 0x14.....	34
Figure 35 – Compare PMIC Config to a File, 0x15.....	34
Figure 36 – Carefully Read, then Confirm to Proceed.....	35
Figure 37 – Programming 0x13 Succeeded	35
Figure 38 – Power Down Initiated Through Short Press of SW4.....	37
Figure 39 – Voucher Confirmation	41
Figure 40 – Generate Node-Locked.....	41
Figure 41 – Select Host Information	42

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.

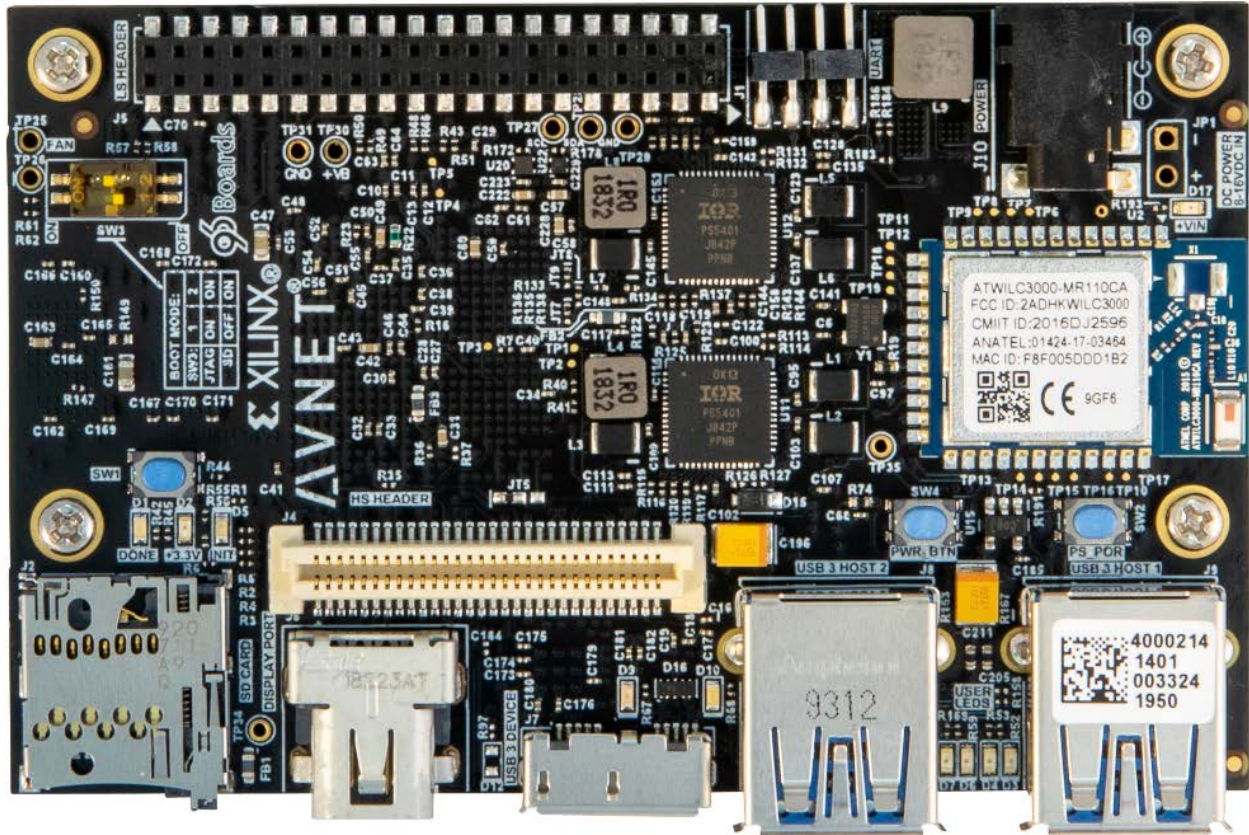


Figure 1 – Ultra96-V2

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
- 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

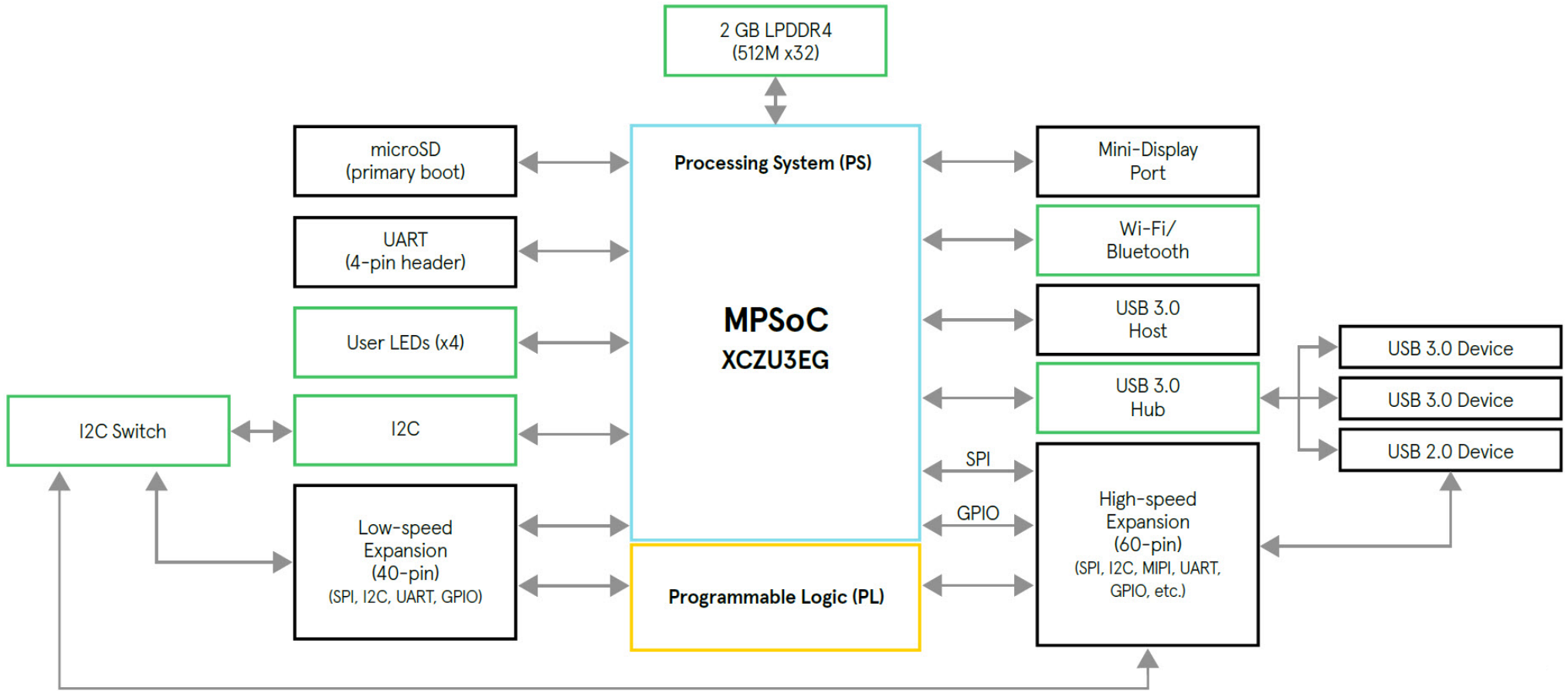


Figure 2 – Ultra96-V2 Block Diagram

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.

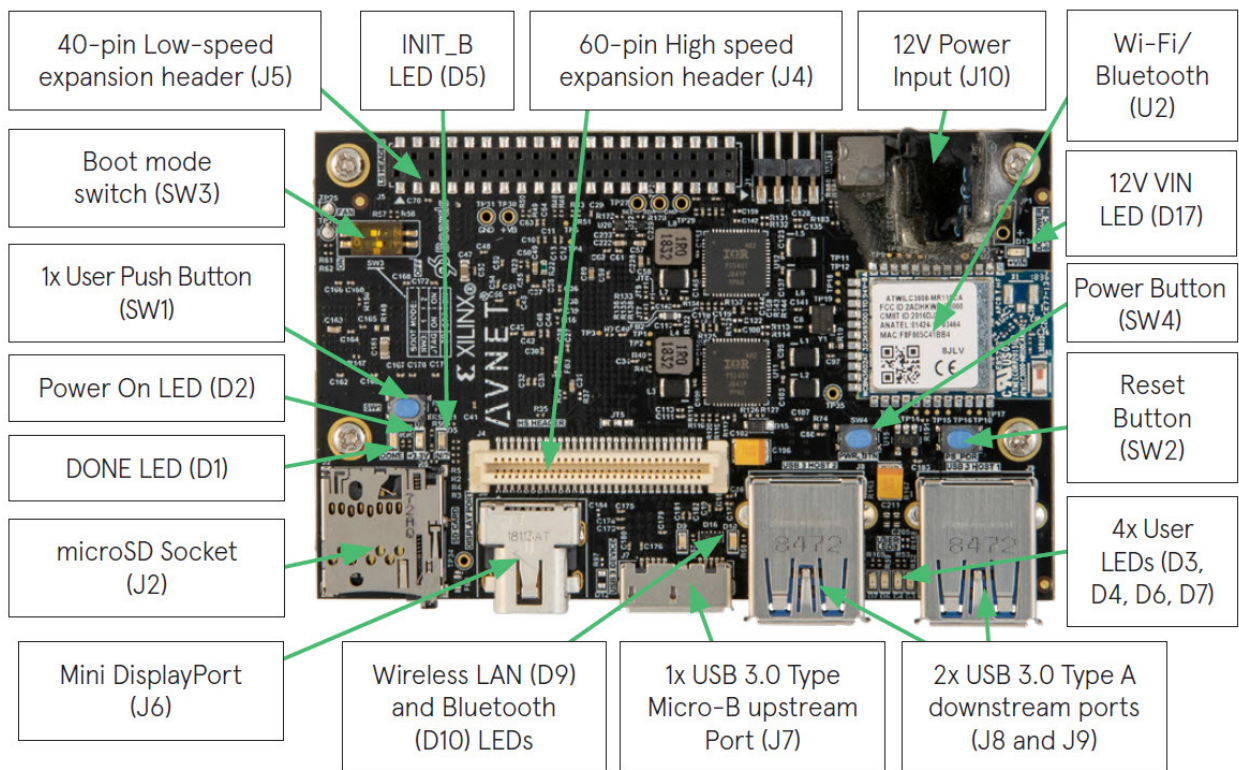


Figure 3 – Ultra96-V2 Topology

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 \(517 MB\)](http://avnet.me/ultra96-v2-oob)
MD5 Checksum for Image: d2676f376e74940e7cb02cbee6aecd49
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: tssh2.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.

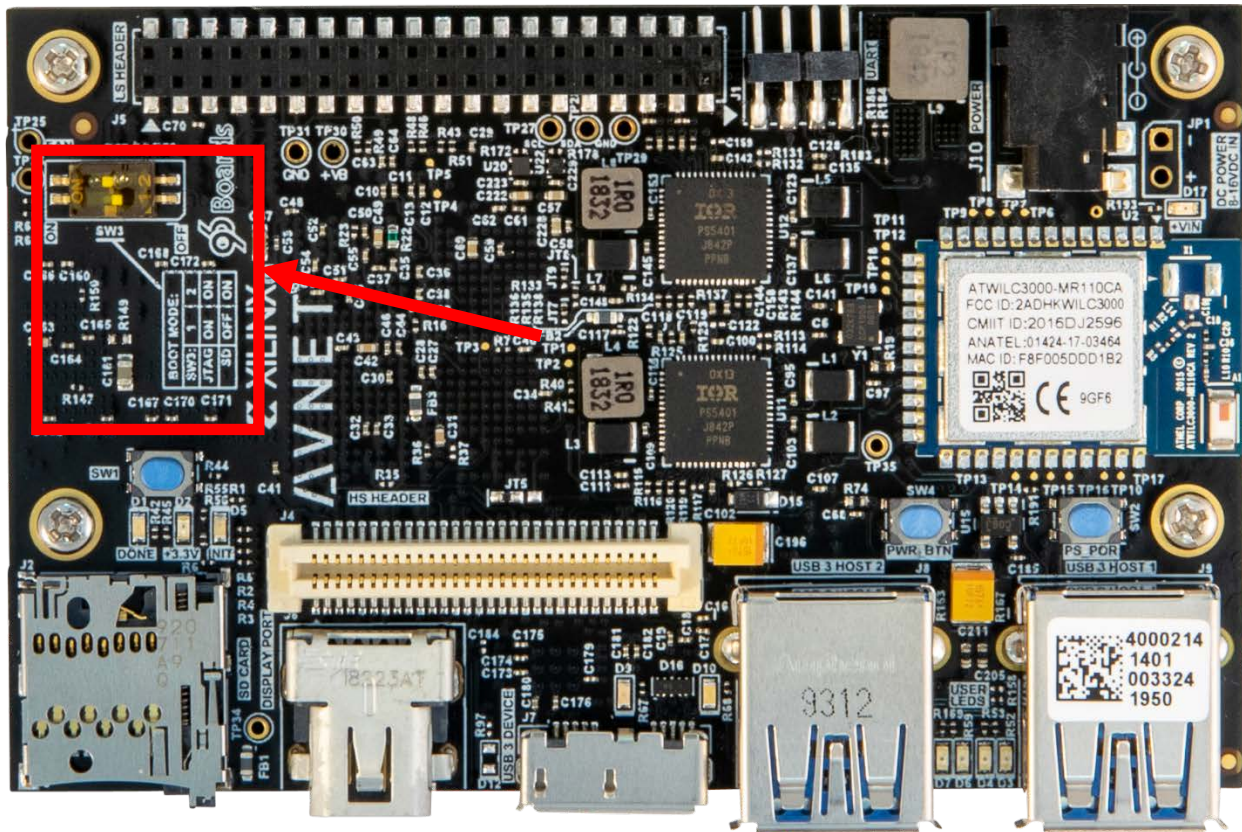


Figure 4 – Ultra96-V2 SW3 Boot Mode Switch Location

- 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>

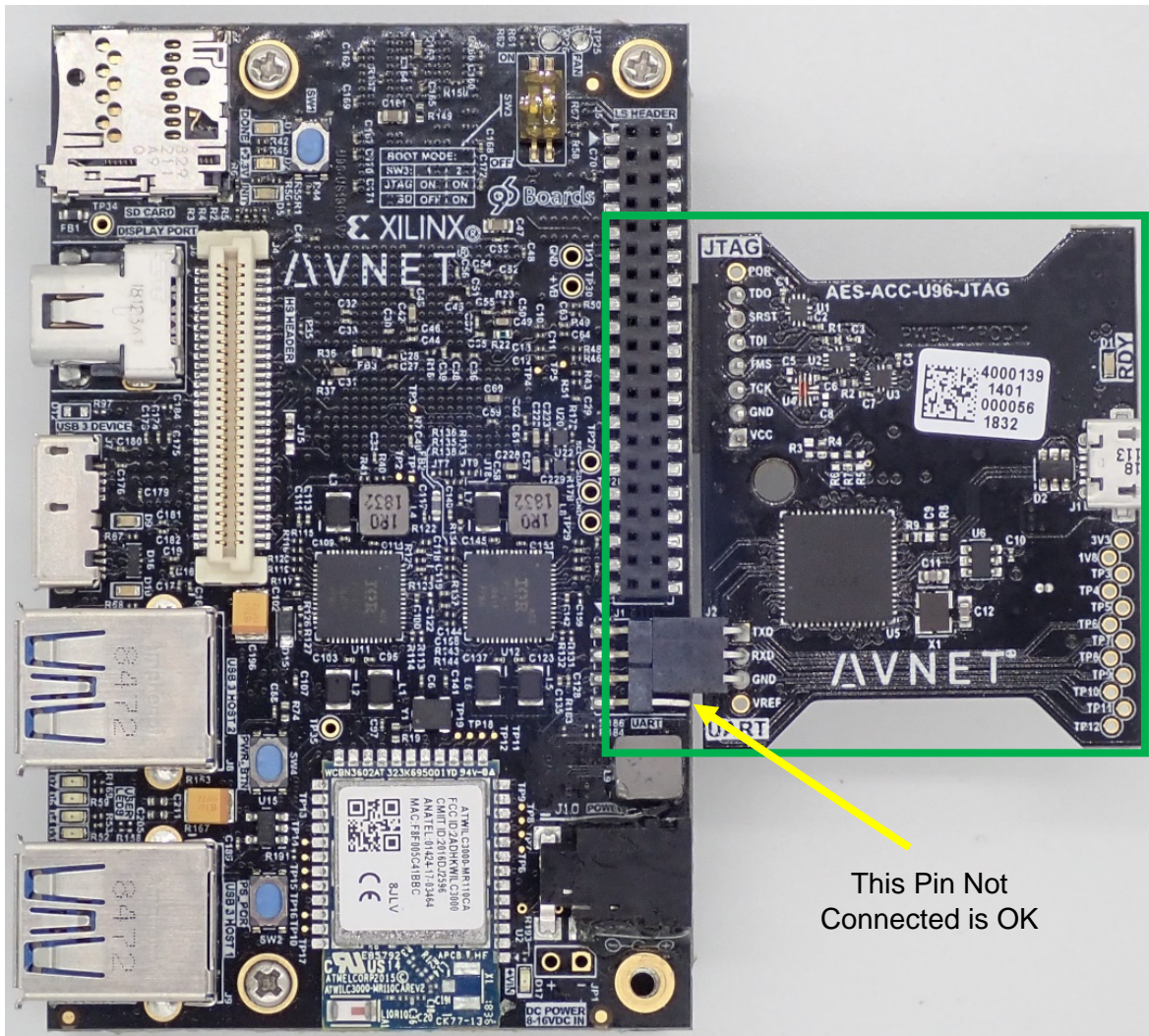


Figure 5 – Ultra96-V2 with older 3-pin JTAG/UART Pod

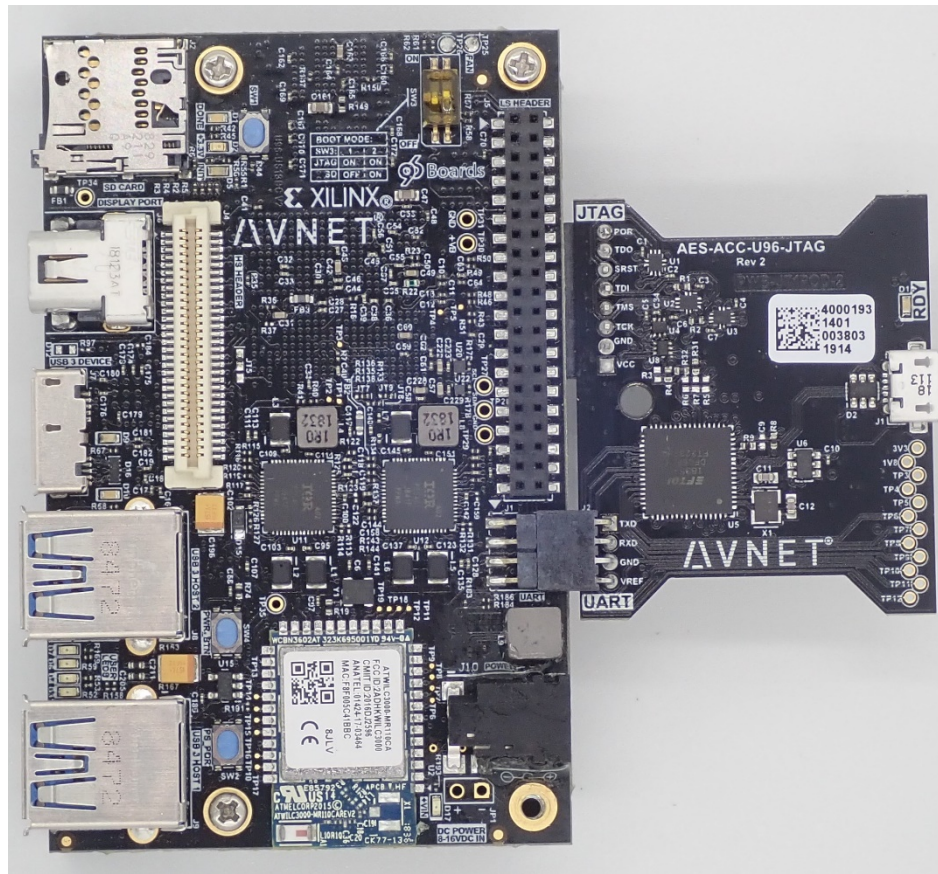


Figure 6 – Ultra96-V2 with newer 4-pin JTAG/UART Pod

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 FT232H 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).

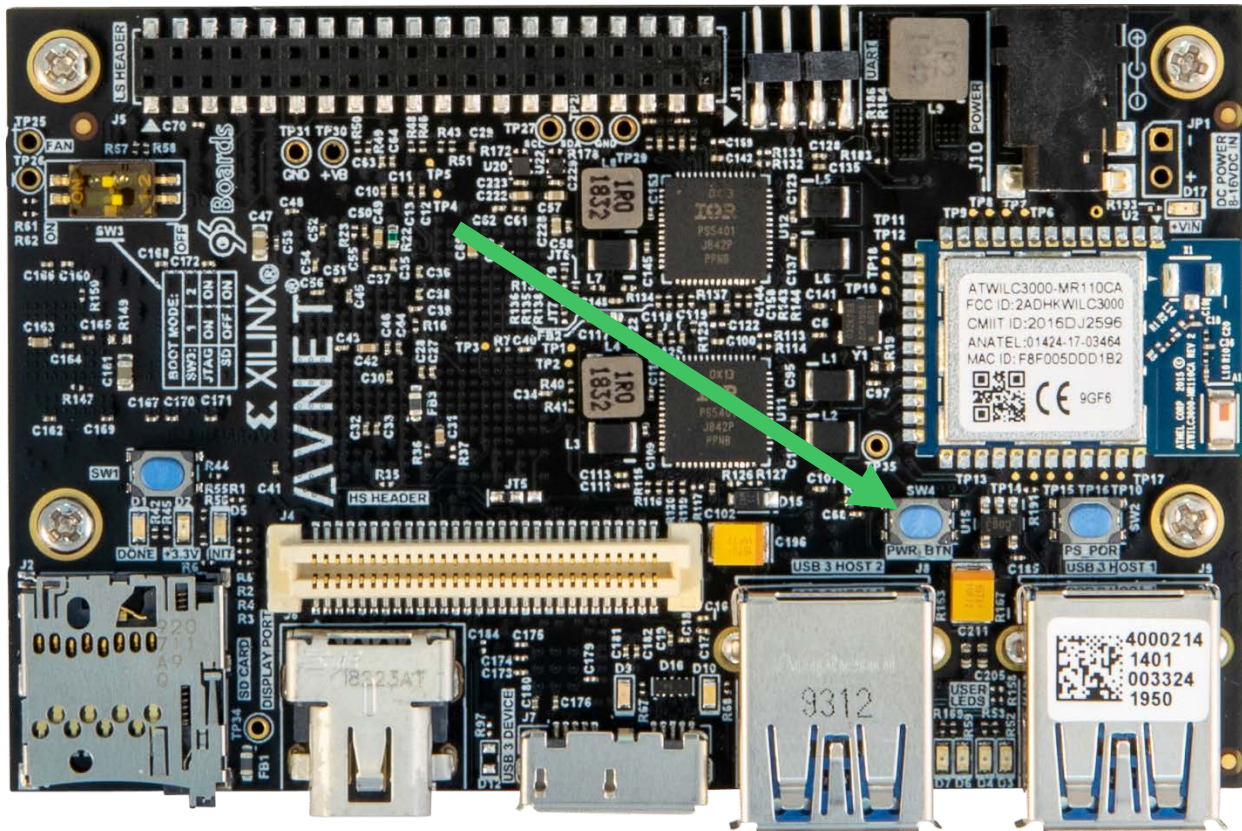


Figure 7 – Press the SW4 Power On Button

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.

- 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.

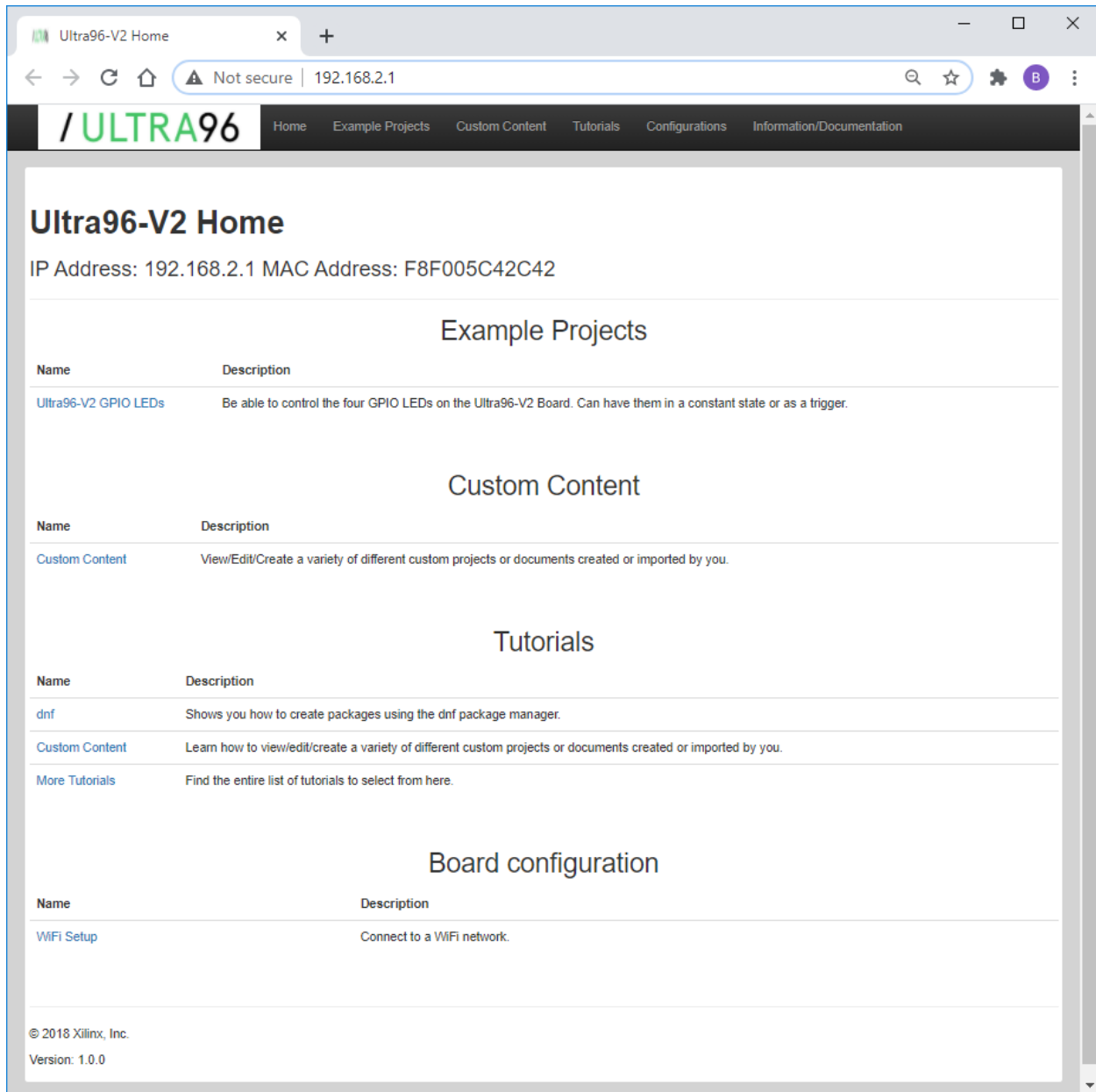


Figure 9 – Connected to Ultra96-V2 Webserver

9 Ultra96-V2 GPIO LEDs Example Project

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

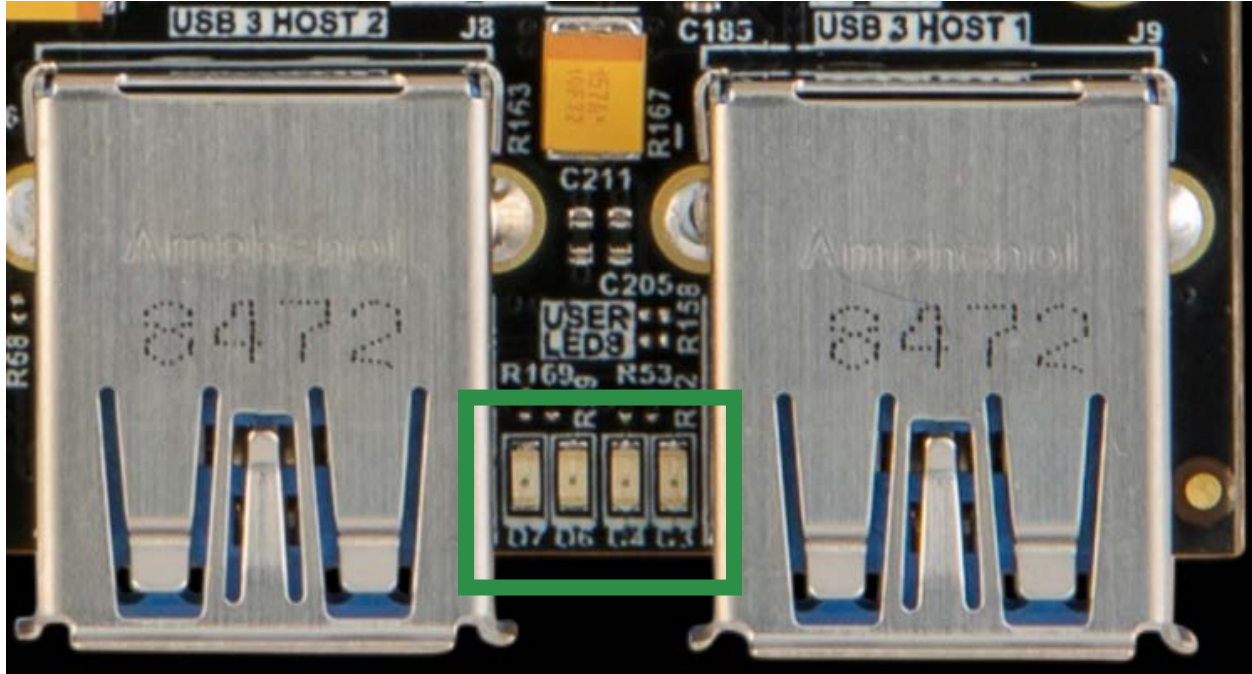


Figure 10 – Ultra96-V2 User LEDs

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

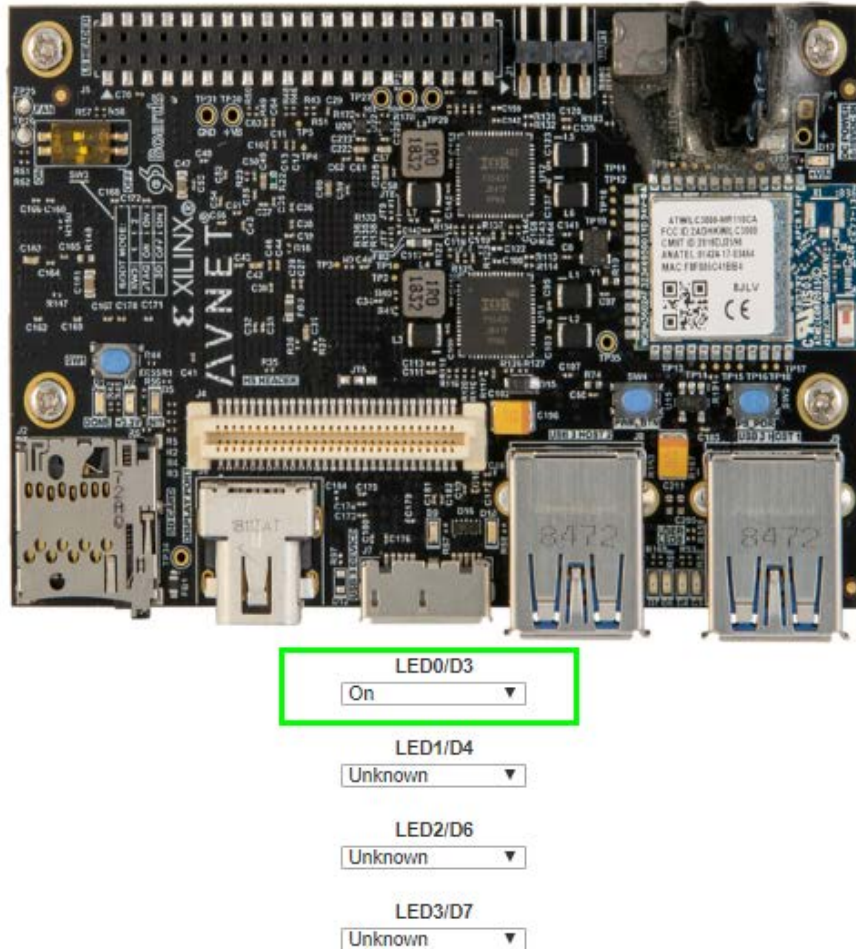
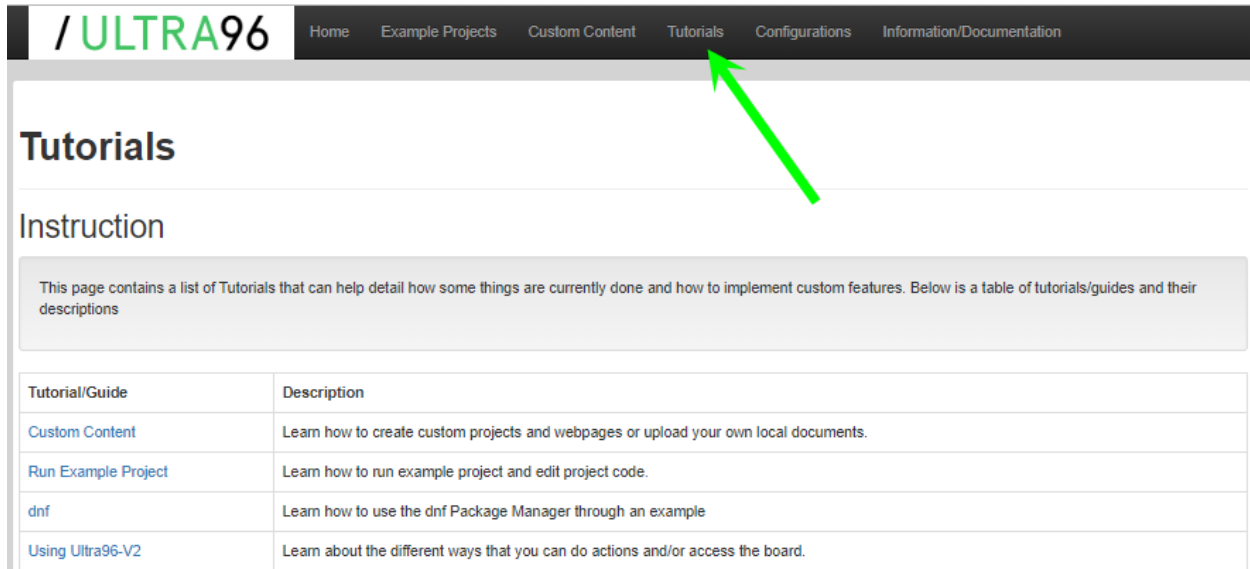


Figure 12 – LED0/D3 Changed to Steady-state ON

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.



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.

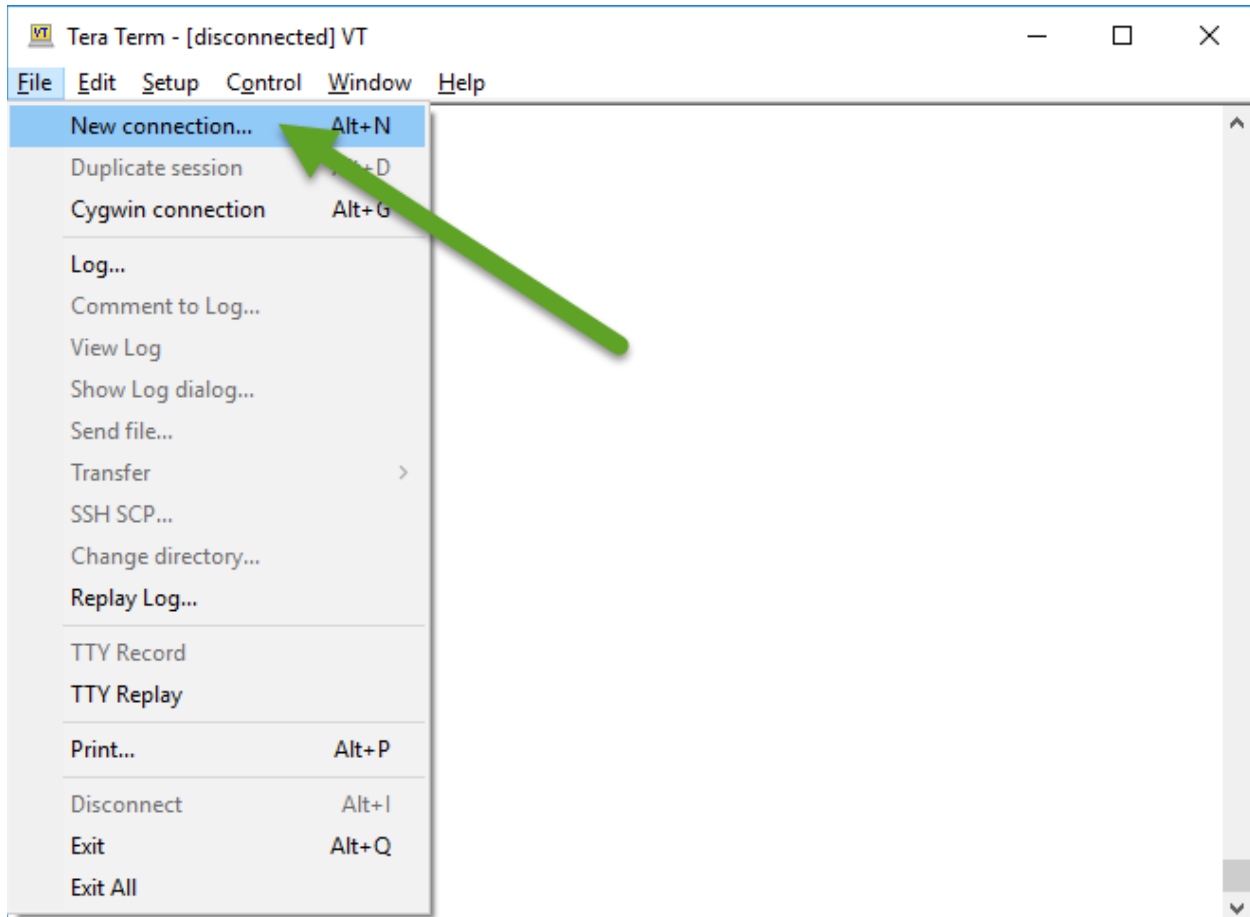


Figure 14 – TeraTerm New Connection

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.

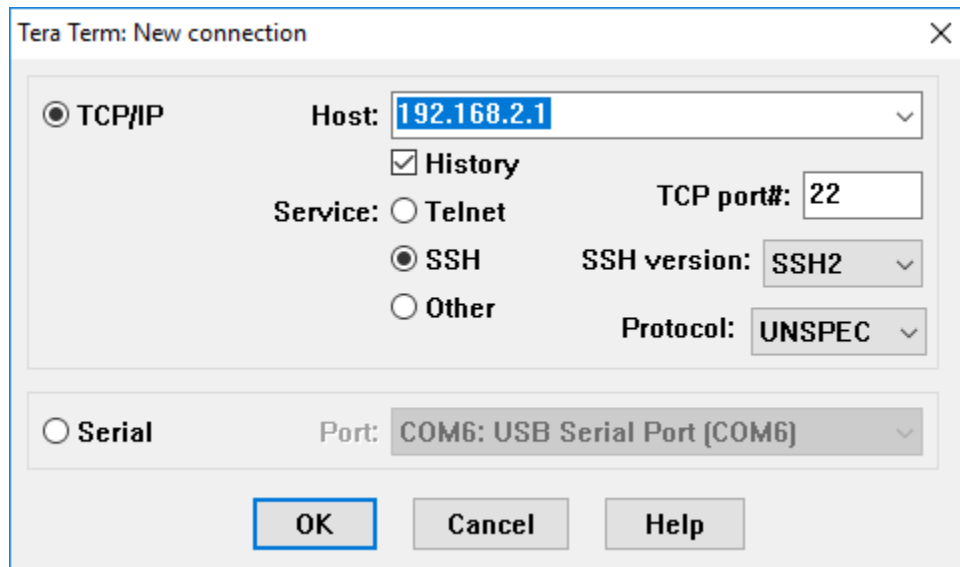


Figure 15 – SSH Terminal Settings

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

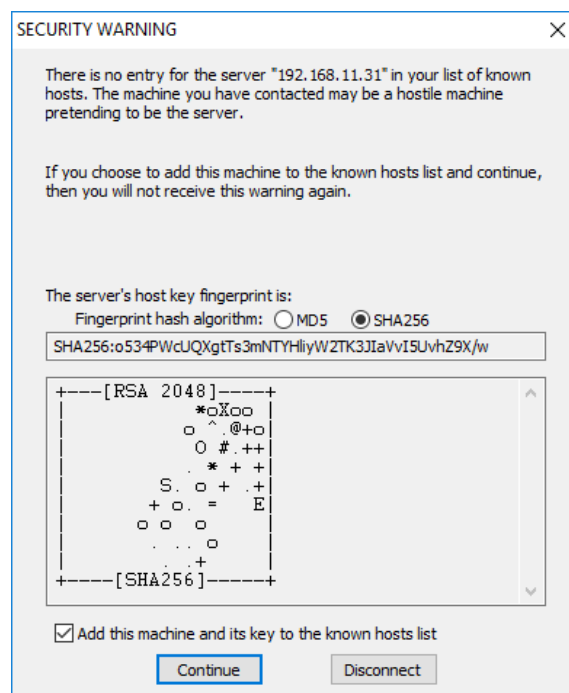


Figure 16 – Click Continue

- 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**.

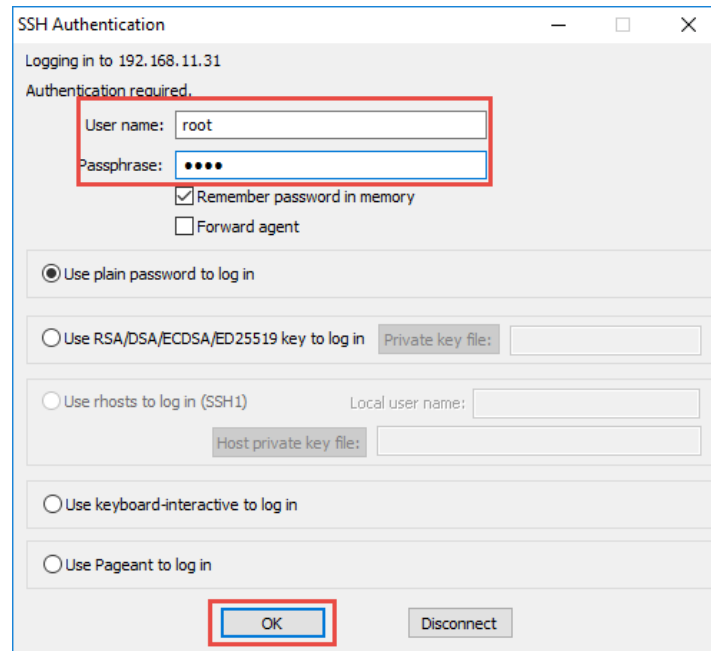


Figure 17 – SSH Authentication

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

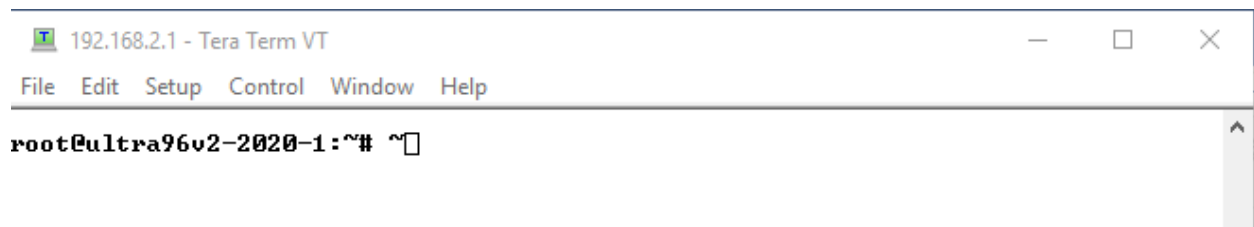
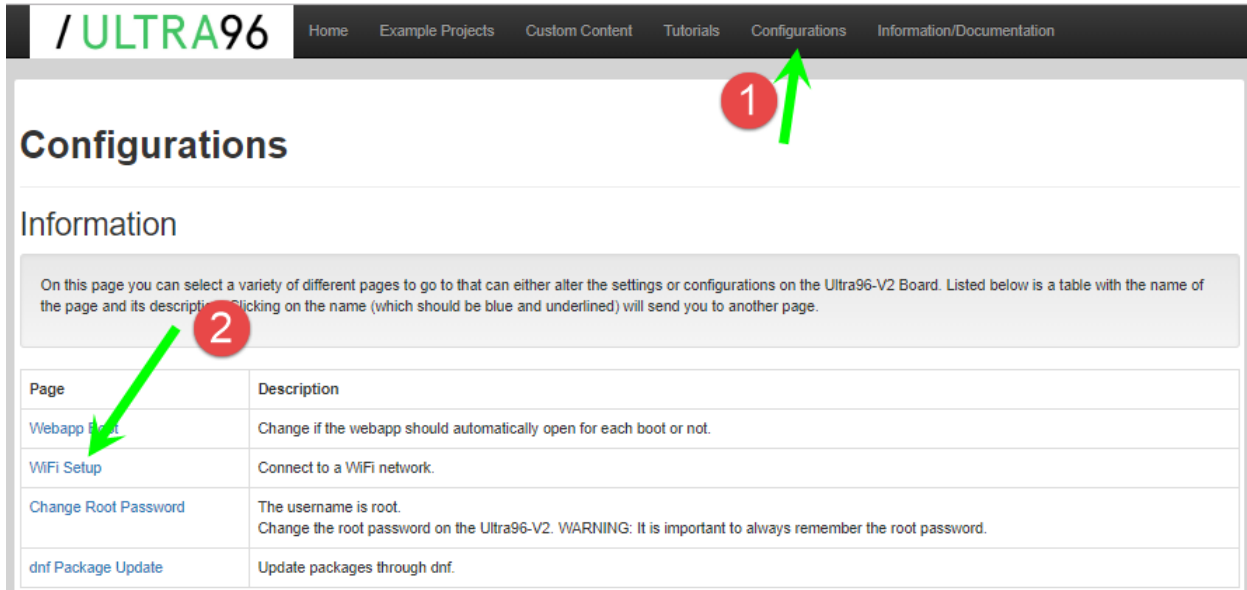


Figure 18 – Ultra96-V2 Terminal

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**.

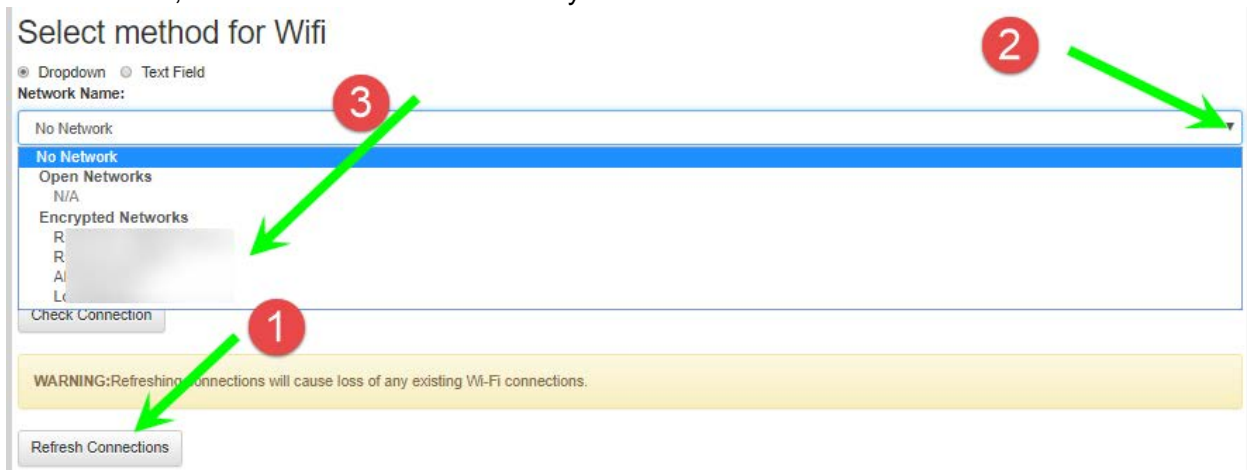


The screenshot shows the ULTRA96 web interface. The top navigation bar includes links for Home, Example Projects, Custom Content, Tutorials, Configurations, and Information/Documentation. The 'Configurations' link is highlighted with a red circle and a green arrow labeled '1'. Below the navigation bar, the 'Configurations' page title is displayed. Underneath, there is an 'Information' section with a paragraph explaining the page's purpose. A table lists various configuration options, with 'WiFi Setup' highlighted by a red circle and a green arrow labeled '2'.

Page	Description
Webapp Auto	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.



The screenshot shows the 'Select method for Wifi' interface. At the top, there are radio buttons for 'Dropdown' (selected) and 'Text Field'. Below this is a 'Network Name:' label and a dropdown menu. The dropdown menu is open, showing 'No Network' selected, with other options like 'Open Networks' and 'Encrypted Networks' visible. A red circle and green arrow labeled '3' point to the dropdown menu. Below the dropdown is a 'Check Connection' button. A yellow warning box contains the text: 'WARNING: Refreshing connections will cause loss of any existing Wi-Fi connections.' Below the warning box is a 'Refresh Connections' button, which is highlighted with a red circle and a green arrow labeled '1'. A red circle and green arrow labeled '2' point to the 'Refresh Connections' button.

Figure 20 – Discovering Networks

3. If a passphrase is required, **first** select the radio button for Text Field. 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:

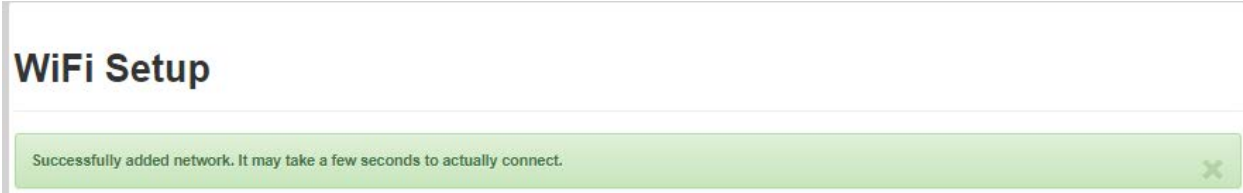


Figure 21 – Successfully Connected

At this point, the Access Point may no longer work. To connect back to the board, you must now connect to the IP address 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
            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::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.

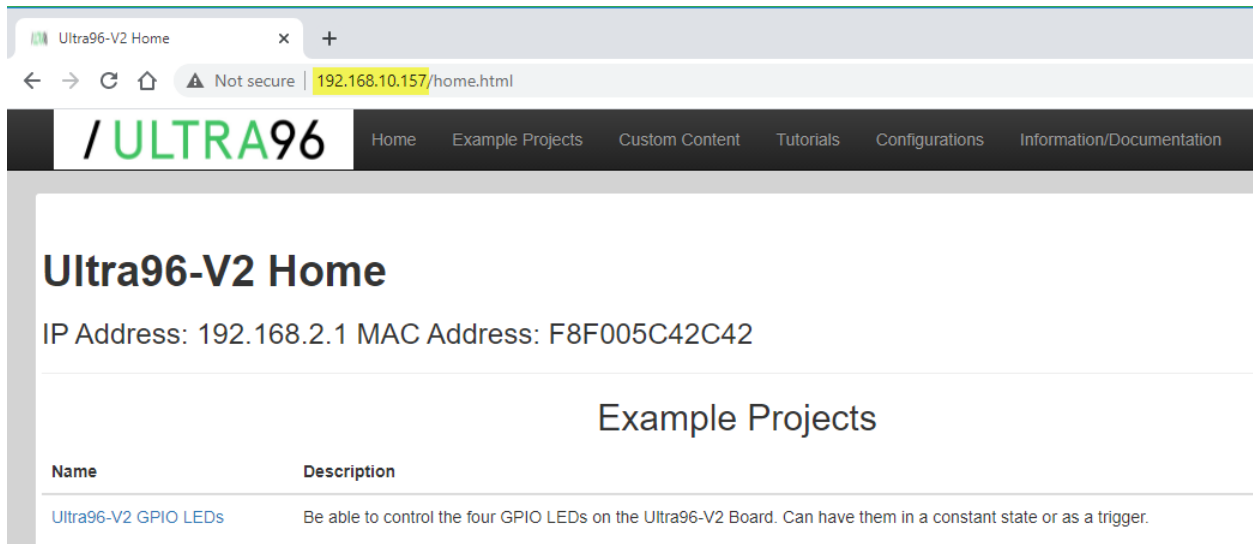


Figure 23 – Connected to Ultra96-V2 Via External Wi-Fi

13 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//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 24 – 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.

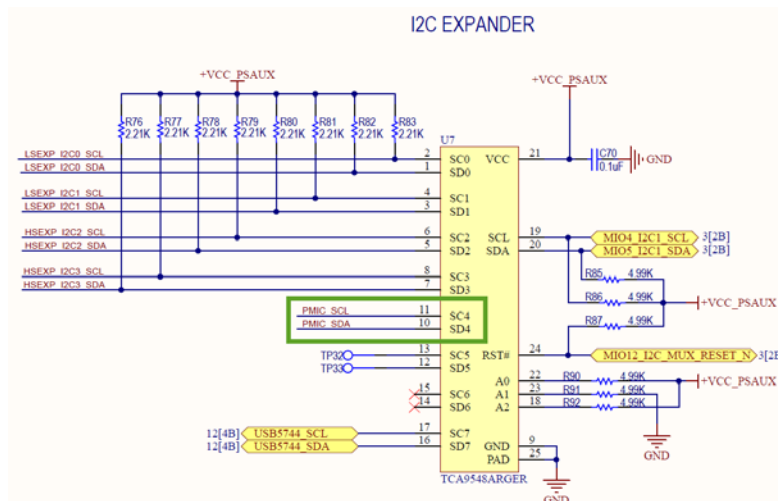


Figure 25 – PMIC on I2C Mux Channel 4

- 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 26 – PMICs on I2C-6

- 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 U (min = +0.50 U, crit max = +24.00 U)
vout1:        5.05 U (crit min = +4.00 U, min = +4.25 U)
              (max = +5.75 U, crit max = +6.00 U)
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 27 – Telemetry from PMIC 0x15/45

```

irps5401-i2c-6-43
Adapter: i2c-0-mux (chan_id 4)
vin1:      5.03 U (min = +4.12 U, crit max = +5.50 U)
vin2:      4.97 U (min = +4.12 U, crit max = +5.50 U)
vin3:      5.06 U (min = +4.12 U, crit max = +5.50 U)
vin4:      5.00 U (min = +4.12 U, crit max = +5.50 U)
vin5:      4.95 U (min = +0.00 U, crit max = +6.00 U)
vout1:     1.79 U (crit min = +1.65 U, min = +1.70 U)
           (max = +1.90 U, crit max = +1.95 U)
vout2:     1.20 U (crit min = +1.10 U, min = +1.15 U)
           (max = +1.25 U, crit max = +1.30 U)
vout3:     1.10 U (crit min = +1.00 U, min = +1.05 U)
           (max = +1.15 U, crit max = +1.20 U)
vout4:     847.00 mU (crit min = +0.65 U, min = +0.74 U)
           (max = +0.99 U, crit max = +1.00 U)
vout5:     3.30 U (crit min = +2.48 U, min = +2.90 U)
           (max = +3.73 U, crit max = +4.14 U)
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 28 – Telemetry from PMIC 0x13/43

```

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

```

Figure 29 – Telemetry from MPSoC Sysmon

```

irps5401-i2c-6-44
Adapter: i2c-0-mux (chan_id 4)
vin1:      5.06 U (min = +4.12 U, crit max = +5.50 U)
vin2:      5.06 U (min = +4.12 U, crit max = +5.50 U)
vin3:      5.09 U (min = +4.12 U, crit max = +5.50 U)
vin4:      5.03 U (min = +4.12 U, crit max = +5.50 U)
vin5:      4.96 U (min = +4.50 U, crit max = +5.50 U)
vout1:     1.80 U (crit min = +1.65 U, min = +1.70 U)
            (max = +1.90 U, crit max = +1.95 U)
vout2:     847.00 mU (crit min = +0.75 U, min = +0.80 U)
            (max = +0.90 U, crit max = +0.95 U)
vout3:     3.30 U (crit min = +3.10 U, min = +3.15 U)
            (max = +3.45 U, crit max = +3.50 U)
vout4:     851.00 mU (crit min = +0.75 U, min = +0.80 U)
            (max = +0.90 U, crit max = +0.95 U)
vout5:     1.20 U (crit min = +0.90 U, min = +1.05 U)
            (max = +1.35 U, crit max = +1.50 U)
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 30 – Telemetry from PMIC 0x14/44

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

14 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.

14.1 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 31 – PMIC Config Files

The naming convention for the files is as follows.

<board>_<device>_<channel>_<datecode in YYYYMMDD>

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

6. 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.

7. 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 32 – PMICs Detected On I2C Bus 6

8. 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

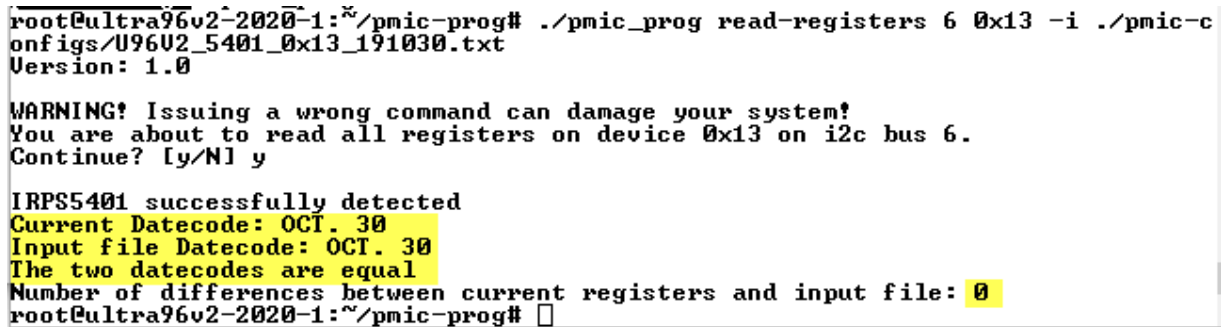
WARNING! Issuing a wrong command can damage your system!
You are about to read byte at address 0x002B on device 0x13 on i2c bus 6.
Continue? [y/N] y

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

Figure 33 – Example datecode October 30 for Device 0x13

- 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
```



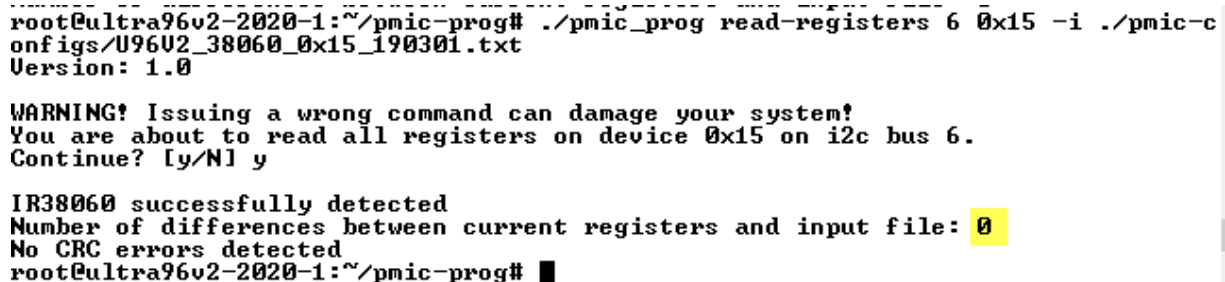
```
root@ultra96v2-2020-1:~/pmic-prog# ./pmic_prog read-registers 6 0x13 -i ./pmic-configs/U96V2_5401_0x13_191030.txt
Version: 1.0

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 34 – 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-configs/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 35 – Compare PMIC Config to a File, 0x15

14.2 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 36 – 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 37 – 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).

15 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 38 – 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.

16 Getting Help and Support

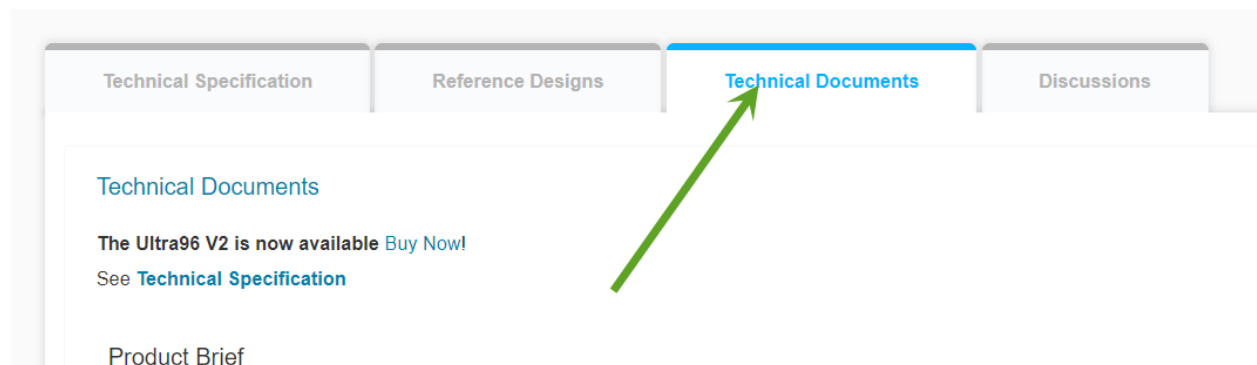
16.1 Avnet Support

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.

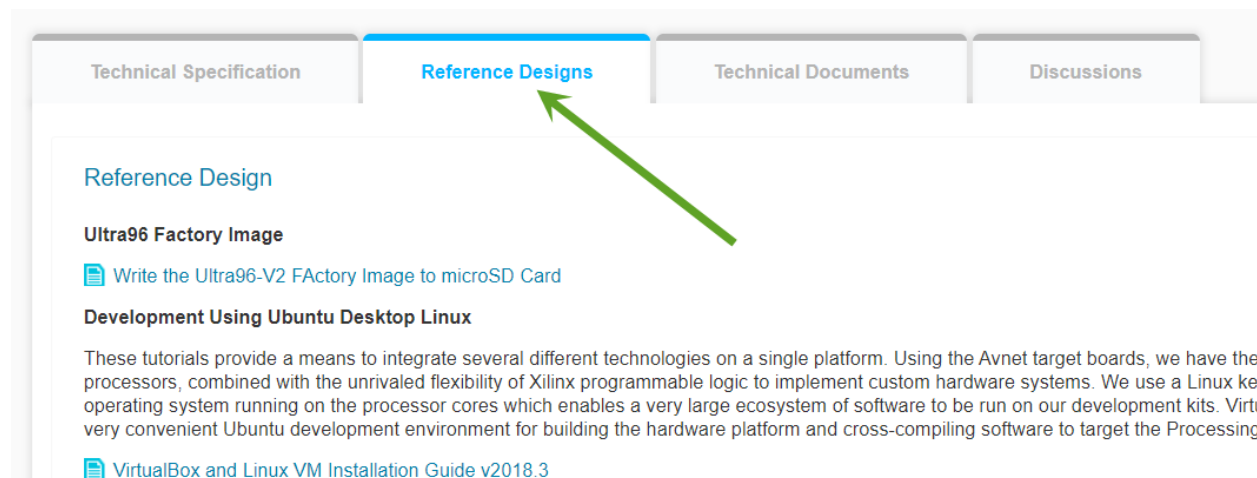
To access Ultra96-V2 collateral, please visit the community support page at:

<http://avnet.me/ultra96-v2>

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



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



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](http://avnet.me/TTC_on_Demand).

16.2 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

17 Installing and Licensing Xilinx Software

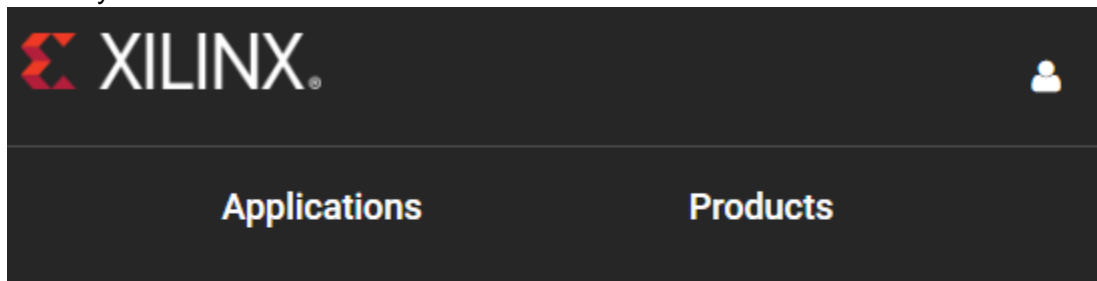
17.1 Install Vivado Design Suite, Design Edition

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



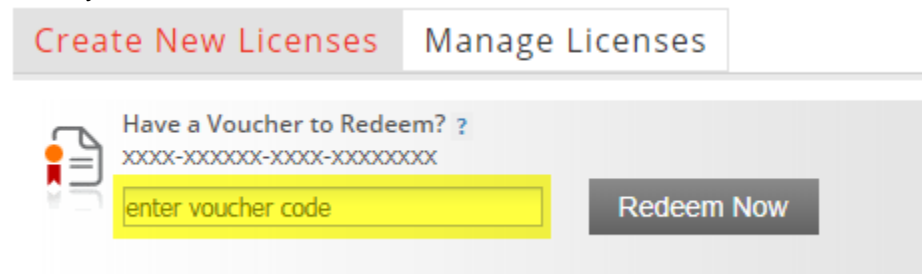
[Home](#) : [Support](#) : [Product Licensing](#)

Product Licensing

Account:



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



- At the confirmation screen, click Yes.

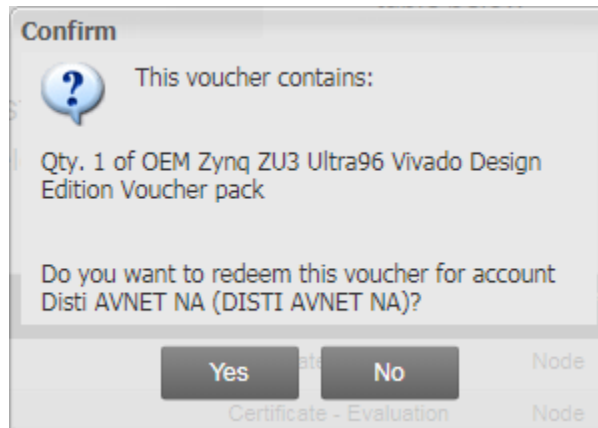


Figure 39 – Voucher Confirmation

- 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
<input type="checkbox"/> OEM Zynq ZU3 Ultra96 Vivado Design Edition Voucher pack	Certificate - Full	Node	1/1	Current	23 May 2019

Figure 40 – 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
<input checked="" type="checkbox"/>	OEM Zynq ZU3 Ultra...	Full	1/1	23 MAY 2019	1	

2 SYSTEM INFORMATION

License	Node
Host ID * ?	<input type="text" value="JSHOM00NBE24974 - Windows 64-bit - Disk - 4003dac8"/>

3 COMMENTS

Comments ?		
------------	--	--

[Cancel](#)

Figure 41 – 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)**.

18 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.

19 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.

20 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>