Introduction

This User Guide (UG) for the OSD3358-SM-RED platform is intended to serve as a resource to find information regarding design and usage of the hardware and software aspects of the board. It will cover topics ranging from getting started with the platform to using different sensors and peripherals that are present on board.

This document will be updated as required to improve or add information. Please make sure to look for updates and sign up for document change notifications on Octavo Systems website to get up to date info.

Some relevant resources for support and development are given below:

- OSD3358-SM-RED board design files: https://octavosystems.com/files/osd3358-sm-red-eagle-files/
- Sign up for OSD3358-SM-RED latest software updates: https://octavosystems.com/doc-change-signup/
- OSD3358-SM datasheet: https://octavosystems.com/docs/osd335x-sm-datasheet/
- OSD3358-SM RED platform getting started guide: https://octavosystems.com/docs/osd3358-sm-red-quick-start-guide/
- OSD3358-SM Application notes:
    a. OSD335x-SM layout guide: https://octavosystems.com/app_notes/osd335x-sm-layout-guide/
    b. OSD335x Family Pin Assignments Compared to AM335x: https://octavosystems.com/app_notes/osd335x-family-pin-assignments/

Notice: The information provided within this document is for informational use only. Octavo Systems provides no guarantees or warranty to the information contained.

Revision History

Revision	Details	Date	Author
Initial draft		09-19-2017	Neeraj Dantu

Contents

1 Powering the OSD3358-SM-RED

Figure 1 shows the OSD3358-SM-RED and its power inputs. The board can be powered through any of three inputs:

1. Micro-USB cable connected to the micro-USB port
2. 5V AC adapter connected to the barrel jack
3. Single cell Li-Ion or Li-Polymer battery connected to the battery header

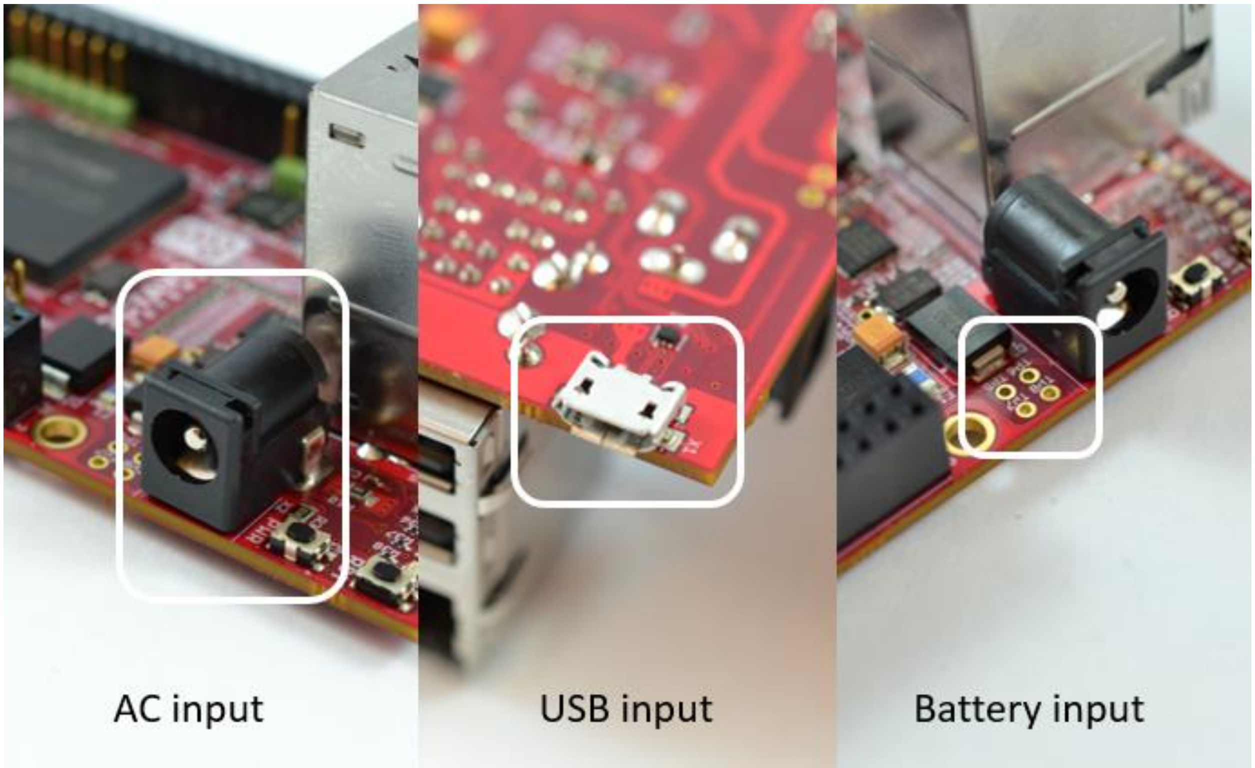

Figure 1: Picture of RED board highlighting power inputs

1.1 Power up procedure

The following steps describe the power up sequence of the board when powered via USB, AC Adapter or battery inputs.

1. Connect your preferred power source:
   a. USB:
      i. Connect the micro-USB side of the USB cable to the micro-USB port X1
      ii. Connect the USB side to one of the USB ports on a laptop/computer or other USB power source.

Figure 2: Powering the board using a USB cable

b. AC adapter:
   i. Plug in the 5V AC adapter with 2.5 mm x 5.5 mm barrel into the power connector X2.

Figure 3: Powering the RED board using an AC-DC adapter

c. Battery input terminals:
   i. Connect the battery input terminals to the battery terminals as shown below.

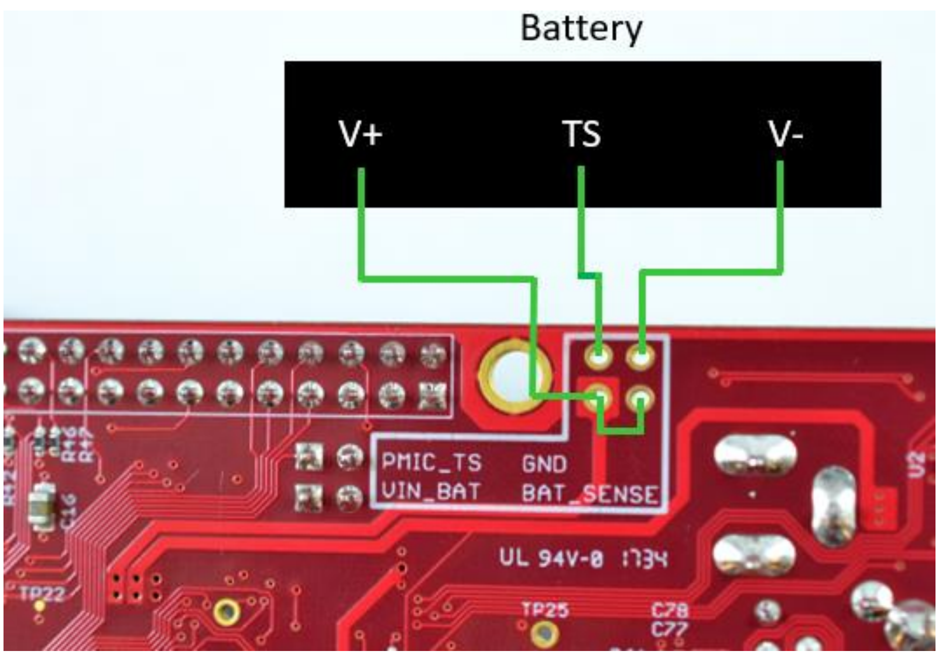

Figure 4: OSD3358-SM-RED battery connections (back-side of board)

NOTE: Notes on connecting the PMIC_TS input of the SiP can be found in "Battery pack temperature monitoring" section of the TPS65217C PMIC datasheet. If the battery does not have a temperature output, the battery temperature monitoring mechanism of the PMIC can be bypassed by connecting a 10KOhm resister between PMIC_TS (TP7) of the board and GND.

2. LED D1 will turn on immediately after power is applied to the board.

Figure 5: RED board boot up - LED D1 ON

3. LEDs D4 – D7 turn on in the sequence indicated below indicating Linux is booting.

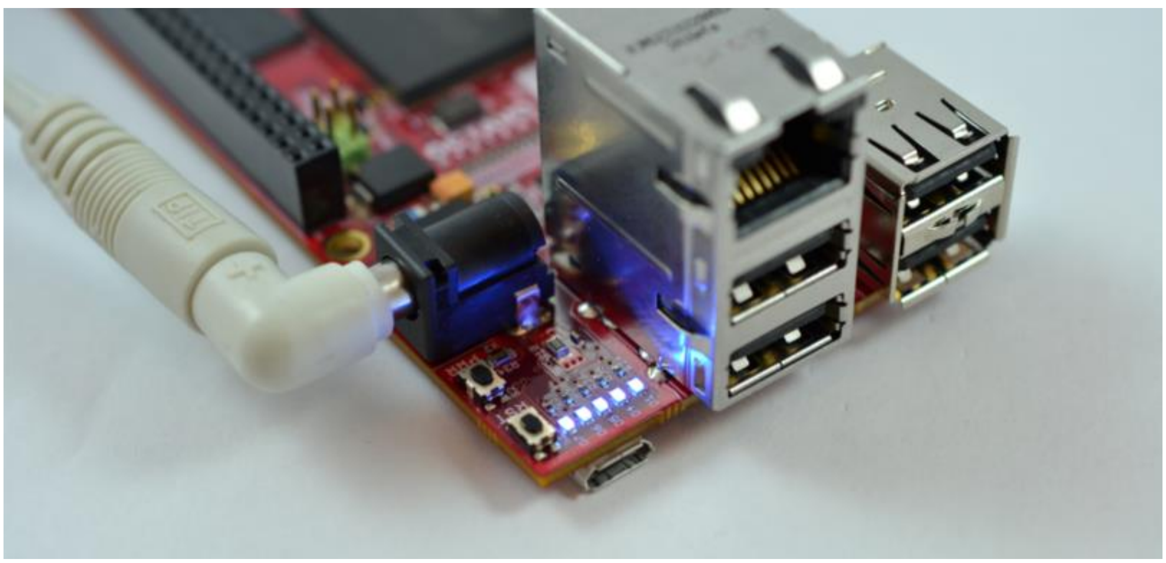

Figure 6: RED board boot up - LEDs D4 - D7 turn ON

2 OSD3358-SM-RED Use Cases

The OSD3358-SM-RED Platform can be used in 3 different modes. A description of each mode and instructions on how to set up each mode is described below.

2.1 Standalone boot up

The board can be used as a standalone platform for software development. To boot it up as a standalone system, the following components will be needed.

1. 5V AC adapter
2. HDMI/DVI-D monitor
3. Micro HDMI to HDMI cable and/or HDMI to DVI-D adapter
4. Wired/wireless USB keyboard and mouse
5. Ethernet cable or WiFi USB adapter connected to a network for internet access

The following steps describe the procedure to setup the board as a standalone software development tool.

1. Connect the micro HDMI connector of the HDMI cable to X6 connector on board.
2. Connect the HDMI connector of the HDMI cable into the monitor. HDMI to DVI-D convertor may be necessary if the monitor only has a DVI-D input.

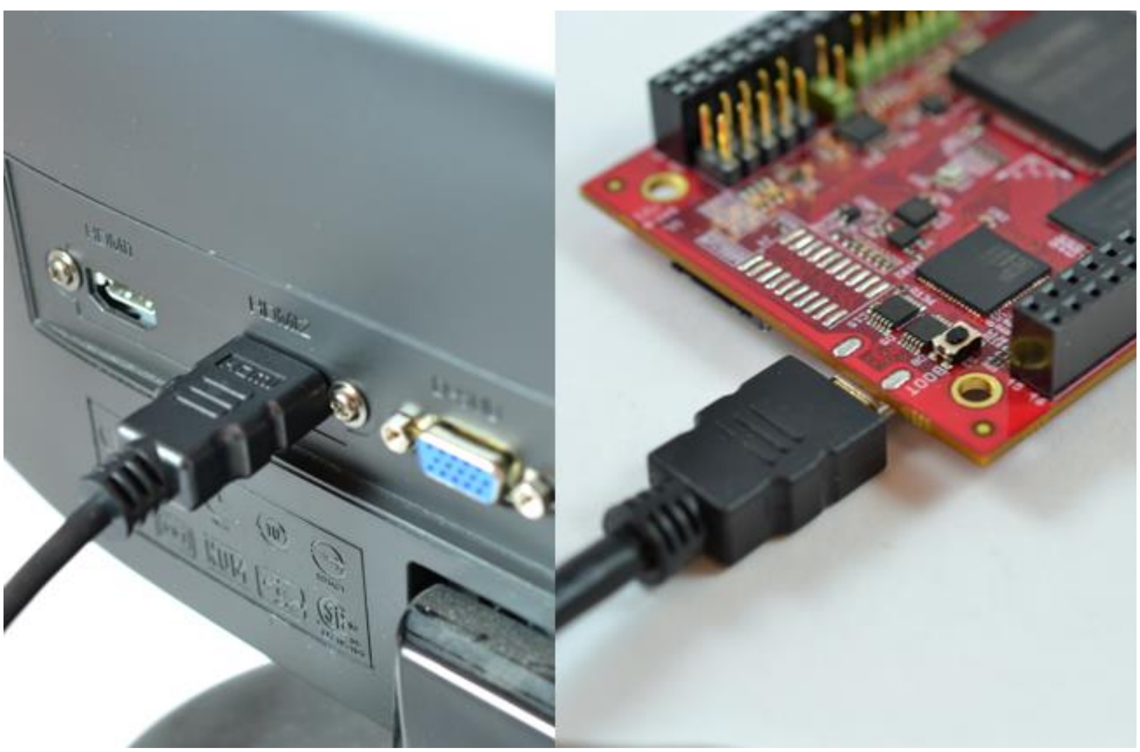

Figure 7: Connecting OSD3358-SM-RED to a monitor via HDMI cable

3. Connect the wired/wireless USB keyboard and mouse inputs to any of the four USB ports available on X4 and X5 connectors of the board.

Figure 8: Wireless adapter for keyboard and mouse plugged in to RED board

4. If internet connectivity is required:
    a. a WiFi-USB adapter can be plugged into one of the available USB ports on X4 or X5 connector. A list of compatible adapters can be found Compatible WiFi adapters: http://www.elinux.org/Beagleboard:BeagleBoneBlack#WIFI_Adapters. *OR*
    b. An ethernet cable can be plugged into the ethernet connector receptacle on X4.

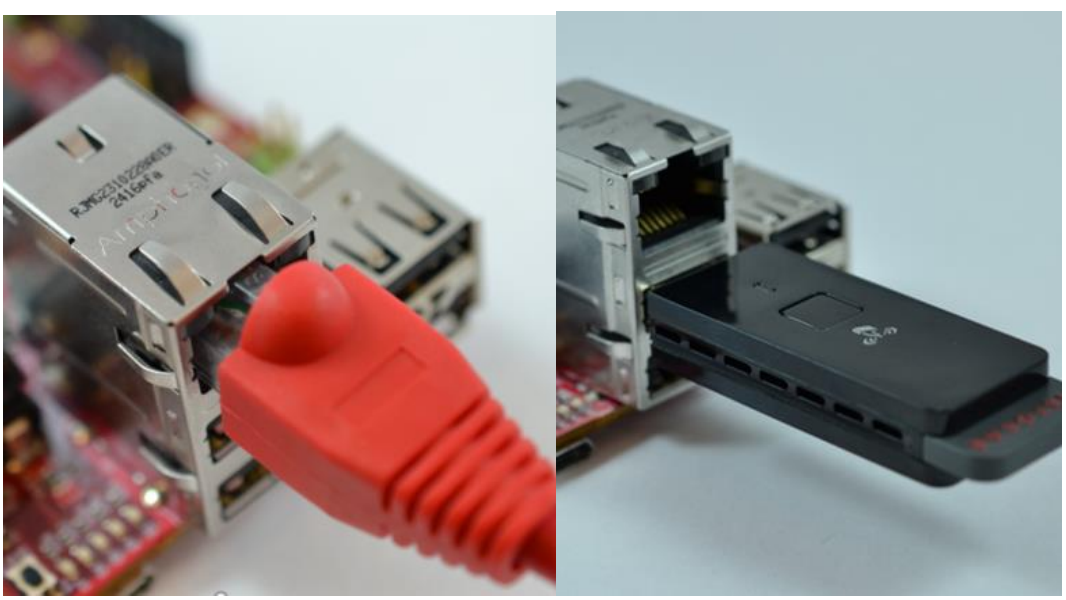

Figure 9: Ethernet cable/Wi-Fi dongle plugged in to the RED board

5. Plug the 5V AC adapter into the power connector X2.

Figure 10: AC-DC adapter plugged in to RED board

6. The board will go through the default boot up process
7. After the board boots up, the monitor screen should show a Linux desktop environment as shown below.

Figure 11: OSD3358-SM-RED desktop environment

8. Click on the Start>System Tools>Q Terminal to open up a command terminal.

Figure 12: Shell command terminal on booted desktop

2.2 Booting as USB client connected to a computer/laptop

The OSD3358-SM-RED platform can also be powered using a computer/laptop's USB port. The micro-USB to USB cable provided in the box can be used to connect the RED board to a computer/laptop. Unlike the standalone mode, this setup does not require additional hardware. The following steps describe the procedure to power the board as a USB slave and set up the development environment.

1. Connect the micro-USB side of the USB cable to the micro-USB port X1
2. Connect the USB side to one of the USB ports on a laptop/computer.

Figure 13: Powering RED board using a USB cable

3. Wait for the board to show up as a mass storage device on the computer/laptop

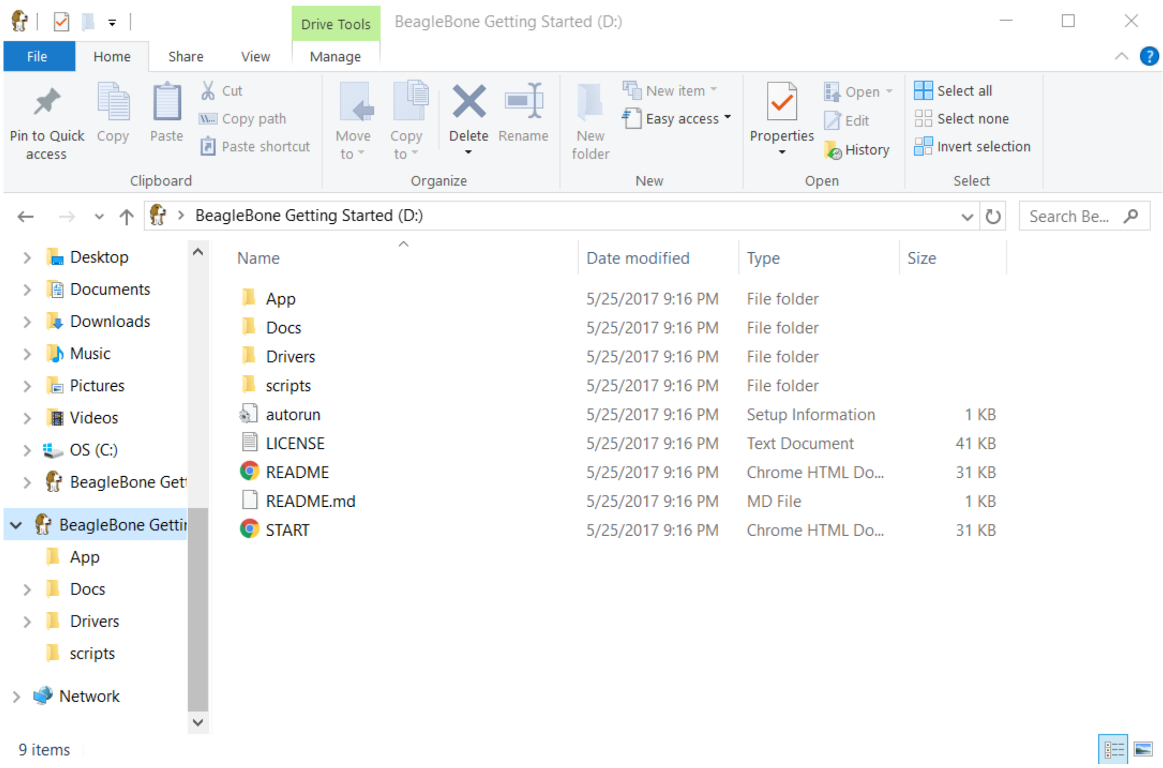

Figure 14: OSD3358-SM-RED USB mass storage filesystem window

4. Open a browser (Firefox/Chrome) and access the url: http://192.168.7.2. The webpage hosted on the webserver of the board should indicate that the board is connected as shown below.

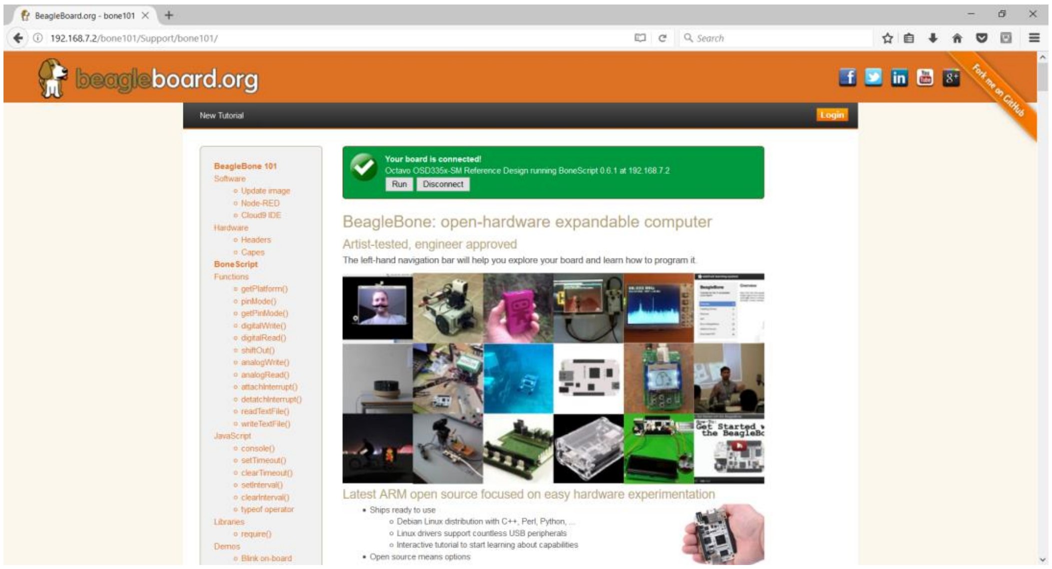

Figure 15: OSD3358-SM-RED home webpage indicating a connection to the board

5. Access the cloud9 IDE through the url: http://192.168.7.2:3000/. Cloud9 IDE is an opensource web based development environment supporting several languages. The environment should look like the following Figure with workspace containing examples that can be executed. Cloud9 also loads a shell terminal that can be seen at the bottom of the following figure.

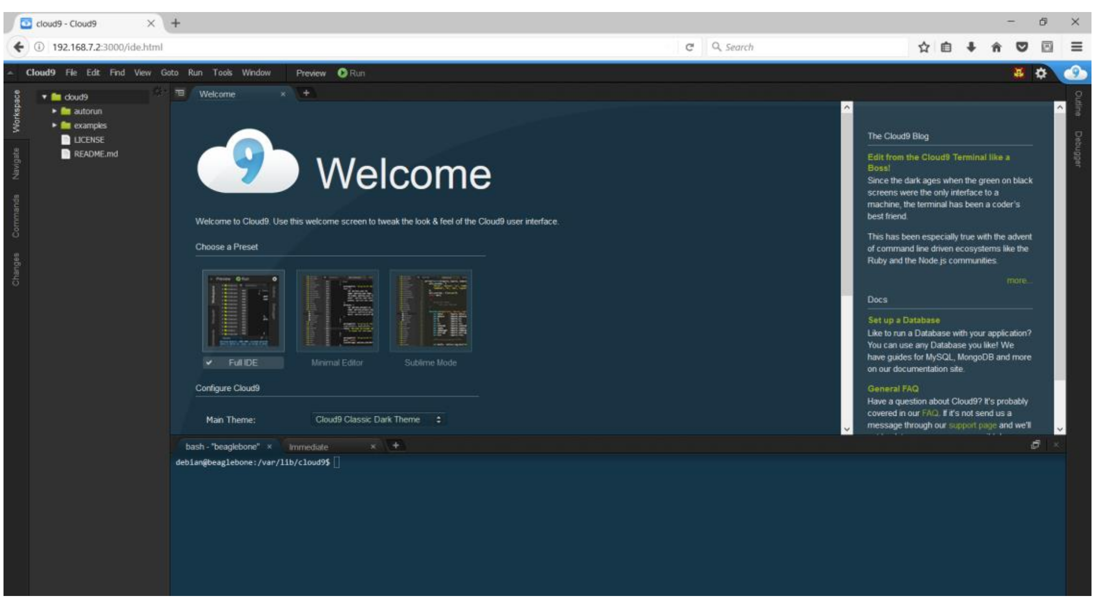

Figure 16: Cloud9 IDE environment

2.3 Boot messages through UART

The processor on the OSD3358-SM-RED platform (RED board) sends boot messages through the UART0 port, J3. This is useful in embedded Linux development or if there are boot problems to verify proper initialization of all components on the board. After boot, the UART0 port will function as a Linux terminal. Accessing this requires a serial connection to a host device with a serial port. A USB-UART serial adapter can also be used for host devices with a USB port. A terminal program, such as Putty is required for communication.

The following steps describe the procedure to access the boot messages during boot and the terminal through UART0 port using a standard computer/laptop as a host device (Note: Given the computer/laptop has no native serial ports, a USB-UART serial adapter must be used):

1. Connect the UART side of the adapter to header J3 of the RED board. Make sure to check for proper orientation of the cable to connect Tx and Rx signals of the UART0 port appropriately to the adapter's signals.

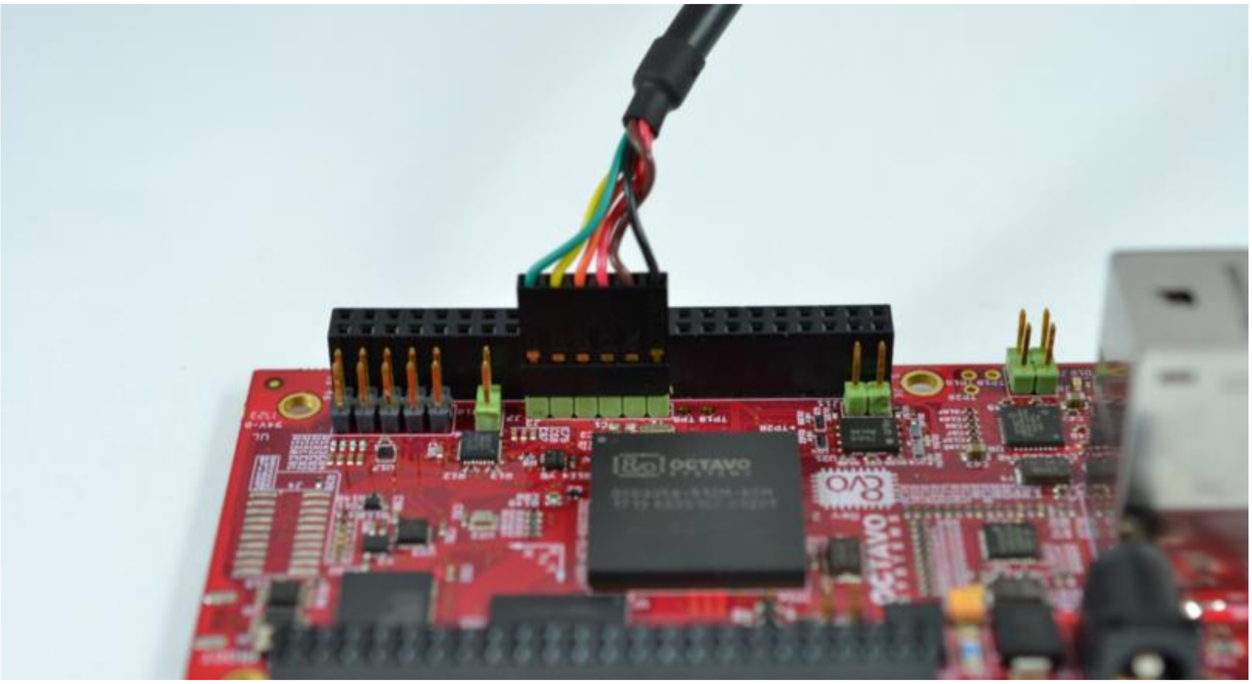

Figure 17: UART adapter cable plugged in to RED board

2. Connect the USB side of the adapter to the computer/laptop

Figure 18: UART-USB adapter plugged in to the computer

3. Check which COM port of the computer/laptop the adapter is connected to using the device manager.

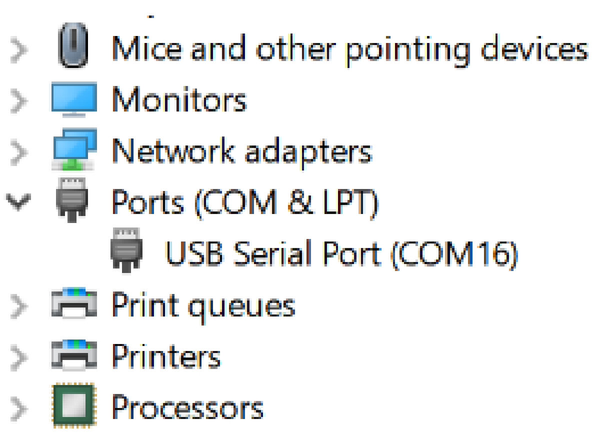

Figure 19: UART adapter detected as COM port 16

4. Open a serial console (opensource applications like Putty come with one). Select the serial communication option and specify the port and speed of the protocol as shown below and click on open.

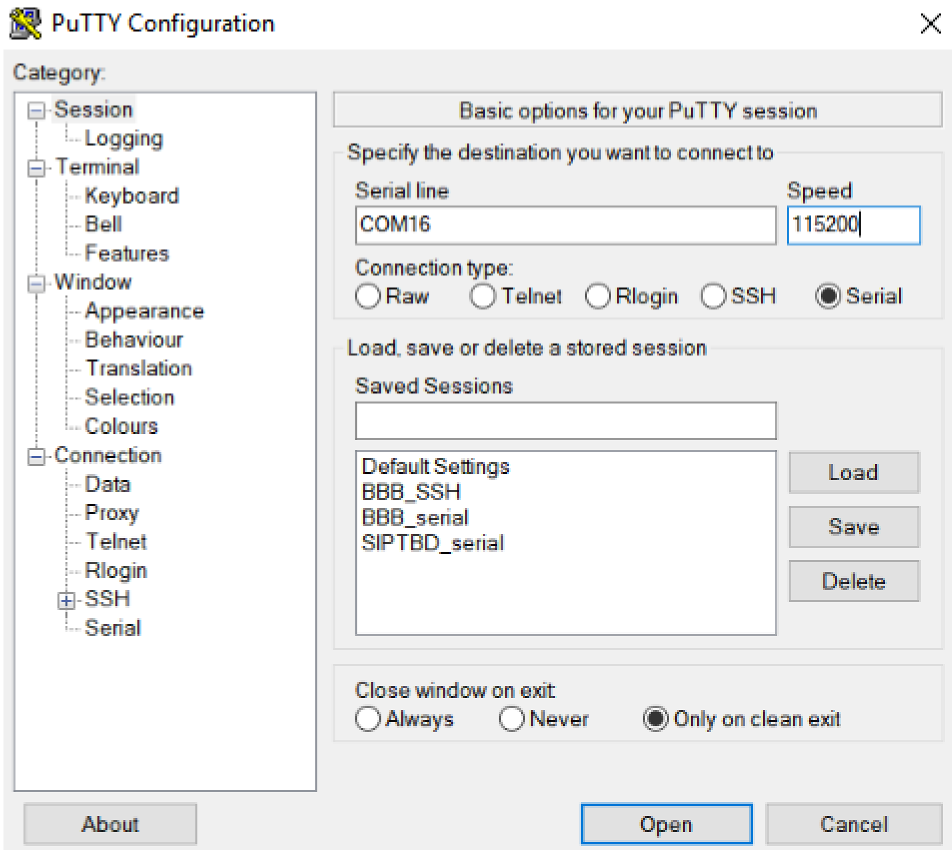

Figure 20: PUTTY configuration for serial communication

5. After a new serial console is open, connect the board to a power supply through one of the three power inputs of the board. The boot messages should begin to scroll on the console. After boot, the console will ask for secure shell login. You can login to the shell using the credentials – **debian:temppwd** .

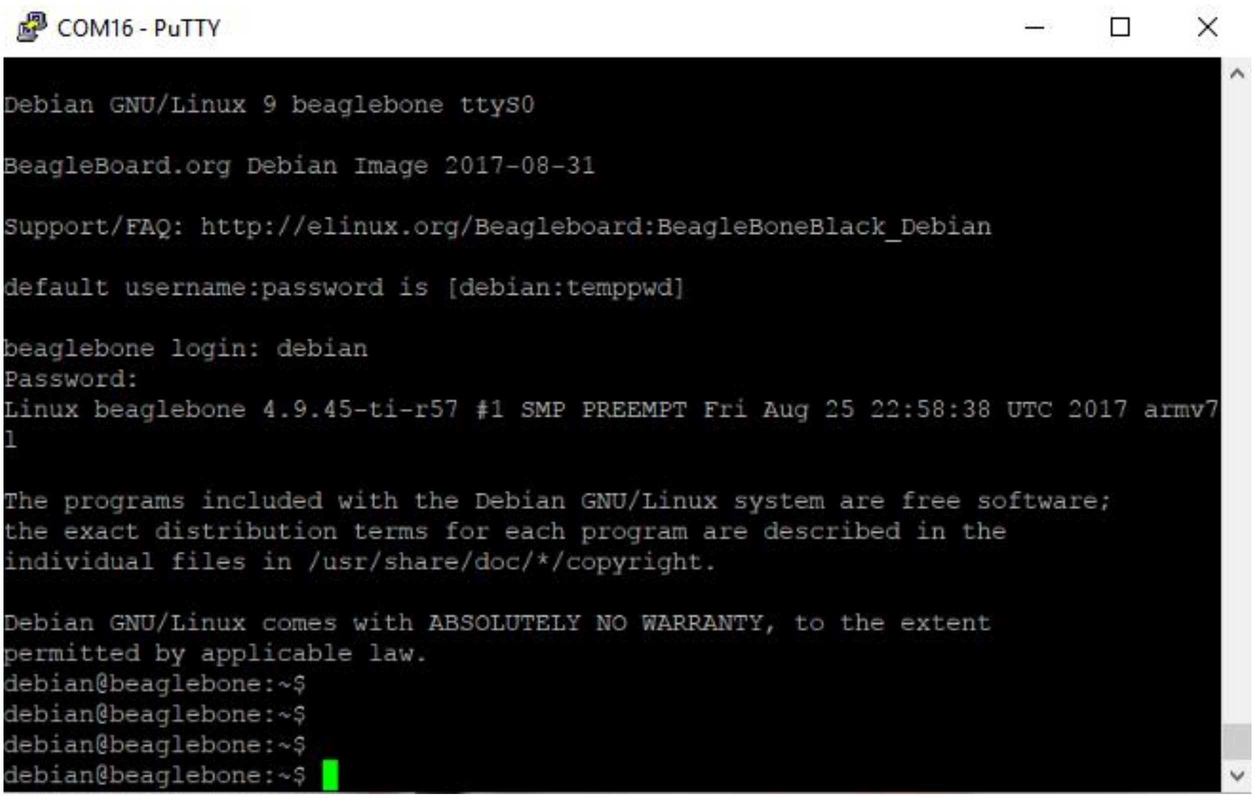

Figure 21: Debian Linux terminal after boot

2.4 User LEDs

While the LEDs D4 through D7 can be controlled as desired, they have the following standard function when the system is booted.

- D4: Activity indicator for MMC1 interface (eMMC)
- D5: Activity indicator for the kernel
- D6: Activity indicator for MMC0 interface (SD card)
- D7: Heartbeat indicator from Linux kernel

2.5 Powering down/ power cycling

Once the system has completely booted, the board can be safely powered down in the following way:

1. Execute the shutdown command on Linux command line. Wait for LEDs D5 and D7 to turn off. Unplug the power source.

The board can be reset using one of the following methods:

1. Hold the PWR button S2 pushed for 8 seconds
2. Push the RST button S1

3 Overview of OSD3358-SM-RED

The OSD3358-SM-RED is a Reference, Evaluation, and Development platform for the OSD335x-SM family of devices from Octavo Systems. The OSD335x-SM integrates the AM335x processor from Texas Instruments (TI) along with the TI TPS65217C PMIC, the TI TL5209 LDO, up to 1 Gigabyte (GB) of DDR3 Memory, a 4 Kilobyte (KB) EEPROM for non-volatile configuration storage and resistors, capacitors, and inductors into a single 21mm x 21mm design-in-ready package. The AM335x is a low cost, low power ARM® Cortex® A8 based processor that is widely used in the industry as well as the open source and hobbyist communities.

The platform offers access to many peripherals of the processor and functions as a foundation to swiftly design and develop applications based on the OSD335x-SM. It comes out of the box with software compatible with Beagleboard.org which has an active open source community and support.

3.1 Specification

Figure 22 shows the block diagram of the OSD3358-SM-RED.

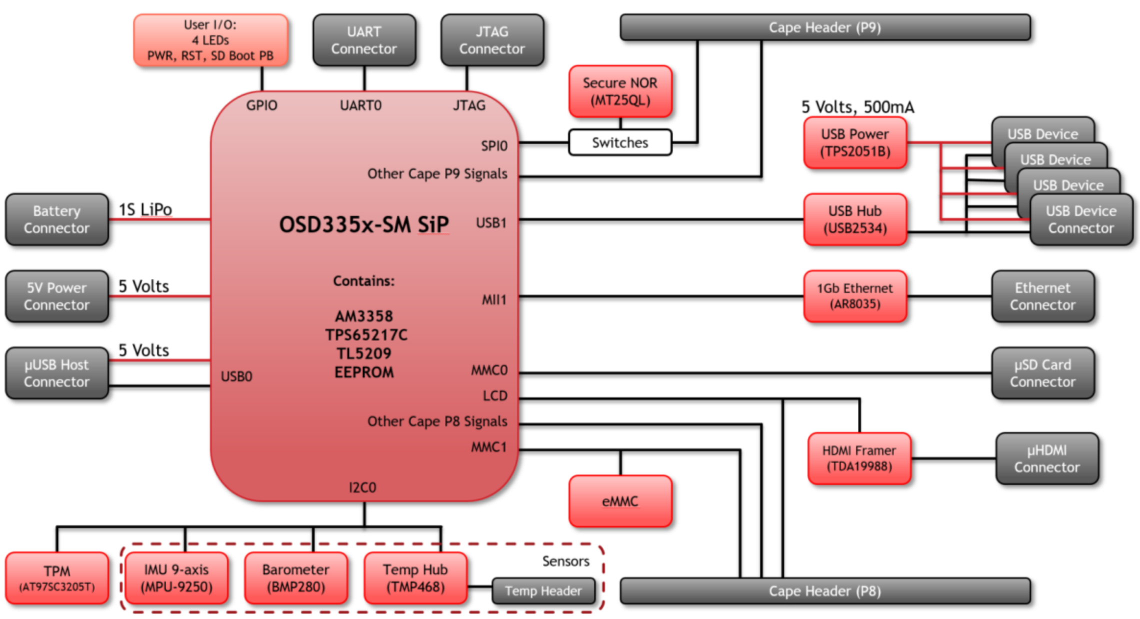

Figure 22: OSD3358-BSM RED Block Diagram

The following table provides an overview of the specification of the board.

Table 1: OSD3358-SM-RED specification

PART	PROPERTY	VALUE
OSD3358-512M-BSM (U1)	Processor	TI Sitara AM3358 ZCZ 1GHz SGX530 3D Graphics accelerator
	Real-time processor	2 Real-time Programmable Units (PRU)
	Memory size	512MB DDR3L, 400MHz
	Power Management	TI TPS65217C TI TL5209 LDO
	Non-volatile memory	4KB EEPROM
	Power inputs	AC (from AC-DC adapter), USB & Battery
Power Jack (X2)	Dimensions	2.5 mm x 5.5 mm barrel jack
Power button (S2)	Function	Power cycle the board/power the board ON/OFF
Reset button (S1)	Function	Holds the processor in reset
Boot button (S3)	Function	Used for SD-card boot
User LEDs (D4 – D7)	Number	4
eMMC (U7)	Part number	SDIN8DE2-16G
	Memory Size	16GB
4 port USB host ports (U8)	Max. current output	500mA
Micro-USB client port (X1)	Default Max. input current	500mA
HDMI Framer (U10)	Audio supported resolutions	• 1920 x 1080 • 1280 x 720 • 720 x 480 • 640 x 480
Ethernet port (X5)	Supported speeds	10BASE-Te/100BASE-Tx/1000Base-T IEEEE 802.3 compliant
Debug (J3 and J4)	ports	JTAG, UART (Boot messages)
JTAG connector (J4)	Part number	FTR-110-03-G-D-06
ADC	Number of inputs	7 available. 1 PMIC voltage monitoring input
	Reference voltage	1.8V
	Resolution	12 bit
NOR Flash (U21)	Part number	S25FL127S
	Memory size	128MB
	Supply voltage	3.3 V
	Sector size	64KB

	Package	8-pin W-PDFN
TPM (U16)	Part number	AT97SC3205T
	I2C bus	I2C0
	I2C address	0x29
	Supply voltage	3.3V
	Package	32-pin QFN
Temperature hub (U17)	Part number	TMP468
	Number of channels	9: 1 board monitoring channel, 4 channels available through expansion header and 3 channels unused
	Local and remote diode accuracy	±0.75˚C
	Local sensor accuracy	±0.35˚C
	Supply voltage	3.3V
	I2C bus	I2C0
	I2C address	0x48
Barometer (U22)	Part number	BMP280
	Pressure range	300 – 1100 hPa
	Relative accuracy	±0.16 Pa
	Supply voltage	3.3V
	I2C bus	I2C0
	I2C address	0x76
IMU (U23)	Part number	MPU-9250
	Gyroscope word length	16-bits
	Accelerometer word length	16-bits
	Magnetometer word length	14-bits
	Supply voltage	3.3V
	I2C bus	I2C0
	I2C address	0x68
PCB	Layers	4
	Minimum trace width	6mils
	Minimum via drill	12mils
	Minimum via diameter	24mils
Board	Size	108 mm x 55 mm
	Maximum height	32.15 mm
	Weight	60 grams

> ***NOTE:*** The NOR flash and eMMC parts are replaceable with equivalent footprint compatible parts.

3.2 Processor (U1)

The OSD3358-SM-RED board uses the OSD3358-512M-BSM (U1) System-in-Package (SiP) device from Octavo Systems. The SiP package integrates TI's AM3358 processor, TI power management system, 512MB of DDR3 memory and 4KB EEPROM in 21mm x 21mm BGA package. A list of references for each of the main components integrated in the OSD3358-SM is given below:

- AM335x datasheet: http://www.ti.com/lit/ds/sprs717j/sprs717j.pdf
- AM335x Technical Reference Manual: http://www.ti.com/lit/ds/sprs717j/sprs717j.pdf
- TPS65217C Datasheet: http://www.ti.com/lit/ds/symlink/tps65217.pdf
- Powering AM335x with TPS65217x: http://www.ti.com/lit/ug/slvu551i/slvu551i.pdf
- TL5209 LDO Datasheet: http://www.ti.com/lit/ds/symlink/tl5209.pdf
- 4KB EEPROM Datasheet: http://www.microchip.com/wwwproducts/en/24LC32A

> ***NOTE:*** The 4KB EEPROM sits on the I2C0 line of the AM3358 processor. Along with the board information that is needed to boot Linux, it is pre-programmed with 256 bytes of device specific information described in OSD3358-SM datasheet (https://octavosystems.com/docs/osd335x-sm-datasheet/).

3.3 PC USB interface (X1)

The board has a micro-USB OTG port X1 (shown below) that can connect to the USB port of a laptop/computer. In addition to powering the board, this port can also be used to access the board as a storage device or through an RNDIS ethernet connection. SSH clients such as PUTTY (http://www.putty.org/) can be used to connect and run programs through a secure shell command prompt. In addition, the Beagleboard compatible Linux image allows a web browser (Chrome or Firefox) to browse to the web server running on the board and access Cloud9 IDE, an easy to use web-based programming platform can also be used for software development.

Figure 23: USB OTG port of the RED board

NOTE: OSD3358-SM-RED Platform is compatible specifically with Linux images that can be installed on the boards Beaglebone Black, Beaglebone Black Wireless, Beaglebone Blue. Suitable modifications to the device tree need to be made for complete functionality. Please see Section 5.2 for details.

3.4 Serial debug interface (J3)

The UART0 port of the processor is exposed as a 1x6 header J3 (shown below). This interface serves as a debug port that can be accessed via USB port of a laptop/computer using a USB to Serial interface adapter. The processor sends status messages to this port during boot up allowing for debug and development.

Figure 24: UART header (boxed)

3.5 EMMC (U7)

A 16GB embedded Multi-Media Card (eMMC), U7, is connected to 8-bit wide MMC1 port of the processor (shown below). It contains the Linux image that the board will boot from by default. The board can be made to boot from other sources and the Linux image on the eMMC can be updated as needed.

Figure 25: eMMC (boxed)

3.6 Cape Headers (J5 & J6)

Two 2x23 cape headers J5 and J6 (shown below) serve as expansion interfaces to the processor exposing a wide range of peripherals of the processor. They are compatible with Beaglebone Black headers P8 & P9 (http://elinux.org/Beagleboard:Cape_Expansion_Headers) meaning capes for the Beaglebone Black are compatible and work with the OSD3358-SM-RED board. More information on these headers can be found in Section 4.22.

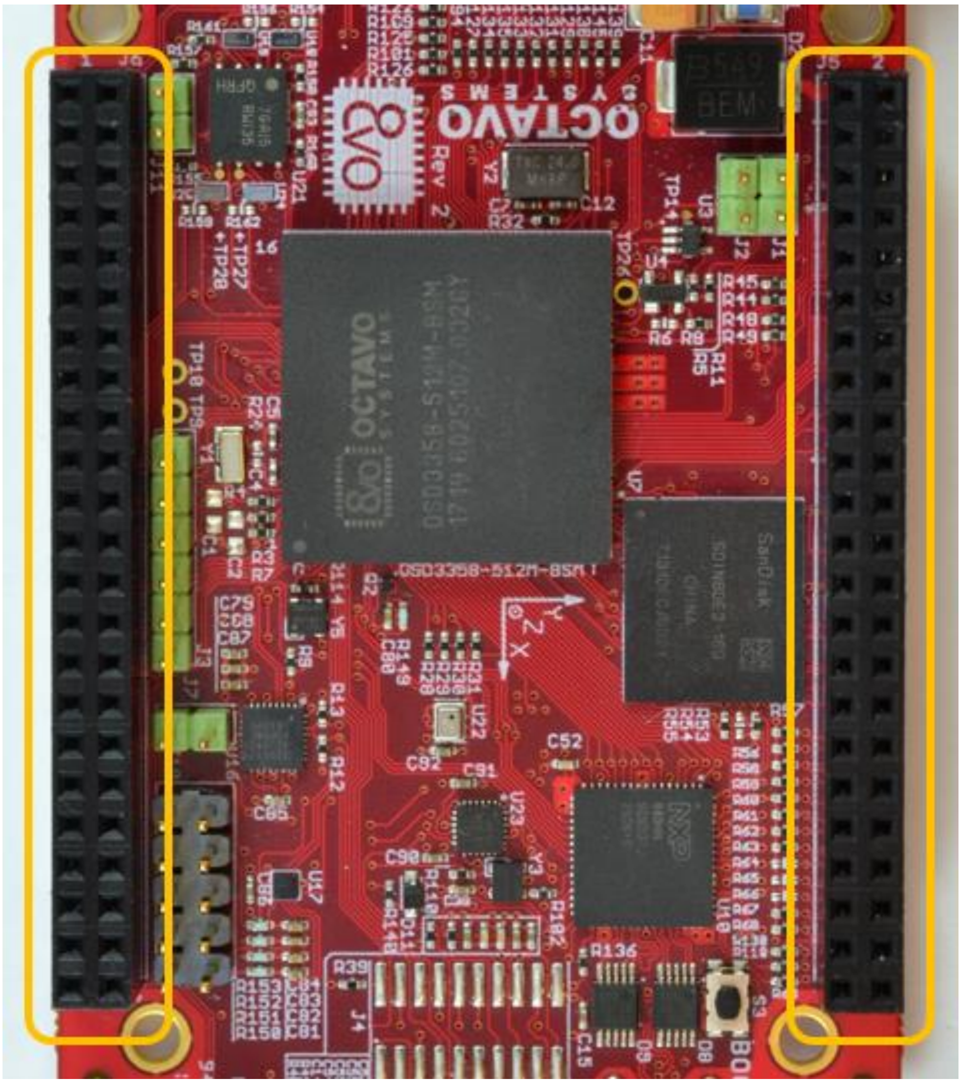

Figure 26: Cape headers

3.7 JTAG (J4)

A 20 pin CTI JTAG footprint J4 (shown below) is exposed for mounting a JTAG header on the board. This port can be used for software development and debug by connecting to the board through a JTAG emulator.

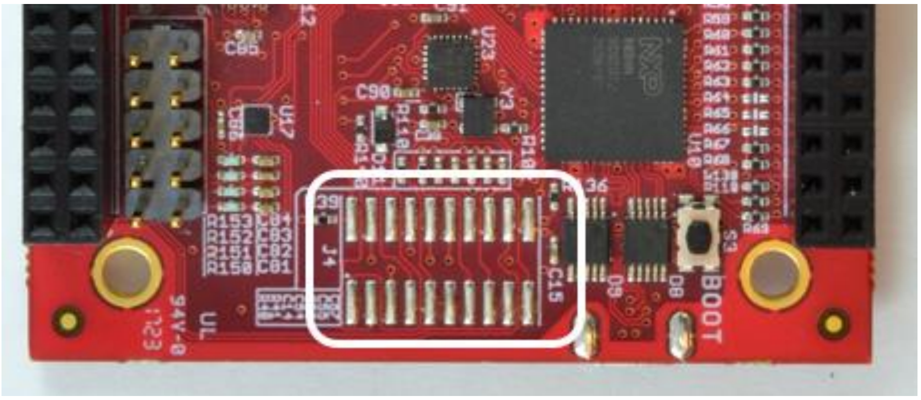

Figure 27: JTAG header footprint

In order to use the JTAG footprint it is recommended to use the emulators and JTAG header parts listed below (The JTAG footprint comes with solder on the pads to make it easy to solder the JTAG connector):

JTAG HEADER:

1. Samtec Part No: FTR-110-03-G-D-06

Emulators:

1. Blackhawk XDS2xx USB debug probe (http://www.blackhawk-dsp.com/products/USB200.aspx)
2. Blackhawk XDS100v2 USB debug probe (http://www.blackhawk-dsp.com/products/usb100.aspx)

Software:

1. TI Code Composer Studio (http://www.ti.com/tool/ccstudio)
2. IAR Embedded Workbench (https://www.iar.com/iar-embedded-workbench/)

3.8 Micro-SD card slot (X3)

A Micro-SD card slot X3 (shown below) interfaces the processor with a micro-SD card. The slot is connected to 4-bit wide MMC0 interface of the processor. A micro-SD card inserted in the slot can be used as a storage device, a boot source or can be used to flash the on board eMMC to update the software image of the board.

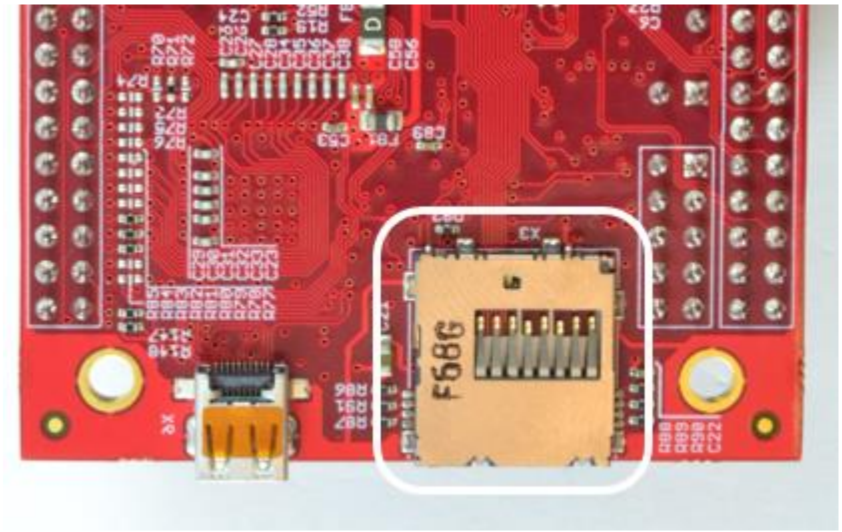

Figure 28: Micro-SD card slot

3.9 HDMI interface (X6)

An HDMI port X6 (shown below) provides an HDMI connection interface to the processor's 16-bit LCD controller through an HDMI framer (U10).

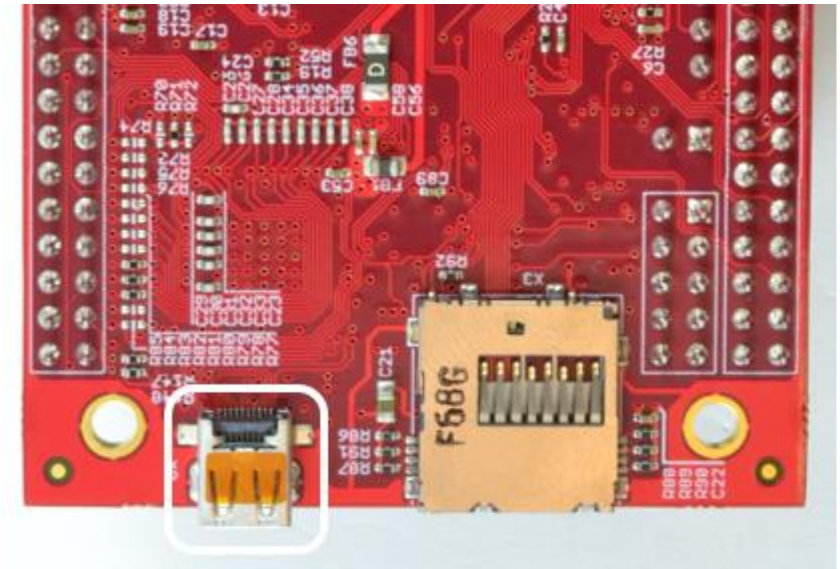

Figure 29: HDMI port

The following resolutions are supported via the preloaded software image.

- 1280 x 1024
- 1440 x 900
- 1024 x 768
- 1280 x 720

HDMI Framer TDA199988 Datasheet:
https://media.digikey.com/pdf/Data%20Sheets/NXP%20PDFs/TDA19988_DS_21_july_2011.pdf

> ***NOTE:*** When the HDMI interface is used, the LCD controller pins that are multiplexed with other functions and pinned out to the headers should not be used. When the HDMI interface is not used, the LCD pins can be used for their multiplexed functions through the headers J4 and J5.

3.10 Gigabit ethernet (X4)

The board has a Gigabit Ethernet interface port X4 (Shown below). The ethernet PHY is capable of 10/100/1000 Mbps speeds and is connected to the RGMII port of the processor. The Cloud9 IDE, web-server and the secure shell command prompt can be accessed by connecting an ethernet cable to this port in addition to the micro-USB port.

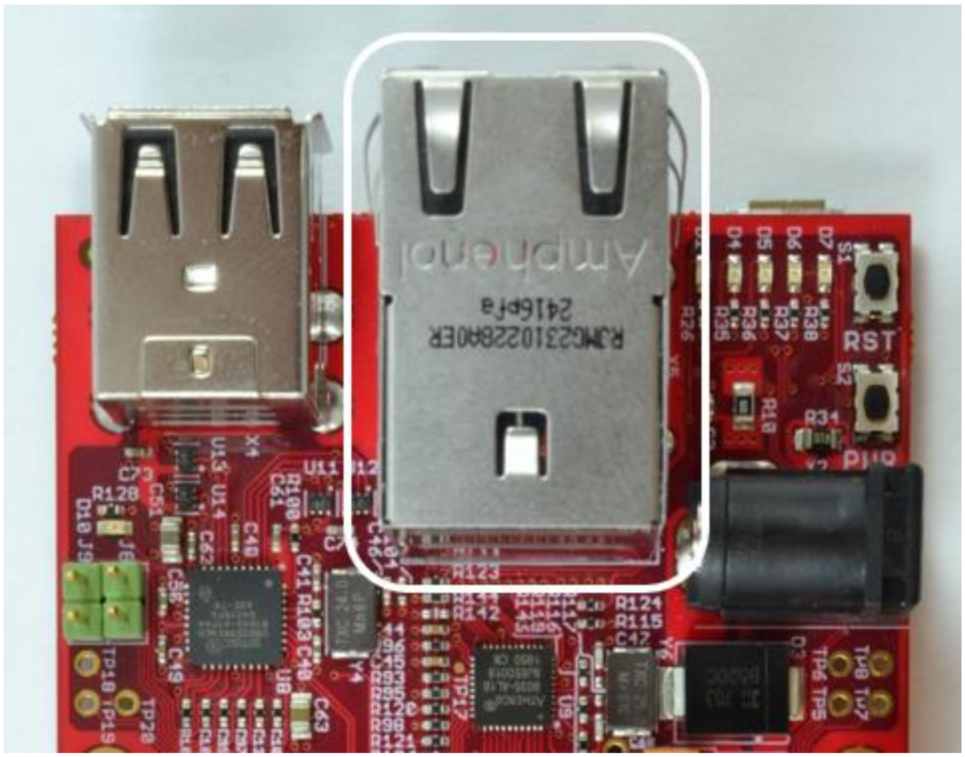

Figure 30: Ethernet port

Ethernet PHY AR8035 Datasheet:
https://media.digikey.com/pdf/Data%20Sheets/CSR%20PDFs/AR8035_DS_(Atheros)_Mar2011.pdf

3.11 4 Port USB hub (X4 & X5)

The board comes with a total of 4 USB Type-A ports in connectors X4 and X5 (shown below) facilitated by a USB hub controller that is connected to USB1 port of the processor. These ports can be used to connect USB storage devices and other USB gadgets such as WiFi adapters and wireless/wired keyboard and mouse. The 4 ports can supply a combined current of 500mA to the devices that are connected to them when the board is powered through AC input. While they cannot supply 500mA when the board is powered using the USB port of a laptop/computer, low powered USB devices can still be used.

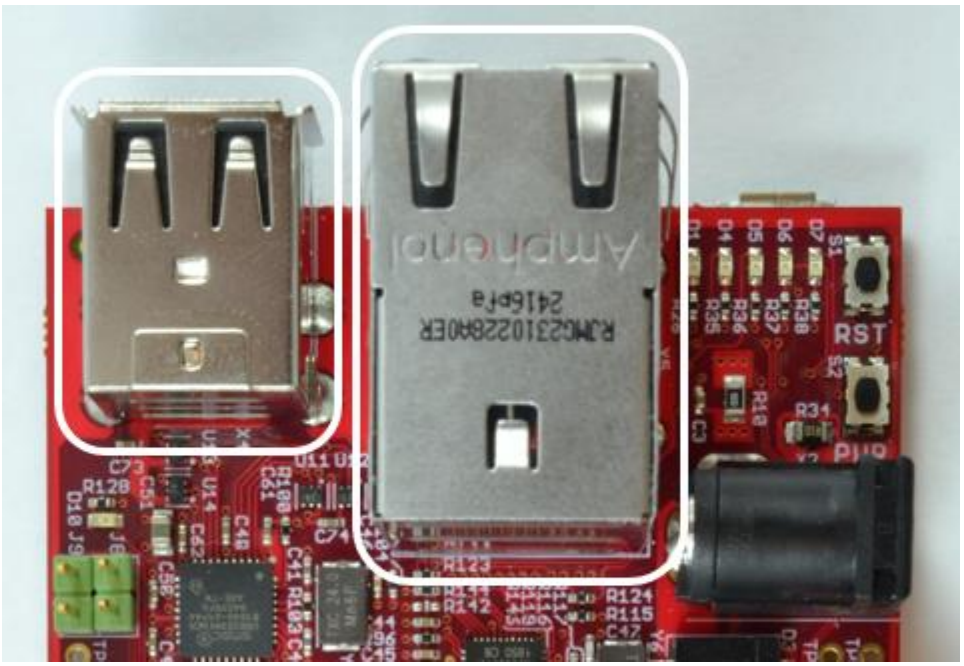

Figure 31: USB ports

USB port hub USB2534-1080AEN datasheet:
http://ww1.microchip.com/downloads/en/DeviceDoc/00001713A.pdf

3.12 Inertial Measurement Unit (IMU) (U23)

The board comes equipped with a 9 axis IMU (shown below). The MPU-9250 (U23) sits on the I2C0 line of the processor. It has a 3-Axis gyroscope, a 3-Axis accelerometer and a 3-Axis magnetometer inside and functions as a 9-Axis motion tracking device. More information on how to use it is given in Section 5.3.

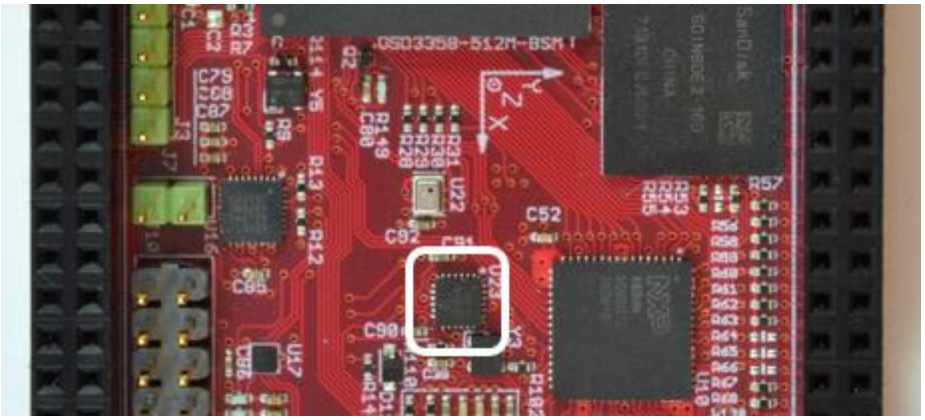

Figure 32: IMU on board

IMU MPU9250 datasheet: https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf

3.13 Temperature sensor hub (U17)

A high accuracy 9 channel temperature sensor TMP468 (U17) (shown below) is connected to I2C0 interface of the processor. 4 channels of the sensor are exposed using a 2x5 header P3 (shown below). 1 remote channel is utilized to measure temperature on board and 1 channel internal to the temperature sensor gives the temperature of the sensor itself. More information on how to use it can be found in Section 5.3.

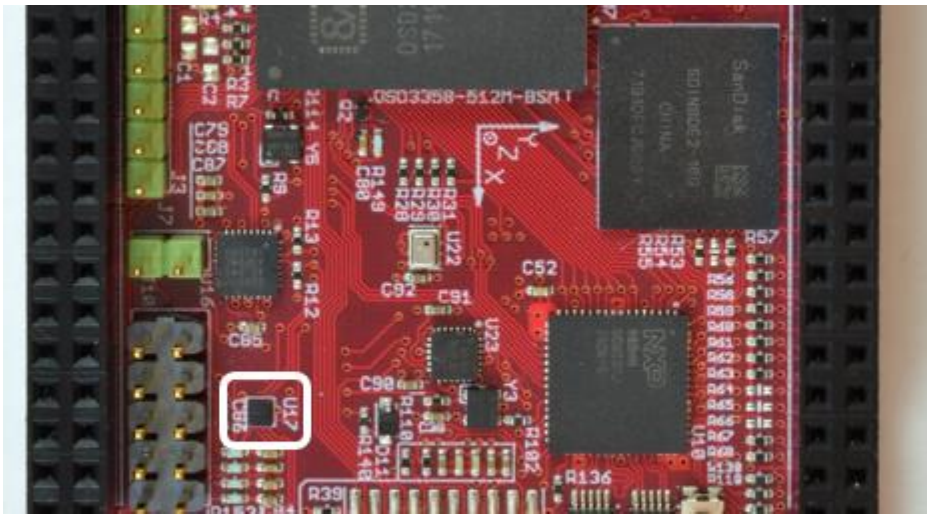

Figure 33: Temperature sensor on board

Temperature sensor TMP468 datasheet: http://www.ti.com/lit/ds/symlink/tmp468.pdf

3.14 Pressure sensor (U22)

A Piezo-resistive pressure sensor BMP280 (U22) (shown below) is connected to the processor through I2C0 interface. The pressure sensor can deliver high precision pressure measurements that aid in altitude measurements. More information on this device and how to use it can be found in Section 5.3.

Figure 34: Pressure sensor on board

Pressure sensor BMP280 datasheet: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280_DS001-11.pdf

3.15 Trusted Platform Module (TPM) (U16)

A Trusted Computing Group (TCG) compliant TPM v1.2 U16 (shown below) on board facilitates development of security software. Applications supported by the TPM include secure cryptographic key generation, secure boot, authentication and random number generation. The TPM sits on I2C0 interface of the processor. While the board does not come with any software support for secure applications out of the box, inclusion of the TPM enables development of such software solutions.

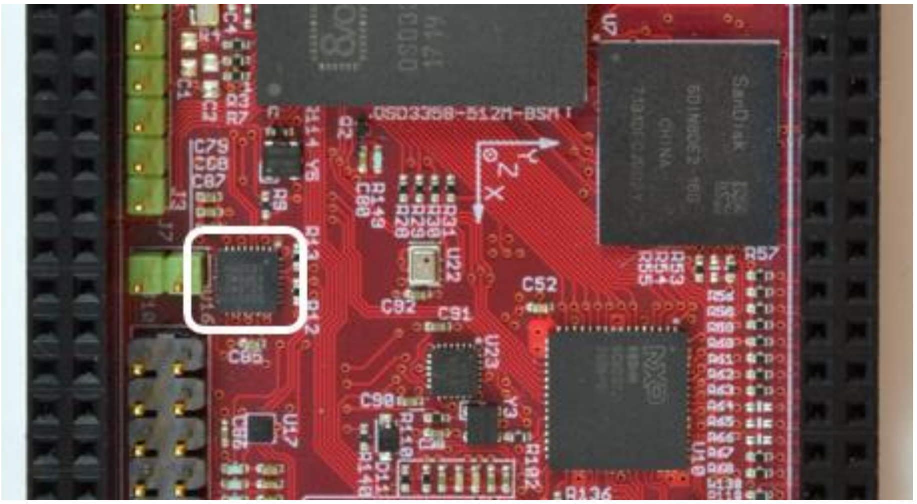

Figure 35: TPM on board

TPM AT97SC3205T datasheet: http://www.atmel.com/Images/Atmel-8883S-TPM-AT97SC3205T-Datasheet-Summary.pdf

3.16 NOR flash (U21)

A 128MB Serial NOR Flash U21 (shown below) is connected to the processor via SPI0. This non-volatile storage space can be utilized by applications running on the processor when needed and can also be used for secure boot in conjunction with the TPM. The SPI0 serial interface used to communicate with the Flash is multiplexed with the I2C1 and UART2 interfaces on header J6. When using the NOR Flash, the 2 pins of the jumper J11 should be shorted together or the FLASH_EN signal controlled by the processor should be pulled high. This will disable the use of the I2C1 and UART2 interfaces through J6. If the Flash is not used, the pins of the jumper J11 must not be populated in order to use the I2C1 and UART2 interfaces of the processor through header J6.

Figure 36: NOR flash on board

NOR flash S25FL127S datasheet: http://www.cypress.com/file/177961/download

4 Detailed hardware design

The following sections describe the hardware design of the OSD3358-SM-RED platform.

4.1 Power management

The following figure shows the power system of the OSD3358-SM-RED board. The voltage rails, their current sourcing capabilities and the components that are powered by them are also shown.

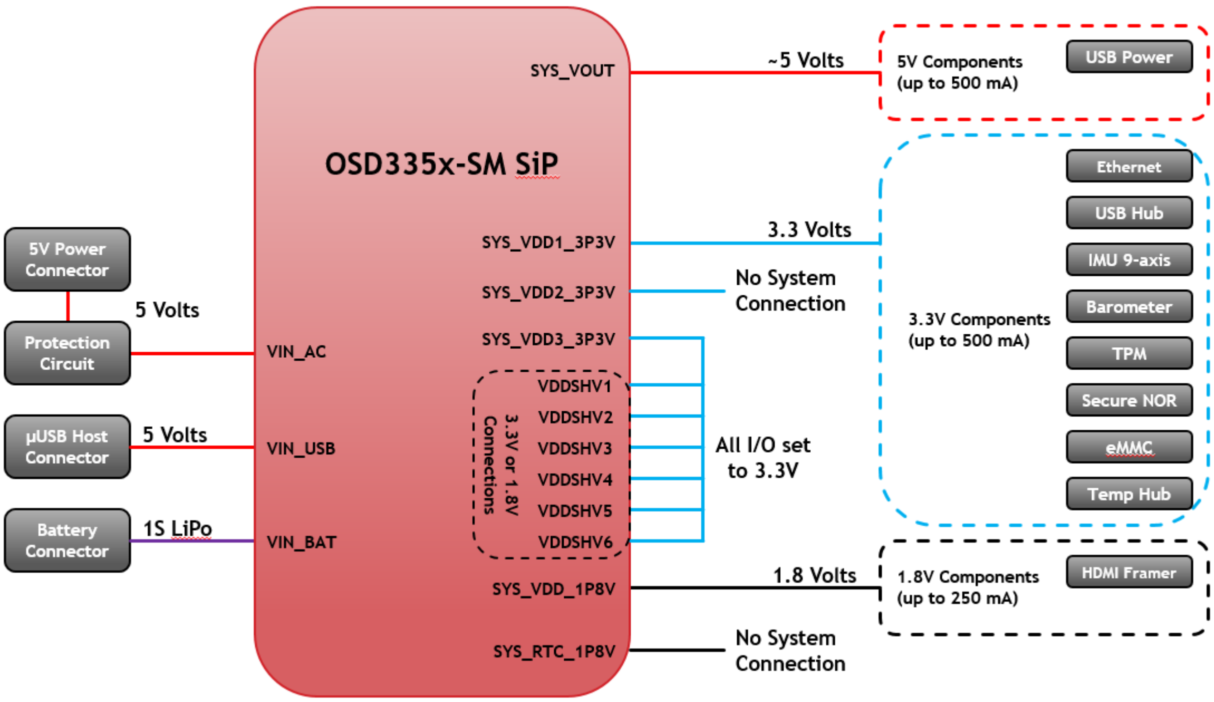

Figure 37: Power system outline of OSD3358-SM-RED

The power system of the platform is comprised of :

1. AC, USB, battery inputs
2. protection circuit on AC input
3. The PMIC TPS65217C and LDO TL5209 inside the OSD3358-512M-BSM.

The PMIC is responsible for powering the AM335x processor and the DDR3 as well as provide output power for other system needs. It provides configurable power-up and power-down sequencing required by the processor and monitors the processor input voltage levels. It contains 3 DC-DC power converters, 2 LDOs and 2 load switches that are configured as LDOs and used as power supplies. It can be powered by any combination of a 5V AC adapter, USB port, or Single Cell Li-Ion battery. Figure 38 shows the power system of OSD335x-SM including connections between the PMIC and various power domains of the processor.

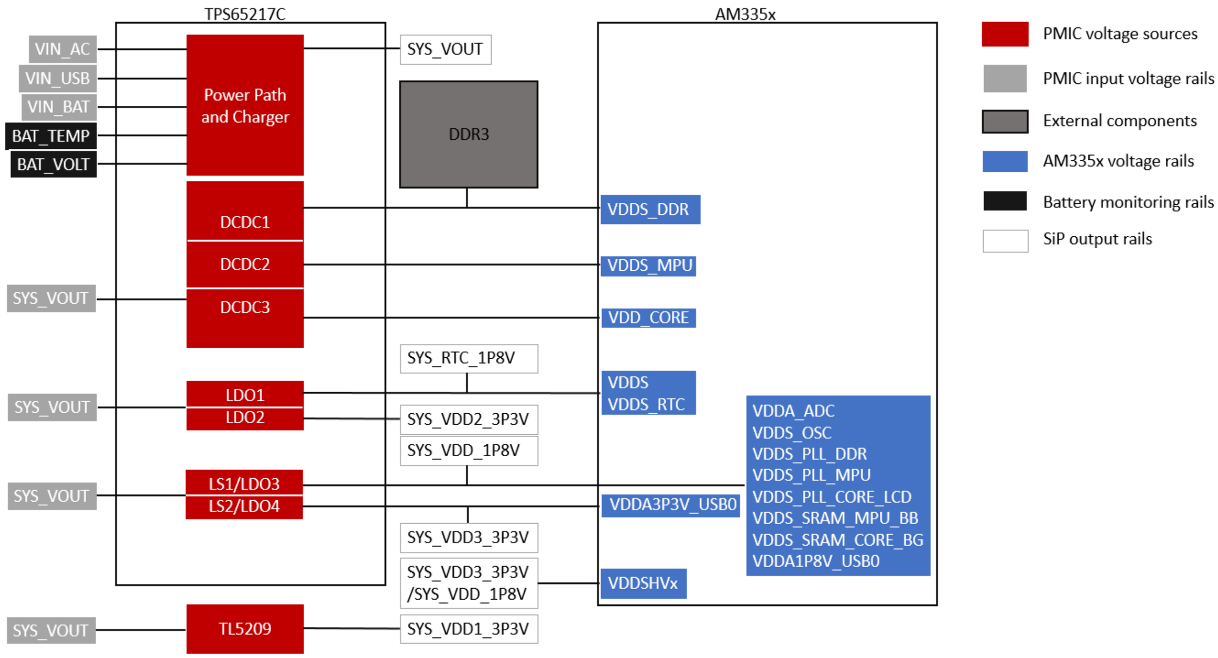

Figure 38: Power system outline of OSD335x-SM

Table 2 shows the voltage output levels of each of the voltage sources which can be measured on the respective test point. Each of the output voltages can be changed dynamically using I2C commands when the PMIC is in active mode.

Table 2: PMIC Voltage outputs

TPS65217C Voltage Source	OSD335x-SM Voltage rail	Test Point Label	Voltage (V)
DCDC1	VDDS_DDR	TP1	1.5
DCDC2	VDD_MPU	TP3	1.1
DCDC3	VDD_CORE	TP2	1.1
LDO1	SYS_RTC_1P8V	TP14	1.8
LDO2	SYS_VDD2_3P3V	TP13	3.3
LDO3	SYS_VDD_1P8V	TP15	1.8
LDO4	SYS_VDD3_3P3V	TP16	3.3

4.2 Additional functions of the PMIC

The PMIC also performs some tertiary functions. A few of the key functions are listed below. For a complete explanation of all the functions provided by the TPS65217C PMIC refer to the datasheet (http://www.ti.com/product/TPS65217/datasheet).

1. The PMIC retrieves the processor from OFF or SLEEP mode upon detecting a falling edge on PWR_BUT signal. It also power cycles the processor if PWR_BUT signal is held low for more than 8 seconds by holding the power button S2 down.
2. It provides an active low wake up signal (PMIC_NWAKEUP) which is de-asserted when a wakeup event is detected and is connected to EXT_WAKEUP pin of the processor.
3. The PMIC has an interrupt pin (PMIC _NINT) to signal an event or fault condition to the processor via EXTINT pin. The pin is released when the processor reads the INT register.
4. PMIC provides a linear charger for Single Cell Li-Ion batteries and allows charging of the battery and powering of the system at the same time. More information on this feature can be obtained from the datasheet.
5. The TPS65217C provides protection to AM335x and itself in the event of catastrophic situations like an unexpected short or excessive current leakage. Please refer to the OSD335x Power application note (https://octavosystems.com/app_notes/osd335x-power-application-note/) for information on how to design the power system for an application with OSD335x.

4.3 Input power

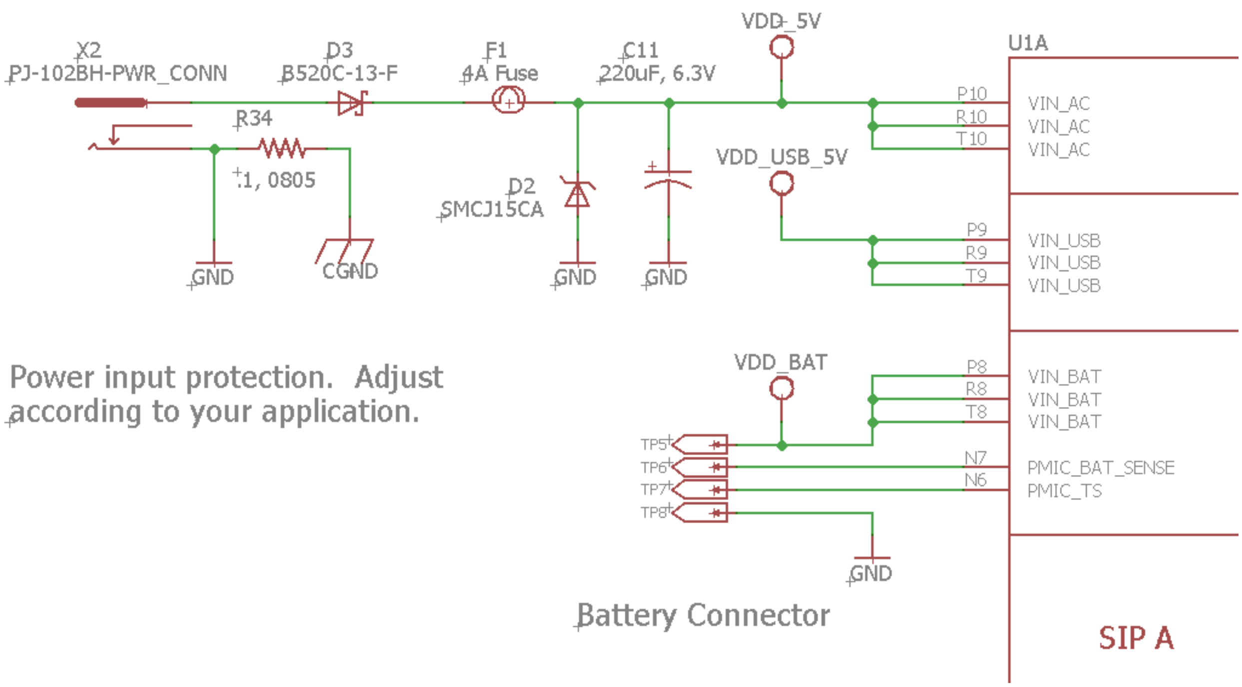

Figure 39: OSD3358-SM-RED input power circuit

Figure 39 shows power inputs of the OSD3358-SM-RED platform. A combination of the AC, USB and battery inputs can be used to power the board. To protect the board from power surge, reverse polarity and transient input voltage behavior, the AC adapter input signal goes through an input protection circuit shown in Figure 39. To power the board using a 1S Li-Po battery, the following hook up is required.

Table 3: OSD3358-SM-RED battery connections

Pin	Function	Hook up
VIN_BAT (TP5)	Battery input	Positive battery terminal
BAT_SENSE (TP6)	Battery voltage sense input	Positive battery terminal
PMIC_TS (TP7)	Battery temperature sense input	NTC thermistor to sense battery temperature. Works with 10k and 100k thermistors
GND (TP8)	Digital system ground	Battery negative terminal

The hook up is shown in the following Figure:

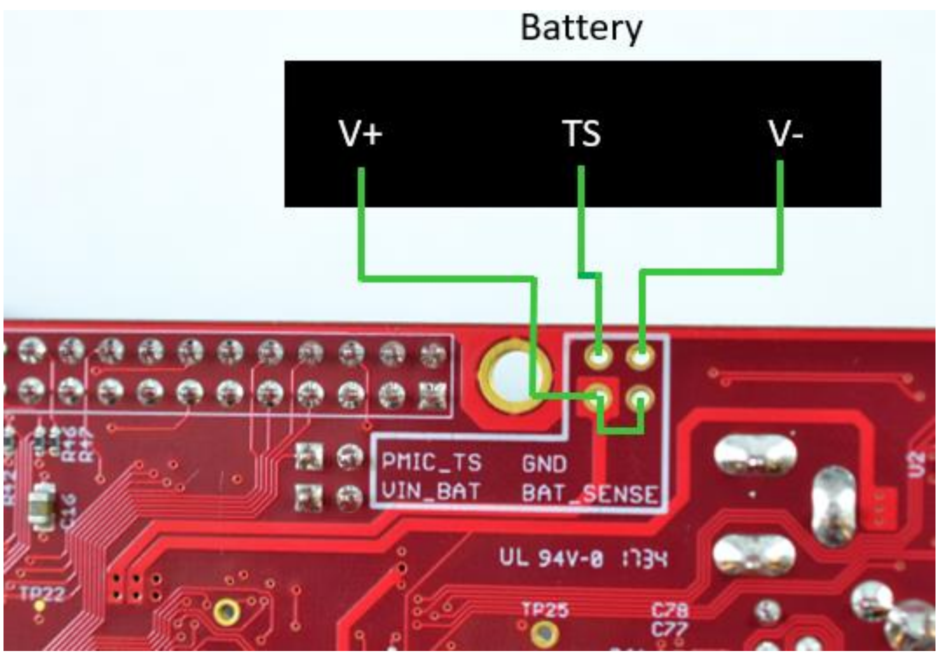

Figure 40: Battery hookup

NOTE: Due to the input voltage requirements of the TL5209 LDO that supplies SYS_VDD1_3P3V of the OSD335x-SM, the voltage of the battery that will power the board needs to be greater than 3.7 V. If the voltage of the battery falls below this threshold, SYS_VDD1_3P3V begins to go out of regulation resulting in some of the on board components powered by this voltage rail not working in spec. This includes the SD card slot and the eMMC, which are the main boot sources for the board.

4.4 PMIC-AM3358 interface

As discussed in sections 4.1 and 4.2, functions of the PMIC include: proper power up of the processor, monitoring of the AM3358 power rails, interrupt generation and reset management of the processor. To perform these functions, the PMIC and AM335x must be connected. These connections can be found in the datasheet (https://octavosystems.com/docs/osd335x-sm-datasheet/). Figure 41 shows some of these connections in OSD335x-SM-RED schematics.

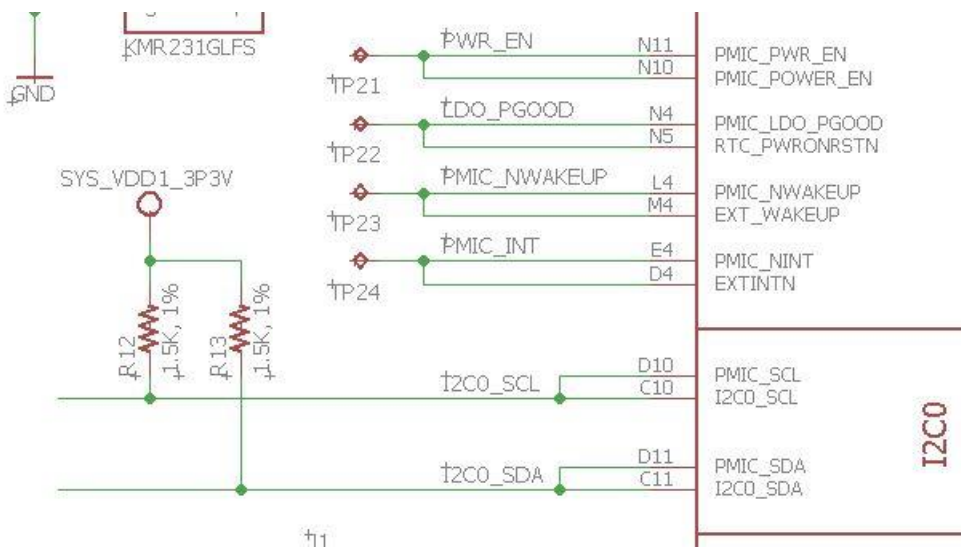

Figure 41: OSD3358-SM-RED PMIC - AM3358 interface

NOTE: I2C0 lines are internally pulled up to 3.3 V using 4.7K resistors. In addition, 1.5K resistors were used to pull up the I2C lines in the RED board. Please see: (http://www.ti.com/lit/an/slva689/slva689.pdf) for more information on how to calculate I2C pull ups.

NOTE: Test points TP21, TP22, TP23 and TP24 are provided to monitor these signals for debug purposes.

Figure 42 shows the reset circuit in the design schematic. The PMIC's PGOOD output controls the PWRONRST input of the processor. So, the processor is held in reset until PMIC_PGOOD comes up which means all the power rail outputs of the PMIC are up and stable.

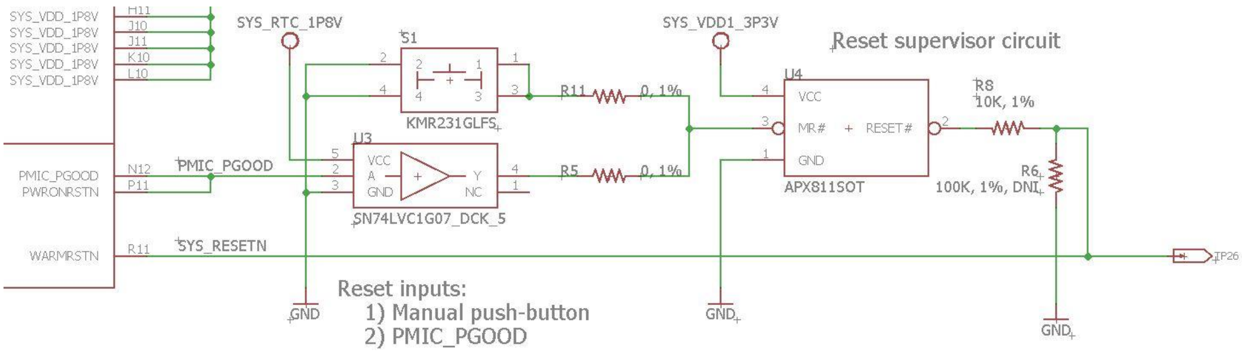

Figure 42: OSD3358-SM-RED reset circuit

> ***NOTE:*** PWRONRSTN holds all the processor systems under reset except for RTC module. The RTC module has its own power-on-reset input, the RTC_PWRONRSTN pin of OSD335x.

Figure 42 also shows the reset mechanism for the WARMRSTN input of the processor. WARMRSTN input is a soft reset and does not affect most of the power/boot systems of the processor unlike PWRONRSTN. While PMIC_PGOOD and PWRONRSTN operate on 1.8V logic, WARMRSTN is 3.3V logic input. Therefore, an open drain buffer U3 is used to drive the WARMRSTN input. WARMRSTN can also be pulled low using the reset switch S1. As the switch output is susceptible to ground bounce, causing multiple or partial reset, a reset supervisor U4 is used in between the output of the switch and WARMRSTN input. For more information on the reset circuit and the functions of each signals, please refer to the 'OSD335x Design Tutorial series' 'OSD335x Reset Circuitry' article (https://octavosystems.com/app_notes/osd335x-design-tutorial/bare-minimum-boot/reset-circuitry/)

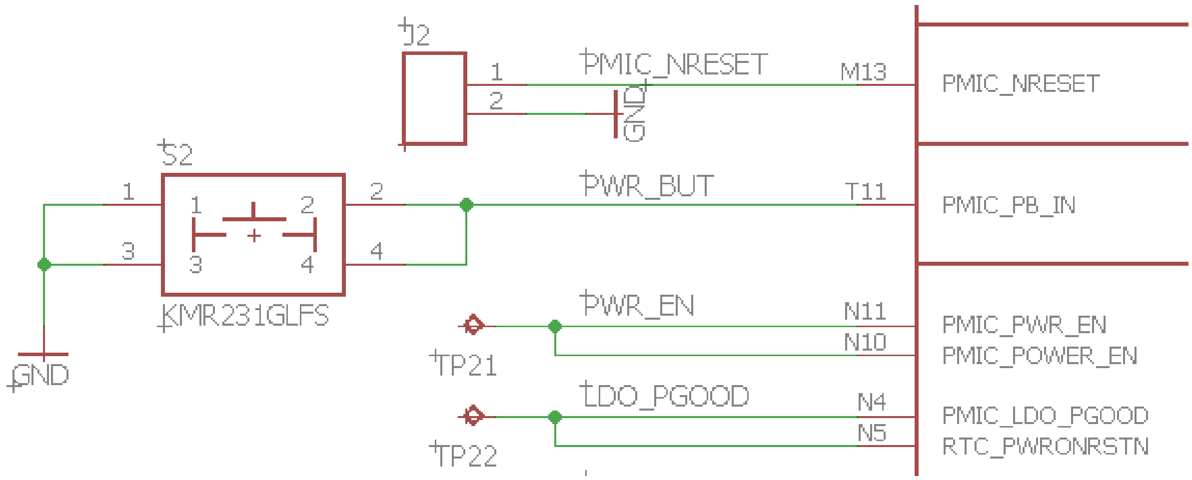

Figure 43: OSD3358-SM-RED power button

The OSD335x-SM-RED board also provides a power button S2 connected to the PMIC_PB_IN (active low push button input of PMIC). PMIC_PB_IN has the following functions:

- The PMIC will be powered up from OFF or SLEEP mode upon detecting a falling edge on PWR_BUT signal (hold the power button, S2 down).
- PMIC is power cycled/reset when PWR_BUT (Switch S2) is held low for more than 8 s. All rails will be shut down by the sequencer and all register values are reset to their default values. Rails not controlled by the sequencer are shut down immediately. After 1 sec of RESET state, the board powers up again.
- If the PMIC_IN_PB_IN pin is kept low for an extended amount of time, the device will continue to cycle between ACTIVE and RESET state, entering RESET every 8 seconds.

> ***NOTE:*** Holding the power button low for 8 seconds on some boards will power off the system instead of power cycling it depending on current draw.

Figure 43 also shows a 2 pin header J2. Pin 1 of the jumper is connected to PMIC_NRESET and pin 2 is connected to GND. The PMIC can be put in reset state by using a jumper to connect the 2 pins of the header. The PMIC will remain in reset until the PMIC_NRESET input is released from the pull down to GND and for a minimum of 1 sec.

4.5 Clamping circuit

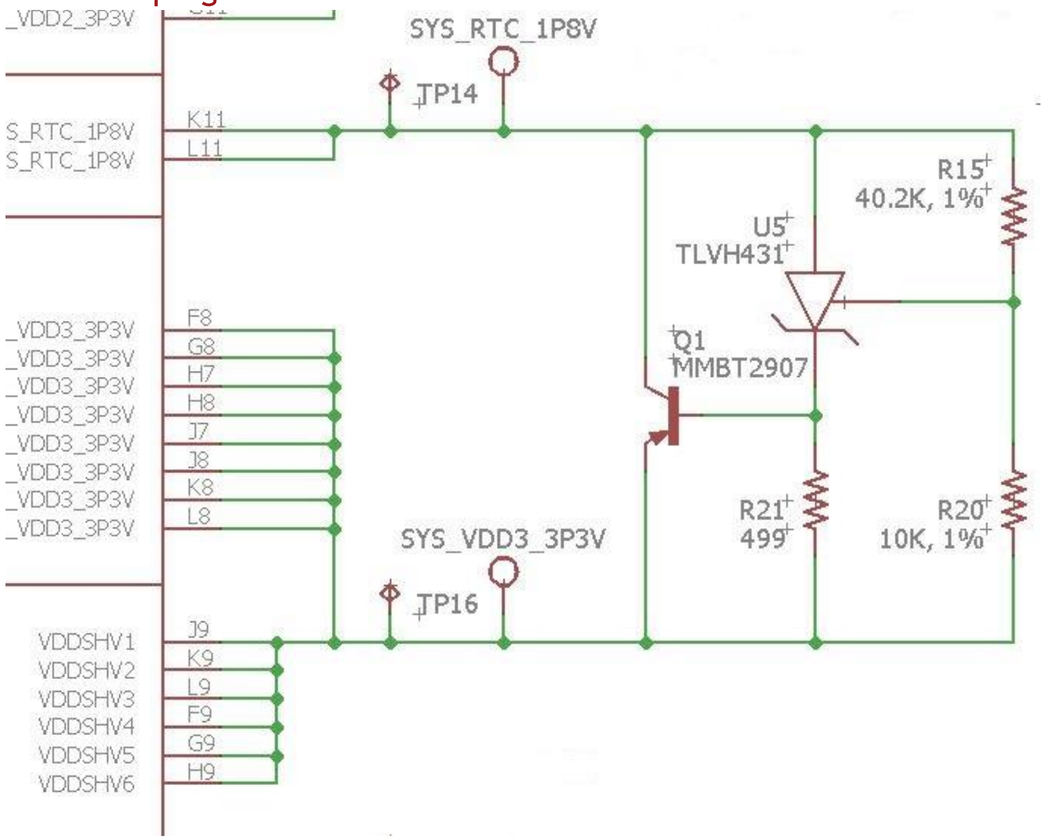

Figure 44: OSD3358-SM-RED clamping circuit

Figure 44 shows a clamping circuit designed to prevent a PMIC shutdown fault. The AM335x datasheet requires that the voltage difference between the power rails VDDS (1.8V) and VDDSHVx [1-6] (3.3V) of the AM335x processor be less than 2V during the entire power-down sequence. A detailed description of the clamping circuit and its operation can be found in “The OSD335x Reference Design Lesson 1 Power Circuitry Part 4: Clamping Circuit” (https://octavosystems.com/app_notes/osd335x-reference-design-lesson-1-power-circuitry-part-4-clamping-circuit/).

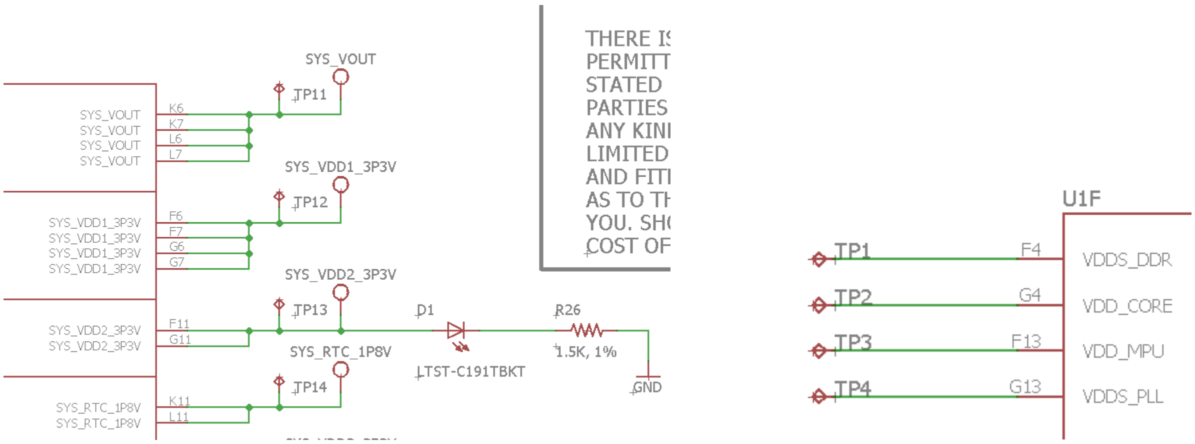

Figure 45: OSD3358-SM-RED power rails

Figure 45 shows the main power rails of the PMIC connected to test points on the OSD3358-SM-RED board schematic. This facilitates debug of the hardware system. The test points allow probing of the power rails and verification of the boot up sequence. To give an initial indication that the board is booting up, LED D1 is connected to SYS_VDD2_3P3V, since this power output will come up as part of the boot sequence.

4.6 Clock inputs to the processor

OSD335x-SM has 2 clock inputs, OSC0 and OSC1.

4.6.1 OSC0:

This input is the main clock source for the processor and is the reference clock to all subsystems except RTC. This clock input is a requirement for all designs with the OSD335x-SM. For the OSD3358-SM-RED, this clock is 24MHz.

4.6.2 OSC1:

This input is the clock source for RTC system of the processor. The clock input for OSC1 should be 32.768KHz generated by a single ended clock source or crystal oscillator. This clock input is optional and can be removed if the RTC subsystem of the processor is not used.

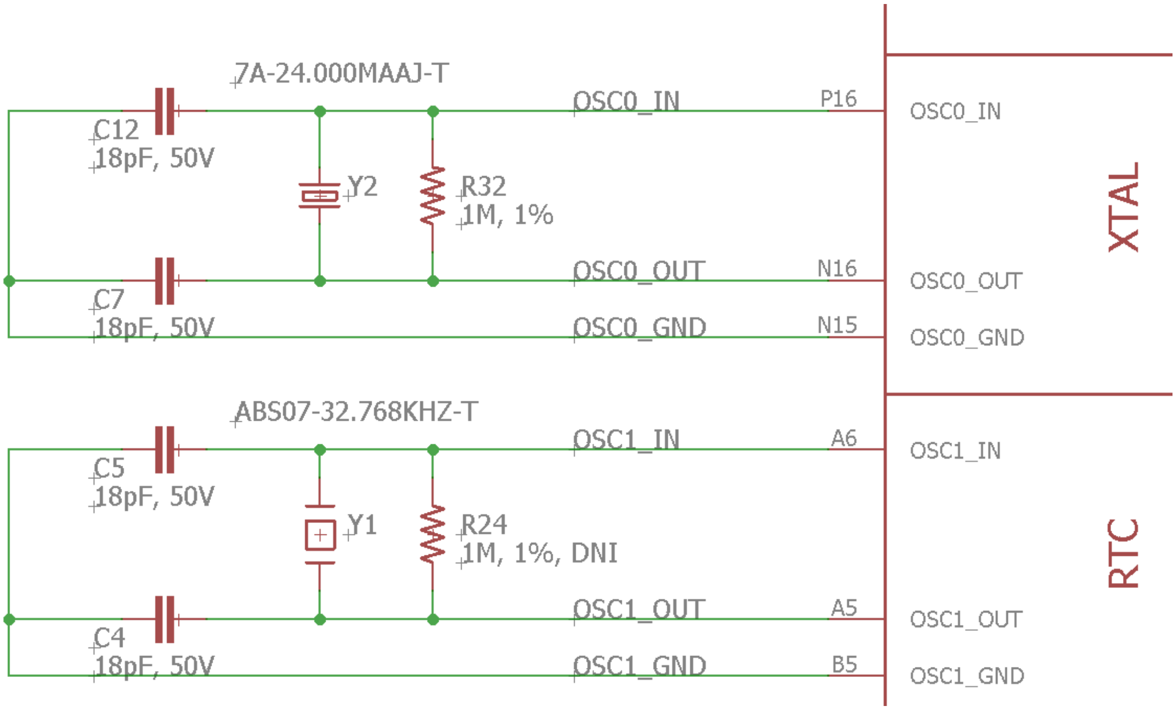

Figure 46: OSD3358-SM-RED clock circuit

Figure 46 shows the circuit for OSC0 and OSC1 clock inputs of OSD335x-SM in the OSD3358-SM-RED Board. More information on the inputs and circuit design can be found in AM335x TRM (http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf), AM335x datasheet (http://www.ti.com/lit/ds/sprs717j/sprs717j.pdf) and Reference Design Clock circuitry article (https://octavosystems.com/app_notes/osd335x-design-tutorial/bare-minimum-boot/clock-circuitry/).

NOTE: The version of the PMIC used within the OSD335x-SM is not capable of supporting RTC-only low power mode. However, the internal RTC LDO is still used to power the RTC module by pulling RTC_KALDO_ENN (M16) to GND through a 10K resistor.

4.7 Boot configuration

The ROM code checks the SYSBOOT configuration pins to communicate information to the processor such as, the OSC0 clock frequency, system boot source and boot sequence of the board. The boot configuration functionality is multiplexed with the LCD interface pins of the processor. The SYSBOOT pins are latched on the rising edge of PWRNONRSTN signal. After boot, the pins can be used for LCD interfacing or other multiplexed functions. Table 26-7 of the TRM (http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf) describes the function of each boot configuration pin. For the OSD3358-SM-RED Platform, the following boot configuration was chosen.

Table 4: OSD3358-SM-RED sysboot pin functions

Pins	Value	Function
SYSBOOT[15:0]	0x401C	
SYSBOOT[15:14]	01b	Input crystal frequency of OSC0 input is 24MHz
SYSBOOT[13:12]	00b	Reserved
SYSBOOT[11:6]	000000b	Don’t care for ROM code for the OSD3358-SM-RED boot modes
SYSBOOT[5]	0b	CLKOUT1 disabled
SYSBOOT[4:0]	See Table 5	

SYSBOOT[4:0] controls the boot sequence. The OSD3358-SM-RED has the ability to select multiple boot sequences. The boot button S3 can be used to modify SYSBOOT[2] to change the boot sequence of the processor. The boot button S3 must be pressed before power is applied so that the correct value can be latched during power up. The boot sequence changes are described in the following table allowing the processor to boot from the SD card slot even when there is a valid image in the eMMC.

Table 5: OSD3358-SM-RED boot order

System change	Default	Boot button held down
SYSBOOT[4:0]	11100b	11000b
Boot sequence priority	• MMC1 • MMC0 • UART0 • USB0	• SPI0 • MMC0 • USB0 • UART0

Figure 47 shows the boot configuration circuit in the OSD3358-SM-RED Board schematics.

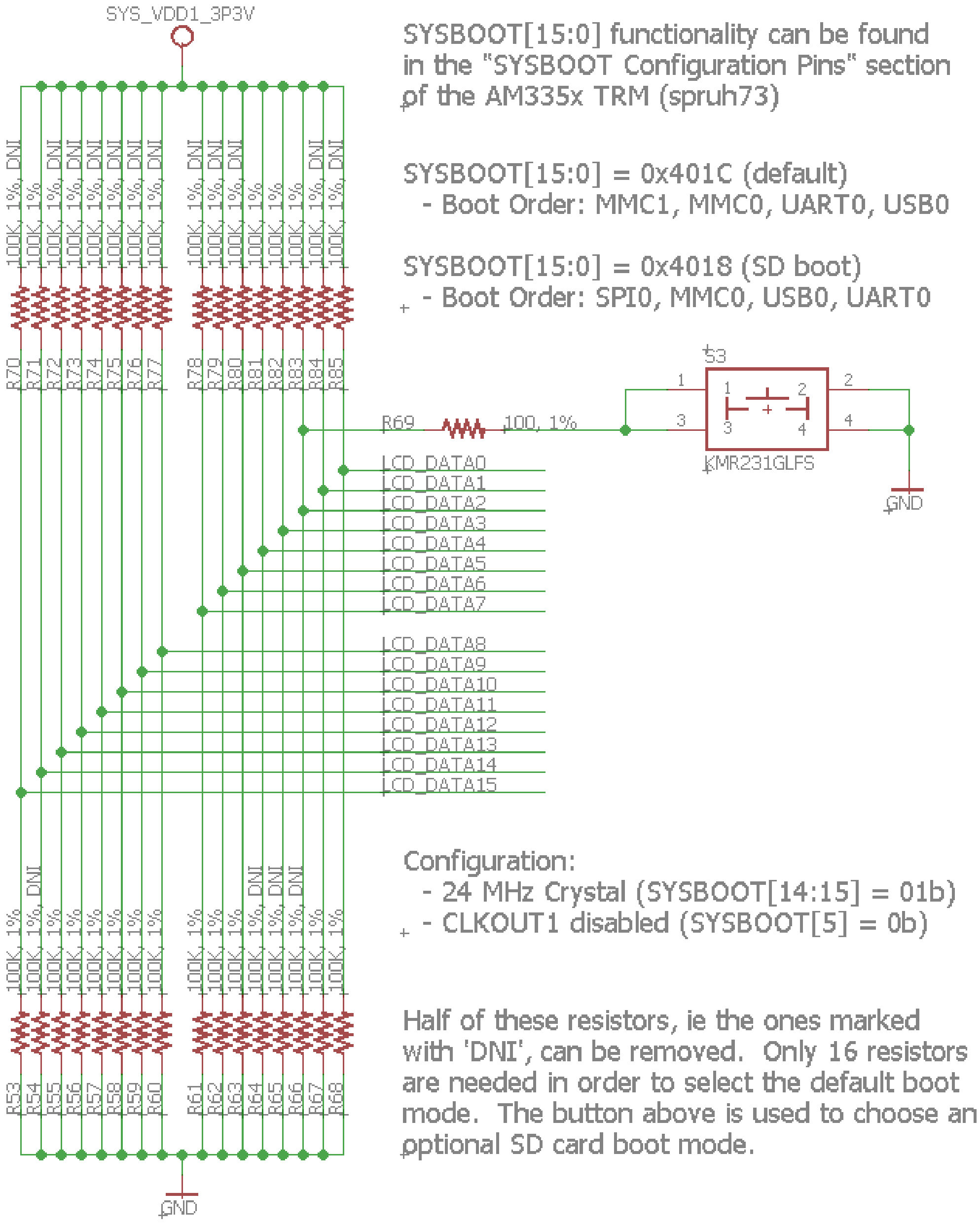

Figure 47: OSD3358-SM-RED boot configuration

4.8 UART0 interface

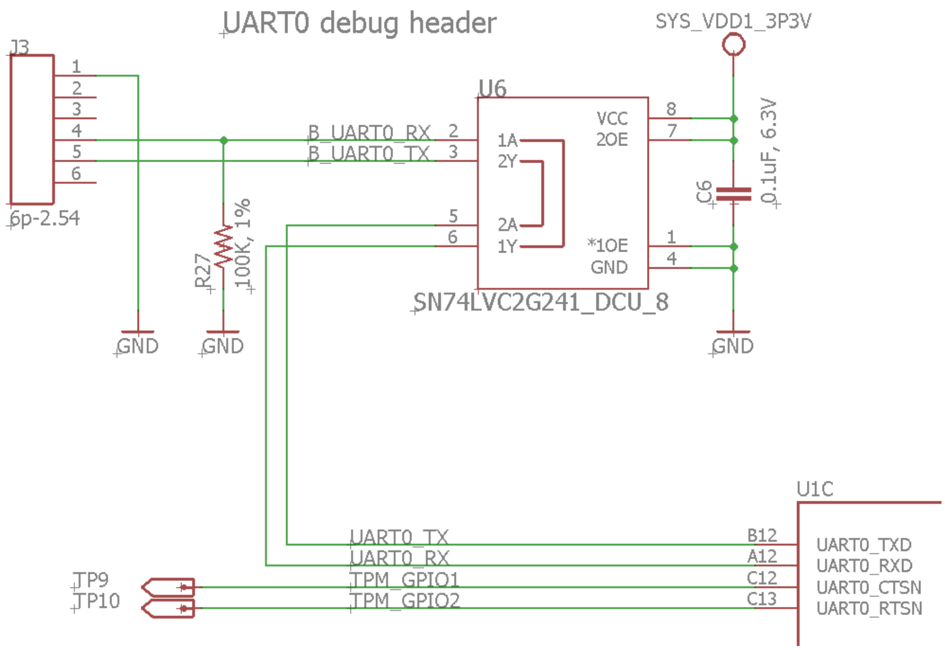

Figure 48: OSD3358-SM-RED UART interface

Figure 48 shows UART0 interface of OSD3358-512M-BSM connected to header J3 on the board through a dual buffer. This port can be used for debug by monitoring boot messages the processor sends through the UART0 interface. A USB to TTL adapter may be required to communicate to the processor through this port using a laptop/computer.

4.9 eMMC

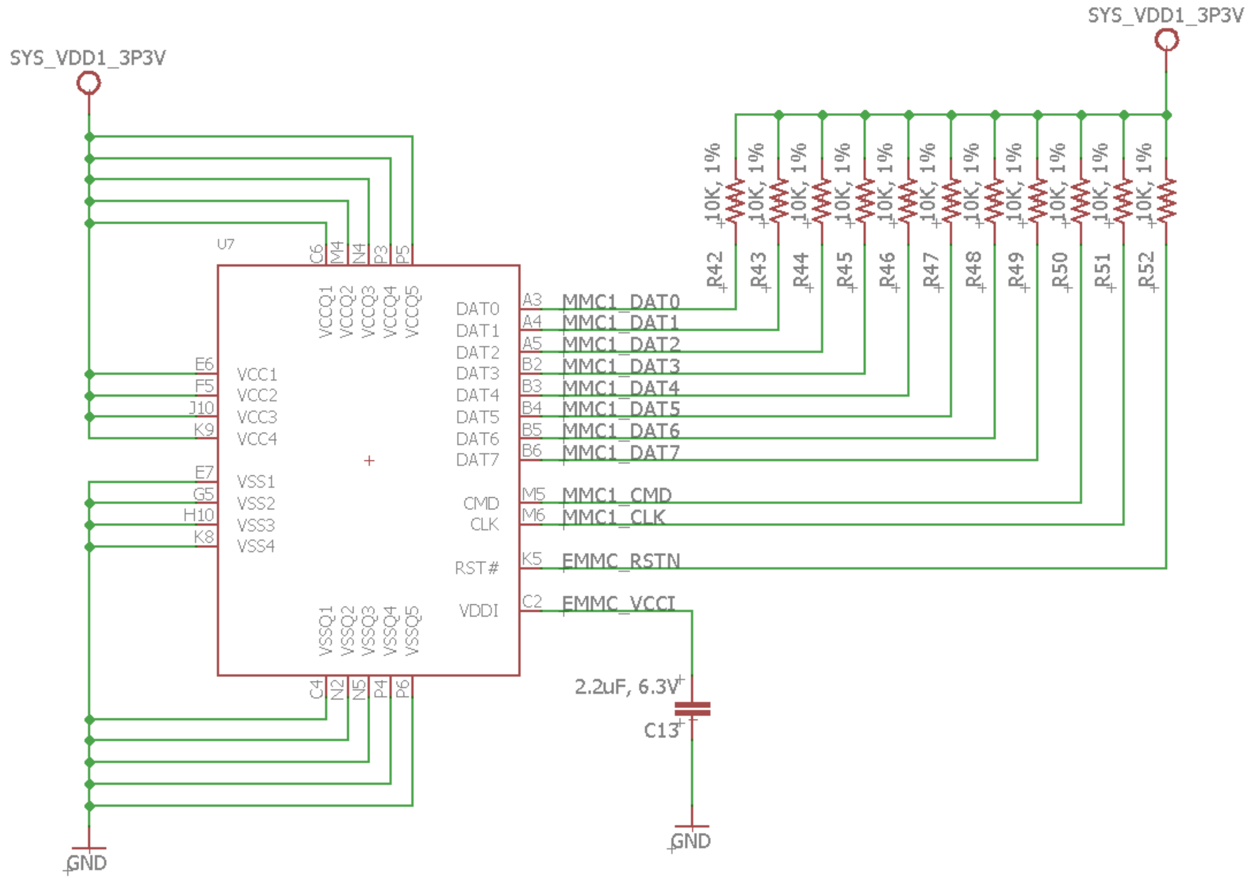

Figure 49: OSD3358-SM-RED eMMC circuit

Figure 49 shows eMMC U7 connected to the processor through MMC1 interface. As discussed in section 3.5, eMMC is the primary boot source for the processor. The Linux kernel and file system needed to boot Linux are stored in here. The ROM code reads the second stage bootloader in the eMMC and passes control to it. The ROM only supports 4 bit mode. But, after initial boot, the eMMC can be interfaced in 8 bit mode. MMC1_CLK is the clock input and MMC1_CMD is the command input to the eMMC. Pull up resistors are used on all the signals to increase rise time on the signals and compensate for any capacitance on the board. More information on booting of the board and board software can be found in Section 5.

4.10 MicroSD card slot

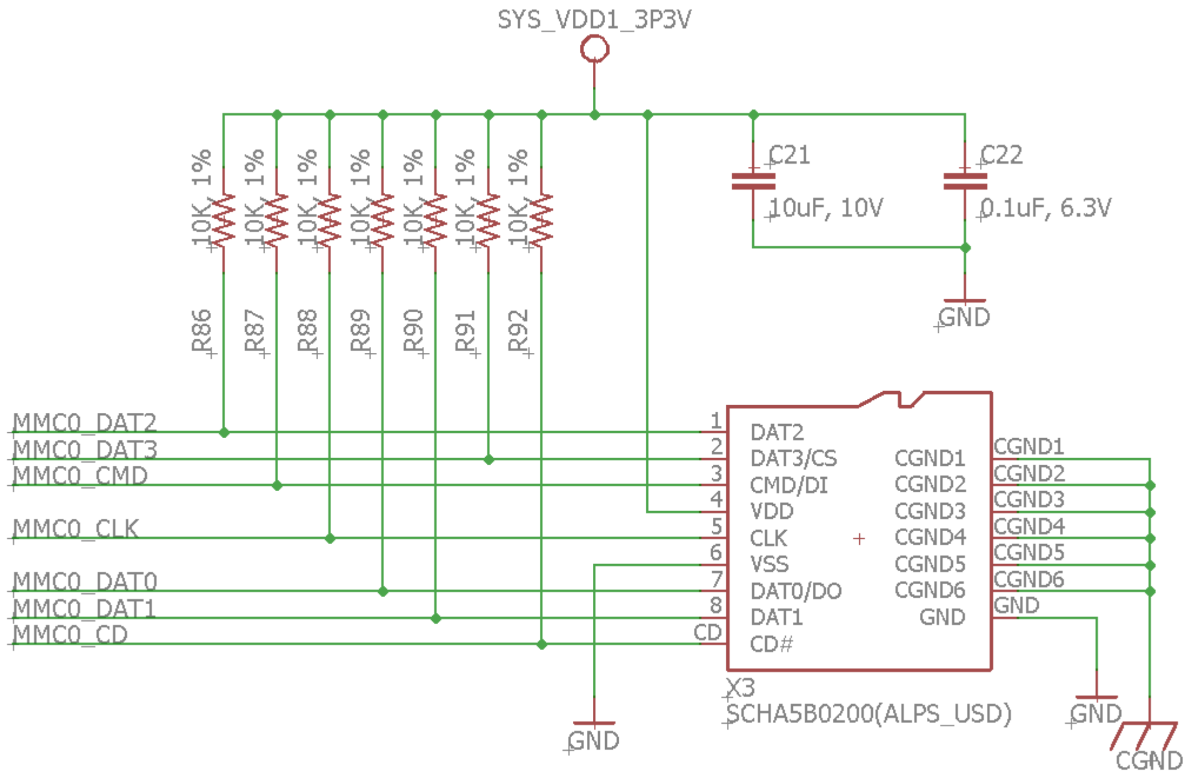

Figure 50: OSD3358-SM-RED micro-SD card slot circuit

Figure 50 shows the MicroSD card slot X3 connected to the processor on MMC0 interface on the OSD3358-SM Evaluation Board schematic. This slot can be used to boot the board from a microSD card, to flash the on board eMMC or as a normal storage device. Unlike MMC1, this is a 4 bit interface which can cause it to be slower than the eMMC interface. Similar to the MMC1 interface, the SD card interface also has a CLK and a CMD line. The MMC0_CD is a card detect signal that is pulled down when a microSD card is inserted and so, the processor is aware of the card when it is. Pull up resistors are used here as well to increase rise times and compensate for board capacitance. More information on booting of the board and board software can be found in Section 5.

4.11 JTAG

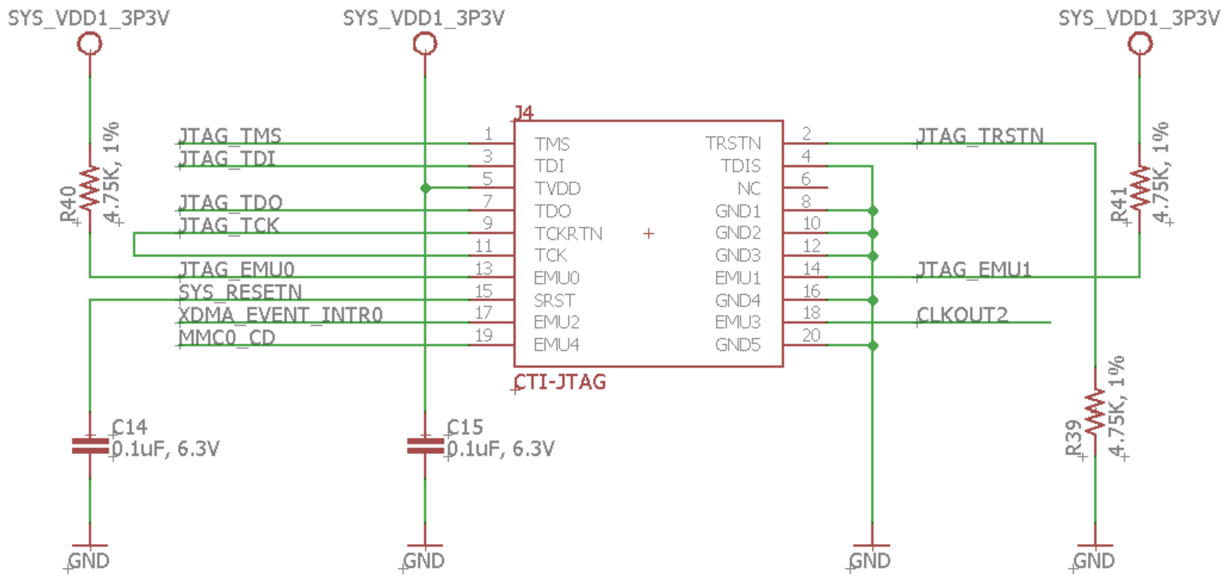

Figure 51: OSD3358-SM-RED JTAG circuit

A JTAG header footprint J4 is provided on the board to facilitate software development and debug using JTAG emulators. More information on how to use this is provided in Section 3.7.

4.12 ADC

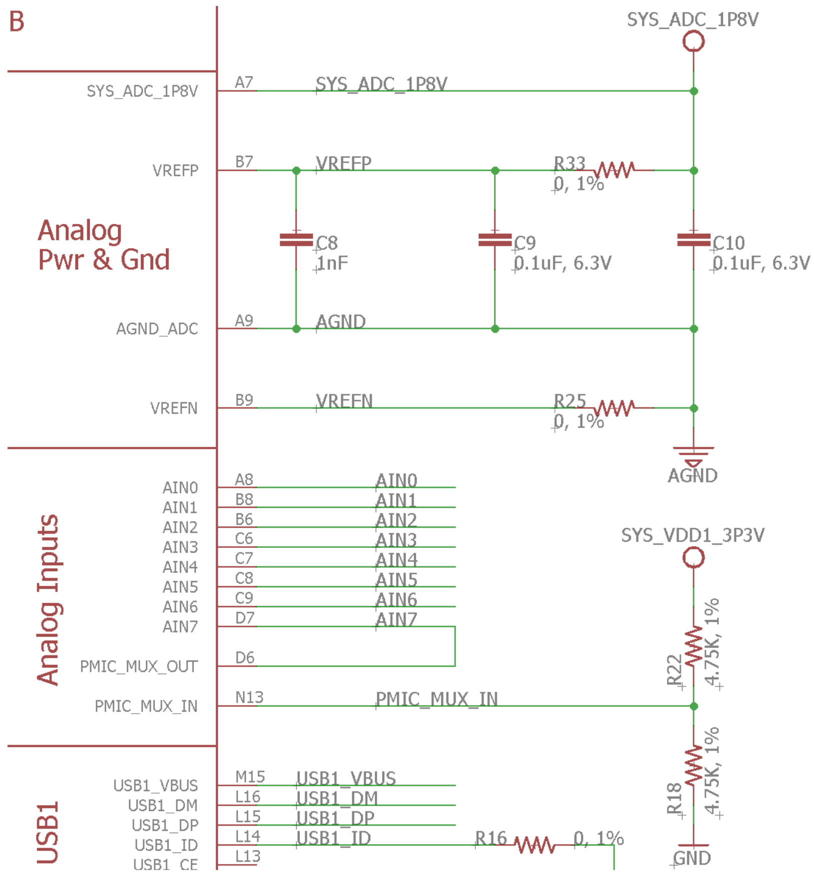

Figure 52: OSD3358-SM-RED ADC circuit

Figure 52 shows ADC circuit on the OSD3358-SM-RED platform schematics. On the RED Board, the positive reference voltage VREFP of the ADC is connected to SYS_ADC_1P8V and the negative reference voltage VREFN of the ADC is connected to AGND (Analog GND). This provides a 1.8V range for the 12-bit ADC module in the OSD335x-SM. The external resistors and capacitors are used to filter the noise on the reference voltages.

Of the total 8 analog inputs (AIN0 – AIN7), seven (AIN0 – AIN6) are brought out to expansion headers on the board and can be used. AIN7 is connected to PMIC_MUX_OUT to monitor the internal voltages of the PMIC. The PMIC has an analog multiplexer that allows critical voltages of the system to be monitored. An additional PMIC_MUX_IN input can be used to input any critical system level voltage that needs to be monitored. The PMIC's I2C interface can be used to select the voltage that needs to be monitored and AIN7 can be used to read the voltage. More information on the analog multiplexer can be sound in the PMIC datasheet (http://www.ti.com/lit/ds/symlink/tps65217.pdf).

4.13 Evaluation board USB client

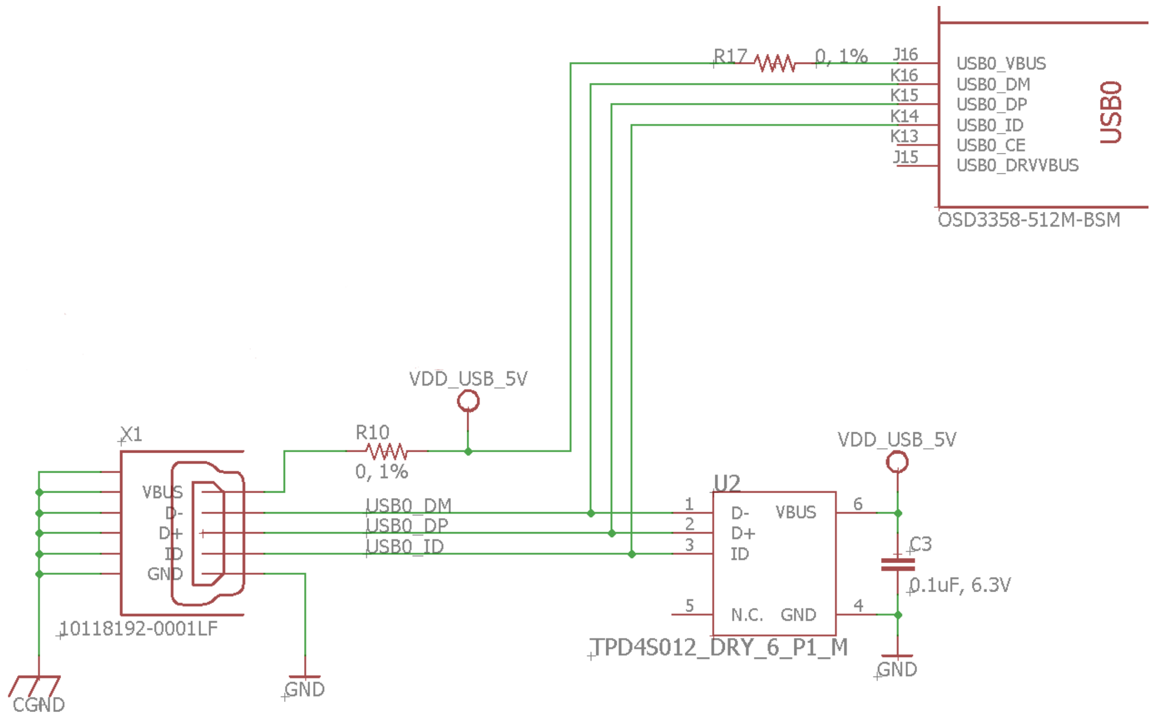

Figure 53: OSD3358-SM-RED USB client circuit

Figure 53 shows USB client port X1 connected to USB0 interface of the processor in the OSD3358-SM-RED Board schematic. As described in Section 2, this port can be used to power the board through a laptop/computer’s USB port. A secure shell program can be used to get to the Linux command prompt from a laptop/computer. U2 provides ESD protection for the USB signals.

4.14 4 port Evaluation board USB host

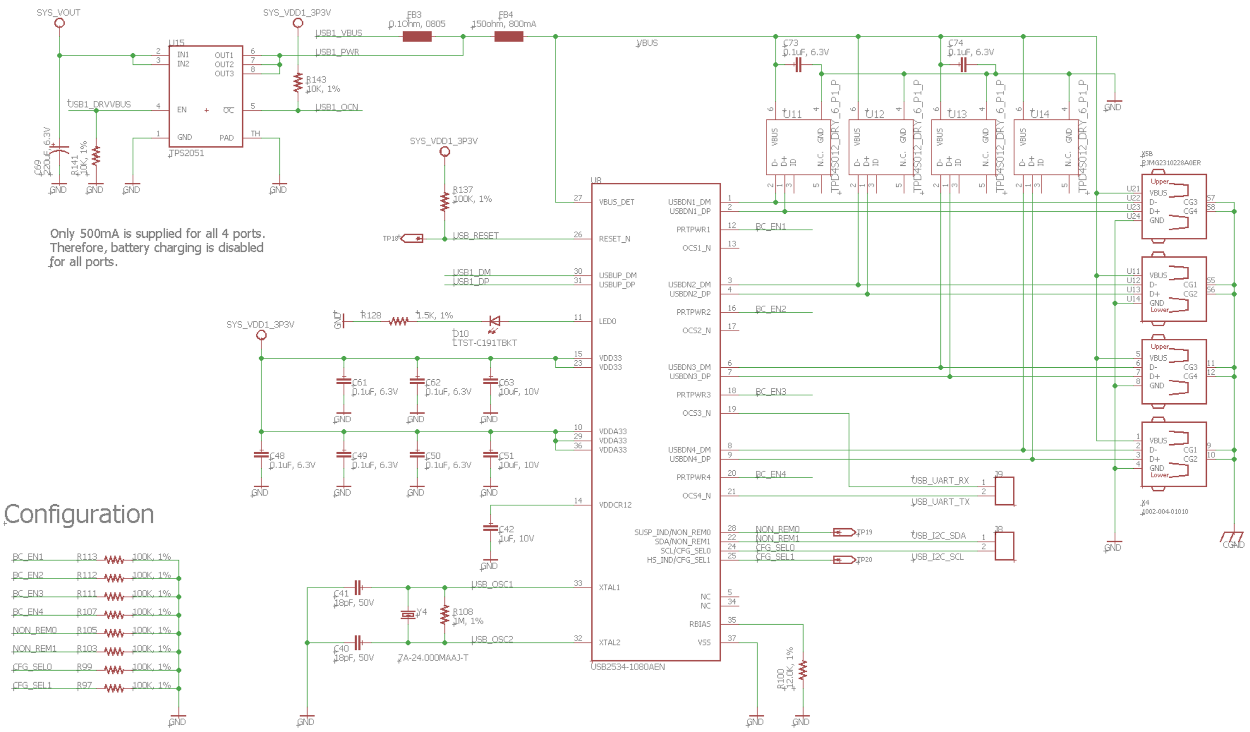

Figure 54: OSD3358-SM-RED USB host hub circuit

Figure 54 shows the 4 port USB hub U8, 4 USB ports part of X4 and X5 connectors and a power distribution switch U15 that combine to form the USB host interface for the OSD3358-SM-RED Board. The USB hub chip converts USB1 interface of the processor into 4 independent USB connection ports each of which have ESD protection and can be used to connect USB peripherals simultaneously. The hub chip has UART and I2C interfaces which can be used to configure it. Battery charging is disabled on all the ports because of the 500mA limit that can be supplied from U15. A 24 MHz crystal is used to supply clock to the hub controller USB2534 (U8) that is powered by SYS_VDD1_3P3V. The four ports can be used simultaneously, however the combination of the attached peripherals cannot exceed the 500mA current draw limit.

4.15 Ethernet

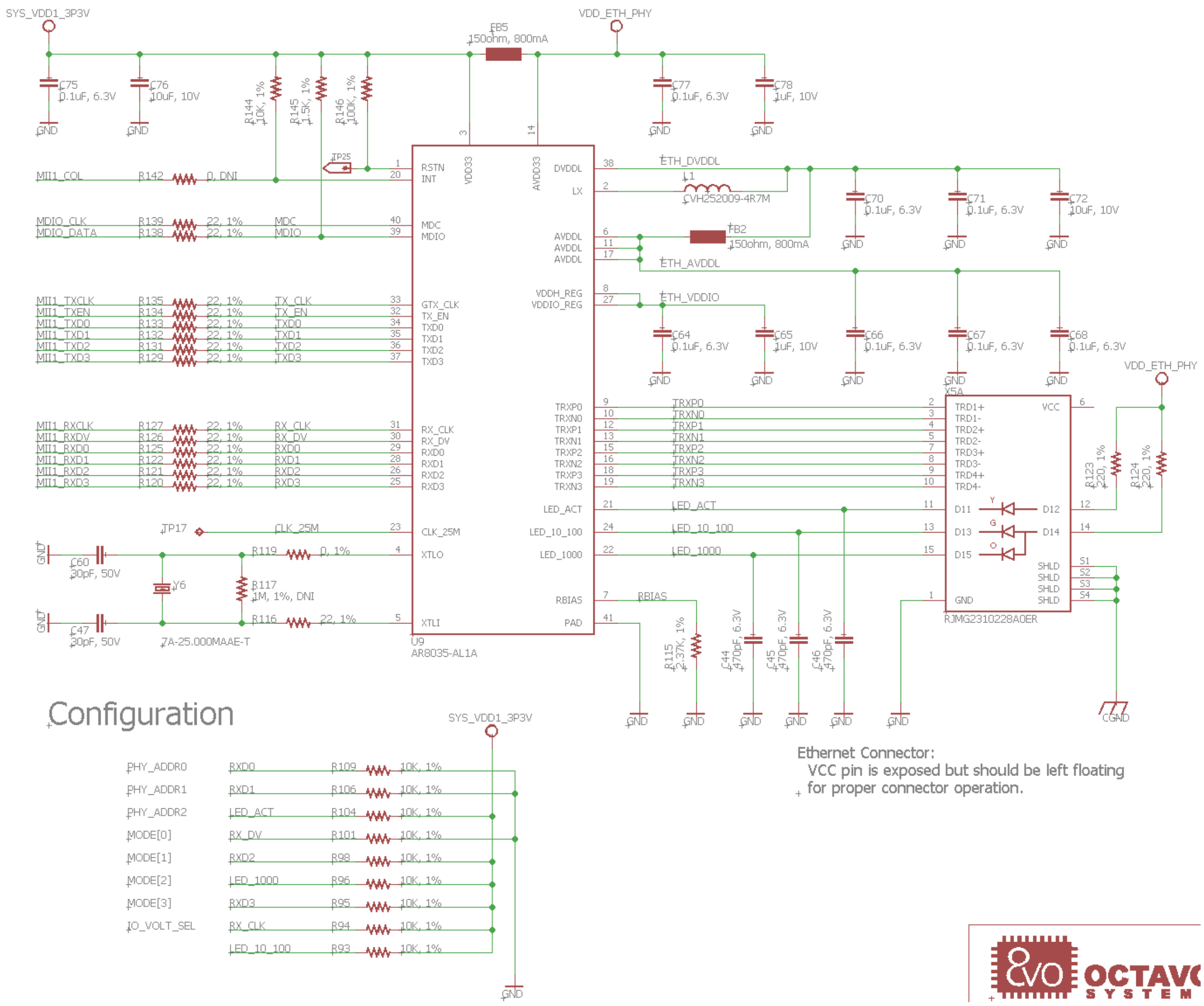

Figure 55: OSD3358-SM-RED ethernet circuit

The OSD3358-SM-RED Platform is capable of Gigabit ethernet with the AR8035 Gigabit Ethernet PHY connected to OSD335x-SM via RGMII1 port. The PHY interface with the processor consists of 4 data lines for transmission and reception along with clock and control lines for each. MDIO interface is used by the processor for instruction/configuration of the PHY. A 25MHz crystal oscillator Y6 supplies clock to the PHY. X5 contains an ethernet connector with integrated magnetics with 2 LEDs indicating mode of operation.

More information on ethernet interface can be found at the following resources:

1. AR8035-AL1A Datasheet (https://media.digikey.com/pdf/Data%20Sheets/CSR%20PDFs/AR8035_DS_(Atheros)_Mar2011.pdf)

2. RJMG2310228A0ER Datasheet (https://media.digikey.com/pdf/Data%20Sheets/Amphenol%20PDFs/RJMG2310228A0ER_Dwg.pdf)

4.16 HDMI interface

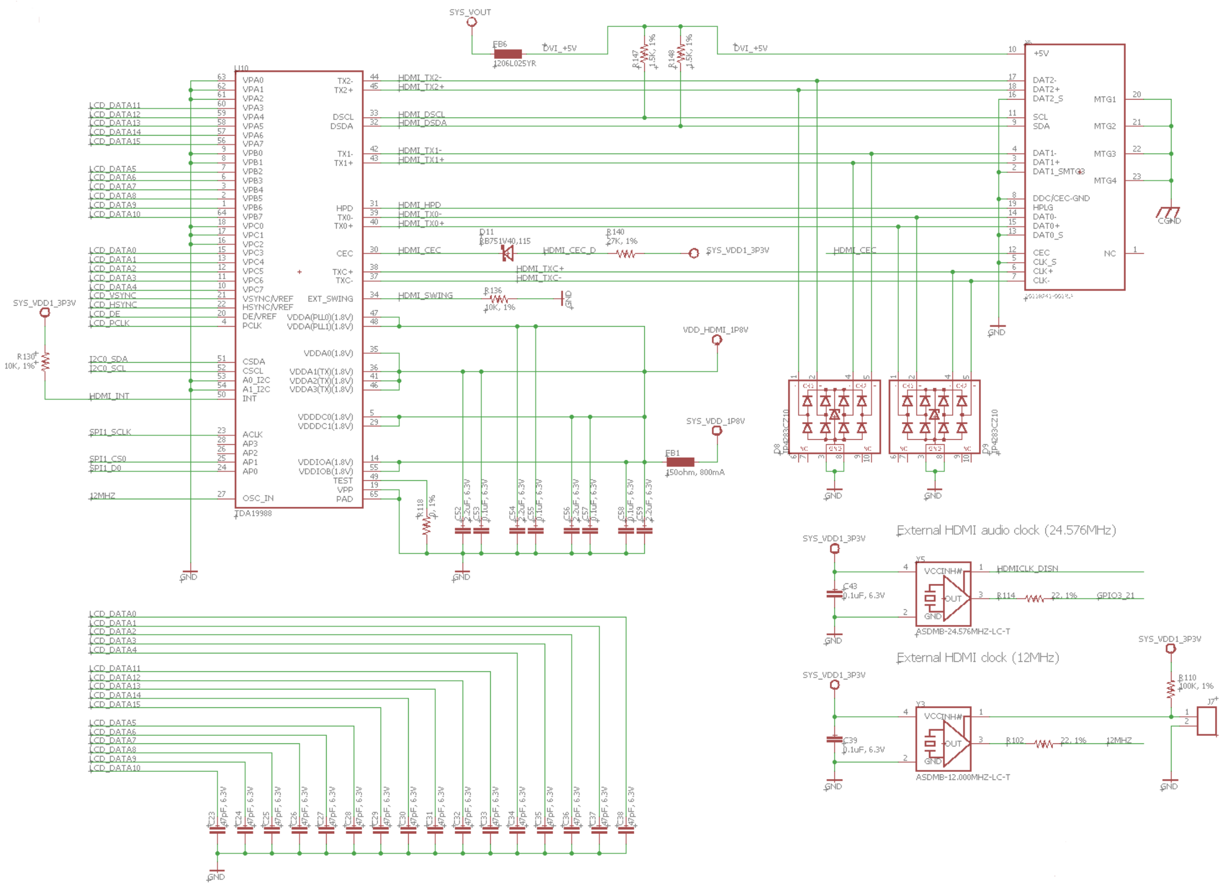

Figure 56: OSD3358-SM-RED HDMI circuit

The OSD3358-SM-RED Platform will be able to drive an HDMI monitor with on board TDA1998 HDMI framer U10 and micro-HDMI connector X6. The framer converts the processor 16-bit LCD interface to drive HDMI monitors. The framer is powered by SYS_VDD1_3P3V and can be configured using the I2C0 interface. A 12 MHz oscillator Y3 supplies the main clock input to the framer while the 24.576MHz crystal provides audio clock. The interface between the micro-HDMI connector X6 and the HDMI framer U10 is ESD protected using D8 and D9.

4.17 9-Axis IMU

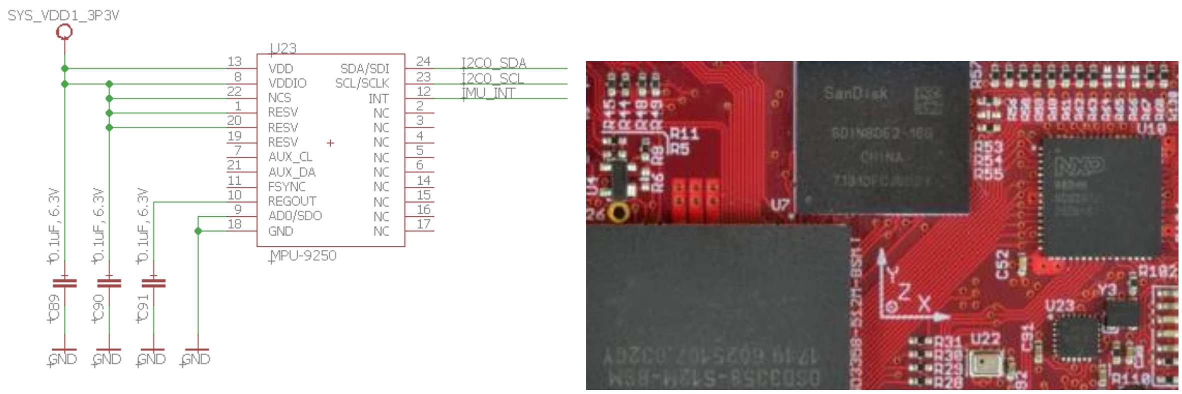

Figure 57: OSD3358-SM-RED IMU circuit

Figure 57 shows a 9-Axis IMU U23 powered by SYS_VDD1_3P3V rail and connected to the processor on the I2C0 interface. This sensor sits on the horizontal axis of the board that cuts the board in half. The chip is located at coordinates (3485 mil, 1075 mil) from the bottom left of the board oriented in the same way as shown in Figure 57. It is comprised of a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer making the device a 9-axis motion tracking sensor. Each of the axes have a 16-bit ADC to digitize the sensor outputs.

4.18 Temperature sensor hub

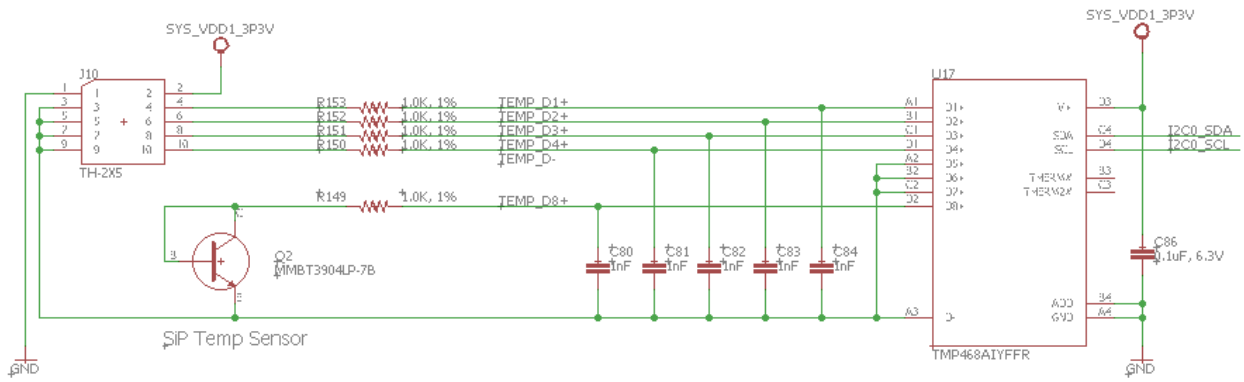

Figure 58: OSD3358-SM-RED temperature sensor hub circuit

The OSD3358-SM-RED Platform also has a temperature sensor hub (shown in Figure 58) sitting on the I2C0 interface of the processor. The hub has 8 sensor inputs, out of which 4 (D1 – D4) are brought out to header J10. An additional input D8 is used to measure temperature on board near the SiP using Q2. The 3 remaining inputs are not routed out due to the size of the package of the sensor hub. The hub is powered by SYS_VDD1_3P3V and can be used to monitor thermal behavior across multiple locations of the system. More information setting up remote measurements can be found in the temp sensor datasheet (http://www.ti.com/lit/ds/symlink/tmp468.pdf). The sensor has a resolution of 13 bits (0.0625C) and can measure temperatures ranging from -40C to 125C.

4.19 Barometer

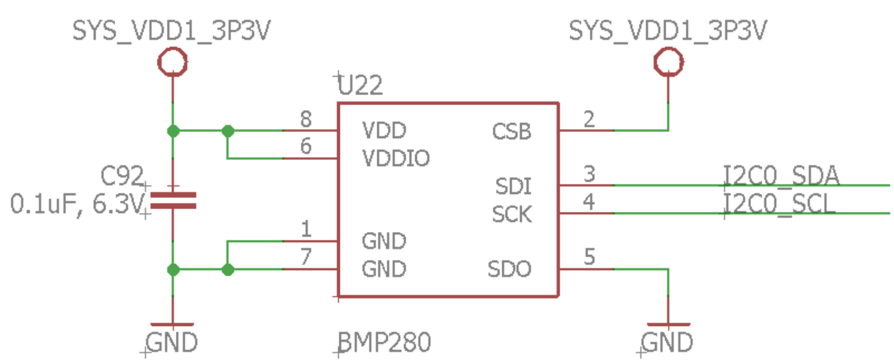

Figure 59: OSD3358-SM-RED pressure sensor circuit

Figure 59 shows BMP280 (U22) connected to the processor on the I2C0 interface. This is a barometric pressure sensor with a range of 300 – 1100 hPa. Along with an additional temperature sensor, it has noise filtering circuitry that can be used to reject short term disturbances. The sensor can measure with a 1m altitude accuracy.

4.20 Trusted Platform Module (TPM)

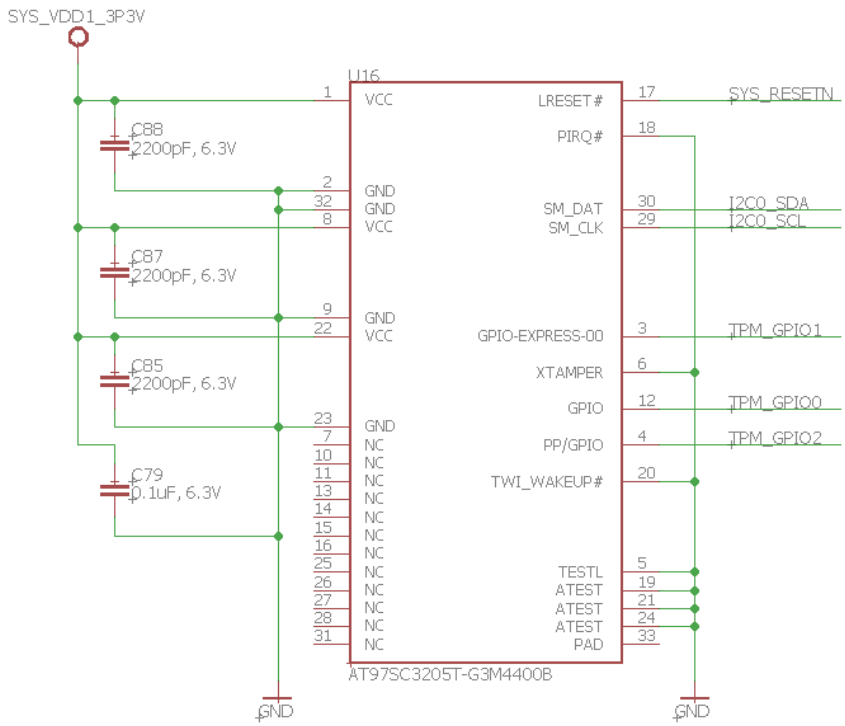

Figure 60: OSD3358-SM-RED TPM circuit

A Trusted Platform Module (TPM) is present on I2C0 interface of the processor. It is powered by SYS_VDD1_3P3V and can be used to develop security applications for the processor. Coupled with the NOR Flash device on SPI0 bus, the TPM can be used to implement secure boot and other secure application environments. The board does not come with any security applications out of the box.

4.21 EEPROM

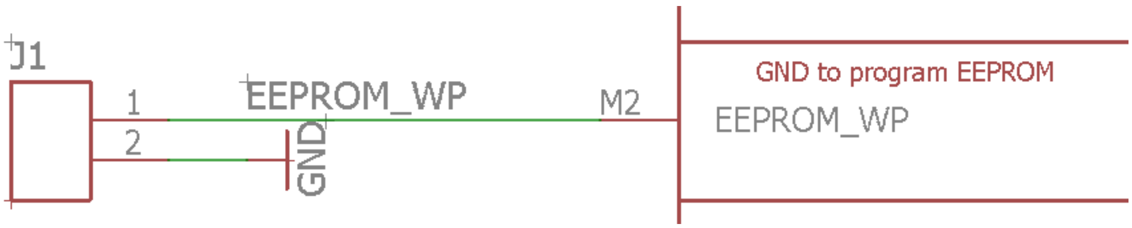

Figure 61: OSD3358-SM-RED EEPROM write protect circuit

The OSD335x-SM device has an EEPROM integrated inside the OSD335x-SM on the I2C0 line. EEPROM's write protect pin is brought out as a ball M2 (Shown in Figure 61) and can be used to write to the EEPROM. By default, the EEPROM holds some board identification and device configuration information. The board specific contents are given below:

Name	Description	Size (bytes)	Contents
Header	Header Value designates section is valid	9	0xAA, 0x55, 0x33, 0xEE
Board Name	Board name	8	Name in ASCII: A335BNLT

The device specific information and its format can be found in the OSD3358-SM datasheet(https://octavosystems.com/docs/osd335x-sm-datasheet/)

4.22 Header descriptions

The OSD3358-SM-RED board has 2 main expansion headers J5 and J6 that have the same pinout as Bealgebone Black headers P8 and P9 if the jumper J11 is unpopulated. The pinout is shown in the Figure 62. Detailed information regarding the pinout and multiplexing of each pin is described in section 'Connectors' of the Bealglebone Black reference manual.

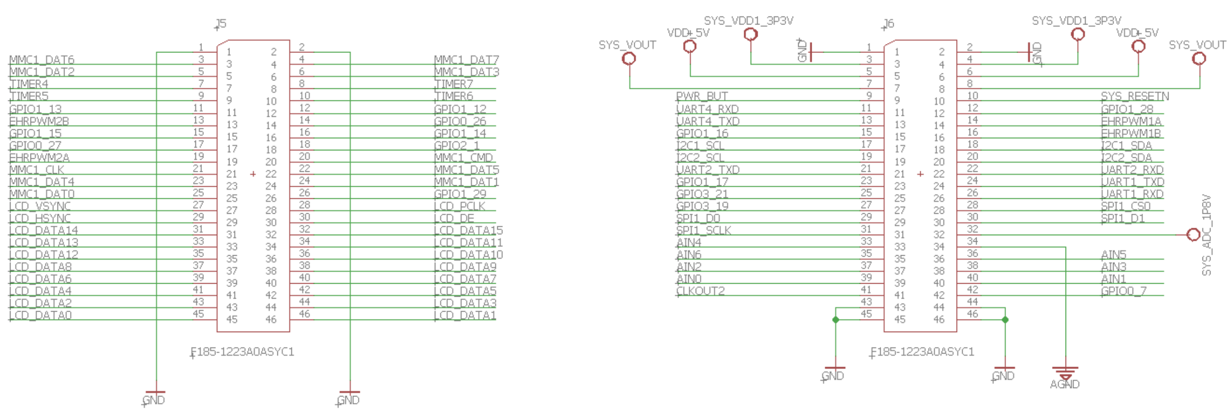

Figure 62: OSD3358-SM-RED headers

4.23 On board connectors and jumpers

The following table details the functions of all the headers and jumpers.

Table 6: Connector/jumper descriptions

Designation	Function	
J1	Populating this jumper grounds the EEPROM write protect	
J2	Populating this jumper grounds the PMIC_NRESET pin of the OSD3358-SM	
J3	**Function**	UART0 Header
	Pin 1	GND
	Pin 2	NC
	Pin 3	NC
	Pin 4	UART0_RX
	Pin 5	UART0_TX
	Pin 6	NC
J4	Footprint for JTAG header is provided. Populate with the recommended part in section 3.7	
J5	Same pin map as header P8 of the Beaglebone Black	
J6	Same pin map as header P9 of the Beaglebone Black	
J7	Populating this jumper disables the 12MHz external HDMI clock from Y3	
J8	**Function**	I2C communication port for the USB port hub U8
	Pin 1	SDA pin of U8
	Pin 2	SCL pin of U8
J9	**Function**	UART communication port for the USB port hub U8
	Pin 1	UART_RX pin of U8
	Pin 2	UART_TX pin of U8
J10	**Function**	Temp sensor channel header
	Pin 1	GND
	Pin 2	SYS_VDD1_3P3V
	Pin 3	D- pin of U17
	Pin 4	D+ of channel 1
	Pin 5	D- pin of U17
	Pin 6	D+ of channel 2
	Pin 7	D- pin of U17
	Pin 8	D+ of channel 3
	Pin 9	D- pin of U17
	Pin 10	D+ of channel 4
J11	Populating this jumper enables the NOR flash SPI communication with the OSD3358-SM *NOTE: populating this header breaks BeagleBoard cape compatibility of the OSD3358-SM-RED board	

4.24 Test points

There are several test points provided on the OSD3358-SM board in order to facilitate debug. The following table provides a description of the available test points and their signal names.

Table 7: OSD3358-SM-RED test points description

Designation/label	Signal name
TP1	VDDS_DDR
TP2	VDD_CORE
TP3	VDD_MPU
TP4	VDDS_PLL
TP5 (VIN_BAT)	VIN_BAT (PMIC Battery voltage input)
TP6 (BAT_SENSE)	BAT_SENSE (PMIC Battery sense input)
TP7 (TEMP_SENSE)	TEMP_SENSE (PMIC Temperature sense input)
TP8	GND
TP9	TPM_GPIO1
TP10	TPM_GPIO2
TP11	SYS_VOUT
TP12	SYS_VDD1_3P3V
TP13	SYS_VDD2_3P3V
TP14	SYS_RTC_1P8V
TP15	SYS_VDD_1P8V
TP16	SYS_VDD3_3P3V
TP17	CLK_25M (Ethernet PHY)
TP18	USB_RESET (USB port hub)
TP19	NON_REM0 (USB port hub)
TP20	CFG_SEL1 (USB port hub)
TP21	PWR_EN
TP22	LDO_PGOOD
TP23	PMIC_NWAKEUP
TP24	PMIC_INT
TP25	RSTN (Ethernet PHY)
TP26	SYS_RESETN
TP27	SPI0_SCLK
TP28	SPI0_D1(MOSI)
TP29	SPI0_D0(MISO)
TP30	SPI0_CS0

5 Board images and software features

The OSD3358-SM-RED board is loaded with a Beagleboard compatible Linux image out of the box. This pre-installed Linux image on the board is a modified version of Debian LXQT image available on Beagleboard website (https://beagleboard.org/latest-images). Modifications to the default image were made to customize the image to work with the hardware on the board. The following changes were made to the default Linux image:

1. A new device tree specific to the OSD335x-SM-RED platform is used to make Linux aware of the board specific hardware
2. The Roboticscape library (http://www.strawsondesign.com/#!manual-install) that comes installed by default is modified to work with the sensors on board. Note that the present version of the Roboticscape library is not compatible with the modified library that comes installed out of the box on the RED board.

5.1 Boot up

The board goes through multiple stages during boot to a Linux environment. These stages are described below:

1. TI Boot ROM: The processor contains a boot ROM that performs some peripheral configuration required for boot up and passes the control to the next stage bootloader
2. MLO: This stage sets up pin-muxing, initializes the clocks and memory, and loads the next stage bootloader
3. U-boot: This is the primary Linux bootloader. It looks for a boot source and reads the boot configuration text file (uEnv.txt) for boot arguments from the boot source. It specifies the root file system, performs additional initialization, loads and passes control to the Linux kernel. A device tree binary file (hardware specific board configuration file) specified in the configuration text file, is also passed to the kernel for additional setup.
4. Linux kernel: Decompresses and loads the kernel into memory, performs peripheral initializations and mounts the file system that contains all the Linux applications.

Useful resources:

1. Linux core U-Boot User's Guide: http://processors.wiki.ti.com/index.php/Linux_Core_U-Boot_User%27s_Guide
2. Das U-Boot home page: http://www.denx.de/wiki/U-Boot/WebHome
3. Building U-Boot, Root Filesystem and Linux Kernel: https://eewiki.net/display/linuxonarm/BeagleBone+Black

5.2 Device tree

The OSD3358-SM-RED board has a device tree file that tells the Linux kernel about the hardware on board. Derivative designs from the OSD33358-SM-RED will need to modify the device tree appropriately. A Flattened Device Tree (FTD) is a data structure that describes the hardware on the board to the Linux kernel. A compiled version, .dtb (Device Tree Binary) of this file specified in the boot configuration file uEnv.txt located in /boot/ folder communicates which peripherals are available and enabled, device drivers that need to be loaded, default pin mux settings after boot, etc. The device Tree Binary file is a compiled version of a .dts(Device Tree Source) file. A Device Tree Compiler (dtc) is necessary to convert the source file into the binary file. The device tree source files for the OSD335x devices and the OSD3358-SM-RED board are linked below. The best way to write custom device trees is by using the dtb-rebuilder and is also linked below.

Useful resources:

1. OSD3358-SM-RED device tree files: https://github.com/octavosystems/OSD335x-Device-Tree
2. Device tree usage: http://elinux.org/Device_Tree_Usage
3. Device tree reference: http://elinux.org/Device_Tree_Reference
4. Device tree rebuilder: https://github.com/RobertCNelson/dtb-rebuilder

5.2.1 Device Tree Overlays

Similar to the capes of Beaglebone Black, capes can be added to the OSD3358-SM-RED platform for enhanced functionality. The board is cape compatible with existing Bealgebone Black capes. So, capes for the Beaglebone Black can be used with the OSD3358-SM-RED board. The additional cape specific setup for Linux is done using Device Tree Overlays. BeagleBone Black compatible expansion capes also normally include an I2C EEPROM (must be on I2C2 in the address range of 0x54 to 0x57), that provides information such as the name of the cape, cape version, part number, and other relevant run-time information. The information required to load the correct Device Tree Overlay for a specific cape is read from the I2C EEPROM by the Cape Manager (http://elinux.org/Capemgr). See the BeagleBone Black SRM (https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual) "Cape Board Support" section for details. Device Tree Overlays can also be loaded (cape is enabled) or unloaded (cape is disabled) at run-time through the cape manager's /sys interface.

Useful resources:

1. Cape Manager: http://elinux.org/Capemgr
2. Using cape manager and device tree overlays: https://github.com/jadonk/validation-scripts/tree/master/test-capemgr
3. Beaglebone Black capes: http://elinux.org/Beagleboard:BeagleBone_Capes
4. Device tree overlays tutorial: https://learn.adafruit.com/introduction-to-the-beaglebone-black-device-tree/exporting-and-unexporting-an-overlay

5.3 Onboard sensor interfacing

As described in section 3 & 4, there are several sensors on OSD3358-SM-RED board. The board is equipped with a software library and examples to interface with the sensors. The library package 'Redperipherallib' is pre-installed on Linux and allows commands line access to these components on the board. The Roboticscape library (http://strawsondesign.com/#!manual-install) that comes pre-installed on the Beagleboard images was used as a base to build the Redperipherallib library. The library is located in **/home/debian/** folder of the Linux environment. The structure of the library is given below:

```
└── Redperipherallib
    ├── debian
    ├── examples
    │   ├── rc_benchmark_algebra
    │   ├── rc_calibrate_gyro
    │   ├── rc_calibrate_mag
    │   ├── rc_check_model
    │   ├── rc_cpu_freq
    │   ├── rc_kill
    │   ├── rc_test_adc
    │   ├── rc_test_algebra
    │   ├── rc_test_barometer
    │   ├── rc_test_drivers
    │   ├── rc_test_filters
    │   ├── rc_test_imu
    │   ├── rc_test_polynomial
    │   ├── rc_test_time
    │   ├── rc_test_tmp
    │   ├── rc_test_vector
    │   ├── rc_uart_loopback
    │   └── rc_version
    └── libraries
        ├── bmp280
        ├── gpio
        ├── math
        ├── mmap
        ├── mpu9250
        ├── other
        ├── pwm
        ├── serial_ports
        └── tmp468
```

Figure 63: Structure of Redperipherallib

- **debian**: Contains Debian specific configuration parameter files
- **examples**: Contains example programs that show how to use the peripherals onboard
- **libraries**: Contains libraries necessary to exercise the peripherals onboard
  - **bmp280**: Contains functions and definitions to interface with the pressure sensor
  - **gpio**: Contains functions and definitions to control the GPIOs of the board
  - **math**: Contains functions and definitions to perform mathematical operations required
  - **mmap**: Contains functions and definitions to perform memory mapping operations
  - **mpu9250**: Contains functions and definitions to interface with the IMU
  - **other**: Contains functions and definitions for additional features of the library
  - **pwm**: Contains functions and definitions to generate a PWM
  - **serial_ports**: Contains functions and definitions to perform I2C and SPI protocol communications

- **tmp468**: Contains functions and definitions to interface with the temperature sensor

5.3.1 Accessing peripherals using shell terminal

The RED peripheral library facilitates command line access to board features. The commands, options and output formats are given below.

1. *rc_benchmark_algebra*
   Function: This command performs several matrix and vector related benchmark tests and returns the time taken to perform each test.
   Options:
   -d: use default matrix size
   -s {size}: specify matrix size
   -h: Display help message
   Result:

```
root@beaglebone:~$ rc_benchmark_algebra -d
Starting
      3888us Time to make random matrix & vector
       474us Time to duplicate matrix
     10849us Time to multiply matrices
       505 MFLOPS multiplying matrices
     20217us Time to find matrix determinant
    228666us Time to invert matrix
     97589us Time to do LUP decomposition
    950841us Time to do QR decomposition
     11431us Time to solve linear system
DONE
```

2. *rc_calibrate_gyro*
   Function: This command performs calibration of the gyro sensor in the IMU
   Options: None
   Result:

```
root@beaglebone:~$ rc_calibrate_gyro
Starting calibration routine

gyro calibration file written
run rc_test_imu to check performance
```

3. *rc_calibrate_mag*
   Funciton: This command calibrates the magnetometer in the IMU
   Options: None
   Result:

```
root@beaglebone:~$ rc_calibrate_mag

This will sample the magnetometer for the next 15 seconds
Rotate the cape around in the air through as many orientations
as possible to collect sufficient data for calibration
Press ENTER to continue or anything else to quit
spin spin spin!!!

keep spinning
you're doing great
keep spinning
you're doing great
keep spinning
you're doing great


Okay Stop!
Calculating calibration constants.....

Offsets X:  71.924 Y:  -1.718 Z: -24.156
Scales  X:   1.956 Y:   0.929 Z:   2.485

magnetometer calibration file written
run rc_test_imu to check performance
```

4. *rc_check_model*
   Function: This command returns the board name that it is currently running on
   Options: None
   Results:

```
root@beaglebone:~$ rc_check_model

Currently running on a:
OCT_OSD3358-SM-RED
```

5. *rc_kill*
   Function: This command kills any existing peripheral library programs that is running onboard
   Options: None
   Results:

```
root@beaglebone:~$ rc_kill
No existing roboticscape program is running.
```

6. *rc_test_adc*
   Function: This command returns the ADC results of 7 channels of the ADC module
   Options: None
   Results:

```
root@beaglebone:~$ rc_test_adc
 adc_0: 3973
 adc_1: 3693
 adc_2: 3930
 adc_3: 2242
 adc_4: 2217
 adc_5: 3304
 adc_6: 3328
```

7. *rc_test_barometer*
   Function: This command runs a test on the barometer and returns the temperature, pressure, altitude and filtered altitude sensor values
   Options: None
   Results:

```
root@beaglebone:~$ rc_test_barometer
  temp  | pressure  | altitude | filtered |
 35.08C | 101.57kpa |  -20.04m |  -19.88m |
```

8. *rc_test_imu*
   Function: This command tests and returns the IMU measurements
   Options:
      None : print sensor values with default settings
      -r: print raw values instead of radians
      -d: print gyro in deg/s instead of radians
      -h: print this help message
   Results:

```
root@beaglebone:~$ rc_test_imu
try 'test_imu -h' to see other options
   Accel XYZ(m/s^2)  |   Gyro XYZ (rad/s)  |  Mag Field XYZ(uT)  |
Temp (C)
```

9. *rc_test_time*
   Function: This command tests the rc time functions of the peripheral library
   Options: None
   Results:

```
root@beaglebone:~$ rc_test_time

testing rc time functions
time to call rc_nanos_since_epoch: 1081ns
time to call rc_nanos_since_boot: 1102ns
time to call rc_nanos_thread_time: 1901ns
```

10. *rc_test_tmp*
    Function: This command tests the temperature sensor and returns the temperature value on channel 8
    Options: None
    Results:

```
root@beaglebone:~$ rc_test_tmp
temperature sensor read value in celsius: 28.750000
```

11. *rc_uart_loopback*
    Function: This command tests the UART protocol capabilities of the library by running a loopback test on UART0 port. UART0_TX and UART0_RX need to be connected for this test to run.
    Options: None
    Results:

```
root@beaglebone:~$ rc_uart_loopback 0
testing UART bus 0
Sending  11 bytes: Hello World
Received 11 bytes: Hello World
```

12. *rc_version*
    Function: This command returns the version of the peripheral library being run on the board
    Options: None
    Results:

```
debian@beaglebone:~$ rc_version
REDPeripherallib 0.3.4 built on Roboticscape 0.3.4
```

5.3.2 Using peripheral library to write applications

All the peripheral access supporting libraries are present in the libraries folder. The Makefile will compile a single shared object robotics_cape.so and install it to /usr/lib. It will also move all the header files to /usr/include/. To use the .so, it is enough to put in #include <robotics_cape.h>.

6 Cape support

The OSD3358-SM-RED board is capable of supporting all the capes of the Bealgebone Black. See the BeagleBone Black SRM (https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual) "Cape Board Support" section for details on capes for Beaglebone Black. The capes can be dynamically loaded/unloaded using device tree overlays.

7 Mechanicals

Figure 64 show the dimensions of the board.

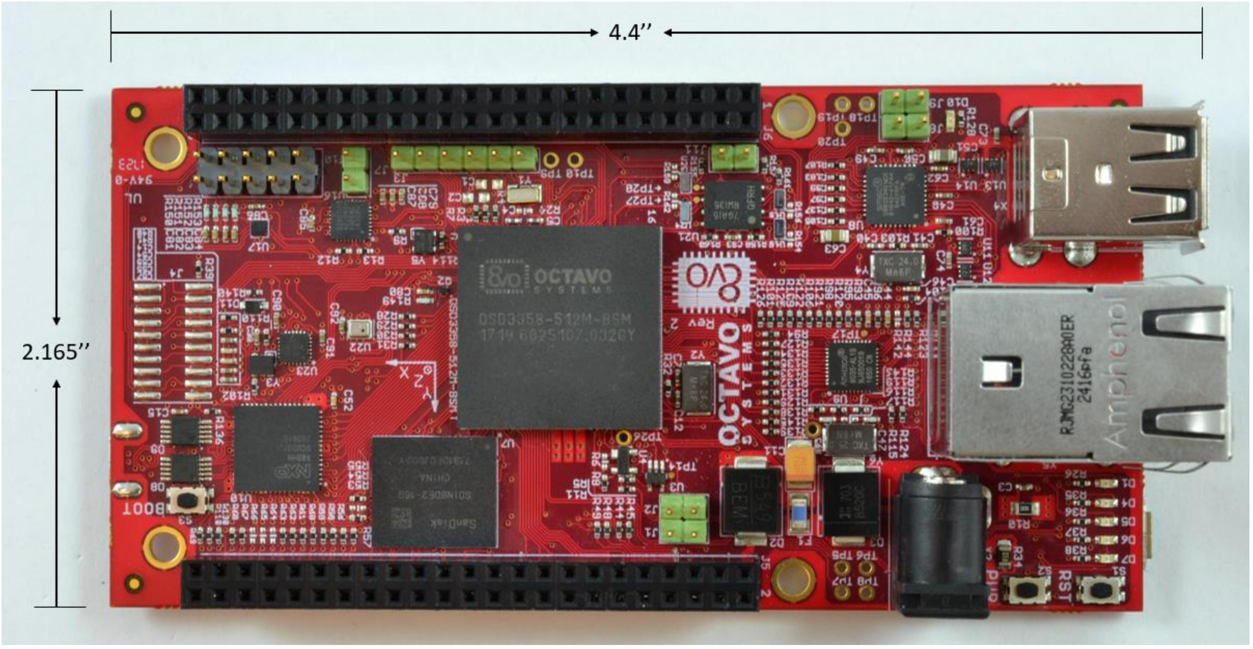

Figure 64: Dimensions of RED board

Figure 65: RED board top view

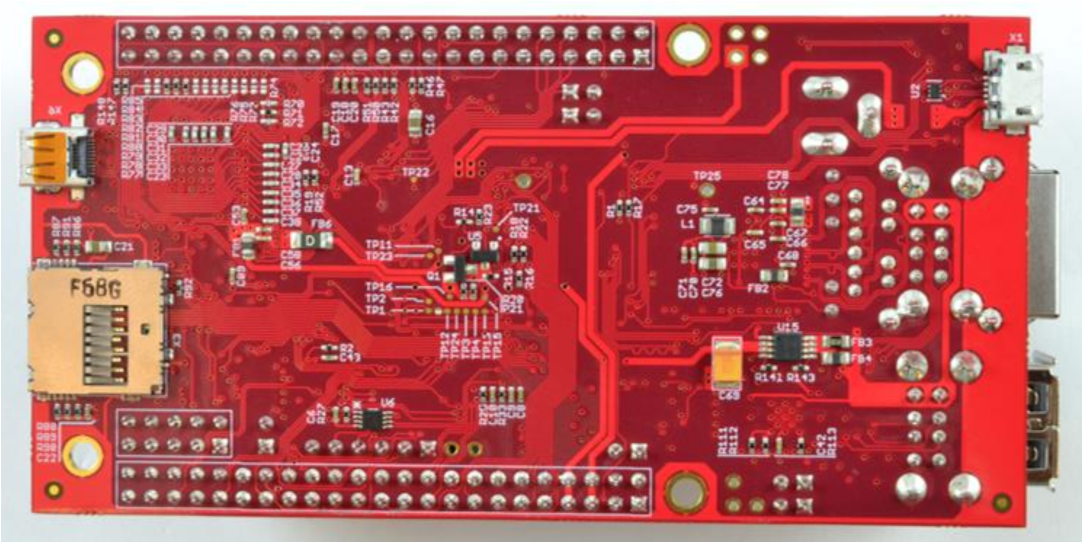

Figure 66: RED board bottom view

8 Support and Troubleshooting

A list of useful support resources is given below:

- OSD3358-SM-RED board design files:
  https://octavosystems.com/files/osd3358-sm-red-eagle-files/
- Sign up for OSD3358-SM-RED latest software updates:
  https://octavosystems.com/doc-change-signup/
- OSD3358-SM datasheet:
  https://octavosystems.com/docs/osd335x-sm-datasheet/
- OSD3358-SM RED platform getting started guide:
  https://octavosystems.com/docs/osd3358-sm-red-quick-start-guide/
- OSD3358-SM Application notes:
  a. OSD335x-SM layout guide:
  https://octavosystems.com/app_notes/osd335x-sm-layout-guide/
  b. OSD335x Family Pin Assignments Compared to AM335x:
- https://octavosystems.com/app_notes/osd335x-family-pin-assignments/
- OSD3358-SM-RED RMA request page: https://octavosystems.com/contact/

9 Reference Documents

1. OSD335x datasheet: https://octavosystems.com/docs/osd335x-sm-datasheet/
2. AM335x datasheet: http://www.ti.com/lit/ds/sprs717j/sprs717j.pdf
3. AM335x Technical Reference Manual: http://www.ti.com/lit/ds/sprs717j/sprs717j.pdf
4. TPS65217C Datasheet: http://www.ti.com/lit/ds/symlink/tps65217.pdf
5. Powering AM335x with TPS65217x: http://www.ti.com/lit/ug/slvu551i/slvu551i.pdf
6. TL5209 LDO Datasheet: http://www.ti.com/lit/ds/symlink/tl5209.pdf
7. 4KB EEPROM Datasheet: http://www.microchip.com/wwwproducts/en/24LC32A
8. Compatible WiFi adapters:
   http://www.elinux.org/Beagleboard:BeagleBoneBlack#WIFI_Adapters
9. PUTTY tool: http://www.putty.org/
10. Beaglebone Black expansion headers:
    http://elinux.org/Beagleboard:Cape_Expansion_Headers
11. HDMI Framer TDA199988 Datasheet:
    https://media.digikey.com/pdf/Data%20Sheets/NXP%20PDFs/TDA19988_DS_21_july_2011.pdf
12. Ethernet PHY AR8035 Datasheet:
    https://media.digikey.com/pdf/Data%20Sheets/CSR%20PDFs/AR8035_DS_(Atheros)_Mar2011.pdf
13. USB port hub USB2534-1080AEN datasheet:
    http://ww1.microchip.com/downloads/en/DeviceDoc/00001713A.pdf
14. IMU MPU9250 datasheet: https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf

15. Temperature sensor TMP468 datasheet: http://www.ti.com/lit/ds/symlink/tmp468.pdf
16. Pressure sensor BMP280 datasheet: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280_DS001-11.pdf
17. TPM AT97SC3205T datasheet: http://www.atmel.com/Images/Atmel-8883S-TPM-AT97SC3205T-Datasheet-Summary.pdf
18. NOR flash S25FL127S datasheet: http://www.cypress.com/file/177961/download
19. OSD335x Power application note: https://octavosystems.com/app_notes/osd335x-power-application-note/
20. I2C pullup calculation: http://www.ti.com/lit/an/slva689/slva689.pdf
21. OSD335x Design Tutorial series - OSD335x Reset Circuitry article https://octavosystems.com/app_notes/osd335x-design-tutorial/bare-minimum-boot/reset-circuitry/
22. OSD335x Reference Design Lesson 1 Power Circuitry Part 4: Clamping Circuit: https://octavosystems.com/app_notes/osd335x-reference-design-lesson-1-power-circuitry-part-4-clamping-circuit/
23. Reference Design Tutorial series - Clock circuitry article: https://octavosystems.com/app_notes/osd335x-design-tutorial/bare-minimum-boot/clock-circuitry/
24. RJMG2310228A0ER Datasheet (https://media.digikey.com/pdf/Data%20Sheets/Amphenol%20PDFs/RJMG2310228A0ER_Dwg.pdf)
25. Beagleboard.org latest images: https://beagleboard.org/latest-images
26. Roboticscape library: http://www.strawsondesign.com/#!manual-install
27. Linux core U-Boot User's Guide: http://processors.wiki.ti.com/index.php/Linux_Core_U-Boot_User%27s_Guide
28. Das U-Boot home page: http://www.denx.de/wiki/U-Boot/WebHome
29. Building U-Boot, Root Filesystem and Linux Kernel: https://eewiki.net/display/linuxonarm/BeagleBone+Black
30. OSD3358-SM-RED device tree files: https://github.com/octavosystems/OSD335x-Device-Tree
31. Device tree usage: http://elinux.org/Device_Tree_Usage
32. Device tree reference: http://elinux.org/Device_Tree_Reference
33. Device tree rebuilder: https://github.com/RobertCNelson/dtb-rebuilder
34. BeagleBone Black SRM: https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual