
Ultra96-PYNQ Documentation

Release v2.4

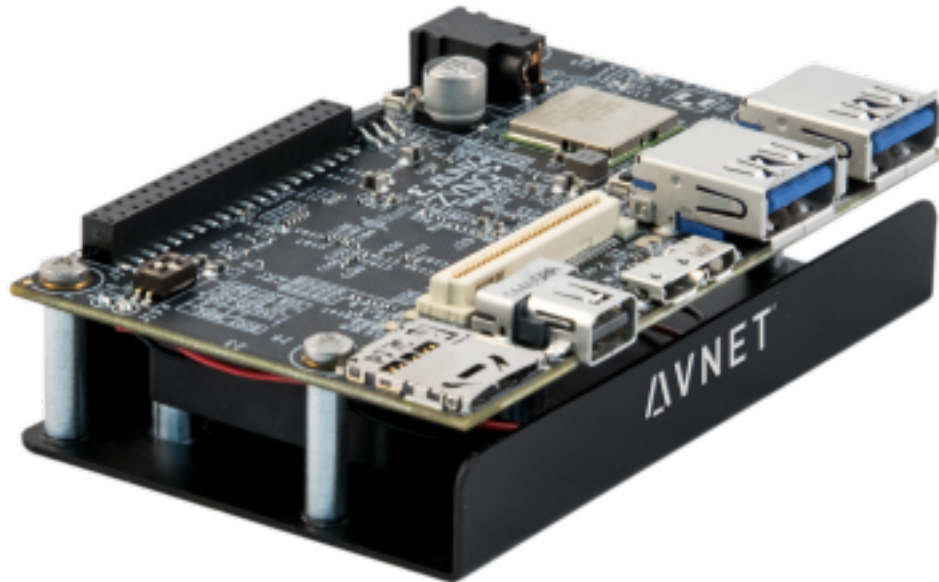
Avnet, Xilinx

May 22, 2020

Contents

1	Getting Started	3
1.1	Getting Started Video	3
1.2	Ultra96 Setup Guide	3
1.3	Connecting to Jupyter Notebooks	5
1.4	Configuring PYNQ	8
2	Sensors96b Overlay	11
2.1	Sensors96 Mezzanine board	11
2.2	Ultra96 overlay diagram	12
2.3	96boards sensors kit	13
2.4	Rebuilding the Overlay	13
3	For More Information	15
3.1	PYNQ	15
3.2	Ultra96	15

This documentation is for the release of PYNQ targeting the Avnet Ultra96 Board. Specifically, a getting started guide is included here and information on the included boot overlay.



For more information on PYNQ, please see the PYNQ readthedocs pages: <https://pynq.readthedocs.io>

This guide will show you how to setup your computer and Ultra96 board to get started using PYNQ.

For information on the Ultra96 board, see the Avnet page: <http://zedboard.org/product/ultra96>

1.1 Getting Started Video

You can watch the getting started video guide, or follow the instructions below.

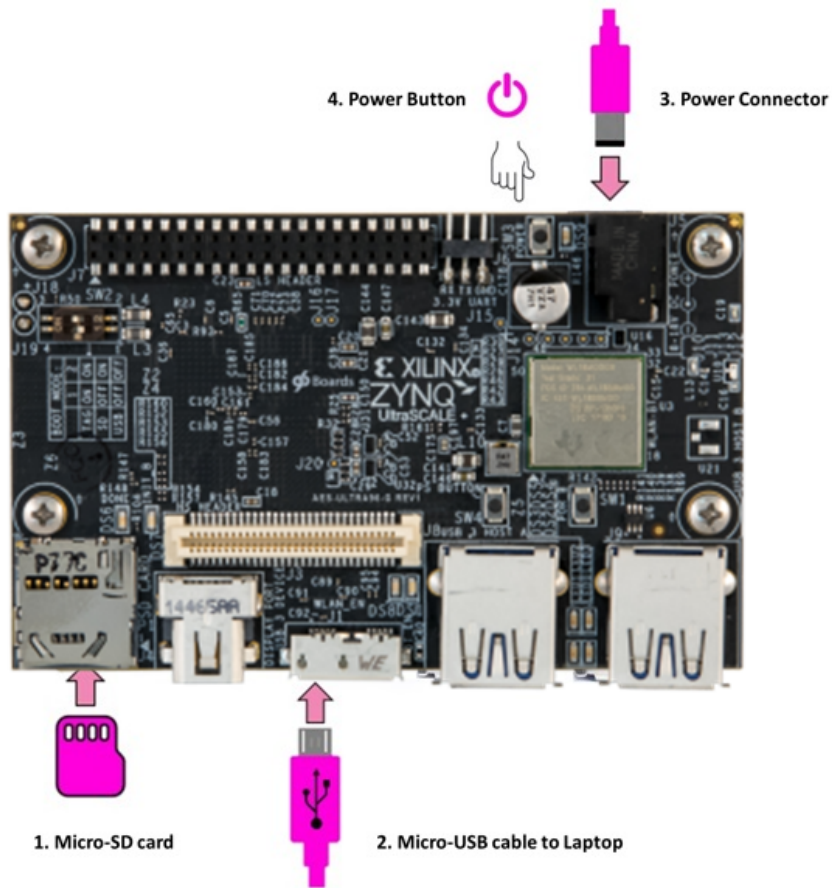
1.2 Ultra96 Setup Guide

The PYNQ environment can be accessed using a web browser over a network connection. You can connect to the board in three main ways:

- USB cable (preferred)
- Use the board's WiFi access point
- Connect a monitor, keyboard and mouse directly to the board.

All three connection methods will be covered below. USB cable is the recommended way to start using PYNQ on the Ultra96.

1.2.1 USB cable board setup



1. Insert the Micro-SD card
2. Connect the Micro USB cable from your PC to the board
3. Connect the power supply to the board
4. Press the SW3 power button to power on the board

1.2.2 Turning on the Ultra96

The green power LED (beside the power switch), and the red Init LED (near the SD Card socket) will turn on when the power button is pressed. After a few seconds, the red LED will switch off, and the green Done LED will switch on. The board will be ready after the blue and orange LEDs switch on.

1.2.3 Connecting to the Ultra96

USB cable

When connected to a computer using the USB cable, the board can act as an Ethernet device (“Ethernet Gadget”). This is the recommended way to connect to the board from your computer, and allows you use a browser on your host computer.

When the USB cable is connected, an Ethernet device should be detected automatically on the host computer (Windows/Linux). A driver should already be available in Windows 10 and Linux. In Windows 7 a driver should start to install automatically when the board is connected.

Skip to *Connecting to Jupyter Notebooks* or keep reading for other connection options.

WiFi

The Ultra96 includes a WiFi chip that can be used to connect to other WiFi networks as a client, or act as a WiFi access point (hotspot).

WiFi access Point

On boot, the board will enable itself as a WiFi access point.

The Access point name will be **pynq_<MAC Address>** where **<MAC address>** is the MAC address for your board.

To connect to the board's WiFi network (access point), open your wireless network configuration, search for and connect to an SSID starting with **pynq_** followed by the MAC address of your board.

Skip to *Connecting to Jupyter Notebooks* or keep reading for other connection options.

Other WiFi networks

You can connect the Ultra96 to other WiFi networks (E.g. your home WiFi network). If you can connect to the board using the USB cable, then you can use the WiFi notebook in the Jupyter home area to configure the board to connect to a WiFi network:

```
<Jupyter Home>/common/wifi.ipynb
```

If you are familiar with configuring network connections in Linux, you can also configure your connection using a terminal.

Skip to *Connecting to Jupyter Notebooks* or keep reading for other connection options.

DisplayPort monitor

You can also use the board like a mini-PC by connecting a keyboard and mouse, and a HDMI or DisplayPort monitor to the mini DisplayPort on the board. The Chromium browser is installed on the board and can be used with PYNQ.

Note that you can try PYNQ examples in this mode, but for optimal performance it is recommended to use a browser from a computer connected to the board.

1.3 Connecting to Jupyter Notebooks

To connect to Jupyter Notebook, on the host computer, open a web browser (Chrome is preferred) and browse to one of the following addresses depending on how you setup your connection to the board in the instructions above:

- USB: <http://192.168.3.1>
- WiFi access point: <http://192.168.2.1>
- Other WiFi: Browse to the IP address of the board <http://<IP address of board>> You can find the IP from a terminal, from your router, or from a network scanning app.

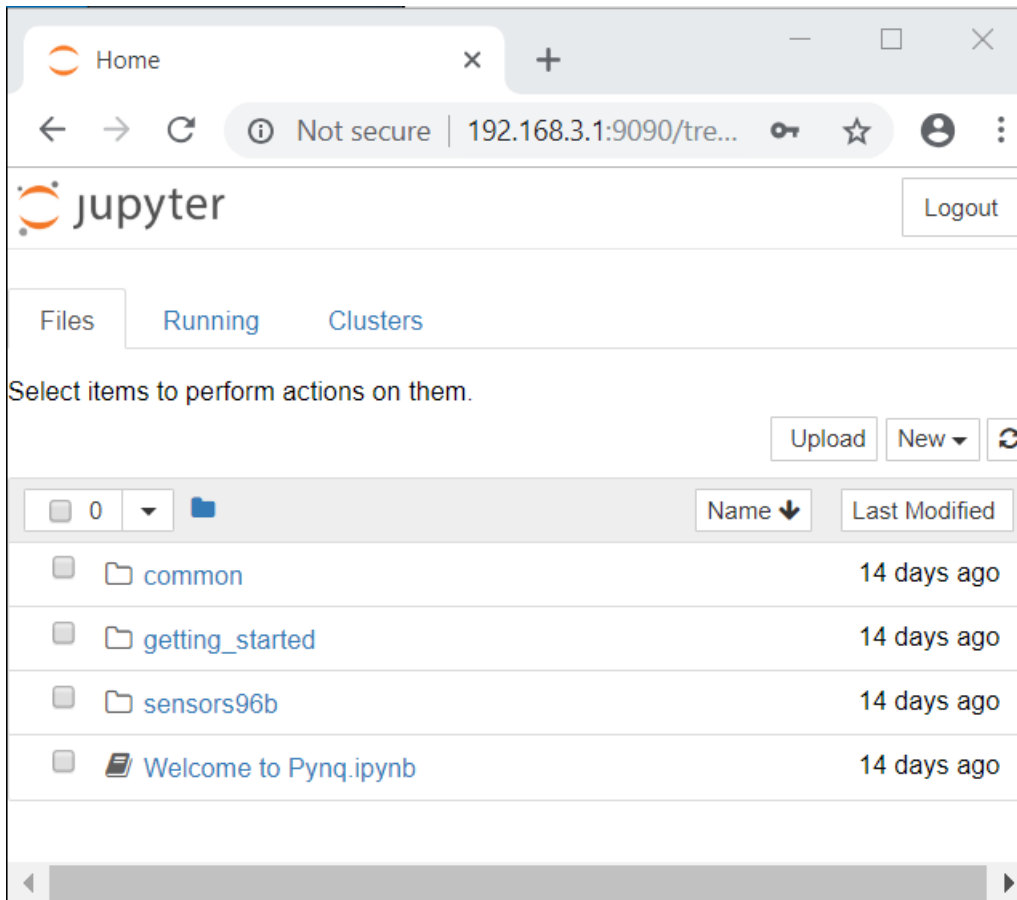
- DisplayPort monitor: from the Chromium browser, browse to <http://192.168.2.1>

It may take a few seconds for your computer to resolve the IP address and connect to the board the first time it is powered on.

Note that the board can have multiple connections to different networks at the same time. E.g. USB and *securbrd* WiFi. The board will have a different IP for each connection, and you can choose to use any of the available connections.

If your board is configured correctly you will be presented with a login screen. Enter the password as **xilinx**

After logging in, you should see the following screen:

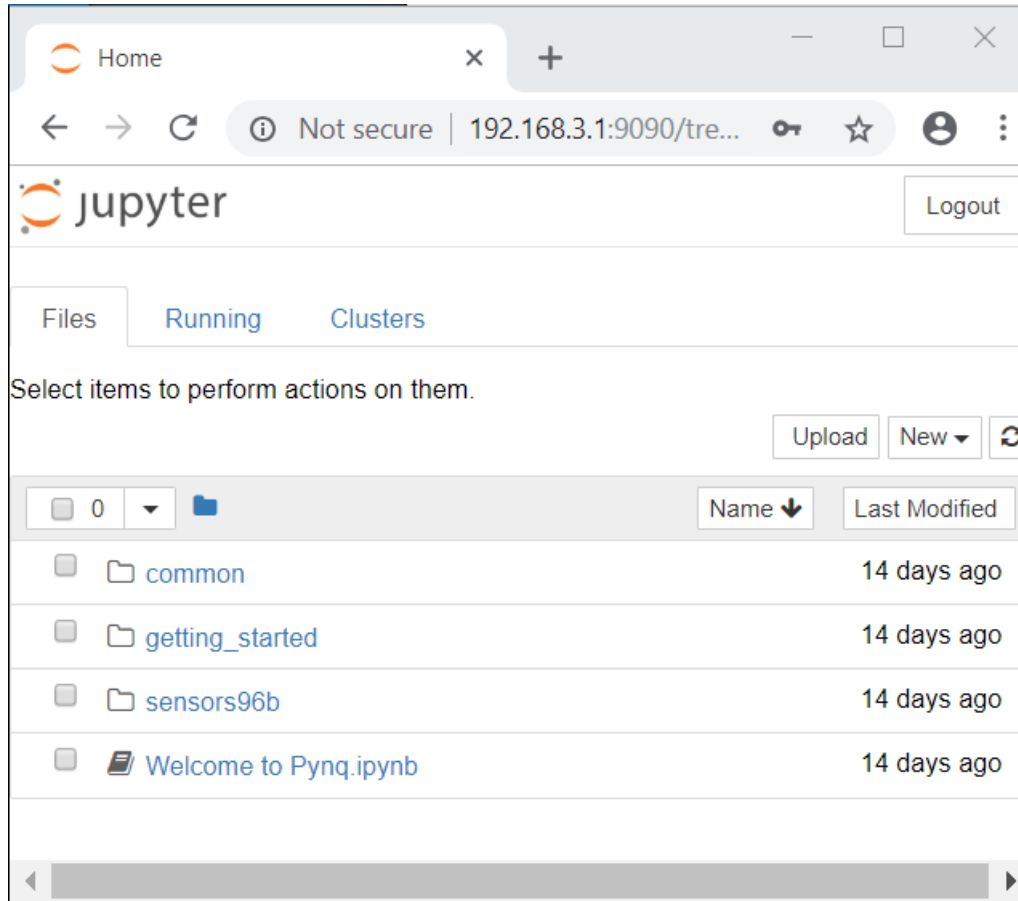


1.3.1 Example Notebooks

PYNQ uses the Jupyter Notebook environment to provide examples and documentation. Using your browser you can view and run the notebook documentation interactively.

To get started using PYNQ, try running the provided example notebooks:

- **getting_started**: includes an introduction to using Jupyter notebook with PYNQ, the Python environment, and how to use some basic features of the current platform.
- **common**: contains example notebooks on how to download an overlay, how to set the Zynq clocks, how to execute Linux shell commands, and how to use USB devices.
- **sensors96b**: includes example notebooks to use peripherals with the mezzanine board.



When you open a notebook and make any changes, or execute cells, the notebook document will be modified. It is recommended that you “Save a copy” when you open a new notebook. If you want to restore the original versions, you can download all the example notebooks from [Avnet’s Ultra96 Github Repository](#).

Serial connection

The board can act as a USB-Ethernet device, a serial port, and a USB mass storage device at the same time over the micro USB cable.

When a USB cable is connected to your computer, the board should be detected as a serial device automatically. Windows 10 and Linux include a serial driver by default. On Windows 7, the device should be detected, but the driver needs to be installed manually. The driver is included on the mass storage device that also appears when the cable is connected. You can install the driver in the *serial_driver* folder.

Note that the USB terminal can be useful for debug, but is not intended to be used to access PYNQ. A browser based terminal is available to run anything directly from a command line.

Connect board to WiFi network

After you have been able to connect to the board using one of the other connection methods, a notebook is available to simplify the process of connecting the Ultra96 to another WiFi network.

```
<Jupyter Home>/common/wifi.ipynb.
```

The notebook includes an example of connecting the board to WiFi hotspots.

Once the board is connected to a WiFi network, you will need to find the IP address of the board before you can connect to it. If your phone is on the same WiFi network as the board and the network allows scanning (E.g. home network), there are network scanner apps that can be used from your phone (e.g. iOS/Android: Fing) that can help identify the IP address of board.

If you know how to configure Linux network connections manually, you can also do this from a terminal.

1.3.2 Accessing files on the board

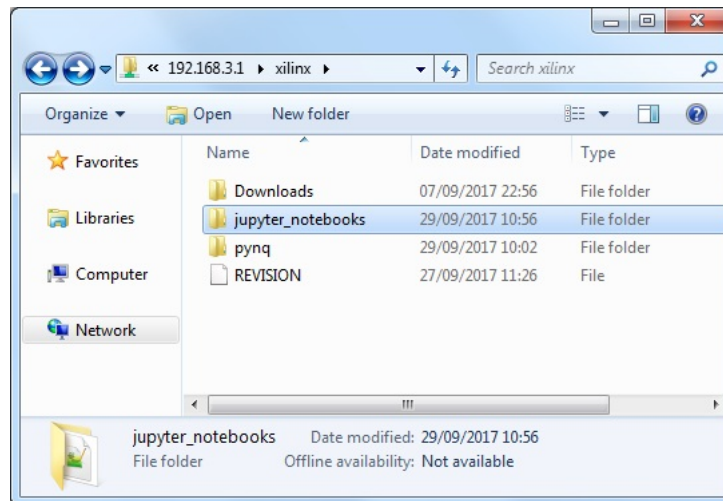
Samba, a file sharing service, is running on the board. This allows you to access the PYNQ home area as a network drive. This can be a convenient way to transfer files, including bitstreams, to and from the board.

Note: In the examples below change the IP address to match your board settings.

To access the PYNQ home area in Windows Explorer type one of the following in the navigation bar.

```
\\192.168.3.1\xilinx # If connected via USB Ethernet
```

If prompted, the username is **xilinx** and the password is **xilinx**. The following screen should appear:



To access the home area in Ubuntu, open a file browser, click Go -> Enter Location and type one of the following in the box:

```
smb://192.168.3.1/xilinx # If connected via USB Ethernet
```

When prompted, the username is **xilinx** and the password is **xilinx**

1.4 Configuring PYNQ

1.4.1 Opening a USB serial terminal

If you can't access the terminal from Jupyter, you can connect the micro-USB cable from your computer to the board and open a terminal. You can use the terminal to check the network connection of the board. You will need to have terminal emulator software installed on your computer. PuTTY is one application that can be used, and is available for free on Windows. To open a terminal, you will need to know the COM port for the board.

On Windows, you can find this in the Windows *Device Manager* in the control panel.

1. Open the Device Manager, expand the *Ports* menu
2. Find the COM port for the *USB Serial Port*. e.g. COM5
3. Open PuTTY

Once PuTTY is open, enter the following settings:

4. Select serial
5. Enter the COM port number
6. Enter the serial terminal settings (below)
7. Click *Open*

Full terminal Settings:

- 115200 baud
- 8 data bits
- 1 stop bit
- No Parity
- No Flow Control

Hit *Enter* in the terminal window to make sure you can see the command prompt:

```
xilinx@pynq: /home/xilinx#
```

You can check the IP address of the board using *ifconfig*:

```
ifconfig
```


96boards is an open set of hardware specifications for ARM based boards to allow manufacturers to provide a common set of features for developers. The Ultra96 follows this specification.

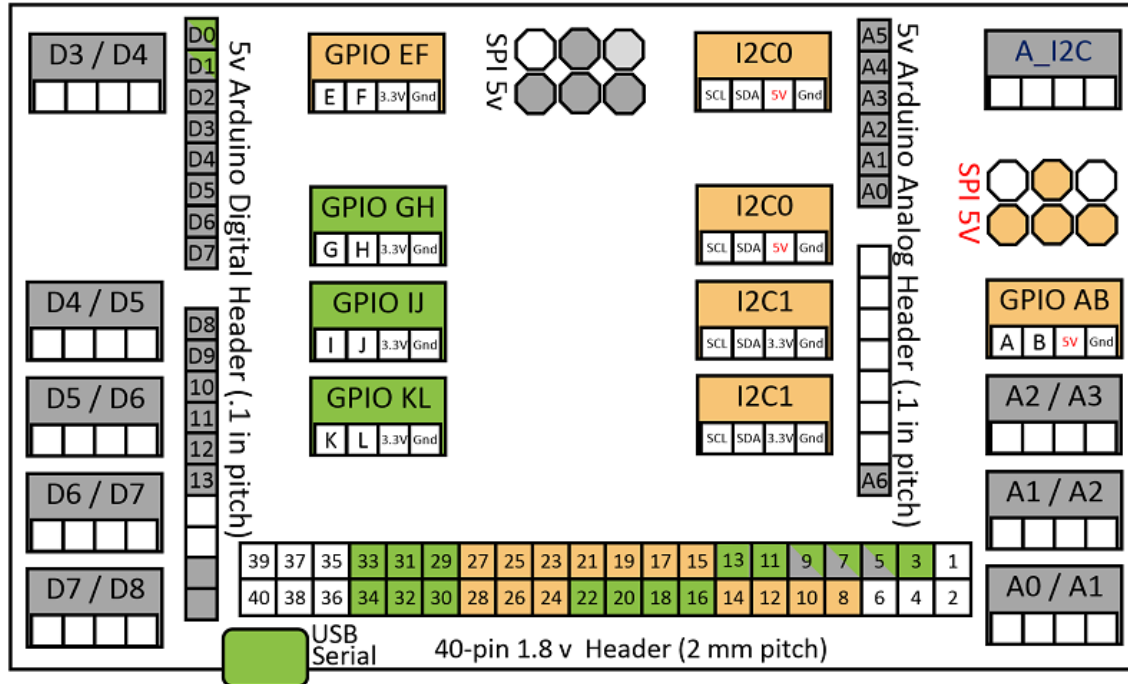
2.1 Sensors96 Mezzanine board

The [Grove starter kit for 96boards](#) is a mezzanine board that provides interfaces that can be used to connect sensors to the 40 pin header on the Ultra96. The board includes an Arduino header, a 40 pin female header, and a number of [Grove 4 pin headers](#).

The mezzanine board has an Arduino compatible microcontroller that is connected to some of the interfaces on the board.

The interfaces on the mezzanine board are connect to the microcontroller, the 40-pin header, or available to both (shared). Of the interfaces connected to the 40-pin header (i.e. available to the Ultra96), some are connected to PS pins, and others are connected to PL pins.

The diagram below shows the interfaces and pin mappings to the PS/PL/Arduino.



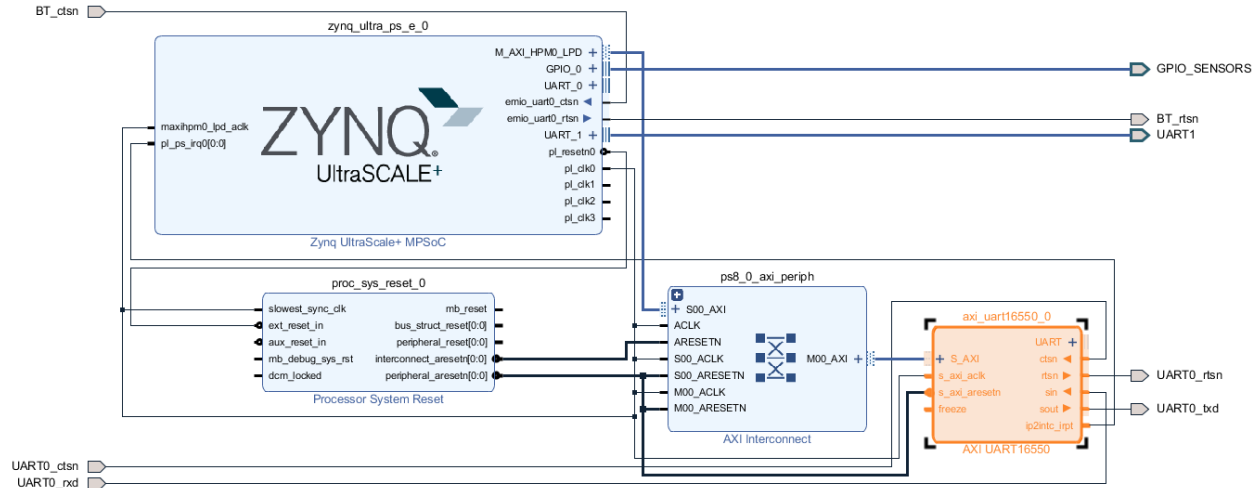
- PL Accessible
- PS Accessible
- Arduino

The following pins are connected to the Ultra96, and available for use with PYNQ:

- Pins 3,5,7-34 on the 40-pin header
- Pins D0, D1 on the Arduino interface (5V)
- I2C0 (5V) and I2C1
- GPIO AB (5V), EF, GH, IJ, KL Grove connectors

2.2 Ultra96 overlay diagram

The sensors96b overlay is an overlay designed to support the [96boards Sensors mezzanine board](#). The overlay is a simple design providing a number of GPIO and a UART to connect to available interfaces on the mezzanine board. 2x PS I2C are connected to I2C0 and I2C1, PS GPIO are connected to GPIO AB and GF, and PS GPIO are connected to GPIO GH, IJ, KL, through the EMIO.



2.3 96boards sensors kit

A number of peripherals and sensors are provided in the 96 boards sensors kit.

For examples of using the boards96 sensors board and peripherals, see the notebooks in the corresponding *sensors96b* directory in the Jupyter home area. Note that some of the peripherals are analog, and can only be used with the ADC on the Arduino (the XADC on the Zynq Ultrascale+ is not connected to the Ultra96 40-pin header).

See the example notebook for details:

```
<Jupyter home>/sensors96b/sensors_mezzanine_examples.ipynb
```

2.4 Rebuilding the Overlay

The project files for the overlay can be found here:

```
https://github.com/Avnet/Ultra96-PYNQ/tree/master/Ultra96/sensors96b
```

2.4.1 Linux

To rebuild the overlay, source the Xilinx tools first. Then assuming Ultra96-PYNQ has been cloned:

```
cd <Ultra96-PYNQ>/Ultra96/sensors96b
make
```

2.4.2 Windows

To rebuild from the Vivado GUI, open Vivado. In the Vivado Tcl command line window, change to the correct directory, and source the Tcl files as indicated below.

Assuming Ultra96-PYNQ has been cloned:

```
cd <Ultra96-PYNQ>/Ultra96/sensors96b
source ./sensors96b.tcl
```

To build from the command line, open the Vivado 2018.3 Tcl Shell, and run the following:

```
cd <Ultra96-PYNQ>/Ultra96/sensors96b
vivado -mode batch -source sensors96b.tcl
```

Note that you must change to the overlay directory, as the .tcl files has relative paths that will break if sourced from a different location.

For More Information

For more information on PYNQ and Ultra96, please see the following links.

3.1 PYNQ

- <http://www.pynq.io>
- <https://pynq.readthedocs.io>
- <https://www.hackster.io/pynq>
- https://groups.google.com/forum/#!forum/pynq_project

3.2 Ultra96

- <http://zedboard.org/product/ultra96>
- <https://www.96boards.org/product/ultra96>
- <https://www.xilinx.com/products/boards-and-kits/1-vad4rl.html>