Keysight U3810A Advanced IoT Teaching Solution

Getting Started Guide

KEYSIGHT TECHNOLOGIES

Notices

Copyright Notice

Manual Part Number

U3810-90002

Edition

Edition 2, May 2021

Printed in:

Printed in Malaysia

Published by:

Keysight Technologies
Bayan Lepas Free Industrial Zone,
11900 Penang, Malaysia

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Declaration of Conformity

Declarations of Conformity for this product and for other Keysight products may be downloaded from the Web. Go to http://www.keysight.com/go/conformity. You can then search by product number to find the latest Declaration of Conformity.

U.S. Government Rights

The Software is “commercial computer software,” as defined by Federal Acquisition Regulation (“FAR”) 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement (“DFARS”) 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight:
(1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or
(2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing else- where in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED “AS IS,” AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUD ING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR OF ANY INFORMATION CON- TAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT SHALL CONTROL.

Safety Information

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Safety Symbols

The following symbols on the instrument and in the documentation indicate precautions which must be taken to maintain safe operation of the instrument.

Caution, risk of danger (refer to this manual for specific Warning or Caution information)

Safety Consideration

Read the information below before using the instrument.

The following general safety precautions must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards for design, manufacture, and intended use of the instrument. Keysight Technologies assumes no liability for the customer's failure to comply with these requirements.

WARNING

To prevent fire or injury:

- Use only the designated AC/DC adapter with the instrument.
- Observe all ratings and markings on the instrument before connecting to the instrument.
- When performing measurements, ensure that the right safety and performance ratings of instrument and accessories are used

CAUTION

Electrostatic discharge (ESD) can result in damage to the components at the exposed area of the educational kit. To prevent electrostatic discharge (ESD):

- Select a static-free work location when installing and removing sensitive component.
- Handle sensitive components to the minimum extent possible with ESD safe practices.
- Transport and store in ESD preventive bags or containers that protect sensitive components from static electricity.

CAUTION

- If the instrument is used in a manner not specified by the manufacturer, the instrument protection may be impaired.
- Always use a dry cloth to clean the instrument. Do not use ethyl alcohol or any other volatile liquid.

Environmental Conditions

The U3810A IoT System Design Module Training Kit is designed to operate under the general environmental requirements stated in the table below.

Environmental condition	Requirement
Temperature	Operating condition 0 to 40 °C Storage condition -40 to 70 °C
Humidity	Operating condition – Up to 80% RH at 25°C (non-condensing) Storage condition – Up to 95% RH at 40°C (non-condensing)
Altitude	Up to 2000 m

Power

The U3810A is generally powered from USB but may optionally be powered by up to 9 VDC / 500mA at its “Input Power” connector

Regulatory Information

The U3810A IoT System Design Module Training Kit complies with the following Electromagnetic Compatibility (EMC) compliance and radio requirements.

EMC compliance

- EN 61326-1:2013
- EN 301 489-1 V2.1.1:2017
- EN 301 489-17 V3.1.1:2017

RF compliance

- EN 300 328: V2.1.1:2016 (Wi-Fi / BLE)

RF health

- EN 50566: 2017 (Wi-Fi / BLE)

Safety compliance

- IEC 61010-1:2010 / EN 61010-1:2010
- IEC 60950-1:2005 + AMD2:2013 / EN 60950-1:2013 **
- IEC 62368-1:2014 / EN 62368-1:2014+A11:2017

RoHS

- EN 50581:2012

Canada

- ICES-003 Issue 6: 2019

United States

- FCC 47 CFR Part 15B

CAUTION

- This U3810A IoT System Design Module Training Kit may experience performance degradation due to connectivity loss with the U3810 CPU when electrostatic discharge (ESD) occurs at levels that exceed 4 kV.
- ESD precautions should be taken when handling the device.
- This equipment is not suitable for use in locations where children are present

Frequency Range

Band	Tx Frequency Range (MHz)	Transmitter Maximum Output Power (dBm)	Maximum RF Output power EIRP (dBm)
Bluetooth® LE	2402 to 2480	12.0	15.0
WiFi 802.11b/g/n	2412 to 2472	16.0	19.0
Zigbee (optional)	2405 to 2480	19.44 dBm Maximum Peak (87.90 mW)	N/A
LoRa (Country Dependent)	433.05 to 434.79	433.30 to 434.54 MHz Adjustable less than 10 dBm (10mW), typically 9.54 dBm (8.99mW) max measured conducted right at module	N/A
	863.00 to 870.00	863.25 to 869.75 MHz Adjustable less than 14.97 dBm (25mW), typically 13.57 dBm (22.8mW) max measured conducted right at module.	N/A
	917 to 923.5 (with U3810A-003)	917 to 923.5 MHz 19.07 dBm (80.9 mW)	N/A

根據NCC低功率電波輻射性電機管理辦法 規定:

第十二條　經型式認證合格之低功率射頻電機，非經許可，公司、商號或使用者均不得擅自變更頻率、加大功率或變更原設計之特性及功能。

第十四條　低功率射頻電機之使用不得影響飛航安全及干擾合法通信；經發現有干擾現象時，應立即停用，並改善至無干擾時方得繼續使用。
前項合法通信，指依電信法規定作業之無線電通信。
低功率射頻電機須忍受合法通信或工業、科學及醫療用電波輻射性電機設備之干擾。

Regulatory Markings

	The *FCC label* or the *FCC mark* is a certification mark employed on electronic products manufactured or sold in the United States which certifies that the electromagnetic interference from the device is under limits approved by the *Federal Communications Commission*.		The CE mark is a registered trademark of the European Community. This CE mark shows that the product complies with all the relevant European Legal Directives.
	This symbol indicates the time period during which no hazardous or toxic substance elements are expected to leak or deteriorate during normal use. Forty years is the expected useful life of the product.	**ICES/NMB-003**	ICES/NMB-003 indicates that this ITE device complies with the Canadian ICES-003. Cet appareil ITE est conforme a la norme NMB-003 du Canada.
	This instrument complies with the WEEE Directive (2002/96/EC) marking requirement. This affixed product label indicates that you must not discard this electrical or electronic product in domestic household waste.	Complies with IMDA standards N0494-20	This label indicates that this product complies with IMDA standards N0494-20
MCMC	Certification mark indicates a product has been certified by appointed Certifying Agency (SIRIM QAS International) as meeting MCMC Technical Codes (TC) that applied to the product.		The RCM mark is a registered trademark of the Australian Communications and Media Authority.
R-NZ	The R-NZ mark is the compliance mark of New Zealand radio communication standard	UK CA	The UKCA (UK Conformity Assessed) marking is a UK product marking that s used for goods being placed on the market in Great Britain (England, Wales and Scotland)

This marking indicates that this product complies with radio communication act B.E.2498.

ADENDO AO Manual

Modelo:U3810A

ANATEL
Agência Nacional de Telecomunicações

04922-20-11059

Para maiores informações, consulte o site da ANATEL www.anatel.gov.br

Este equipamento não tem direito à proteção contra interferência prejudicial e não pode causar interferência em sistemas devidamente autorizados

Language	Declaration U3810A
English	*Hereby, Keysight Technologies declares that the radio equipment type U3810A IoT System Design Module Training Kit is in compliance with Directive 2014/53/EU.*
French	Le soussigné, *Keysight Technologies*, déclare que l'équipement radioélectrique du type *U3810A IoT System Design Module Training Kit* est conforme à la directive 2014/53/UE.
German	*Hiermit erklärt Keysight Technologies dass der Funkanlagentyp U3810A IoT System Design Module Training Kit der Richtlinie 2014/53/EU entspricht.*
Italian	Il fabbricante, *Keysight Technologies*, dichiara che il tipo di apparecchiatura radio *U3810A IoT System Design Module Training Kit* è conforme alla direttiva 2014/53/UE.
Portuguese	O(a) abaixo assinado(a) *Keysight Technologies* declara que o presente tipo de equipamento de rádio *U3810A IoT System Design Module Training Kit* está em conformidade com a Diretiva 2014/53/UE.
Polish	*Keysight Technologies* niniejszym oświadcza, że typ urządzenia radiowego *U3810A IoT System Design Module Training Kit* jest zgodny z dyrektywą 2014/53/UE.
Spanish	Por la presente, *Keysight Technologies* declara que el tipo de equipo radioeléctrico *U3810A IoT System Design Module Training Kit* es conforme con la Directiva 2014/53/UE.

Waste Electrical and Electronic Equipment (WEEE) Directive 2002/ 96/EC

The U3810A IoT System Design Module Training Kit complies with the WEEE Directive (2002/96/EC) marking requirement. This affixed product label indicates that you must not discard this electrical or electronic product in domestic household waste.

Product category

With reference to the equipment types in the WEEE directive Annex 1, this device is classified as a “Monitoring and Control Instrument” product.

The affixed product label is as shown below.

Do not dispose in domestic household waste.

To return this unwanted device, contact your nearest Keysight Service Center, or visit http://about.keysight.com/en/companyinfo/environment/takeback.shtml for more information

Sales and Technical Support

To contact Keysight for sales and technical support, refer to the support links on the following Keysight websites:

- Product-specific information and support, software, and documentation updates
  - www.keysight.com/find/U3813A
  - www.keysight.com/find/U3814A
  - www.keysight.com/find/U3815A
  - www.keysight.com/find/U3816A
  - www.keysight.com/find/U3817A
  - www.keysight.com/find/U3818A
- Worldwide contact information for repair and service
  - www.keysight.com/find/assist

Table of Contents

Overview

Keysight's ready-to-teach Advanced IoT Teaching Lab Solution is designed to assist educators in quickly setting up new engineering courses on the Internet of Things, with the intention of producing students who will fully understand the challenges and requirements of the IoT system design cycle, from design and validation to deployment in the market.

This courseware will also cover critical design considerations that have emerged with the evolution of the Internet of Things, such as cybersecurity, coexistence, compliance and continuity in the following modules:

U3813A/14A IoT System Design and Validation Fundamentals

The U3813A/14A IoT System Design and Validation Fundamentals lab setup is a ready-to-teach package focused on the fundamentals of the Internet of Things and embedded system design. It introduces students to IoT architecture, technologies, standards, wireless protocols, applications, and ecosystems. It also covers IoT embedded system design that includes device cybersecurity basics.

U3815A/16A IoT Wireless Communication and Compliance

The U3815A/16A Wireless Connectivity and Network Security for IoT Frameworks lab setup is a ready-to-teach package started from portion of IoT System Design and Validation Fundamentals. After that move into how to develop typical IoT applications with various types of wireless connectivity and compliance study, it also covers IoT device and network security learning.

U3817A/18A IoT Precision Power Measurement and MEMS Sensors

The U3817A/18A Precision Power Measurement and MEMS sensors lab setup is a ready-to-teach package started from portion of IoT System Design and Validation Fundamentals. Then move into topic of how to characterize the power consumption of IoT devices onboard controllers, sensors and wireless modules, eventually covers sophisticated battery optimization learning involve RF event detector and analysis software.

Intended Use of Getting Started Guide

The Getting Started Guide is intended for use by a University Teaching Lab Manager as a guide for unpacking, set up, verification and maintenance of the Modular prototype kit.

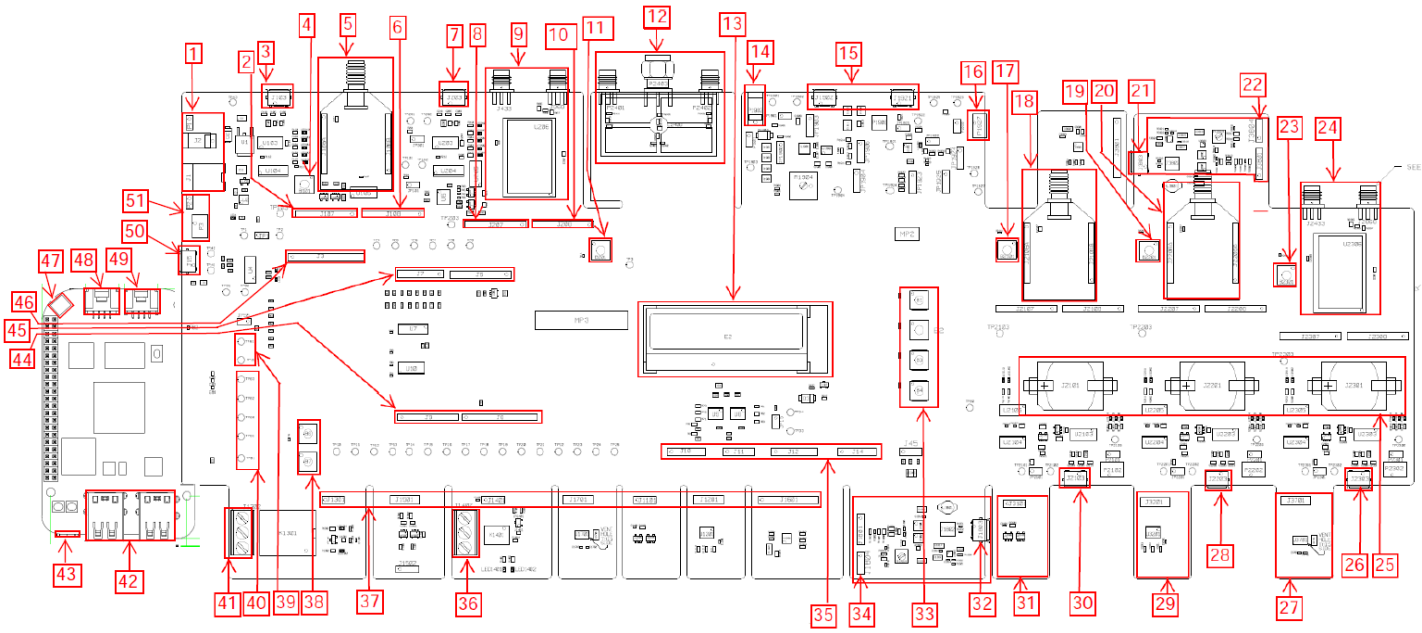
The GSG should be consulted at the beginning of a new semester to assist in returning the Modular prototype kit to its initial factory configuration and for verification before the student(s) receives the kit. In the first lab of each semester the student will run a procedure.

Characteristics and Specifications

For the characteristics and specifications of the U3810A Advanced IoT Teaching Solution, refer to the Data Sheet at https://www.keysight.com/us/en/assets/3120-1243/data-sheets/Advanced-IoT-Teaching-Lab-Solution.pdf.

Component Locations

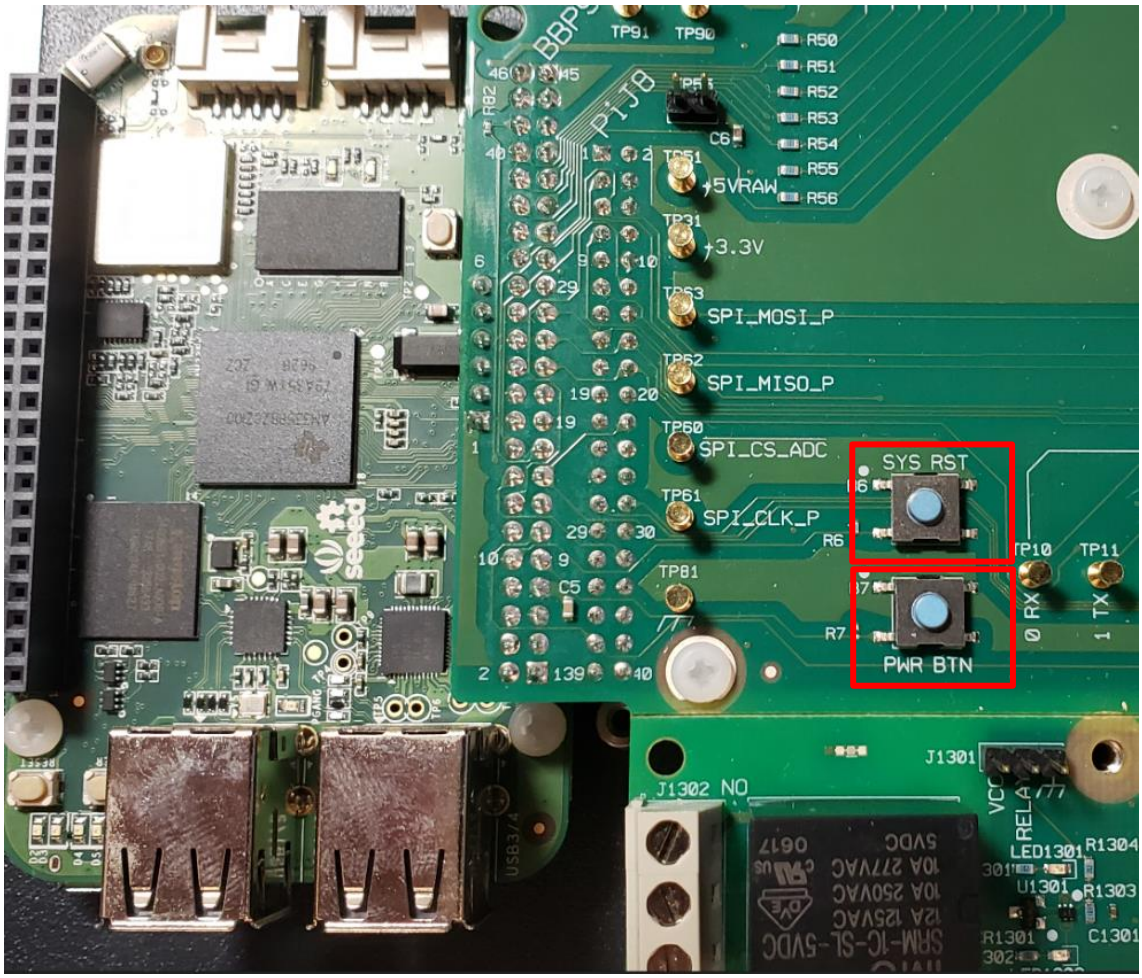
A searchable Component Locator Drawing is available in Appendix C .



	Item		Item
1	DC Power Input Connector	27	Spare Digital pressure sensor (remove)
2	Micro-USB for XB	28	Micro-USB for XB2
3	Power & PWM input for XB	29	Spare Digital Temperature sensor (remove)
4	XB Transceiver RESET Button	30	Micro-USB for XB1
5	XB Transceiver	31	Spare Analog Temperature sensor (remove)
6	AD interface for XB	32	MicroUSB for current sensor
7	USB for LoRa Transceiver	33	Buttons
8	Power & AD interface for LoRa	34	Power & Input for Current Sensor
9	LoRa Transceiver	35	Power, I^2C connector & SPI interface from sensors
10	AD interface for LoRa	36	Load connection for solid state relay
11	LoRa XCVR RESET button	37	Test points for control signals
12	RF Power Splitter	38	Button
13	LCD display	39	Test points for DC Power
14	DC Power Input Connector	40	Test points for control signals
15	Micro-USB power input connector	41	Load connection for relay actuator
16	Battery charger output connector	42	USB BeagleBone
17	XB1 XCVR RESET Button	43	OTG BeagleBone
18	XB1 Transceiver	44	Digital signal interfaces
19	XB2 XCVR RESET	45	Analog signal interfaces and Power
20	XB2 Transceiver	46	Analog Inputs
21	Micro-USB for current sensor	47	Chip Antenna for BT & WLAN
22	Spare Current Sensor (remove)	48	Grove UART - BeagleBone
23	LoRa1 XCVR RESET	49	Grove I^2C - BeagleBone
24	LoRa1 Transceiver	50	UART console
25	Coin cell Battery holder	51	Battery power connector
26	Micro-USB for LoRa1 Transceiver		

Button Functions

Button Name	Function
Power Button (PWR BTN)	Momentarily press PWR BTN. This will start the power down sequence. Once the display goes out, power can be removed. If power is applied and the display is not lit, the PWR BTN can be momentarily pressed to start the boot sequence. Once "Keysight U3810A is displayed, the unit is fully power up.
Reset Button (SYS RST)	Momentarily press the SYS RST button and this will start the system in a boot up sequence. This will override any currently running processes.



Hardware and Software Requirements

NOTE If you intend to configure your kit with a Raspberry Pi instead of BeagleBone, consult the **U3810A - Using Alternate CPUs** for instructions. Raspberry Pi, BBBW (BeagleBone Black Wireless) or BBGW (BeagleBone Green Wireless) are the only CPUs tested at the time of publication. Only Green has been certified by Keysight.

1. Prepare the required items as listed in the "Equipment and Accessories Required" list below.
2. Download the required software installers according to the "Software Required" list and install them on your Windows PC.

Equipment Required

1. 1x Keysight U3810A IoT System Design Module Training Kit with new BeagleBone Wireless CPU
2. 1x Laptop or desktop PC running on Windows 8 or 10 with Internet access (Linux and macOS may work but are not presently on Keysight's list of supported platforms)
3. (Optional) BeagleBone Black Wireless eliminates the two Grove connectors, not used in Keysight labs, but provides the added convenience of connecting a HDMI monitor which provide a text console (like having PuTTY built-in).

Accessories Required

1. 1x Micro-USB cable
2. 1x Analog temp sensor (On-board analog temperature sensor accessory)
3. 1x Digital temp sensor (On-board digital temperature sensor accessory)
4. 1x Digital pressure sensor (On-board digital pressure sensor accessory)
5. 1x IMU (On-board digital accelerometer accessory)
6. 1x Relay actuator (On-board relay actuator accessory)
7. Jumper wires
8. 1x 8GB or larger Micro SD card (Optional)
9. 1x USB thumb drive / memory stick

Software Required

1. PuTTY (http://www.putty.org/)
2. BONE_DRV or BONE_D64 PC driver for RNDIS from BeagleBone storage. Network and Serial Drivers for Mac from https://beagleboard.org/static/Drivers/MacOSX. No drivers needed for Linux.
3. WinSCP (https://winscp.net/eng/download.php)
4. (Optional) balenaEtcher (https://etcher.io/) or WinDisk32 (https://sourceforge.net/projects/win32diskimager/) for writing SD Card

5 (Optional) Keysight BeagleBone Initialization Image: **Keysight_BB_image.img.xz**
This image is a component of the purchased courseware. You can download the image together with the courseware using this process:

a Obtain the Keysight Entitlement Certificate

b Login into Keysight Software Manager at https://ksm.software.keysight.com/. New users will need to create a **myKeysight** account.

c Follow the instructions in the Keysight Entitlement Certificate to redeem and download the image along with the courseware.

Set Up the U3810A System

Configure the Keysight U3810A as a “cape”

Depending on your purchase configuration, your kit will either be supplied with a BeagleBone CPU out of the box or you will supply your own BeagleBone CPU.

Assure that the Keysight U3810A IoT System Design Module Training Kit is configured as a “cape” on top of the BeagleBone CPU.

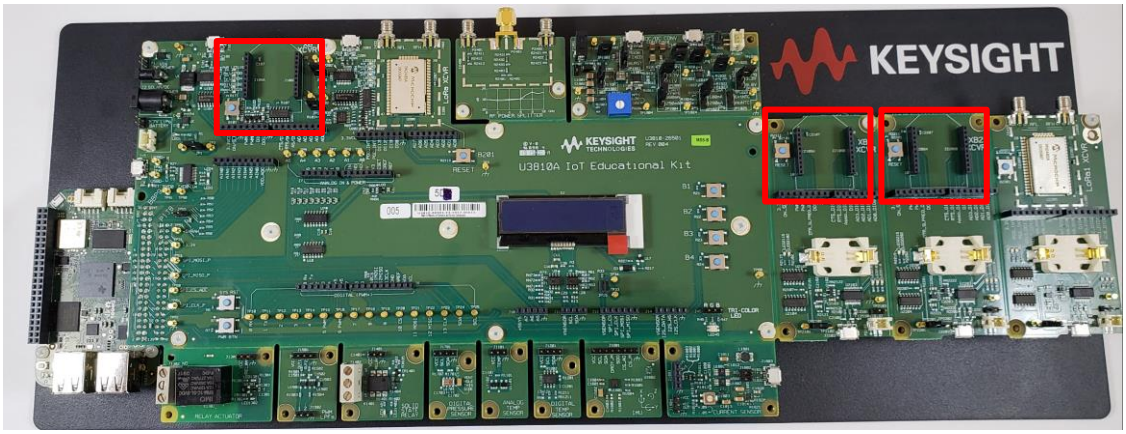
Source: beagleboard.org

1. Remove the U3810A Printed Circuit Assembly (PCA) and Baseplate from its anti-ESD bag.
2. If you are supplying your own BeagleBone CPU, perform the Assembly procedure provided in **Appendix F – Assembly and Disassembly** to install the BeagleBone CPU on the blank baseplate, then installing the U3810A main board and snap-off accessories on top of the BeagleBone CPU.

Sensor and Xbee3 Module Installation

1. Remove the five (spare) snap-offs and place in an anti-ESD bag (provided).

2. Install the three Xbee3 Transceiver Modules in sockets marked **XB**, **XB1**, and **XB2**.

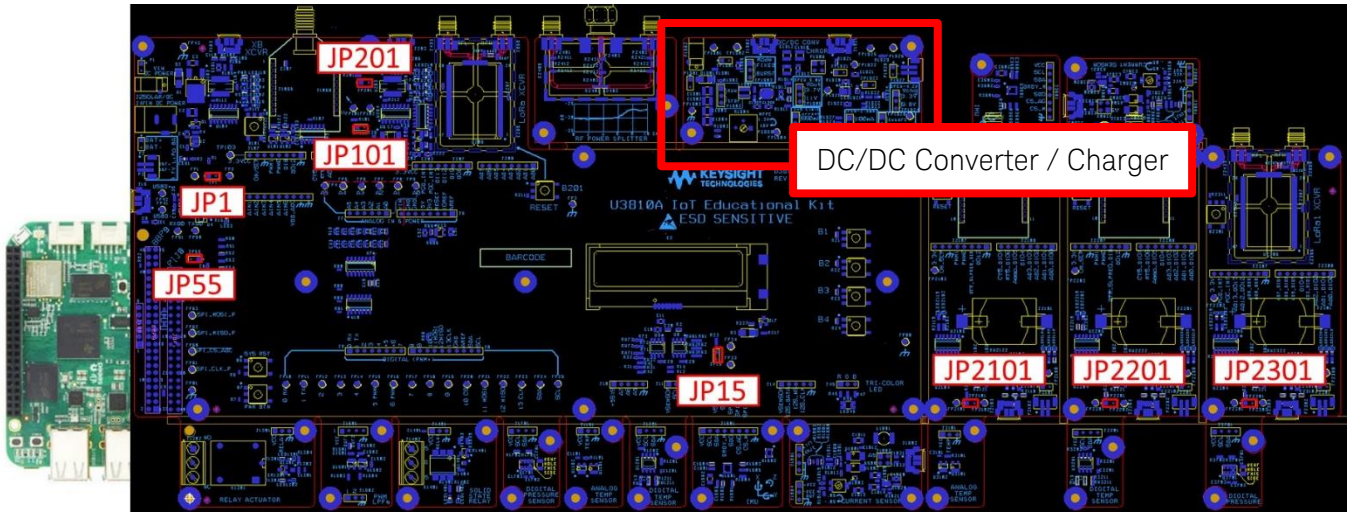


3 Placed the assembly on a location on your worksurface where all components and connectors are accessible for future wiring:

Main Jumpers

Assure the configuration of the Main Jumpers shown below.

Jumper	JP1	JP15	JP55	JP101	JP201	JP2101	JP2201	JP2301
Name	Input Current	Sensor Current	+5VSYS +5VRAW	XB Current	LoRa Current	XB1 Current	XB2 Current	LoRa1 Current
Position	In place	In place	Removed	In place	In place	In place	In place	In place

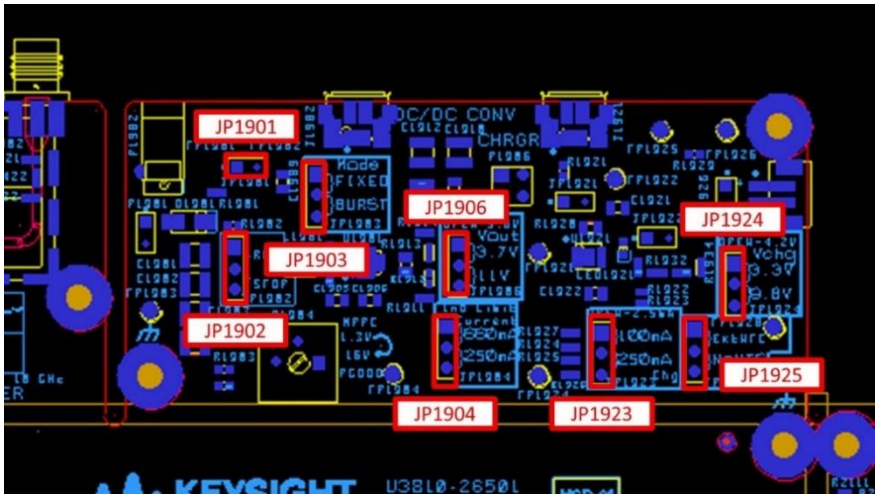


DC-to-DC Converter and Charger Jumpers

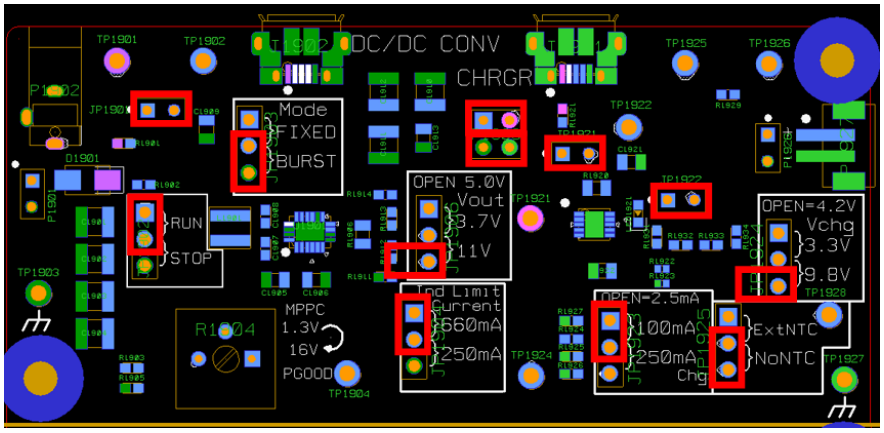
Assure the configuration of the DC-to-DC Converter and Charger Jumpers shown below.

Jumper	JP1901	JP1902	JP1903	JP1904	JP1906	JP1921	JP1922	JP1923	JP1924	JP1925
Name	DC/DC Input Current	DC/DC Run Stop	DC/DC Mode	Inductor Current Limit	DC/DC Vout	Charger Input Current	Charger Output Current	Charger Current	Charge Voltage	NTC Setting
Position	In Place	Up RUN	FIXED	660mA	Open 5.0V	In Place	In Place	Up 100mA	Open 4.2V	Down NoNTC

Jumper locations:



Jumper settings:



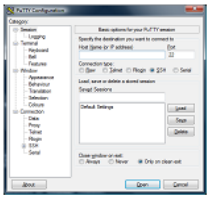
NOTE

In the next steps, perform these tasks which are identical to the tasks contained in Lab 1 of any U381x course, to complete setup.

- Task 1a – Establish Serial Communications between BeagleBone and PC
- Task 1b – Establish Secure Shell (SSH) Communication between BeagleBone and PC
- Task 1d – Configure BeagleBone to Connect to WLAN network
- Task 1e – Copy and Edit Files with WinSCP - only perform these two:
  - Set Up WinSCP
  - Start PuTTY SSH Connection from WinSCP

Establish Serial Communications between BeagleBone and PC

1 If not already done so, download and install PuTTY from http://www.putty.org/. Choose 32-bit or 64-bit, whichever is compatible with your operating system.

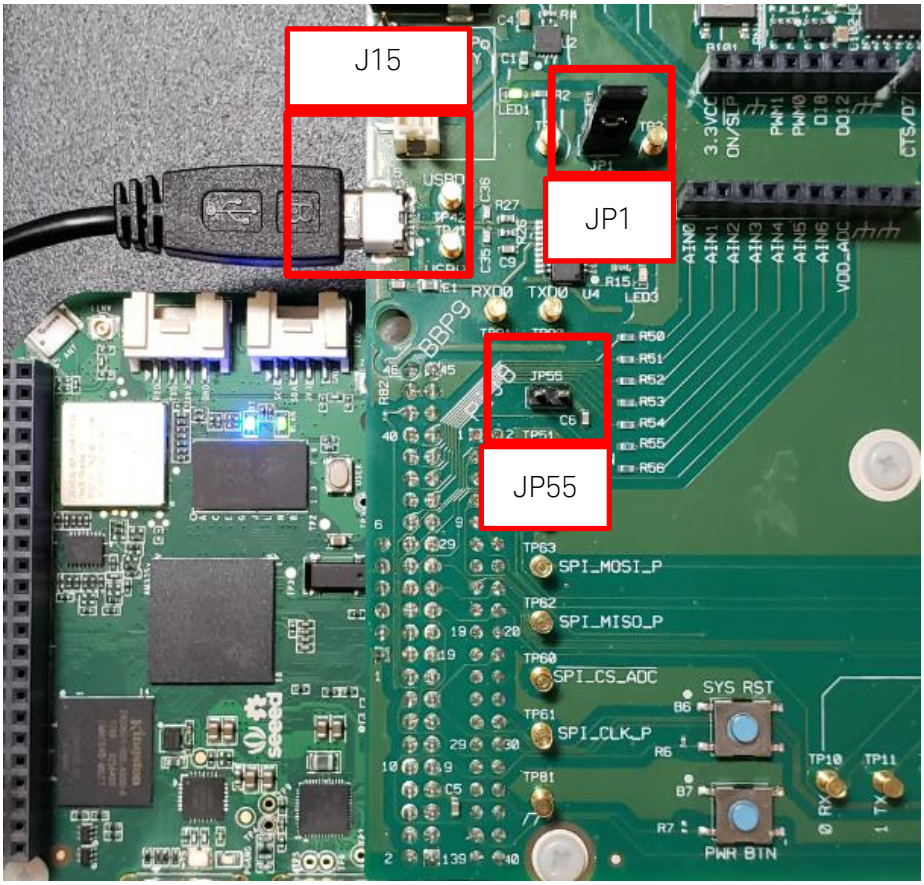


Download PuTTY

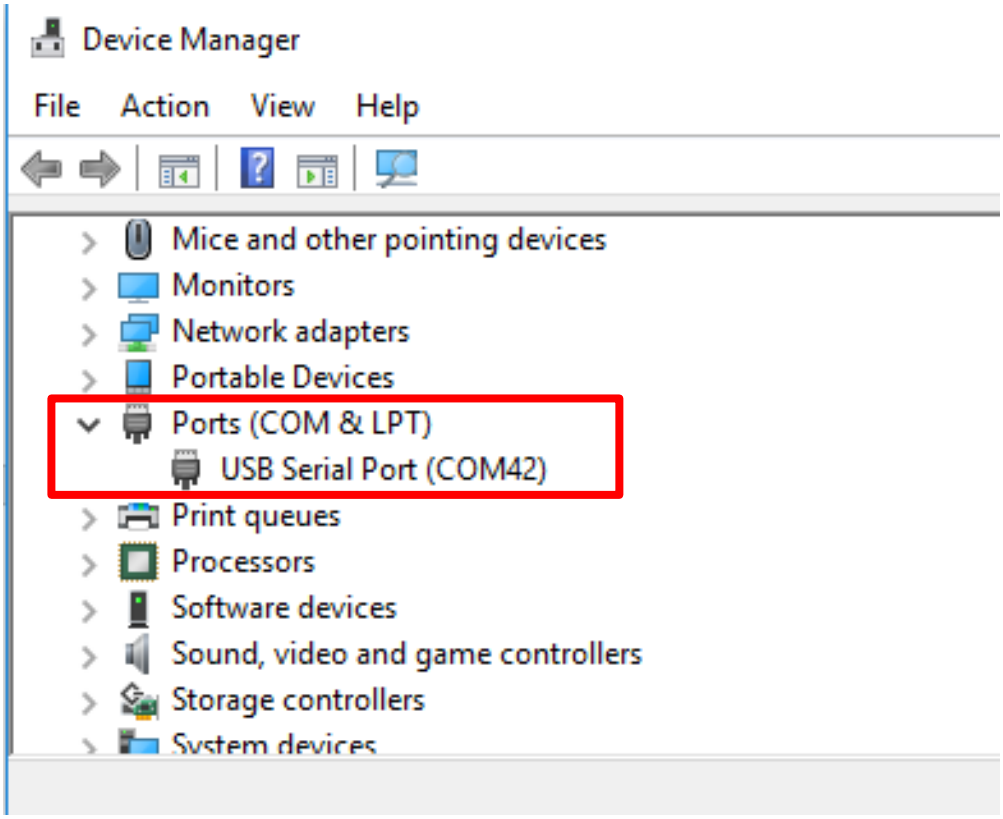
PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

You can download PuTTY here.

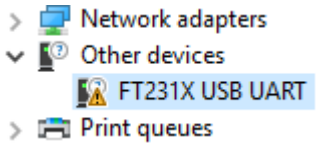
2 Only connect a USB cable from your computer to J15 of the U3810A. The JP1 should be in place and JP55 should not be placed.



3 Press the Windows key and type Device Manager or type the Run and type **devmgmt.msc** to open the Device Manager. Then find the COM port that connects to U3810A. Note down the port number. You will need this to configure the serial communication using PuTTY.

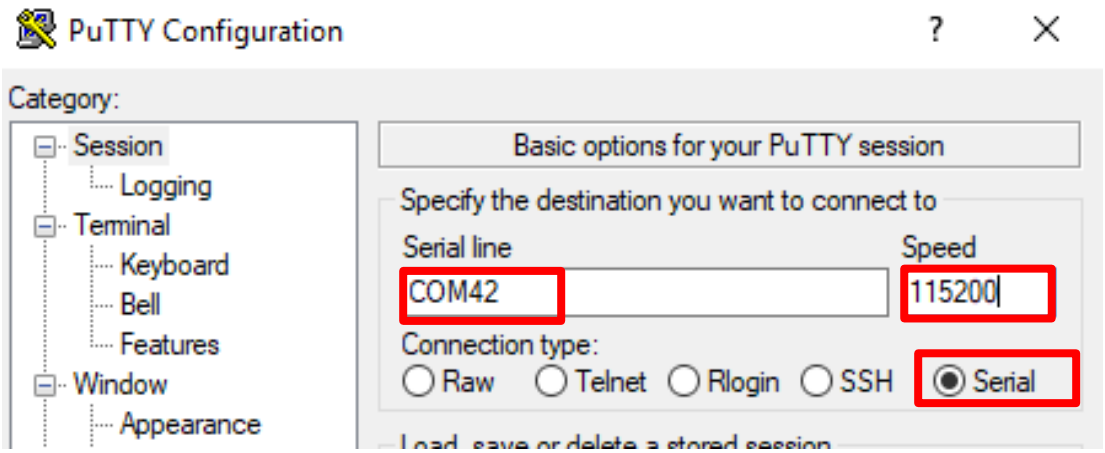


If you see the warning below, the driver was not automatically installed by Windows.

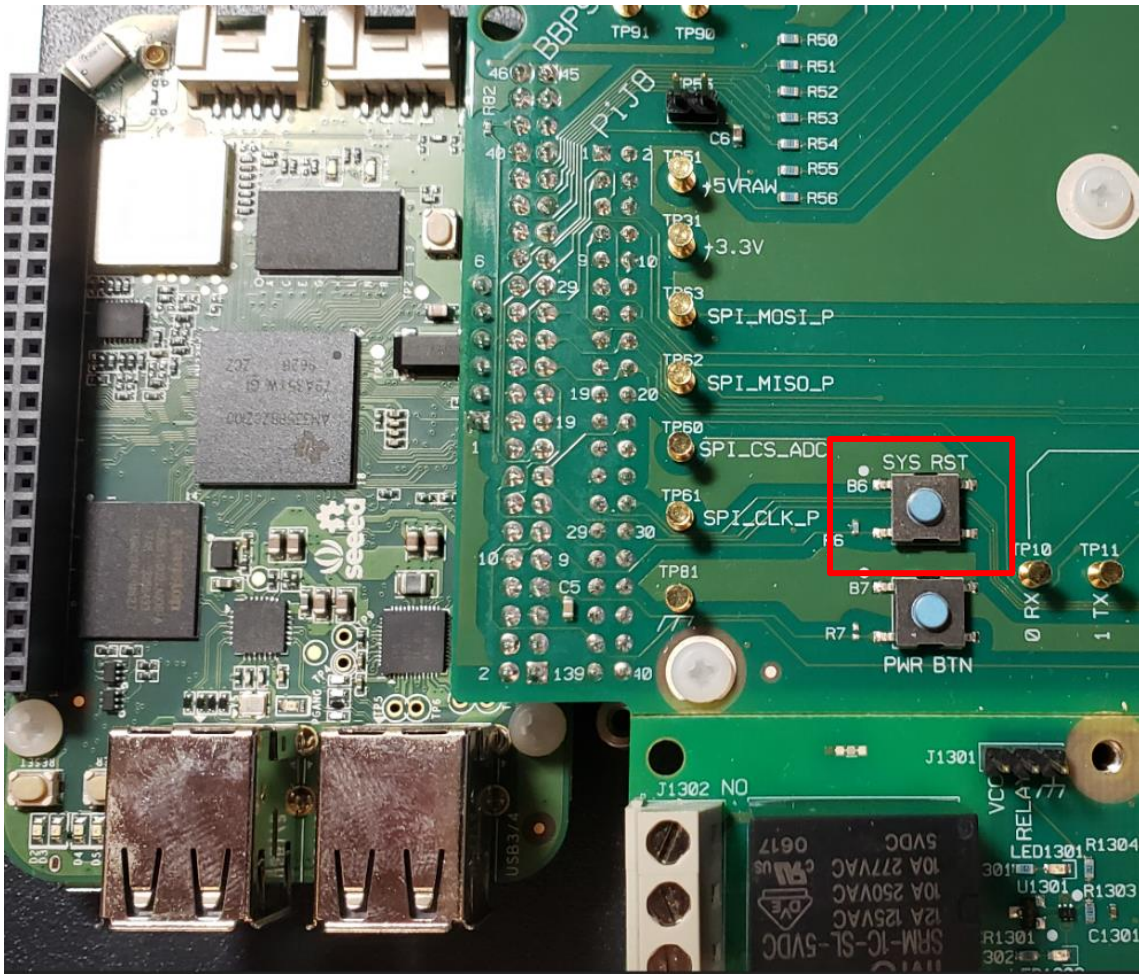


Refer to USB Serial Driver Installation Problems for more information.

4 Run your PuTTY software and connect to the COM port that you have noted down previously using Serial connection at 115200 Baud-rate.



5 While observing the PuTTY window briefly press the **SYS RST** button on the lower left of the U3810A board.



Notice the first action shown in the PuTTY window after the **SYS RST** button has been pressed is to load the U-Boot which is the underlying low-level boot code.

```
U-Boot SPL 2019.04-00002-gbb4af0f50f (Jul 08 2019 - 11:44:39 -0500)
Trying to boot from MMC2
Loading Environment from EXT4... Card did not respond to voltage select!

U-Boot 2019.04-00002-gbb4af0f50f (Jul 08 2019 - 11:44:39 -0500), Build:
jenkins-github_Bootloader-Builder-128

CPU  : AM335X-GP rev 2.1
I2C:   ready
DRAM:  512 MiB
```

Note the SD Card is not present, will cause this error message

NOTE It should be loading a U-Boot version of 2019.03. If an older revision is seen, the BeagleBone will need to be re-initialized by loading a new image from an SD Card. Refer to Appendix A - Initialize BeagleBone with Keysight U3810A Image for more information on how to do it.

If the boot is successful there should be a login prompt with the password hint shown in green highlight.

```
Debian GNU/Linux 9 beaglebone ttyS0
BeagleBoard.org Debian Image 2019-09-01
Keysight U3810A Image Version 3.57 Sept 20th 2019
Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
default username:password is [debian:temppwd]
beaglebone login:
```

NOTE If there is not a line in the serial port login indicating the U3810A Image revision, refer to Appendix A - Initialize BeagleBone with Keysight U3810A Image.

This connection is on the system console what might have messages about the system operation come up If they do hit <**enter**> and start a new line.

---

6 Enter username **debian** and the password **temppwd.** A login message should appear showing the last login and the Keysight U3810A Lab Code Version.

```
Last login: Fri Sep 20 16:39:16 UTC 2019 on ttyS0

The programs included with the Debian GNU/Linux system are free
software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Keysight U3810A Image Version 3.57 Sept 20th 2019
debian@beaglebone:~$
```

7 If there is no Keysight Revision after login, please obtain an SD Card and follow the procedure in Appendix A - Initialize BeagleBone with Keysight U3810A Image.

8 To start with a new version of the lab code run the command **sh LabCodeReset.sh -u**. For this update, enter temppwd for the sudo password.

```
debian@beaglebone:~$ sh LabCodeReset.sh -u
The LabCode.tar.gz has been signed.
Checking on the signature
gpg: assuming signed data in 'checksum'
gpg: Signature made Mon 27 Jul 2020 01:24:21 PM UTC
gpg:         using RSA key E89C4532A5DB38EBE14CF510F55535C5FA4EB16E
gpg: Good Signature from "Copyright Keysight Technologies 2020"
[ultimate]
Checking on SHA256 hash ...
LabCode.tar.gz: OK
Do you want to continue? (y/n) y
you want to erase before refreshing the code? (y/n) y
Erasing LabCode
Refreshing LabCode
LabCode/
LabCode/M1-L2/
LabCode/M1-L2/M1_L2_T2c_RGB_LED_PWM.c
LabCode/M1-L2/M1_L2_E1_buttons.c
...

.KS_Files/etc/environment
.KS_Files/boot/
.KS_Files/boot/uEnv.txt
Updating Keysight Revisions
[sudo] password for debian: temppwd
Reboot is suggested
debian@beaglebone:~$
```

9 After rebooting a logging in, the login information should show the latest version.

```
Last login: Tue Jul 28 13:18:14 UTC 2020 on ttyS0

The programs included with the Debian GNU/Linux system are free
software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Keysight U3810A Image Version 3.81 July 27th 2020
debian@beaglebone:~$
```

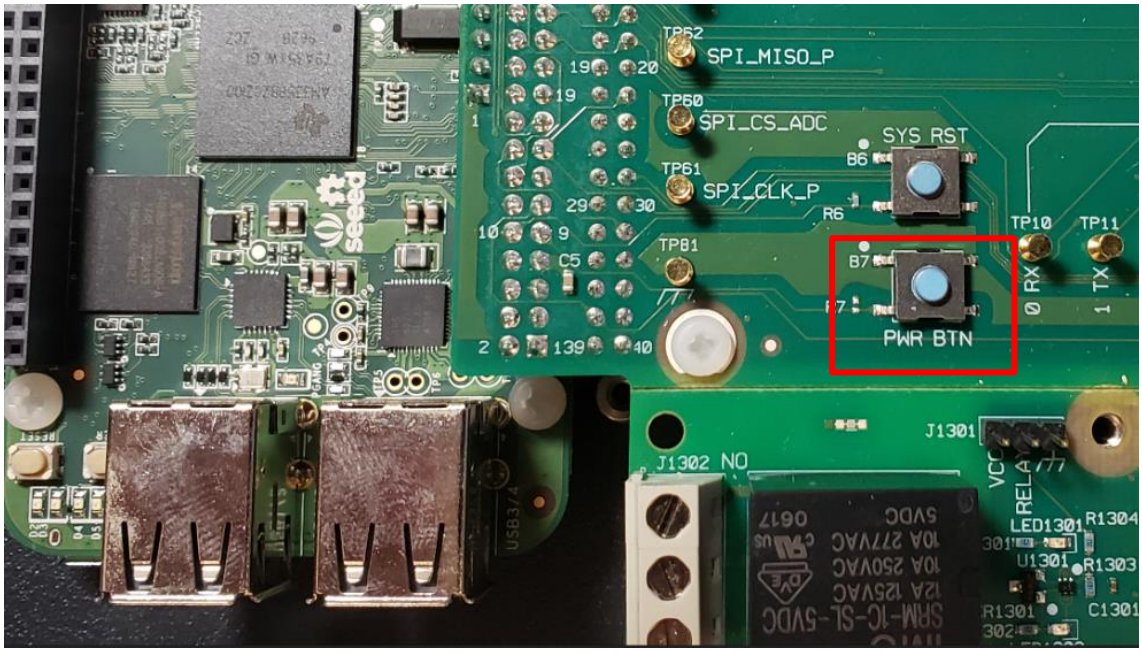
10 Run the **`mraa-gpio list`** command and verify the last few lines of the output look like this.

```
71     GPIO117: GPIO
72      GPIO14: GPIO UART
73     GPIO115: GPIO
74     GPIO113: GPIO SPI
75     GPIO111: GPIO SPI
76     GPIO112: GPIO SPI
77     GPIO110: GPIO SPI
78     VDD_ADC:
79        AIN4: AIO
80     GND_ADC:
81        AIN6: AIO
82        AIN5: AIO
83        AIN2: AIO
84        AIN3: AIO
85        AIN0: AIO
86        AIN1: AIO
87      GPIO20: GPIO
88       GPIO7: GPIO
89         GND:
90         GND:
91         GND:
92         GND:
```

The BeagleBone is fully updated and ready to use.

NOTE Try the command again. This time typing just a few letters followed by **<tab>** to see how command completion works.

11 Power off the system by pushing the **PWR BTN** button or by executing the command **`sudo poweroff`**. The display should blank once the system is completely powered off.

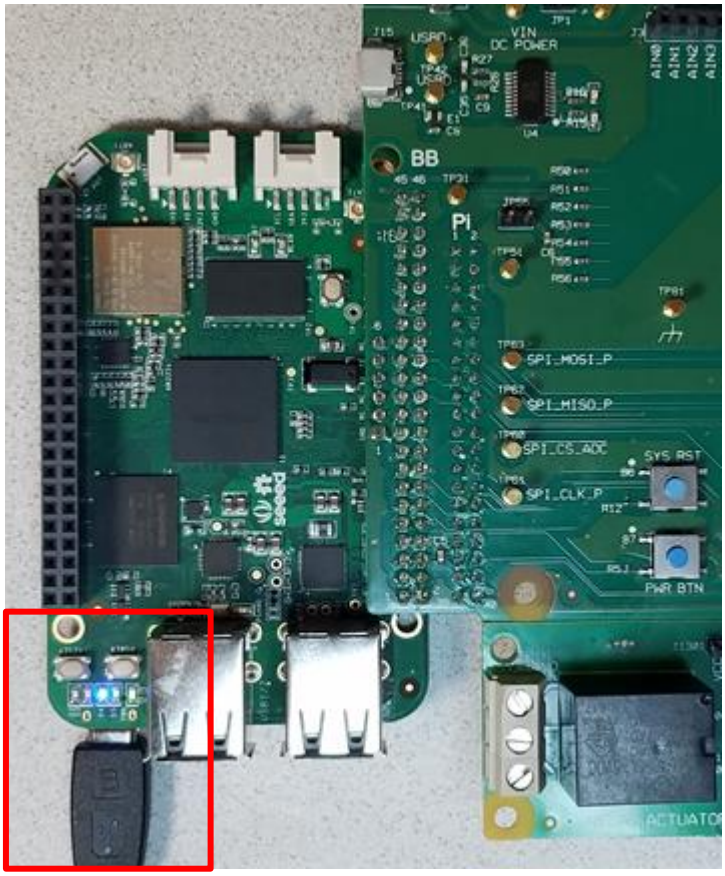


Establish Secure Shell Communication between BeagleBone and PC

In this exercise, you will connect the PC (Host) to Beagle Bone via a USB cable and establish a RNDIS connection. RNDIS is the Remote Network Driver Interface Specification. It defines internet connection via USB and this connection provides a virtual network to the Beagle Bone that supports various network protocols including Secure Shell (SSH) Communication and HTTP. Once the connection is established, a PuTTY terminal using SSH can be used. The local documentation on the webpage can also be explored. The RNDIS Network IP address of the BeagleBone will be **192.168.7.2** while your PC will be at **192.168.7.1.**

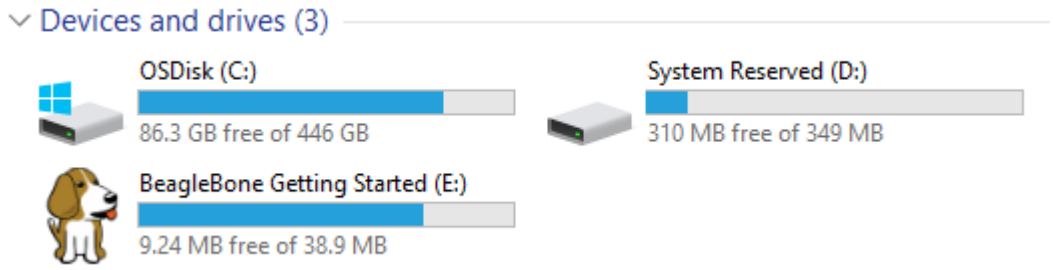
WARNING When JP1 is in place do not connect a USB cable to both the BeagleBone and J15 at the same time, or anomalous behavior may result.

1 Remove the USB cable from **J15** and connect it instead to the BeagleBone CPU USB port to your PC. This will also power up the U3810A. It may take up to one minute to complete the boot process.

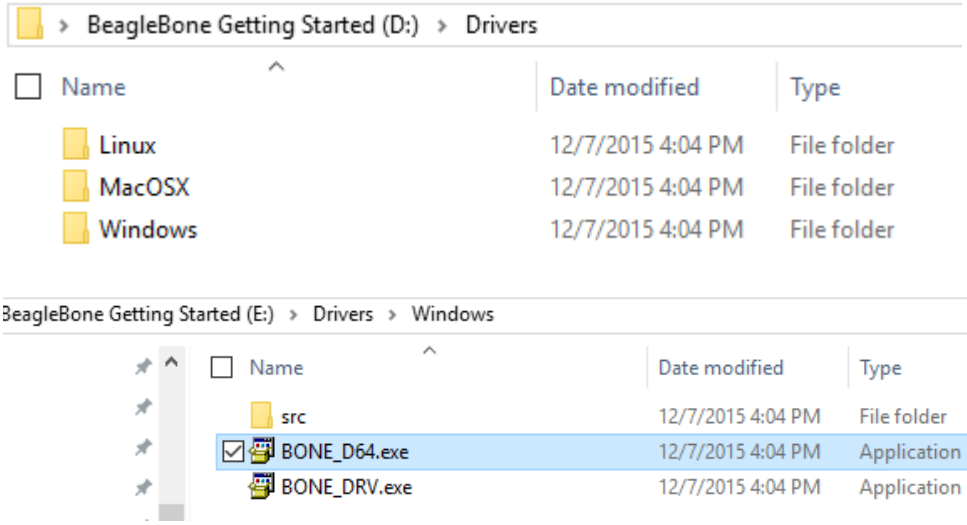


Install RNDIS drivers

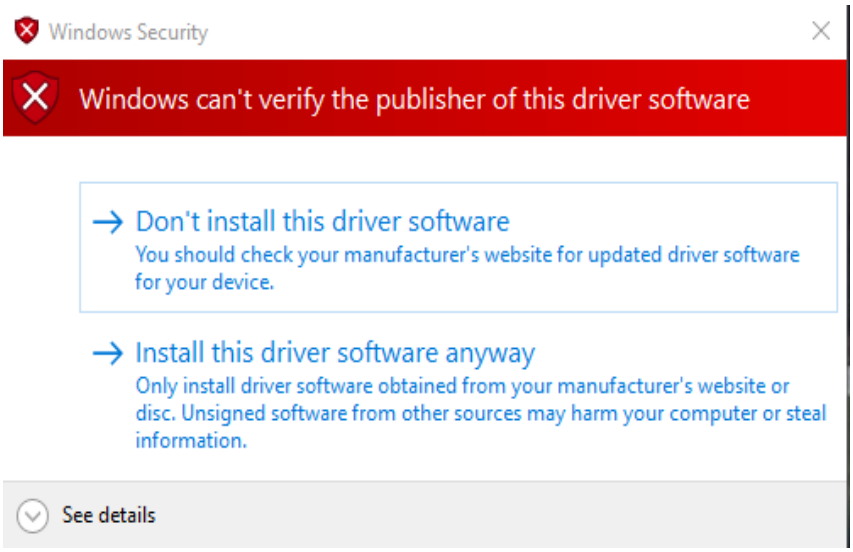
2 If the drivers have not already been installed open the **BeagleBone Getting Started** drive using a file explorer.



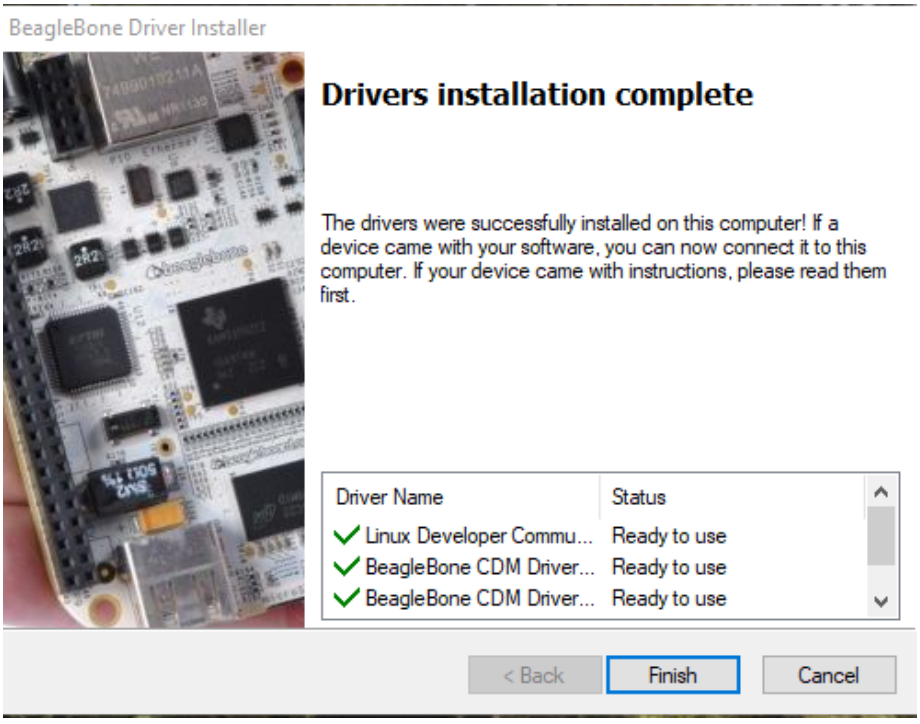
3 Select the driver for your OS from the Drivers folder and install the BONE_D64.exe file.



4 During the installation, Windows 10 users may see this message. Click the **Install this driver software anyway**.



Successful installation message will show the message.



Refer to Troubleshooting Guide - USB RNDIS Drivers Installation for more information if you receive the error below.

Configure RNDIS adapter

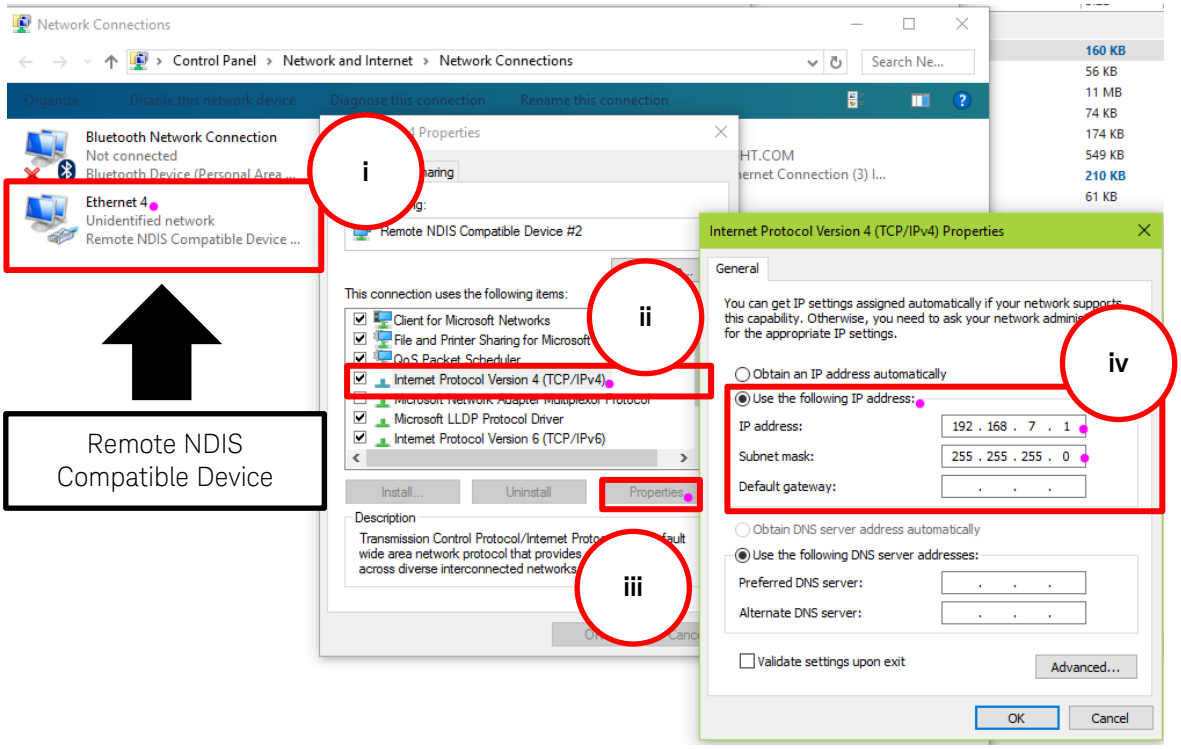
Your PC will need to be on the same subnet using the RNDIS connection. This does not have DHCP, so your PC address needs to be set to **192.168.7.1**.

NOTE You may need to run this step each time you connect a different BeagleBone to your PC. The RNDIS adapter setting can only be **192.168.7.1** while the BeagleBone itself is at **192.168.7.2.**

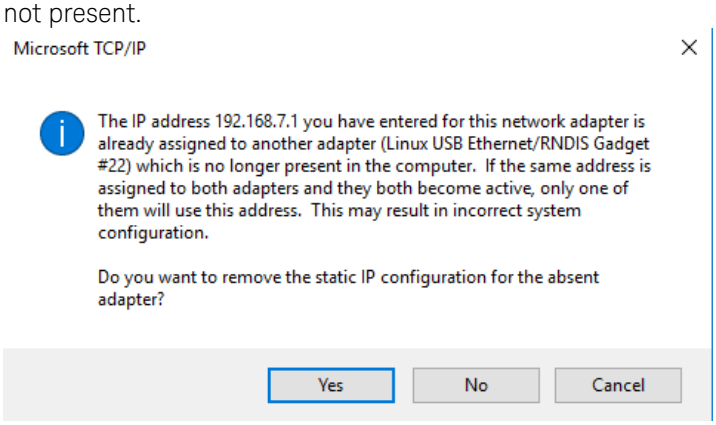
---

5 Go to **Network Settings** and click the **<your Remote NDIS Adapter>**. Click **Internet Protocol Version 4 (TCP/IPv4) > Properties** and set up as shown below.

For Windows 10, go to **Control Panel\Network and Internet\Network and Sharing Center > Change Adapter Settings:**



If you receive the following message, it means that there was a previous BeagleBone or other device on this address. You can click **Yes** if the other device will not be use or No if both devices are not present.



Set Up SSH connection

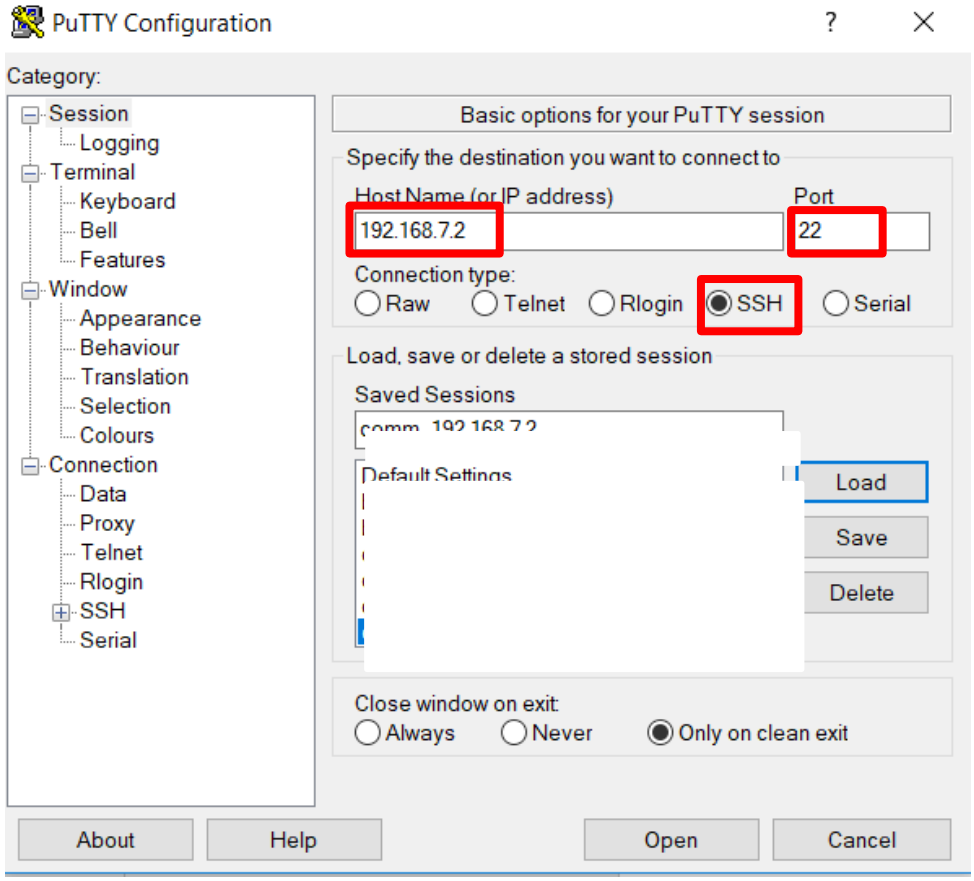
6 To launch the Command Prompt, type cmd on your PC at the search bar and click to launch it.

7 On the Command Prompt, type ping 192.168.7.2 and press Enter. You should see:

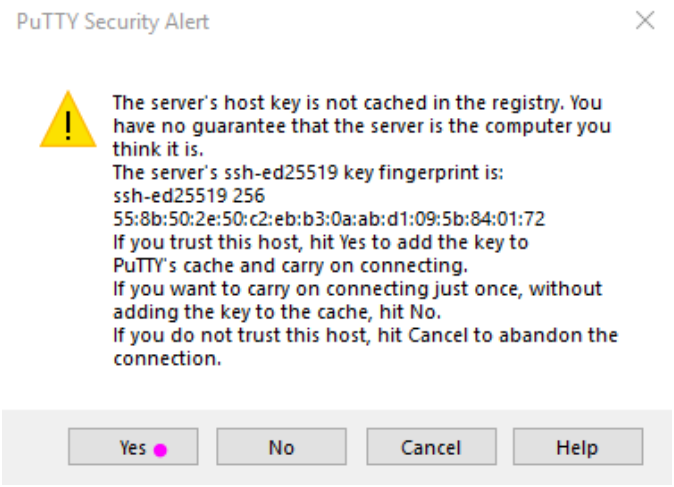
```
C:\Users\Lawrence>ping 192.168.7.2

Pinging 192.168.7.2 with 32 bytes of data:
Reply from 192.168.7.2: bytes=32 time<1ms TTL=64
Reply from 192.168.7.2: bytes=32 time<1ms TTL=64
Reply from 192.168.7.2: bytes=32 time<1ms TTL=64
Reply from 192.168.7.2: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.7.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

8 Once the ping command comes back with a reply and a response time, double-click PuTTY.exe to launch the PuTTY terminal program.

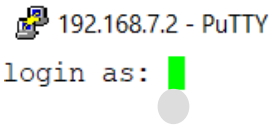
9 A PuTTY Configuration window will pop up to determine the connection type. Select **SSH** for Connection type and enter **192.168.7.2** for the **IP address**.



10 If this is the first time that the computer is connecting to this Beagle Bone, you will receive this message and question to which you should click **Yes**.



11 Click **Open** to open the terminal window. Press **Enter** on the PC keyboard to check and verify connectivity. Otherwise, refer to **Getting Started Guide** to upgrade the firmware.



Default username : **debian**

Default password : **temppwd**

12 Enter **debian** for login to log into the Beaglebone CPU on the U3810A. Debian will require **temppwd** for its password.

Note that the password will appear as blank and unresponsive as you type.

```
login as: debian
Debian GNU/Linux 9

BeagleBoard.org Debian Image 2019-09-01

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

debian@192.168.7.2's password:

The programs included with the Debian GNU/Linux system are free
software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Keysight U3810A Image Version 3.57 Sept 20th 2019
Last login: Fri Sep 20 16:49:15 2019
debian@beaglebone:~$
```

A successful boot will show a login prompt with the password hint shown in green highlight.

```
Debian GNU/Linux 9 beaglebone ttyS0

BeagleBoard.org Debian Image 2020-01-17

Keysight U3810A Image Version 3.63 Jan 17th 2020

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

beaglebone login:
```

NOTE This connection is on the system console what might have messages about the system operation come up If they do, hit <**enter**> and start a new line.

---

13 Enter the username **debian**, <enter> and the password **temppwd**.

14 Run the **`mraa-gpio list`** command and verify the last few lines of the output look like this.

```
71      GPIO117: GPIO
72       GPIO14: GPIO UART
73      GPIO115: GPIO
74      GPIO113: GPIO SPI
75      GPIO111: GPIO SPI
76      GPIO112: GPIO SPI
77      GPIO110: GPIO SPI
78      VDD_ADC:
79         AIN4: AIO
80      GND_ADC:
81         AIN6: AIO
82         AIN5: AIO
83         AIN2: AIO
84         AIN3: AIO
85         AIN0: AIO
86         AIN1: AIO
87       GPIO20: GPIO
88        GPIO7: GPIO
89          GND:
90          GND:
91          GND:
92          GND:
```

The BeagleBone is fully updated and ready to use.

NOTE If the **`mraa-gpio`** command is not present, the Keysight image may not have been loaded correctly. See the section on diagnosing and loading images in the Appendix A – Initialize BeagleBone with Keysight U3810A Image

15 It is best to power off a system with the button or via a command before disconnecting it from the power. To power off the system, push the PWR button or type **`sudo poweroff`**. The display should blank once the system is completely powered off.

Configure BeagleBone to Connect to WLAN network

Once this connection has been established for the first time, it will automatically connect back on subsequent reboots.

1 In the PuTTY terminal window, enter **connmanctl** to start the wireless connection manager.

2 Enter **technologies** to verify the WLAN function is available.

```
debian@beaglebone:~$ connmanctl
connmanctl> technologies
/net/connman/technology/p2p
  Name = P2P
  Type = p2p
  Powered = False
  Connected = False
  Tethering = False
/net/connman/technology/wifi
  Name = WiFi
  Type = wifi
  Powered = True
  Connected = False
  Tethering = False
/net/connman/technology/bluetooth
  Name = Bluetooth
  Type = bluetooth
  Powered = True
  Connected = False
  Tethering = False
connmanctl>
```

NOTE

- It is possible that you may see the following. This is an acceptable behavior and you may proceed:

  ```
  debian@beaglebone:~/LabCode/M3-L7$ connmanctl
  Error getting VPN connections: The name net.connman.vpn was not
  provided by any
  connmanctl>
  ```

- If you see “Powered = False” for WLAN, then it means WLAN is disabled. Enter the **enable wifi** command to enable it.

3 Enter the **scan wifi** command to search the available networks.

```
connmanctl> scan wifi
Scan completed for wifi
```

4 Type the **agent on** command to turn on the connection agent.

```
connmanctl> agent on
Agent registered
```

5 Type the **services** command to view the available SSID's.

```
connmanctl> services
    MRR management service wifi_#######_managed_psk
    dreamx               wifi_1234567890_managed_psk
    MRR Management 2     wifi_########_managed_psk
    PLAZZADPNG           wifi_########_managed_psk
    MRR Management       wifi_########_managed_psk
    MulhafArchitect      wifi_########_managed_psk
    GLOBAL@unifi         wifi_########_managed_psk
    ScienceExplorer      wifi_########_managed_psk
    HUAWEI-B618-1492     wifi_########_managed_psk
    TMSSB2016            wifi_########_managed_psk
    Myreka Office        wifi_########_managed_psk
    pgtopteam            wifi_########_managed_psk
```

6 Select and copy the desired SSID key, type **connect** and paste the selected SSID key. For example:

connect wifi_1234567890_managed_psk

Note on Windows select the key and right-click. On Linux and Mac, you may use middle-click. Enter the SSID passkeys if needed. The result should say "Connected ...".

```
Agent RequestInput wifi_1234567890_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory, Alternates=[ WPS ] ]
  WPS = [ Type=wpspin, Requirement=alternate ]
Passphrase? w1f1p@55w0rd
Connected wifi_1234567890_managed_psk
```

You may connect to a different Access Point using this method.

NOTE

The WLAN network id can be copy and pasted by using the mouse to highlight the section. On a Windows or PuTTY system, right-click the mouse to paste or the middle-mouse-click on a Linux system.

It might take two to three minutes to connect to the WLAN network.

7 Type **Ctrl + C** to exit **connmanctl**. Verify your connection with **ping** by entering **ping www.keysight.com** in PuTTY. Press the Ctrl + C on the keyboard to stop the ping process.

```
debian@beaglebone:/$ ping www.keysight.com

PING e7793.x.akamaiedge.net (23.66.248.80) 56(84) bytes of data.

64 bytes from a23-66-248-80.deploy.static.akamaitechnologies.com
(23.66.248.80): icmp_seq=1 ttl=52 time=102 ms

64 bytes from a23-66-248-80.deploy.static.akamaitechnologies.com
(23.66.248.80): icmp_seq=2 ttl=52 time=125 ms

64 bytes from a23-66-248-80.deploy.static.akamaitechnologies.com
(23.66.248.80): icmp_seq=3 ttl=52 time=256 ms

64 bytes from a23-66-248-80.deploy.static.akamaitechnologies.com
(23.66.248.80): icmp_seq=4 ttl=52 time=182 ms

64 bytes from a23-66-248-80.deploy.static.akamaitechnologies.com
(23.66.248.80): icmp_seq=5 ttl=52 time=102 ms

64 bytes from a23-66-248-80.deploy.static.akamaitechnologies.com
(23.66.248.80): icmp_seq=6 ttl=52 time=127 ms

^C

--- e7793.x.akamaiedge.net ping statistics ---

6 packets transmitted, 6 received, 0% packet loss, time 5007ms

rtt min/avg/max/mdev = 102.375/149.384/256.170/54.741 ms

^Cdebian@beaglebone:/$
```

You might see error or failure in name resolution possibly due to your local network firewall. In this case, it is recommended to use your own mobile hotspot as the internet access for BeagleBone.

NOTE

In case you run into the following problem while setting up WLAN for example

```
connmanctl> scan wifi

Error /net/connman/technology/wifi: Did not receive a reply.
```

Possible causes include: the remote application did not send a reply, the message bus security policy blocked the reply, the reply timeout expired, or the network connection was broken. Try the steps below.

```
connmanctl> tether wifi disable

Disabled tethering for wifi

connmanctl> enable wifi

Error wifi: Already enabled

connmanctl> scan wifi

Scan completed for wifi
```

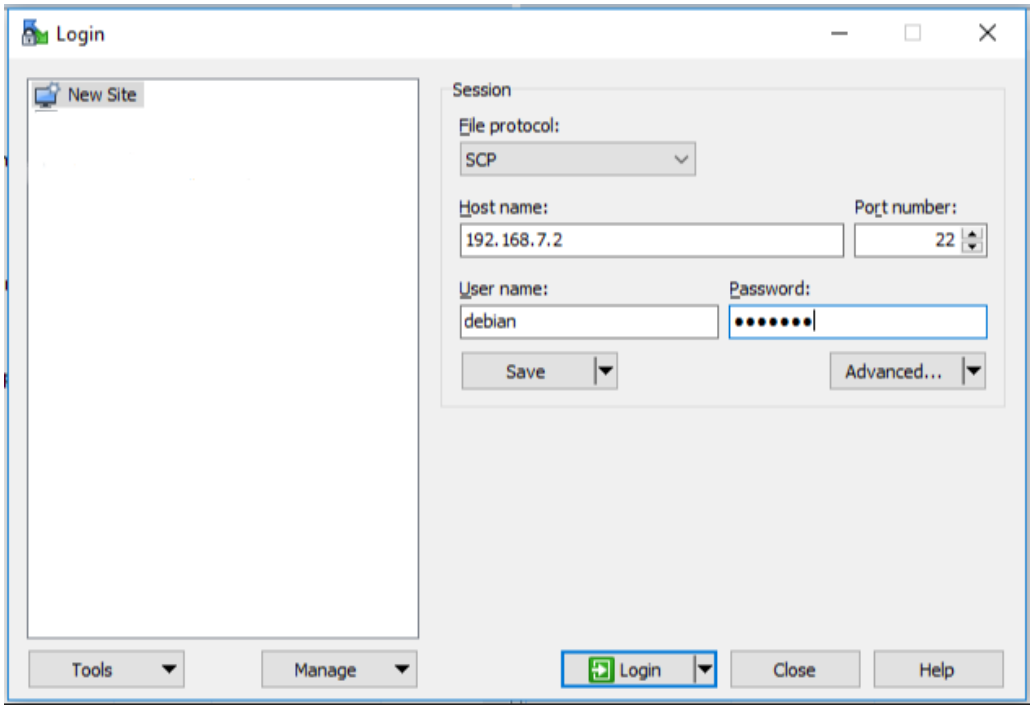
NOTE After power or reset, the boot process may take some time to complete before the 192.168.7.2 port becomes active.

Set Up WinSCP

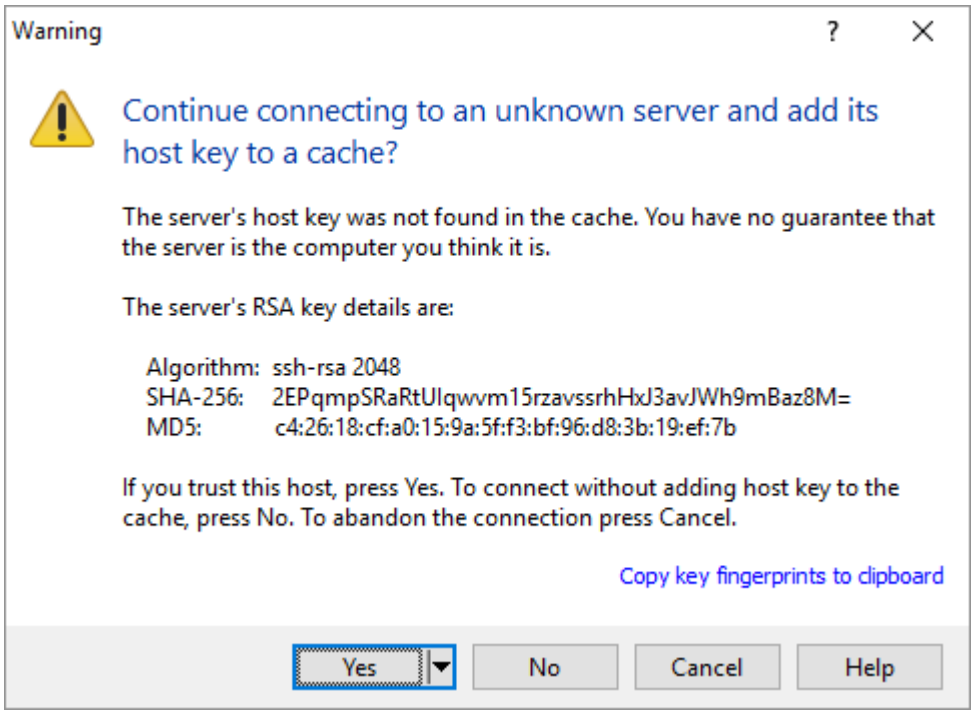
1 For Windows users, download and install a copy of WinSCP from https://winscp.net/eng/download.php. You should see a WinSCP icon on your desktop.

2 Double-click to launch WinWCP and click **New Site**. Then, configure the new site with the following settings.

File Protocol	SCP
Host name	192.168.7.2
Port Number	22
Username	debian
Password	temppwd

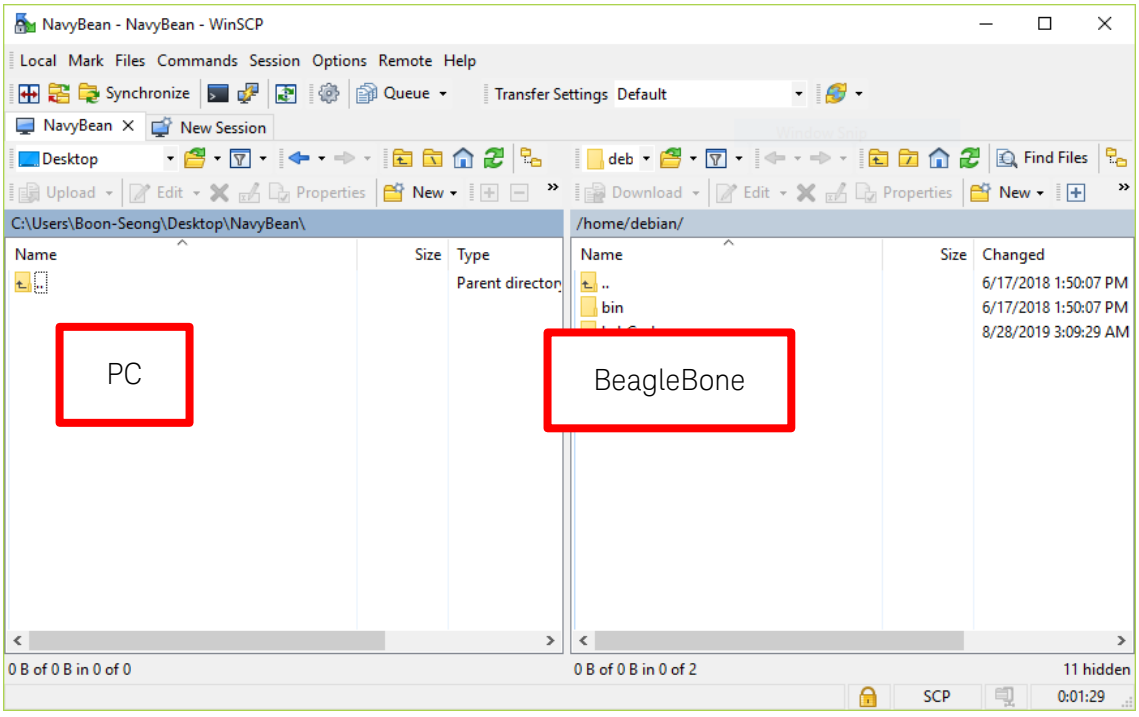


3 First time users who are connecting WinSCP to the BeagleBone, select **Yes** when prompted with a message about connecting to an unknown server.

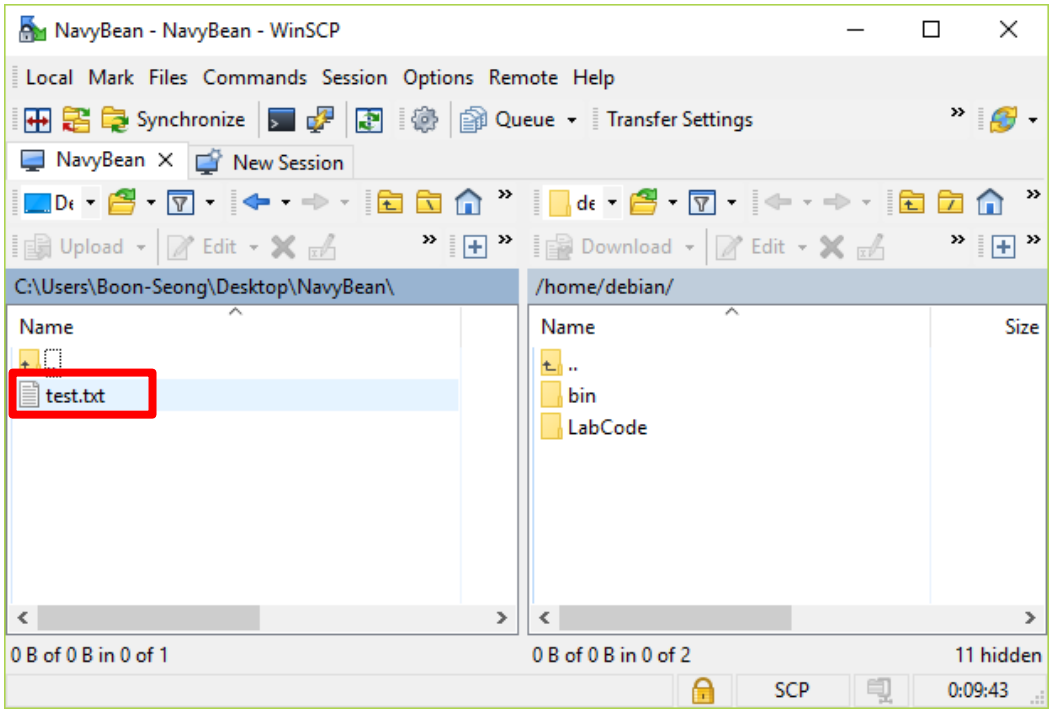


Copy Files with WinSCP

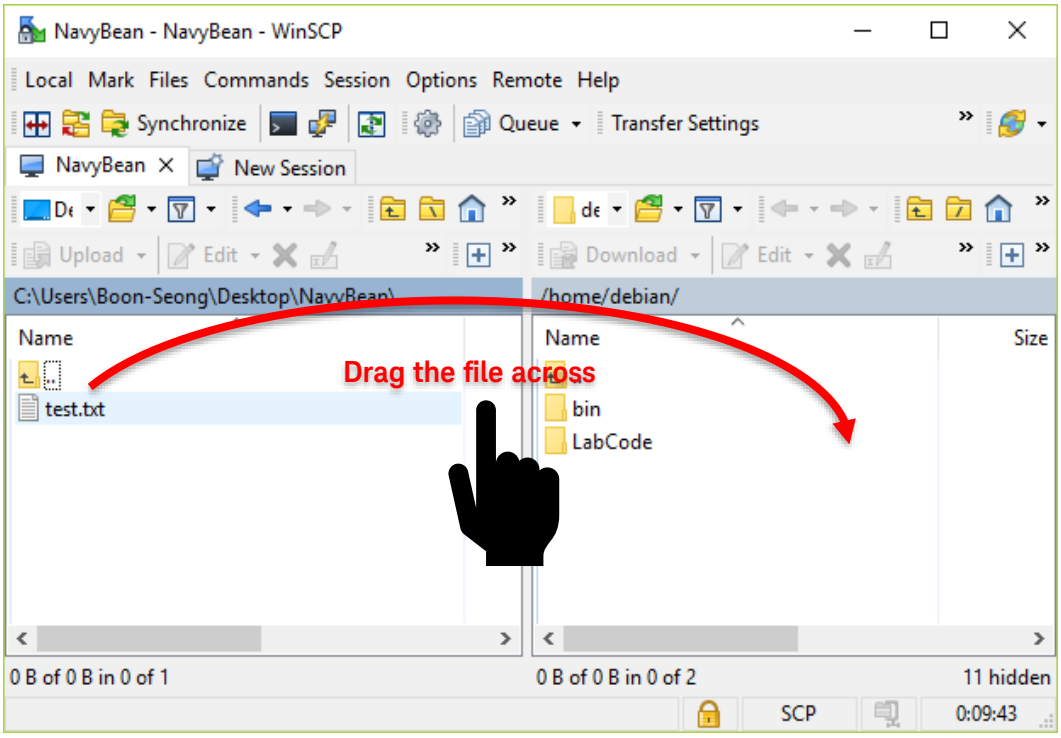
You should see the GUI below where you can drag files across, to transfer it from the PC to the BeagleBone and vice-versa.



4 On your desktop, create a text file “test.txt”.



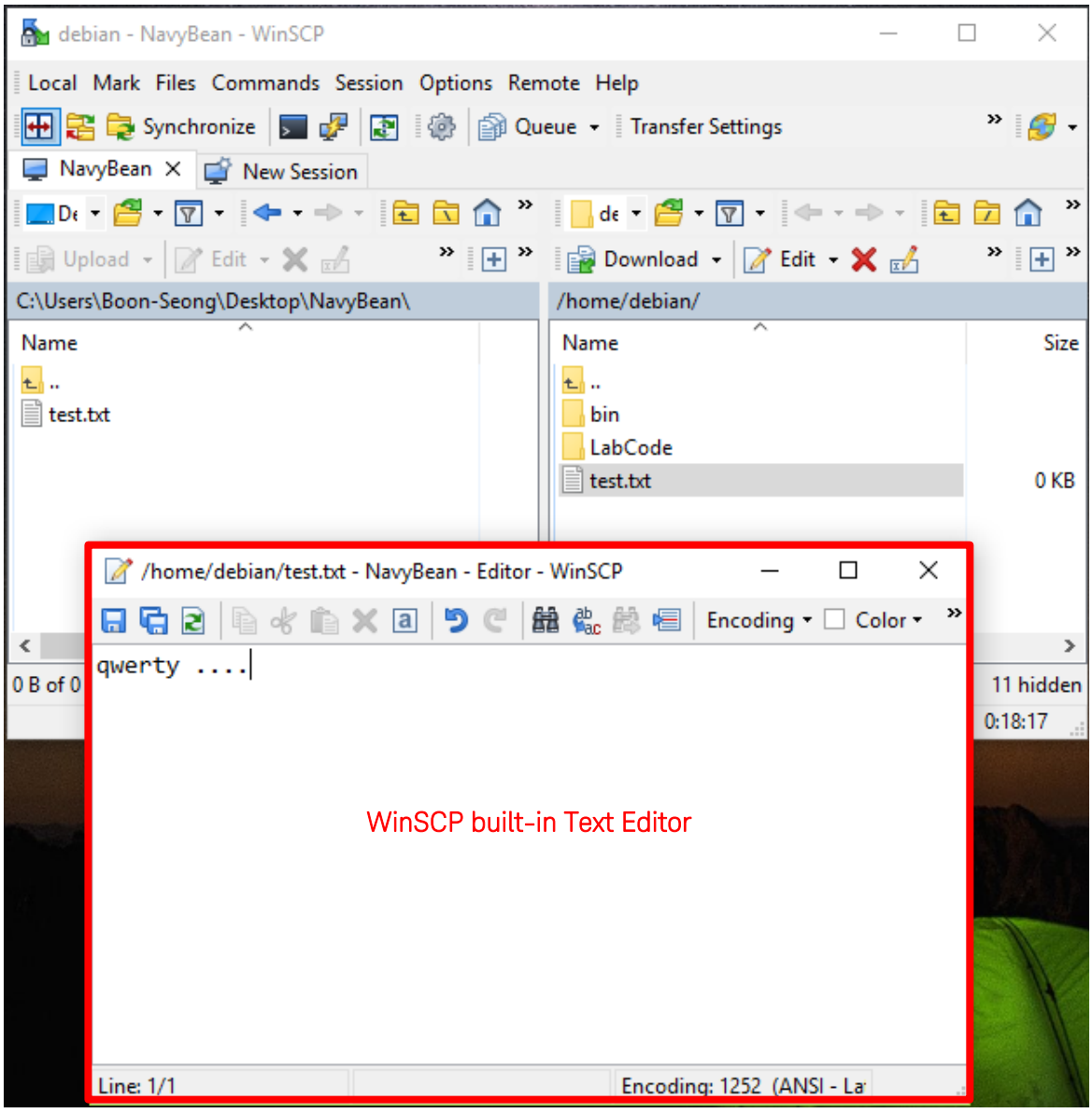
5 Drag the text.txt file across in WinSCP to copy it over to the BeagleBone.



NOTE For Linux based systems, copy the file using **`scp M1-L1.zip debian@192.168.7.2`** command.

Edit Files with WinSCP

6 With the copy of the test.txt file in BeagleBone, right-click the file and click **Edit**. It should prompt a built-in text editor where you will use it to edit shell scripts with a GUI text editor from PC.



7 Save your changes and close the text editor.

NOTE It is recommended to save your changes frequently as you edit the file, to minimize the risk of losing your changes when there are any disconnection between your PC and BeagleBone.

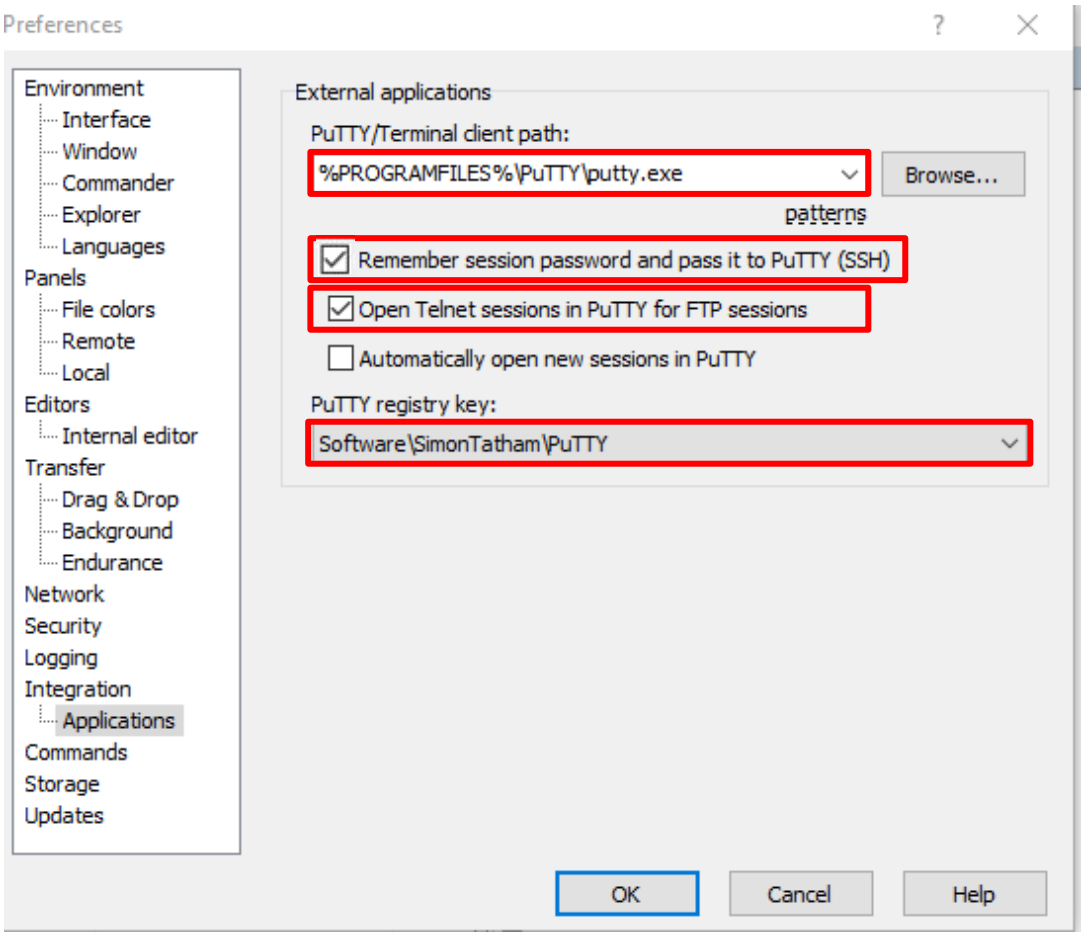
Start PuTTY SSH Connection from WinSCP

If you have PuTTY software installed in your Windows PC, you can integrate PuTTY to WinSCP to easily start the SSH connection with BeagleBone without having to set up the connection properties in PuTTY.

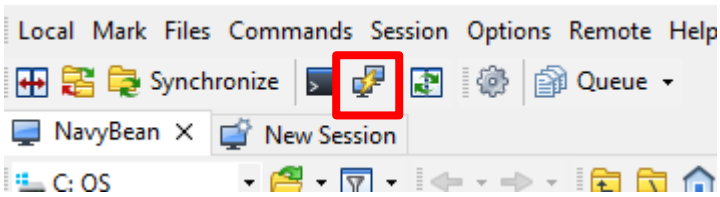
1 In your WinSCP window, go to **Options** > **Preferences**.

2 Go to **Integration** > **Applications**.

3 In the window below, ensure that your settings are as follows. Click **OK** to apply these settings.



4 Click the PuTTY button and PuTTY will automatically log in to BeagleBone with the credentials used in WinSCP.



Use USB Memory Stick to transfer files

This task is optional and is only recommended when there are no direct or wireless connections available to the system. Another method would be to use USB memory stick to transport files.

1 If you are not already logged in, use PuTTY to log in into the BeagleBone module with the following details.

Host Name	192.168.7.2
Port	22
Connection type	SSH
Username	debian
Password	temppwd

NOTE The connection from PuTTY to the BeagleBone uses USB RNDIS device network which is a virtual Ethernet network so that you can use TCP connection to BeagleBone. This method is used since USB2 and its UART will be required for communication to the Xbee3 module in this lab. Use of an IDE is also possible and is covered in the Appendix B – Cloud 9 IDE Usage.

2 With the BeagleBone powered up with a PuTTY terminal open, insert the USB stick into a USB slot on the BeagleBone.

3 Wait for a few seconds before you enter the **`lsblk`** command to list the available devices. You should see similar results as below.

```
debian@beaglebone:~$ lsblk
NAME         MAJ:MIN   RM  SIZE RO TYPE MOUNTPOINT
sda            8:0      1   15G  0 disk
└─sda1         8:1      1   15G  0 part
mmcblk1boot0  179:8     0   2M   1 disk
mmcblk1boot1  179:16    0   2M   1 disk
mmcblk1       179:0     0   3.6G 0 disk
└─mmcblk1p1   179:1     0   3.6G 0 part /
debian@beaglebone:~$
```

NOTE Try to run the **`lsblk`** command before you plug in your USB stick and run the command again with the USB plugged in to know which *sd-* designator is assigned to the USB stick memory.

4 The partition that needs to be mounted is partition 1 of device sda. Use the following command to mount the USB memory stick:

`sudo mount /dev/sda1 /mnt`

5 You can view the content in the USB memory stick on ls /mnt. You can copy files from /mnt using the cp command to your home directory.

NOTE Many systems will automatically mount into /media/usb. To manually mount to this area, first the directory needs to be made and then mount the USB stick to this area.

```
debian@beaglebone:~$ sudo mkdir /media/usb
debian@beaglebone:~$ sudo mount /dev/sda1 /media/usb
```

6 Once the USB memory stick is no longer needed, issue the sudo **umount /mnt** command on this device to flush all the buffers and close the device.

You have now completed the setup of your U3810A.

Students may begin on Lab 1. If the kit has been used by a student before setup and you performed this setup to return the U3810A to factory configuration, it is suggested that you run at least the first test in the next section.

Hardware Verification

WARNING Do not connect voltages greater than 3.3 V to GPIO pins as this may damage the BeagleBone CPU. These over-voltage sources include the VIN pin on the Arduino Shield and DC Power connectors, and +5VRAW and +5VSYS on interface connectors such as J10, JP55, and TP51.

1 If you are continuing this lab from previous lab session;

   a Ensure your U3810A jumper settings are set up according to Main Jumpers
   b Connect your PC to the BeagleBone with a USB cable. Refer to Establish Secure Shell (SSH) Communication between BeagleBone and PC

2 Establish a secure shell communication with the BeagleBone. Refer to latter part of Establish Secure Shell (SSH) Communication between BeagleBone and PC

3 After you have logged into the BeagleBone, run the following commands to go to M1-L1 LabCode directory.

   cd /home/debian/LabCode/M1-L1

   ls -l

   After you have gone into the M1-L1 directory, you should able to see this list of files.

```
debian@beaglebone:~/LabCode/M1-L1$ ls -l
total 76
-rwxr-xr-x 1 debian debian  993 Aug 28  2019 compileCode.sh
-rw-r--r-- 1 debian debian 6841 Aug 28  2019 M1_L1_AccelDisplay.c
-rw-r--r-- 1 debian debian 6152 Aug 28  2019 M1_L1_ADTempDisplay.c
-rw-r--r-- 1 debian debian 3672 Aug 28  2019 M1_L1_ATempDisplay.c
-rw-r--r-- 1 debian debian 2773 Aug 28  2019 M1_L1_ButtonTest.c
-rw-r--r-- 1 debian debian 8026 Aug 28  2019 M1_L1_GyroscopeDisplay.c
-rw-r--r-- 1 debian debian 4834 Aug 28  2019 M1_L1_LCDAnimation.c
-rw-r--r-- 1 debian debian 5293 Aug 28  2019 M1_L1_PressureDisplay.c
-rw-r--r-- 1 debian debian 2953 Aug 28  2019 M1_L1_RelayTest.c
-rw-r--r-- 1 debian debian 7003 Aug 28  2019 M1_L1_RGB_LED_PWM.c
-rw-r--r-- 1 debian debian 2218 Aug 28  2019 M1_L1_TMP36.c
lrwxrwxrwx 1 debian debian   27 Aug 13 20:00 mraa_beaglebone_pinmap.h ->
../mraa_beaglebone_pinmap.h
```

4 The code for this lab can be all compiled by using a shell script. However, the permissions must be set to allow execution of the script. Run the following command to enable execution.

   chmod 755 compileCode.sh

5 Once the permissions have been changed, type **./compileCode.sh** to execute the code. This command will check and ensure that proper MRAA libraries are installed before compiling the code needed in this lab.

```
debian@beaglebone:~/LabCode/M1-L1/M1-L1$ ./compileCode.sh
Starting
Found the proper mraa version

       ***** Compiling LCD Animation
       ***** Compiling Button Test
       ***** Compiling RelayTest
       ***** Compiling TMP36
       ***** Compiling Analog Temp Display
       ***** Compiling Analog Digital Temp display
       ***** Compiling Pressure Display
       ***** Compiling Accel Display
       ***** Compiling RGB_LED_PWM
       ***** Compiling GyroscopeDisplay

Compile complete
```

Test the LCD Display

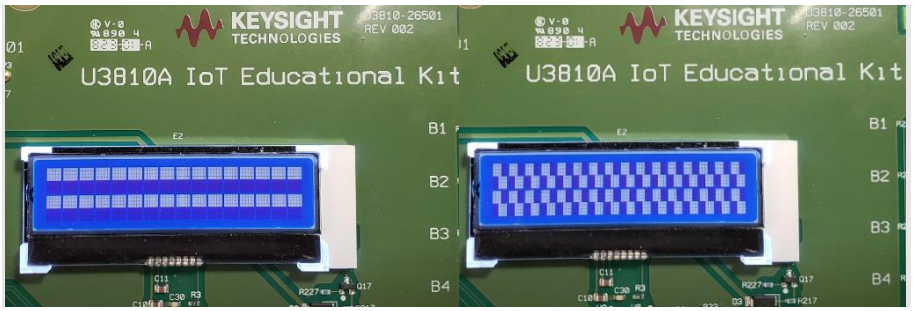
In this task you will compile and run the first program. This is a program that checks the I²C bus and tests the LCD display by an animation of the characters.

1 Type **./LCDAnimation** command to run the LCD Animation code.

NOTE If the file does not exist, enter the following command to recompile the code.

```
gcc M1_L1_LCDAnimation.c -l mraa -o LCDAnimation
```

When the program is executed, you should see an animation on the LCD as shown below.



NOTE If the I²C bus is missing, this error message will be displayed:

```
The I2C Bus 2 is not available. Please check /dev/i2c-2
```

If the LCD display is not detected by the I²C bus, this error message will be displayed:

```
Failed to initialize display
Check i2cdetect -r -y 2 for 0x3E
```

---

2 Press **Ctrl + C** in the PuTTY window to stop the LCD animation program.

```
debian@beaglebone:~/LabCode/M1-L1$ ./LCDAnimation
LCD Animation Program Running...Hit Control-C to Exit.
```

Test the U3810A GPIO Using Buttons and a Relay.

In this section will examine the GPIO functions of the BeagleBone. First, run a program to test out the buttons on BeagleBone. Next, you are going to power up the relay circuit and run another program to control the relay circuit by turning on and off using the button on the BeagleBone.

1 Run the **./ButtonTest** command to run the Button Test code.

NOTE If the file does not exist, enter the following command to recompile the code.

```
gcc M1_L1_ButtonTest.c -l mraa -o ButtonTest
```

NOTE The program will run and if the I²C bus is missing or the display is not detected on the bus and error message will appear.

2 Press each one of the buttons and verify the display shows which button is “on”.

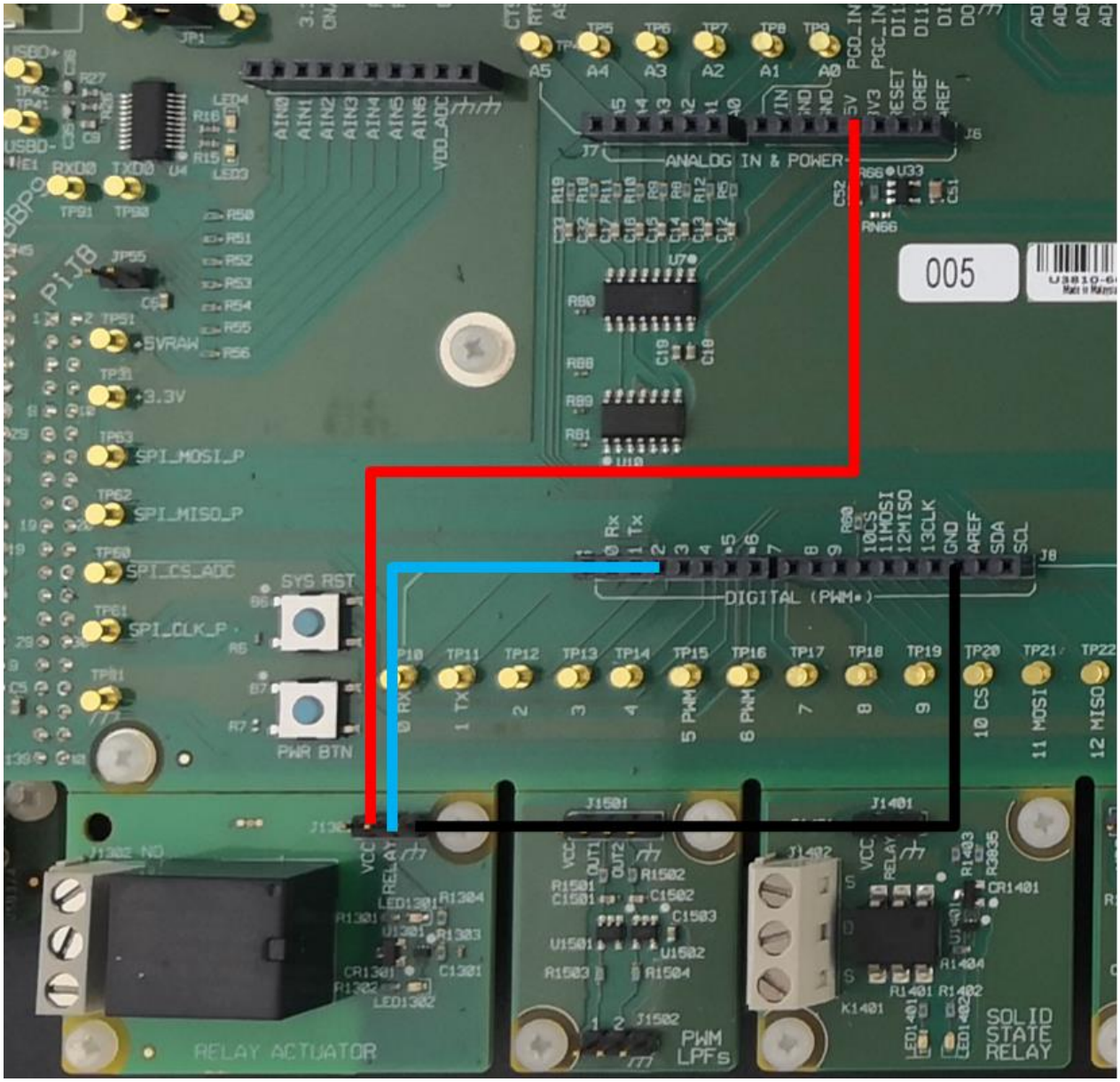
3 Try to press multiple buttons and all the buttons that are pressed should show. When the ButtonTest code is executed, the LCD will display the B1, B2, B3 or B4 buttons depending on which is pressed.

4 Press **Ctrl + C** in the PuTTY window to stop the Button Test program.

Next, you will control a relay using buttons. To do this, the relay will need +5V, ground, and a control signal. The GP2 on **J9 Pin 3 (2)** will provide the control signal. When this signal is asserted low, the relay will turn on.

5 Connect J6, J8, and J9 to the RELAY ACTUATOR according to the table below:

From	To
J8 (GND)	RELAY ACTUATOR J1301 (GND)
J9 (2)	RELAY ACTUATOR J1301 (RELAY)
J6 (5V)	RELAY ACTUATOR J1301 (VCC)



The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

6 Type **./RelayTest** to run the Relay Test code.

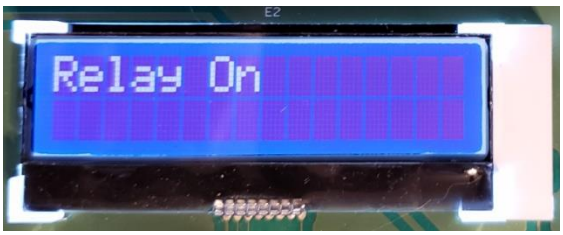
NOTE

If the file does not exist, run the following command to recompile the code.

```
gcc M1_L1_RelayTest.c -l mraa -o RelayTest
```

When you see the following messages in your PuTTY terminal, press **Ctrl + C** to stop the program and re-run it again.

Pressing the B1 button should turn on the relay, you should able to hear the relay switch clicks and LED1302 light up. Press the B2 button to turn it off.



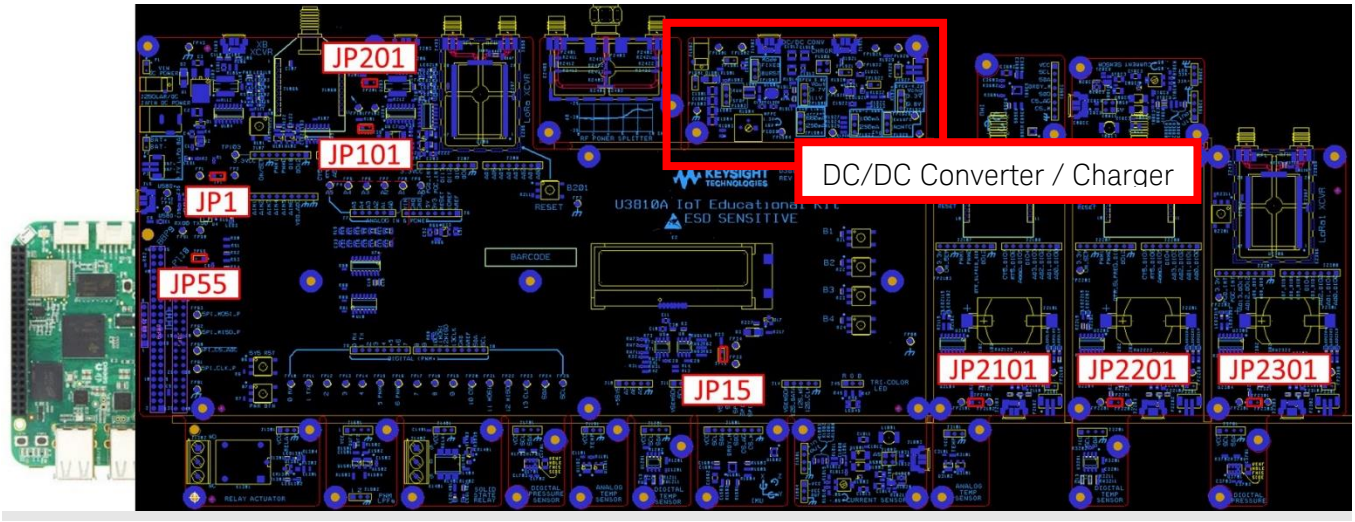
7 Press **Ctrl + C** to exit the RelayTest program and disconnect all jumper wires.

Test SPI by Reading sensor data and display on LCD

NOTE For this part, the JP15 Jumper must be in place.

Before you begin, configure the Keysight U3810A as a “cape” on top of the BeagleBone CPU, and with the jumper configuration shown below:

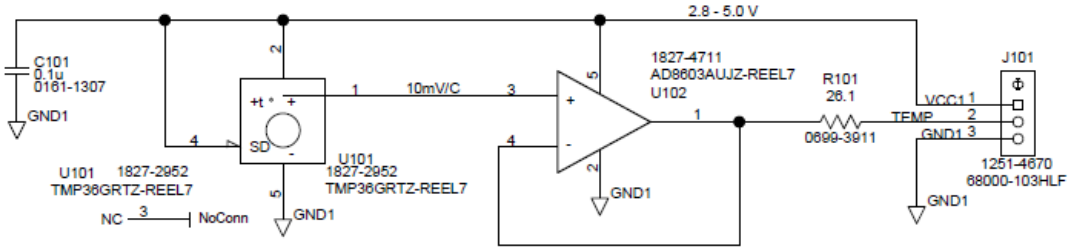
Jumper	JP1	JP15	JP55	JP101	JP201	JP2101	JP2201	JP2301
Name	Input Current	Sensor Current	+5VSYS +5VRAW	XB Current	LoRa Current	XB1 Current	XB2 Current	LoRa1 Current
Position	In place	In place	Removed	In place	In place	In place	In place	In place



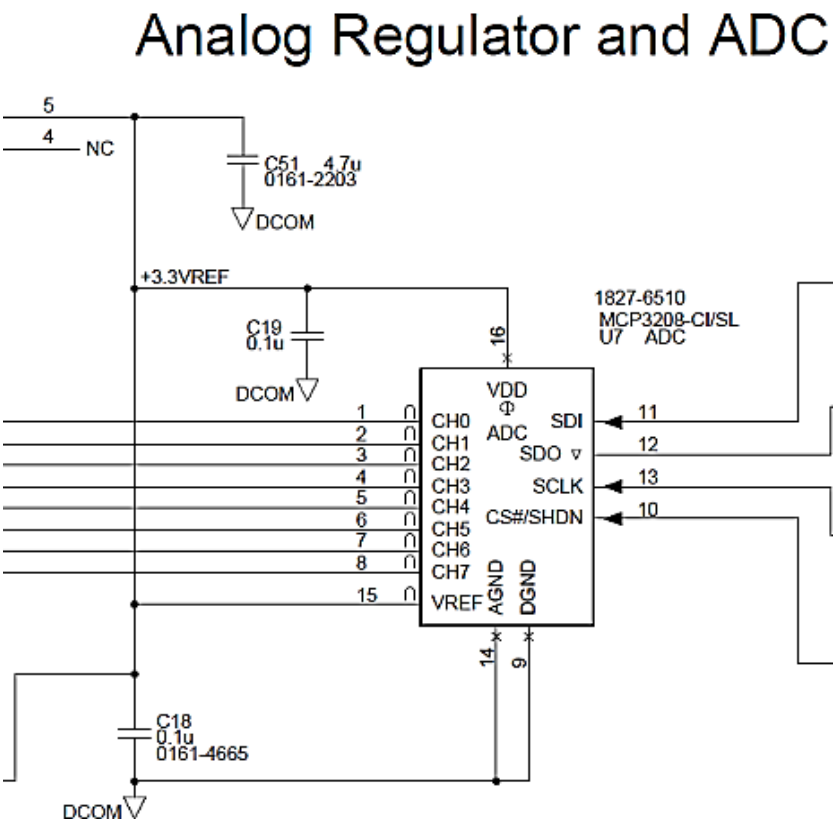
The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

In this section, you are going to test run a ready-made application to capture returned data from the TMP36 analog temperature sensor and display the results on the LCD. The TMP36 analog temperature sensor is a precision integrated-circuit temperature device that operates at 2.7 to 5.5 V and produces an output voltage linearly proportional to the temperature. It measures temperature from -50 °C to 150 °C and produces output voltages from 0.0 to 2.0 Volts. Every 10-mV change in the output voltage represents a temperature change of 1 °C. The TMP36 is based at .5V where 0'C = 500mV

V_{TMP36} = .5V + Temp x .01V/C

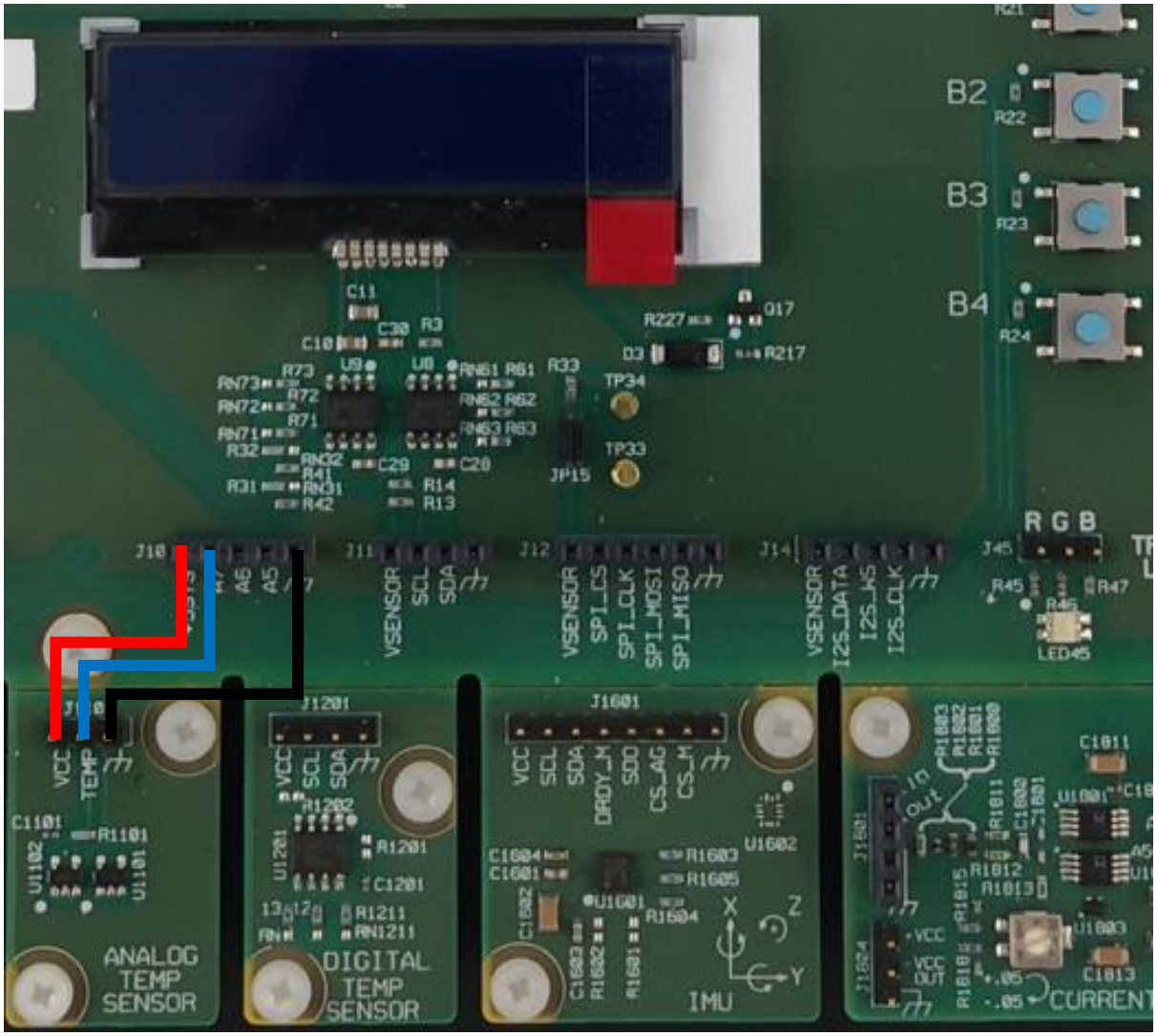


The schematic of the connection is shown below, with the sensor connected to CH7 (A7):



1 Connect the TMP36 ANALOG TEMP SENSOR pins to U3810A J10 pins with jumper wires according to the table and image below.

From	To
J10 (+5SYS)	ANALOG TEMP SENSOR J1101 (VCC)
J10 (A7)	ANALOG TEMP SENSOR J1101 (TEMP)
J10 (GND)	ANALOG TEMP SENSOR J1101 (GND)



The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

2 Run the compiled program tmp36 to get the ADC value reading.
./tmp36

NOTE If the file does not exist, enter the command below to recompile the code.

gcc M1_L1_TMP36.c -l mraa -o tmp36

You should be able to see the returned value from the sensor in the PuTTY terminal program. Take note that the returned value is shown as a digital signal from a scale of 0 to 4095, which represents 0 to 3.3 V. For example, when the TMP36 returns a signal with 0.733 V, it will appear as 911 in the PuTTY terminal.

```
debian@beaglebone:~/LabCode/M1-L1$ ./tmp36
ADC Value at Channel 7: 911
debian@beaglebone:~/LabCode/M1-L1$
```

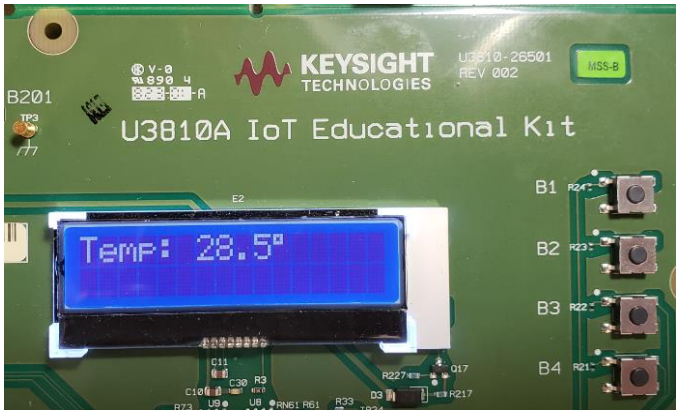
3 The U3810A also has the capability to display a message on the onboard LCD. To display the temperature sensor on the LCD display, compile the ATempDisplay.c inside BeagleBone CPU. Enter the command below.

gcc M1_L1_ATempDisplay.c -l mraa -o ATempDisplay

4 Enter the command below to run the compiled program. The temperature detected by the sensor will appear on the LCD display.

./ATempDisplay

5 Touch the sensor with your fingertip to increase the temperature reading. The temperature reading will increase due to your body temperature.



6 Press **Ctrl + C** to stop the measurement.

Test I²C bus using the LM75 Temperature Sensor

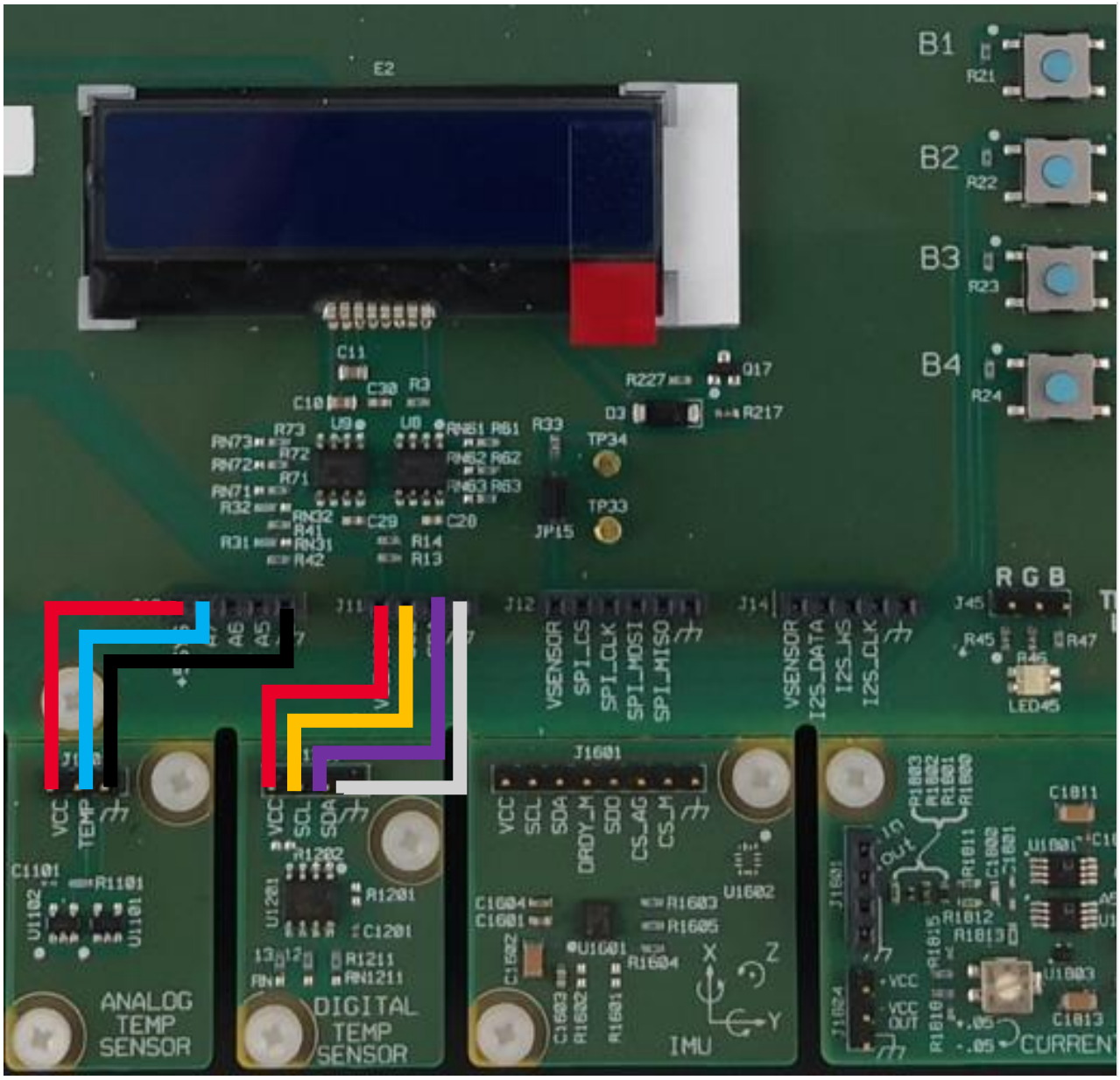
I²C bus devices are located at /dev/i2c*. On the BeagleBone used with the U3810A I²C Bus 1 is connected to the Arduino J8 connector and the J11 connector. The I²C Bus 2 is connected to the LCD Display and the U3810A EEPROMs.

In this section you are going to connect an I²C device to the U3810A. First the connection will be verified with Linux command line functions. Then, run a program that will compare the analog sensor to the digital sensor.

1 Keep the ANALOG TEMP SENSOR wires connected and connect the DIGITAL TEMP SENSOR to J11 according to the table below.

From	To
J11 (VSENSOR)	DIGITAL TEMP SENSOR J1201 (VCC)
J11 (SCL)	DIGITAL TEMP SENSOR J1201 (SCL)
J11 (SDA)	DIGITAL TEMP SENSOR J1201 (SDA)
J11 (GND)	DIGITAL TEMP SENSOR J1201 (GND)

View the jumper wires connection in the next page.



The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

To know more about ground symbols, refer to Appendix G – Electrical Ground Symbols.

2 Run the following command to verify if the sensor is connected correctly.
`i2cdetect -r -y 1`

The result should only show 0x48 which is the Temperature Sensor as shown below.

```
debian@beaglebone:~$ i2cdetect -r -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

NOTE

- The **`i2cdetect`** command is part of the I²C Tools, used to detect the types of devices present.
- The **`i2cdump`** command will dump 256 bytes of data from the specified I²C device.
- The **`i2cget`** gets one byte of an I²C device for its register space.
- The **`i2cset`** can set one byte of an I²C device.

As a test to display the data in the temperature sensor, run the **`i2cdump -y 1 0x48`** command.

```
debian@beaglebone:~$ i2cdump -y 1 0x48
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
10: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
20: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
30: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
40: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
50: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
60: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
70: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
80: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
90: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
a0: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
b0: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
c0: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
d0: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
e0: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
f0: 1c 00 4b 50 ff ff ff a1 1c 00 4b 50 ff ff ff a1    ?.KP...??.KP...?
```

Command	Description
i2cdetect	Detect I²C chips
i2cdump	Examine I²C registers
i2cget	Read from I²C chip registers
i2cset	Set I²C registers

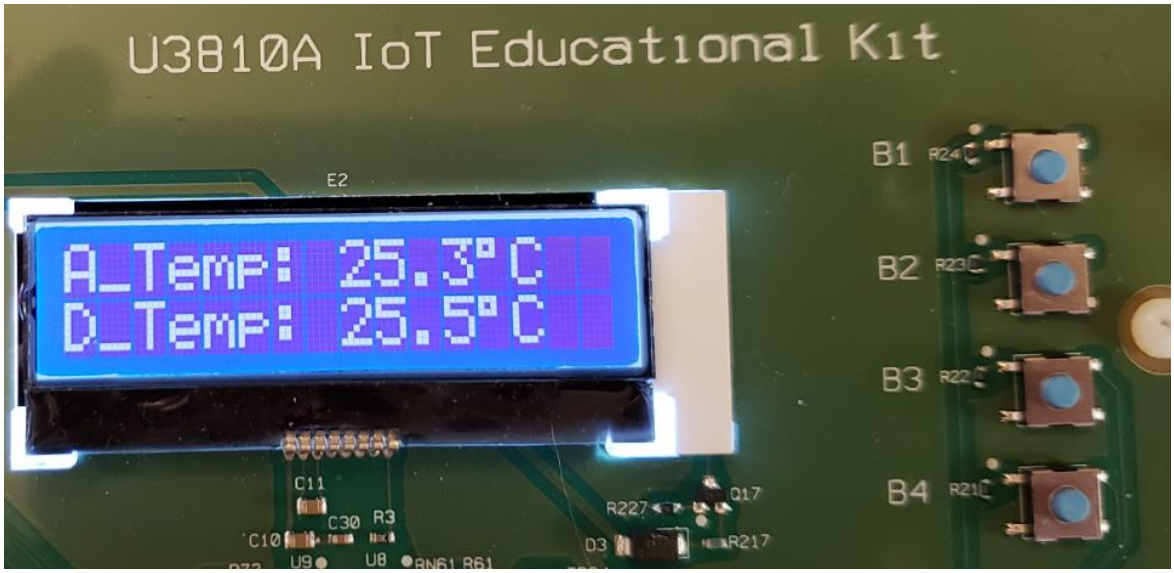
3 Run the compiled program to display both the ANALOG TEMP SENSOR and **DIGITAL TEMP SENSOR** reading on the LCD display.
`./ADTempDisplay`

NOTE

If ADTempDisplay is not already there, use the command below to compile.

`gcc M1_L1_ADTempDisplay.c -l mraa -o ADTempDisplay`

The program will display both TMP36 and LM75 on LCD as shown below.



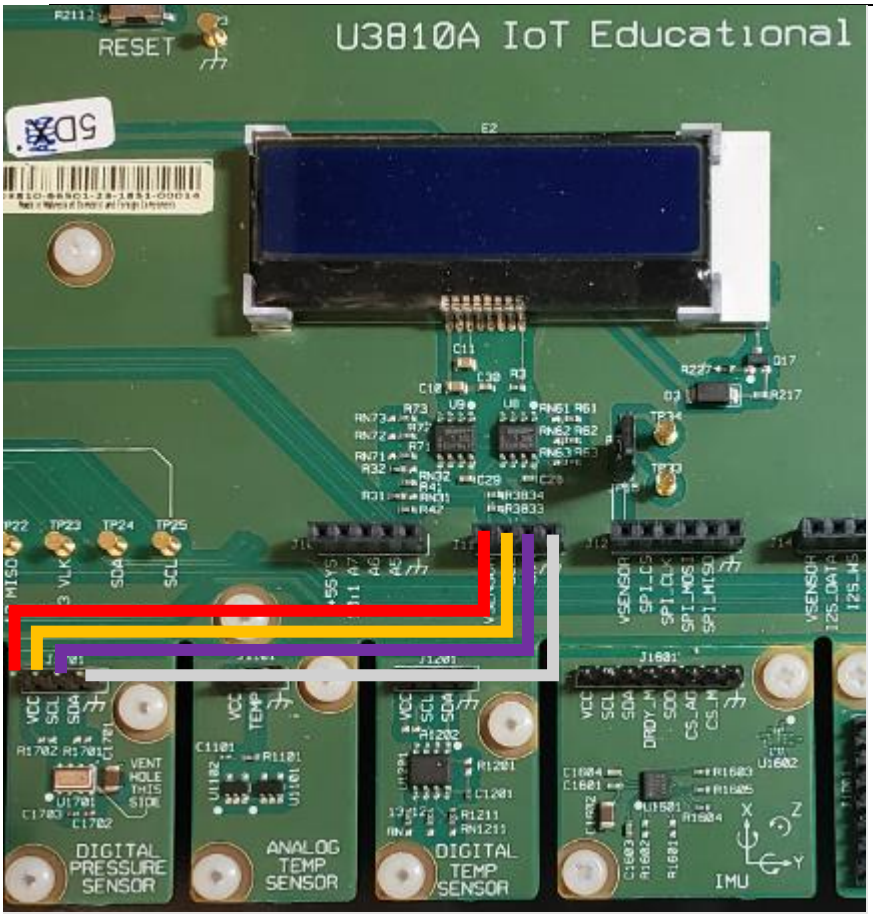
4 Press **Ctrl + C** to exit the program and disconnect all jumper wire connections.

Test I²C bus using the MPL3115A2 Pressure Sensor

In this section, you are going to connect an I²C device to the U3810A. First, the connection will be verified with Linux command line functions. Then, run a program that displays the barometric pressure to the LCD display.

1 Connect the Digital Pressure Sensor to the J11 pins according to the table below.

From	To
J11 (VSENSOR)	DIGITAL PRESSURE SENSOR J1701 (VCC)
J11 (SCL)	DIGITAL PRESSURE SENSOR J1701 (SCL)
J11 (SDA)	DIGITAL PRESSURE SENSOR J1701 (SDA)
J11 (GND)	DIGITAL PRESSURE SENSOR J1701 (GND)



The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

2 Run the following command to test the connection. The result should only show Device 0x60 on the list.

i2cdetect -r -y 1

```
debian@beaglebone:~$ i2cdetect -r -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

3 Run the Barometer code with the command below.
./PressureDisplay

NOTE If the file does not exist, enter the command below to recompile the code.

gcc M1_L1_PressureDisplay.c -l mraa -o PressureDisplay

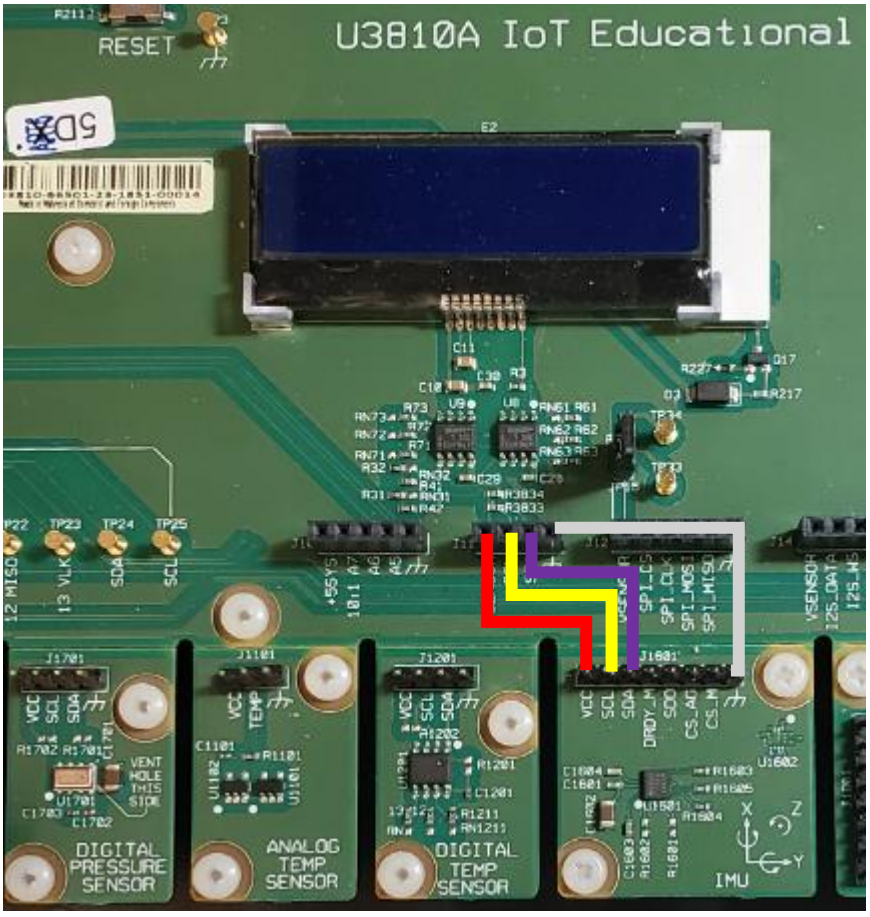
4 Press Ctrl + C to exit the program and disconnect all jumper wires from DIGITAL PRESSURE SENSOR.

Test I²C bus using the LMS9DS1TR IMU Accelerometer.

In this section, you are going to connect an I²C device to the U3810A. First the connection will be verified with Linux command line functions. A program will be run to show the 3-axis acceleration on the LCD display.

1 Connect the IMU to the J11 pins according to the table below. Note the ground pin is not adjacent to the other three pins.

From	To
J11 (VSENSOR)	IMU J1601 (VCC)
J11 (SCL)	IMU J1601 (SCL)
J11 (SDA)	IMU J1601 (SDA)
J11 (GND)	IMU J1601 (GND)



The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

2 Run the following command to test the connection. The result should only show Device 0x6b on the list.

```
i2cdetect -r -y 1
```

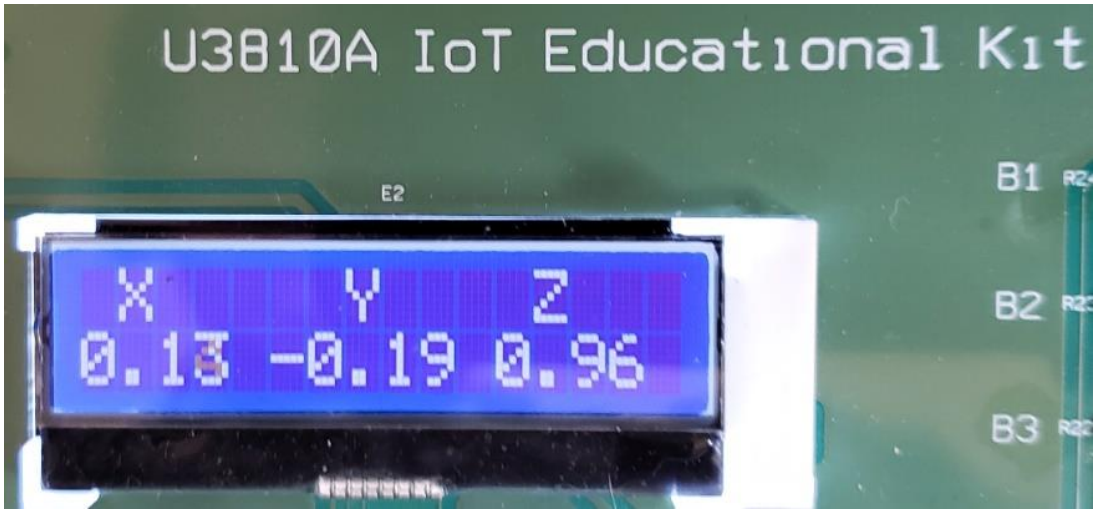
```
debian@beaglebone:~/LabCode/M1-L1$ i2cdetect -r -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 1e --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- 6b -- -- -- --
70: -- -- -- -- -- -- -- --
```

3 Run the Accelerometer code with this command.
./AccelDisplay

NOTE If the file does not exist, enter the command below to recompile the code.

```
gcc M1_L1_AccelDisplay.c -lmraa -o AccelDisplay
```

4 Tilt the board and notice the change 1.0 is equal to 1 g or 9.8m/s^2 due to the earth's gravity.



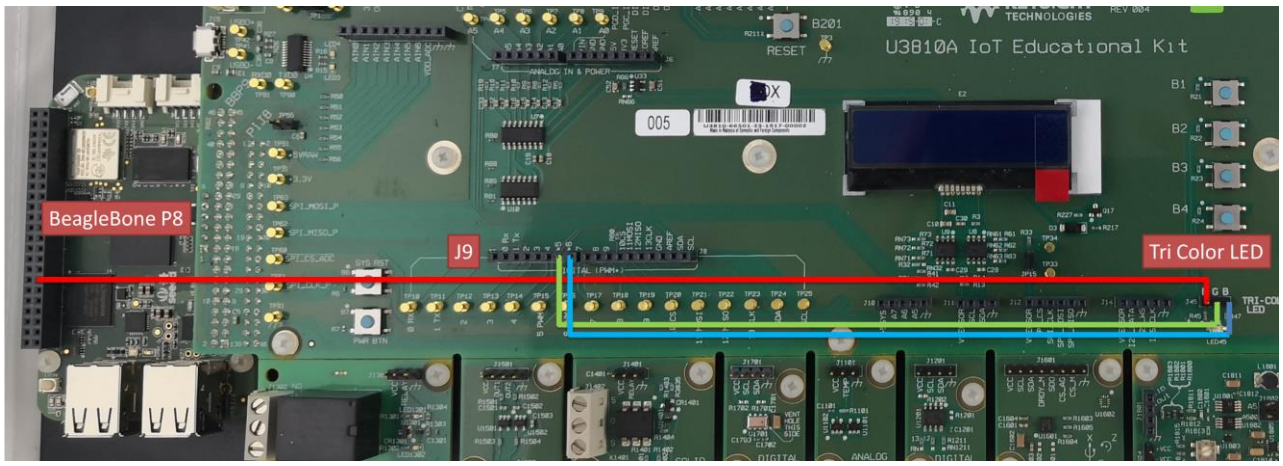
5 Press Ctrl + C to exit the program and disconnect all jumper wires.

Test the PWM functions with the RGB LED.

In this section, you are going to the RGB LED to three different PWM outputs. The PWM outputs will cycle to display the different colors and intensities that can be generated by the RGB LED.

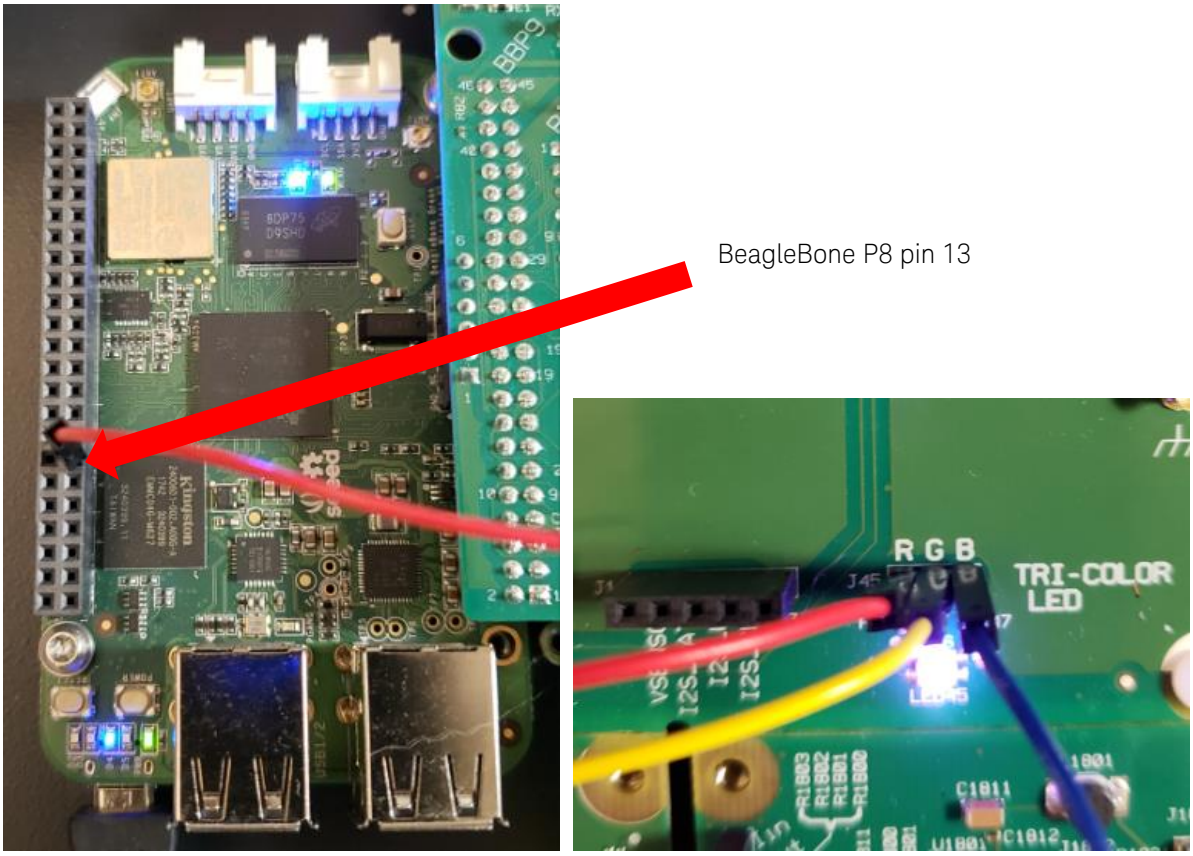
1 Connect the RGB TRI-COLOR LED J45 according to the table below.

From	To
RGB TRI-COLOR LED J45 (R)	BeagleBone P8 pin 13
RGB TRI-COLOR LED J45 (G)	J9 (*5)
RGB TRI-COLOR LED J45 (B)	J9 (*6)



The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

2 Connect a wire from the BeagleBone **P8 pin 13** to the **R** of the **TRI-COLOR LED**.

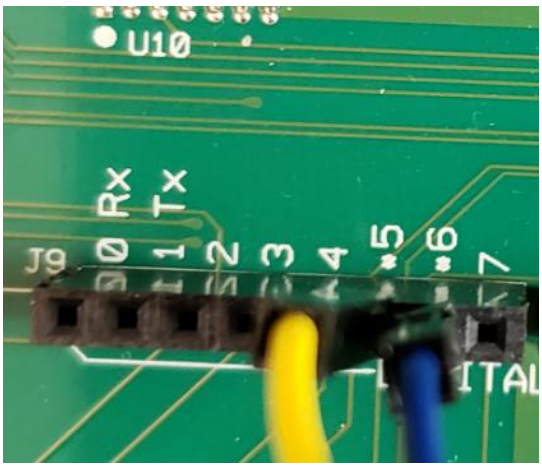


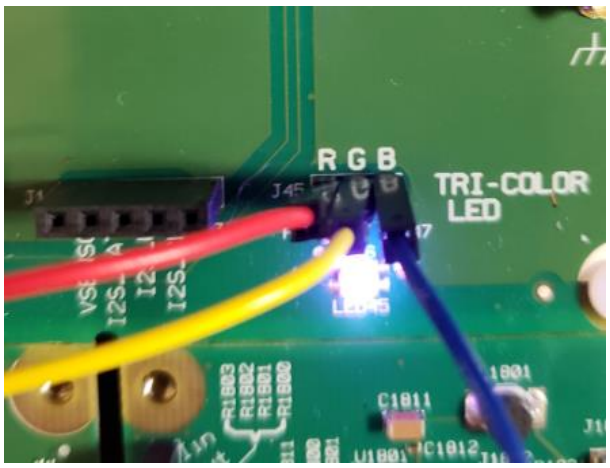
The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

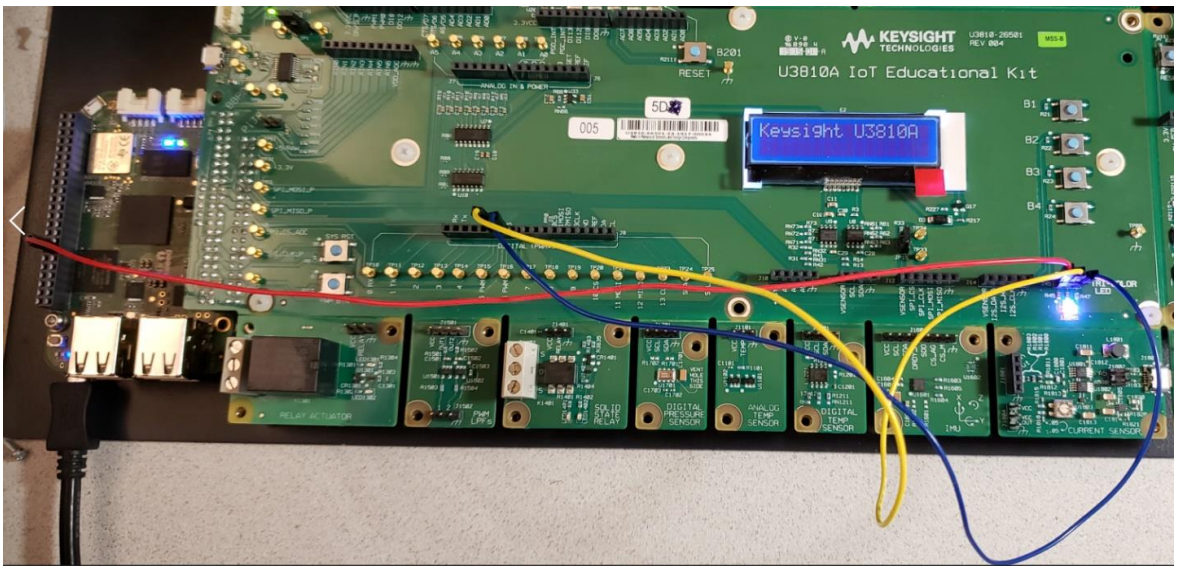
For the location of BeagleBone P8 pin13, refer to Appendix D – BeagleBone Pinouts.

3 Next connect a wire from the *5 pin of J9 to the G of the TRI-COLOR LED.

4 Connect a wire from the *6 pin of J9 to the B of the TRI-COLOR LED.







The diagram above might appear dark in print outs. Refer to Appendix C – Keysight U3810A Technical Documents for the searchable PDF to help you locate the locations of the jumpers, connectors and components.

5 Run the program and observe the different colors on the TRI-COLOR LED. Note this program has a special exit routine to reset the PWM pins back to the default mode.

```
./RGB_LED_PWM
```

NOTE

If the file does not exist, enter the command below to compile the code.

```
gcc M1_L1_RGB_LED_PWM.c -l mraa -o RGB_LED_PWM
```

6 Press Ctrl + C to exit the program and disconnect all jumper wires.

Test the XBee module via BeagleBone Serial Port

1. Enter **`minicom -D /dev/ttyS1 -b 9600`** to open a minicom terminal to /dev/ttyS1.
2. To test the presence of the module, type +++ to get the module into command mode. This will also print back "OK"
3. Press <**CTRL-A**> and then **x** to exit the minicom.

```
debian@beaglebone:~$ minicom -D /dev/ttyS1 -b 9600

Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
Port /dev/ttyS1, 20:30:08

Press CTRL-A Z for help on special keys
+++
   OK
```

Test the LoRa module via BeagleBone Serial Port

1. Open a terminal to /dev/ttyS2 using 57600 baud by entering. **`minicom -D /dev/ttyS2 -b 57600`**
2. With the terminal open, enter a Control-a and the then character e. This will turn on the echo.
3. The LoRa module requires both a Carriage Return and Line Feed at the end of any command. To clear out the buffer, press **Enter** and then a Control-j. You may see "invalid_param"
4. To verify the radio version type **`sys get ver <enter><CTRL-j>`**

```
debian@beaglebone:~$ minicom -D /dev/ttyS2 -b 57600

Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
Port /dev/ttyS2, 19:59:29

Press CTRL-A Z for help on special keys

invalid_param
sys get ver

RN2903 1.0.3 Aug  8 2017 15:11:09
```

5. Press <CTRL-a> and then **x** to exit minicom.

Maintenance

Electrostatic Discharge (ESD) Precautions

Almost all electrical components can be damaged by electrostatic discharge (ESD) during handling. Component damage can occur at electrostatic discharge voltages as low as 50 V.

The following guidelines will help prevent ESD damage during use and service operations:

- Disassemble products only in a static-free work area.
- Use a conductive work area to reduce static charges.
- Use a conductive wrist strap to reduce static charge accumulation.
- Minimize handling.
- Keep replacement parts in original static-free packaging.
- Remove all plastic, foam, vinyl, paper, and other static-generating materials from the immediate work area.

Cleaning

To prevent electrical shock, disconnect the product from AC mains power and disconnect all test leads before cleaning. Clean the outside of the product using a soft, lint-free, cloth slightly dampened with water.

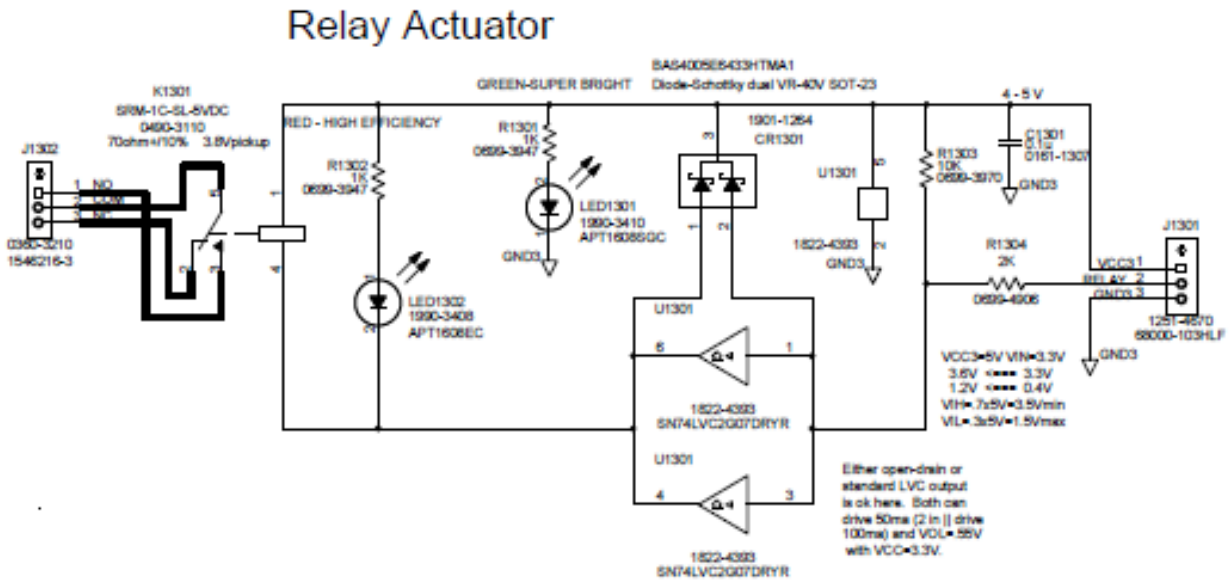
Do not use detergent or solvents.

Do not attempt to clean internally.

If needed, contact a Keysight Technologies Sales and Service office to arrange for proper cleaning to ensure that safety features and performance are maintained.

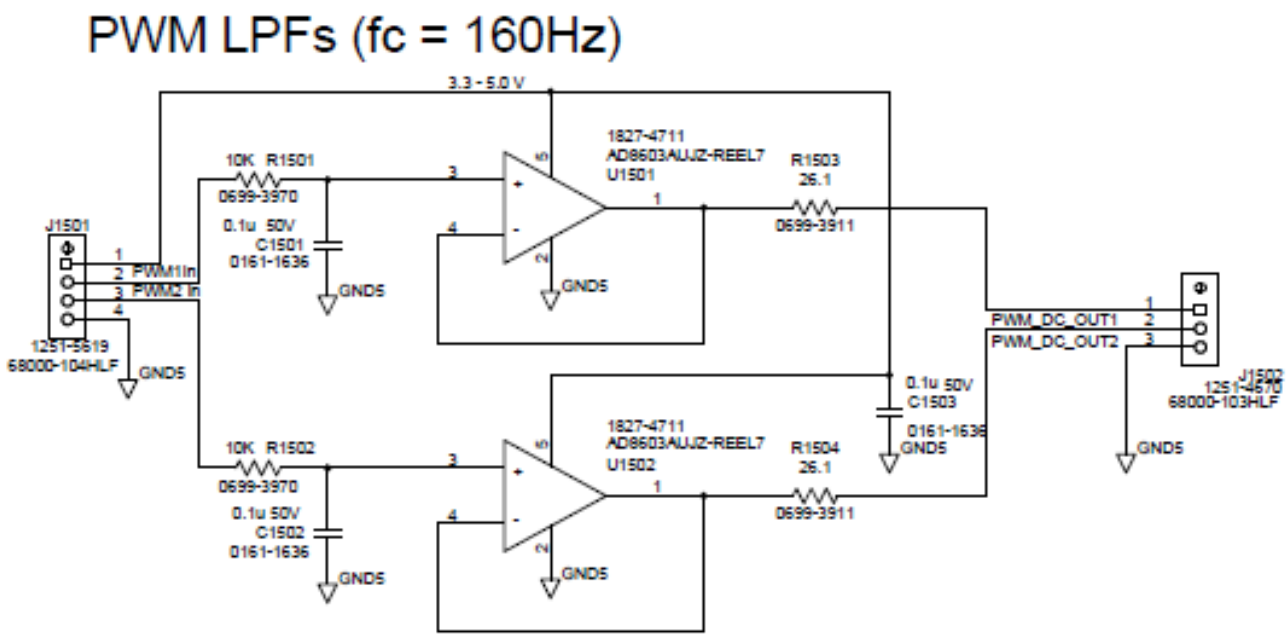
Schematic and Characteristics of Sensors and Actuators

Relay Actuator



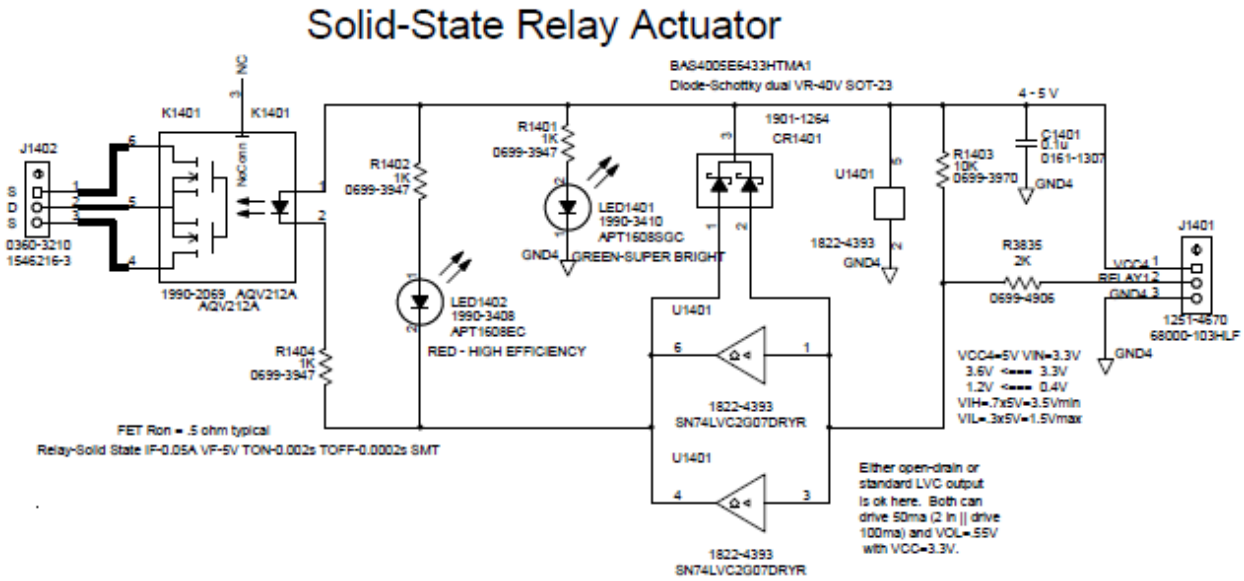
Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	4.0	5.5	5.0	Volts	
Iin	2.0	4.0	3.0	mA	Relay Off
Iin	55.0	90	71.0	mA	Relay On
Vinh	2.1	VCC+0.7		Volts	Input to turn off relay
Vinl	-0.7	1.8		Volts	Input to turn on relay
Operate time		15		ms	
Release time		5		ms	
Rinitial		50		mΩ	
Life expectancy			10^7	ops	< 1A contact current

PWM LPF



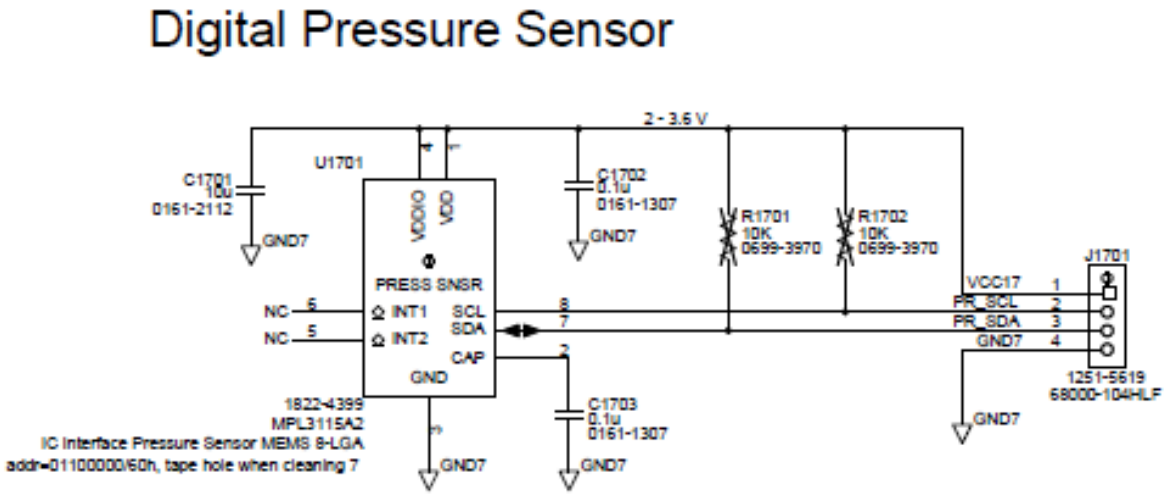
Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	3.3	5.5	5.0	Volts	
Iin		100	80	µA	No load
Filter			160	Hz	3dB, 1 pole RC

Solid State Relay



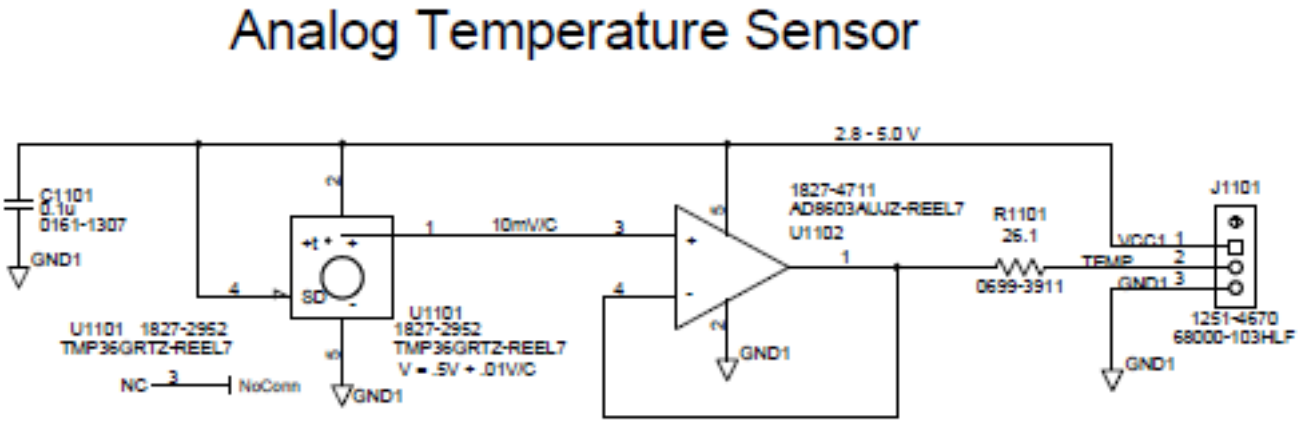
Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	4.0	5.5	5.0	Volts	
Iin	2.0	4.0	3.0	mA	Relay Off
Iin	5.0	8.0	6.0	mA	Relay On
Vinh	2.1	VCC+0.7		Volts	Input to turn off relay
Vinl	-0.7	1.8		Volts	Input to turn on relay
Operate time		2	.65	ms	
Release time		.2	.08	ms	
Rinitial		2.5	.83	Ω	Each FET (2 FETs may be paralleled)
Life expectancy			n/a	ops	No significant degradation with life

Digital Pressure Sensor



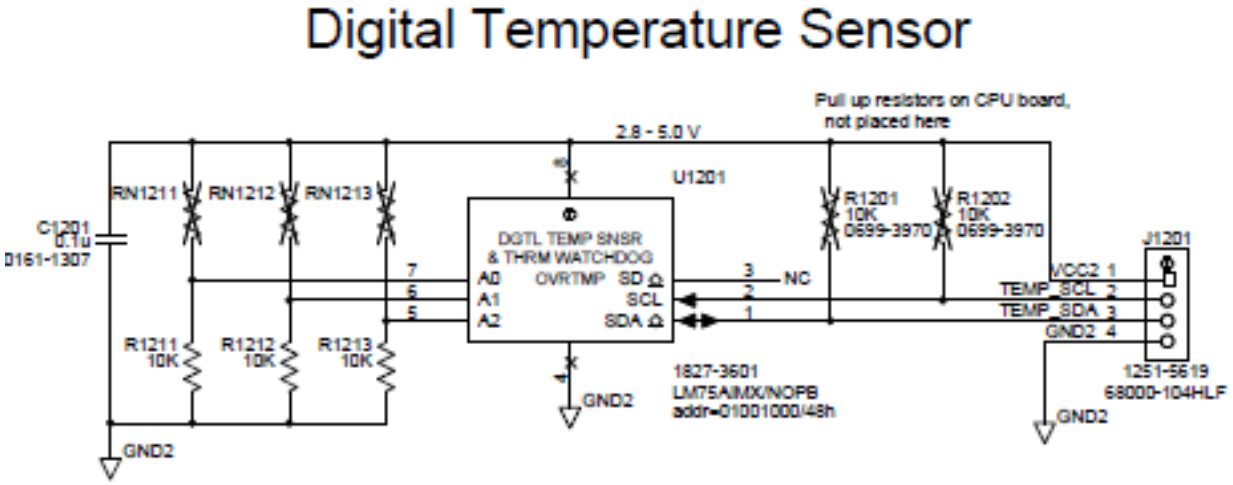
Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	2.0	3.6	3.3	Volts	
Iin		265	40	µA	Relay Off
Vinh			.75	Volts	
Vinl			.3	Volts	
Calibrated range	50	110		kPa	
Noise		19		Pa rms	1x oversample
Absolute accuracy	-.4	.4	+/-.4	kPa	
Relative accuracy			+/-.05	kPa	Relative accuracy during pressure change between 70 to 110 kPa at any constant temperature between –10 °C to 50 °C
Resolution	.25 (.0625)		1.5(.3)	Pa(m)/LSB	
Output data rate			100	Hz	One-shot mode

Analog Temp Sensor



Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	2.7	5.5	3.3	Volts	
Iin		100	80	µA	
V Temp range	0.0	1.5	0.75	V	Output Voltage -40C, 100C, 25C
Accuracy		+/-2	+/-1	C	25C
V Temp			.5+.01/C	V	Vout = .5V + .01V/C

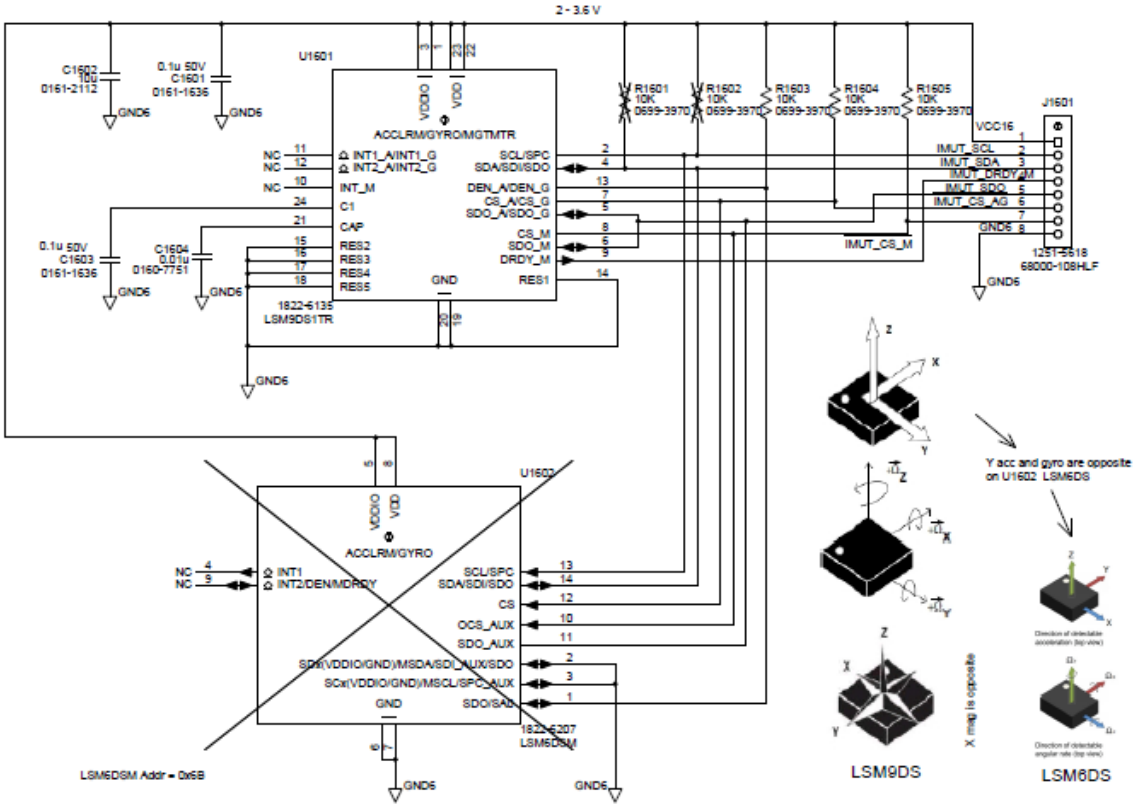
Digital Temp Sensor



Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	2.8	5.5	3.3	Volts	
Iin		.5	.28	mA	I^2C inactive
Output			.5	C/LSB	9-bits 2s complement
Accuracy	-2	2	.5	C	Typ from 0 to 50C, Max -25 to 100C
V Temp			.5+.01/C	V	Vout = .5V + .01V/C

Inertial Measurement Unit

Inertial Measurement Unit (IMU)



Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	2	3.6	3.3	Volts	
Iin			.6/4	mA	accel+mag/gyro
Output data rate			100	Hz	One-shot mode

IMU Characteristics from the LSM9DS1 datasheet:

@ Vdd = 2.2 V, T = 25 °C unless otherwise noted[a]

Table 3. Sensor characteristics

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
LA_FS	Linear acceleration measurement range			±2		*g*
				±4		
				±8		
				±16		
M_FS	Magnetic measurement range			±4		gauss
				±8		
				±12		
				±16		
G_FS	Angular rate measurement range			±245		dps
				±500		
				±2000		
LA_So	Linear acceleration sensitivity	Linear acceleration FS = ±2 *g*		0.061		mg/LSB
		Linear acceleration FS = ±4 *g*		0.122		
		Linear acceleration FS = ±8 *g*		0.244		
		Linear acceleration FS = ±16 *g*		0.732		
M_GN	Magnetic sensitivity	Magnetic FS = ±4 gauss		0.14		mgauss/ LSB
		Magnetic FS = ±8 gauss		0.29		
		Magnetic FS = ±12 gauss		0.43		
		Magnetic FS = ±16 gauss		0.58		
G_So	Angular rate sensitivity	Angular rate FS = ±245 dps		8.75		mdps/ LSB
		Angular rate FS = ±500 dps		17.50		
		Angular rate FS = ±2000 dps		70		
LA_TyOff	Linear acceleration typical zero-g level offset accuracy[2]	FS = ±8 *g*		±90		mg
M_TyOff	Zero-gauss level [3]	FS = ±4 gauss		±1		gauss
G_TyOff	Angular rate typical zero-rate level [4]	FS = ±2000 dps		±30		dps
M_DF	Magnetic disturbance field	Zero-gauss offset starts to degrade			50	gauss
Top	Operating temperature range		-40		+85	°C

1. Typical specifications are not guaranteed
2. Typical zero-*g* level offset value after soldering
3. Typical zero-gauss level value after test and trimming
4. Typical zero rate level offset value after MSL3 preconditioning

a. The product is factory calibrated at 2.2 V. The operational power supply range is from 1.9 V to 3.6 V.

Current Sensor

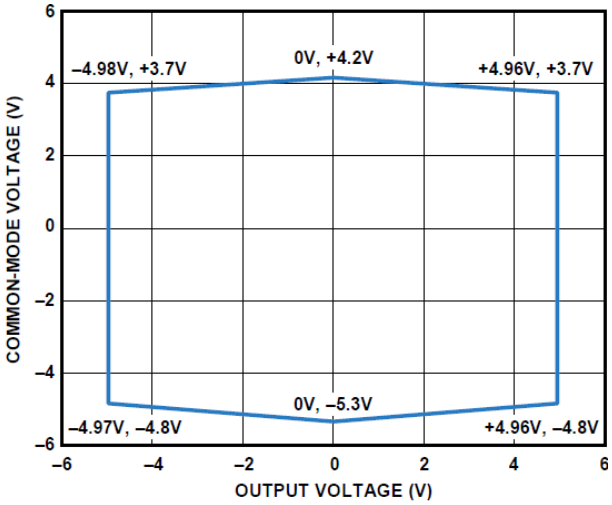
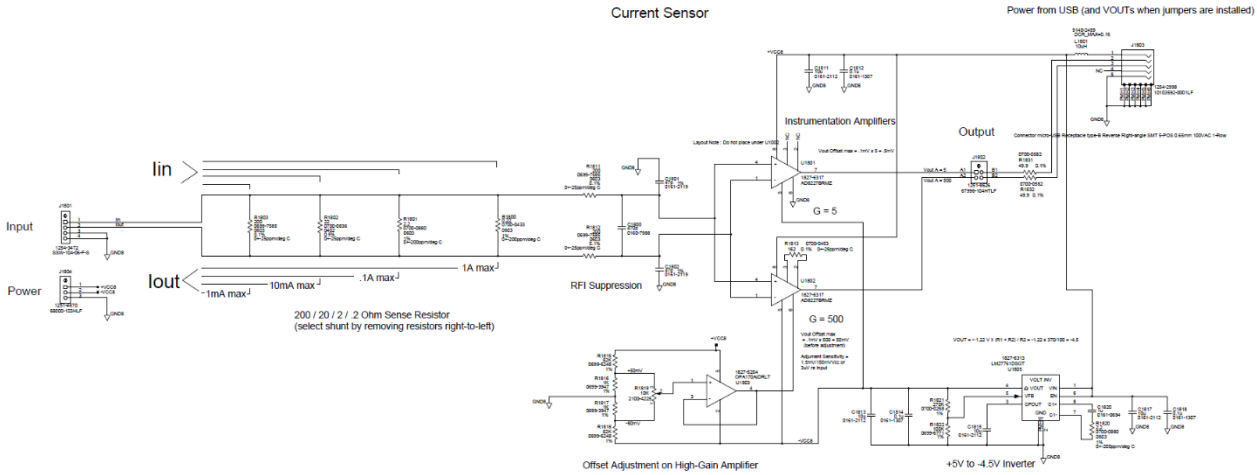
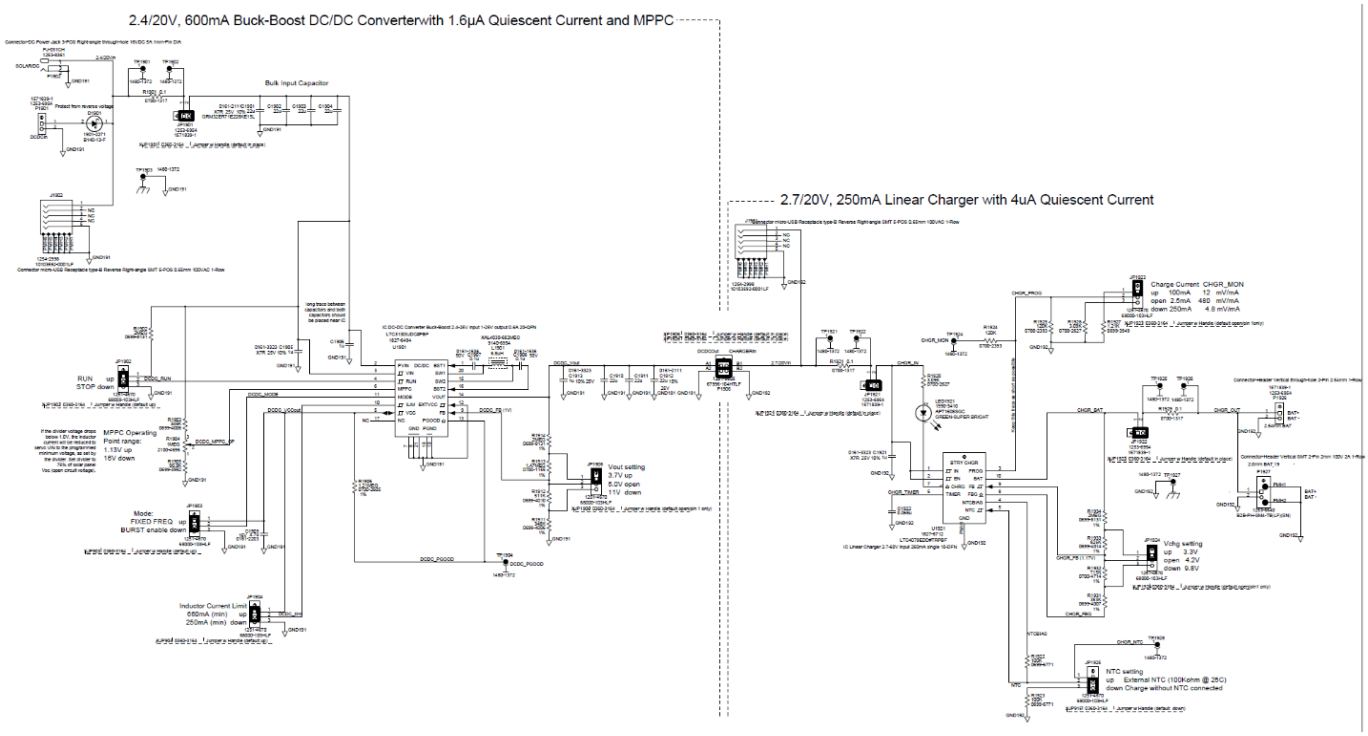


Figure 11. Input Common-Mode Voltage vs. Output Voltage, Dual Supply, $V_S = \pm 5$ V, G = 5



Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	4.75	5.25	5	Volts	
ICC			4	mA	Powered by 5VDC from USB
Rin (shunt)			.2	Ω	All 4 shunt resistors in place
Sensitivity G = 5 **Sensitivity G = 500**			1 100	V/A	All 4 shunt resistors in place
Sensitivity G = 5 **Sensitivity G = 500**			.333 33.3	V/A	Rin in parallel with .1Ω (Req = 66.7mΩ) and all 4 shunt resistors in place
Input voltage range	-4.5	3.95	-4.8 to 4.2	V	For |Vout| < .5V (see graph above)
Vos G = 5 **Vos G = 500**	-100	100 100		µV	G = 5 (re input, no adjust) G = 500 (re input, adjustable)
TCVos		1	.2	µV/C	Vout = .5V + .01V/C
BW G = 5 **BW G = 500**			250 10	kHz	-3dB
VINnoise G = 5 **VINnoise G = 500**			33 2.5	µVrms	

DC/DC Converter and Charger

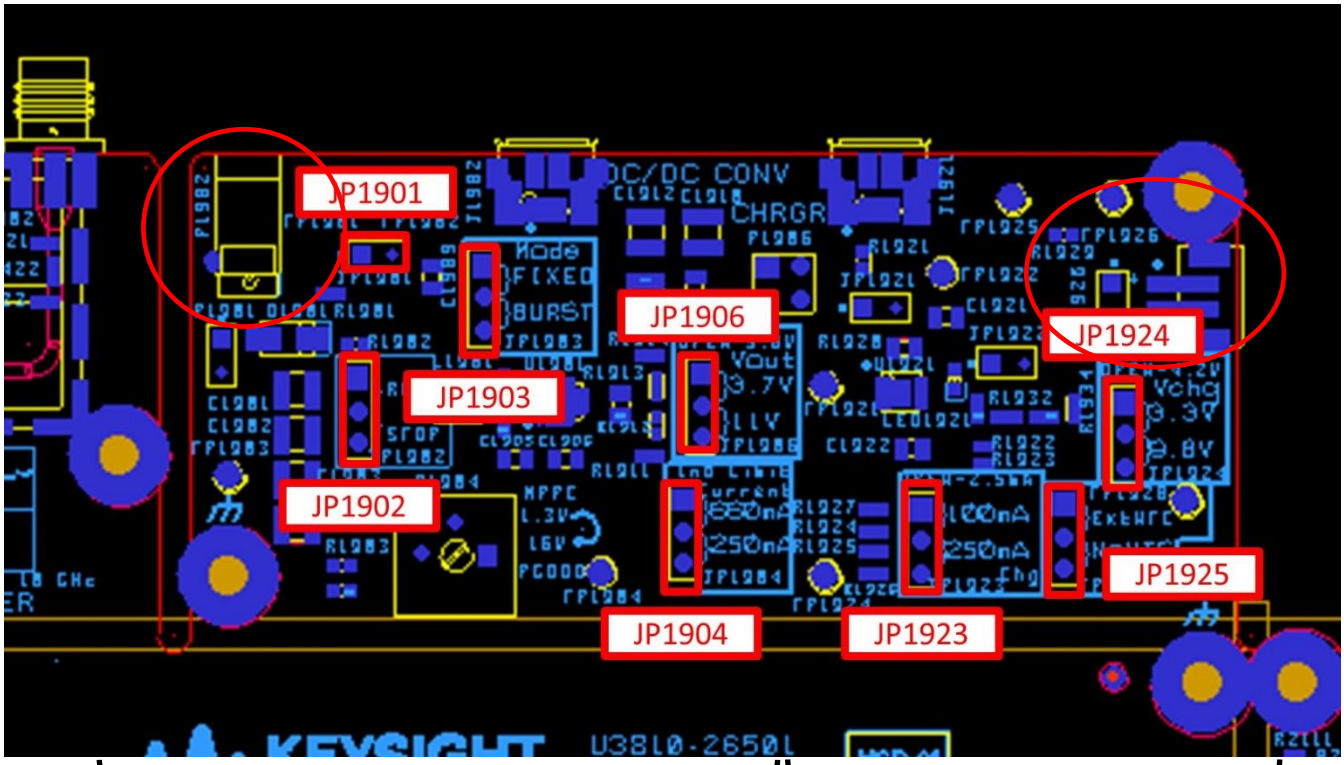


Parameter	Minimum	Maximum	Typical	Units	Condition
DCDC Input	2.4	25		Volts	Fixed or high efficiency Burst mode
DCDC Output			3.7/5/11	Volts	Selectable
MPPC Operating Point range	2.4	16		Volts	Set with potentiometer R1904
Charger Input	2.7	25		Volts	One-shot mode
Charger Output			3.3/4.2/9.8	Volts	Selectable For charging 2xNiMH/1xLiPo/1x NiMH
Charger Output			2.5/100/250	mA	Selectable Current should be < C (mA-hr.)/10

DC/DC Converter / Charger Jumper setting:

Jumper	JP1901	JP1902	JP1903	JP1904	JP1906	JP1923	JP1924	JP1925
Name		RUN/STOP	MODE	Inductor Current Limit	Vout	Charge Current	Vchg	NTC
Position	In place	RUN	FIXED	660mA	Removed (5.0V)	Removed (2.5mA)	Removed (4.2V)	NoNTC

DC/DC Converter / Charger Jumpers drawing:

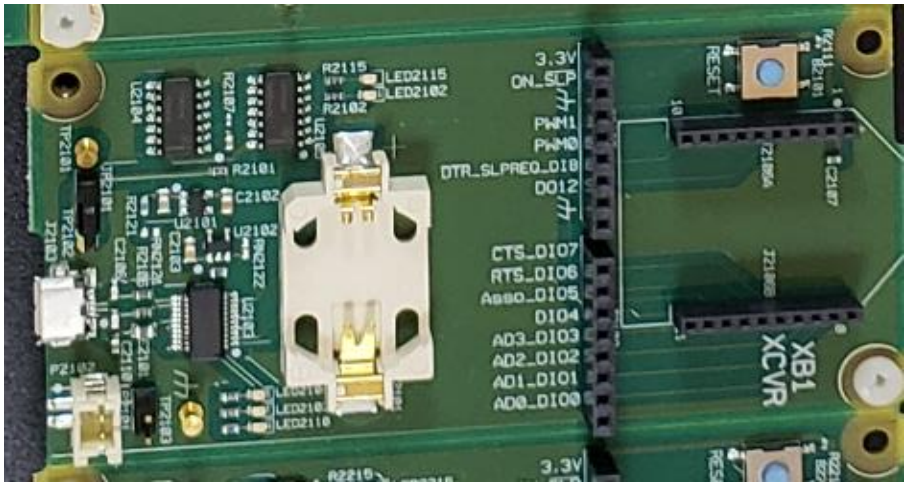


DC/DC CONVERTER CHARGER

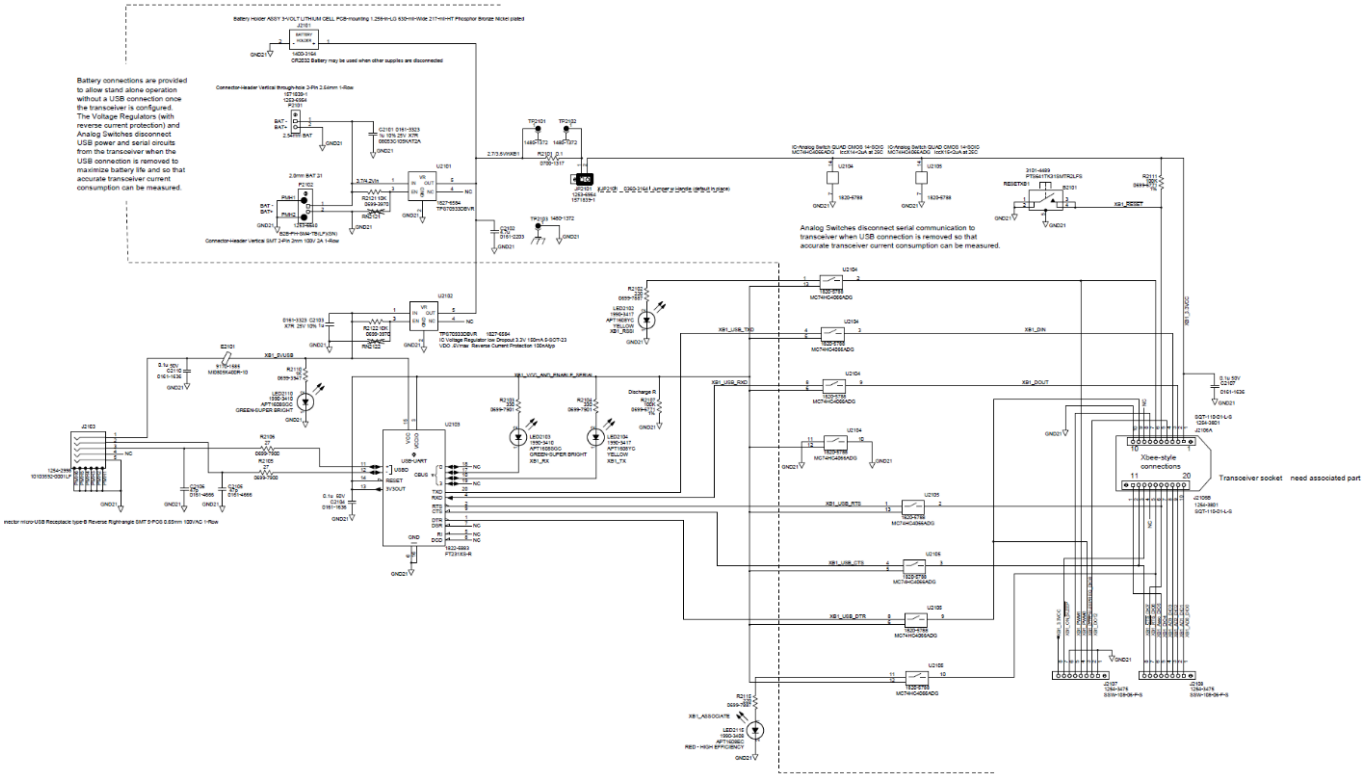
Buck-Boost DC/DC Converter Accessory (left) **interconnect jumper P1906** Charger Accessory (right)

	JP1901	JP1902	JP1903	JP1904	JP1906	P1906	JP1921	JP1922	JP1923	JP1924	JP1925
Name	DC/DC Input Current	DC/DC Run Stop	DC/DC Mode	Inductor Current Limit	DC/DC Vout	DC/DC to Charger	Charger Input Current	Charge Current	Charger Output Current	Charge Voltage	NTC Setting
Position	in place	Up RUN	Down BURST	Up 660ma	Open5.0V Down11V	in place	in place	Up 100ma	in place	Open 4.2V Down 9.8V	Down w/oNTC

XBEE /Multi Radio Module

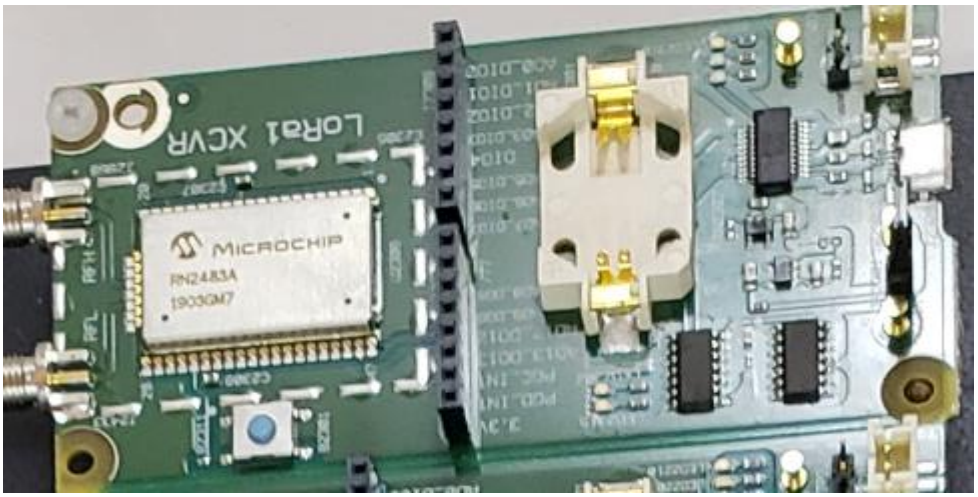


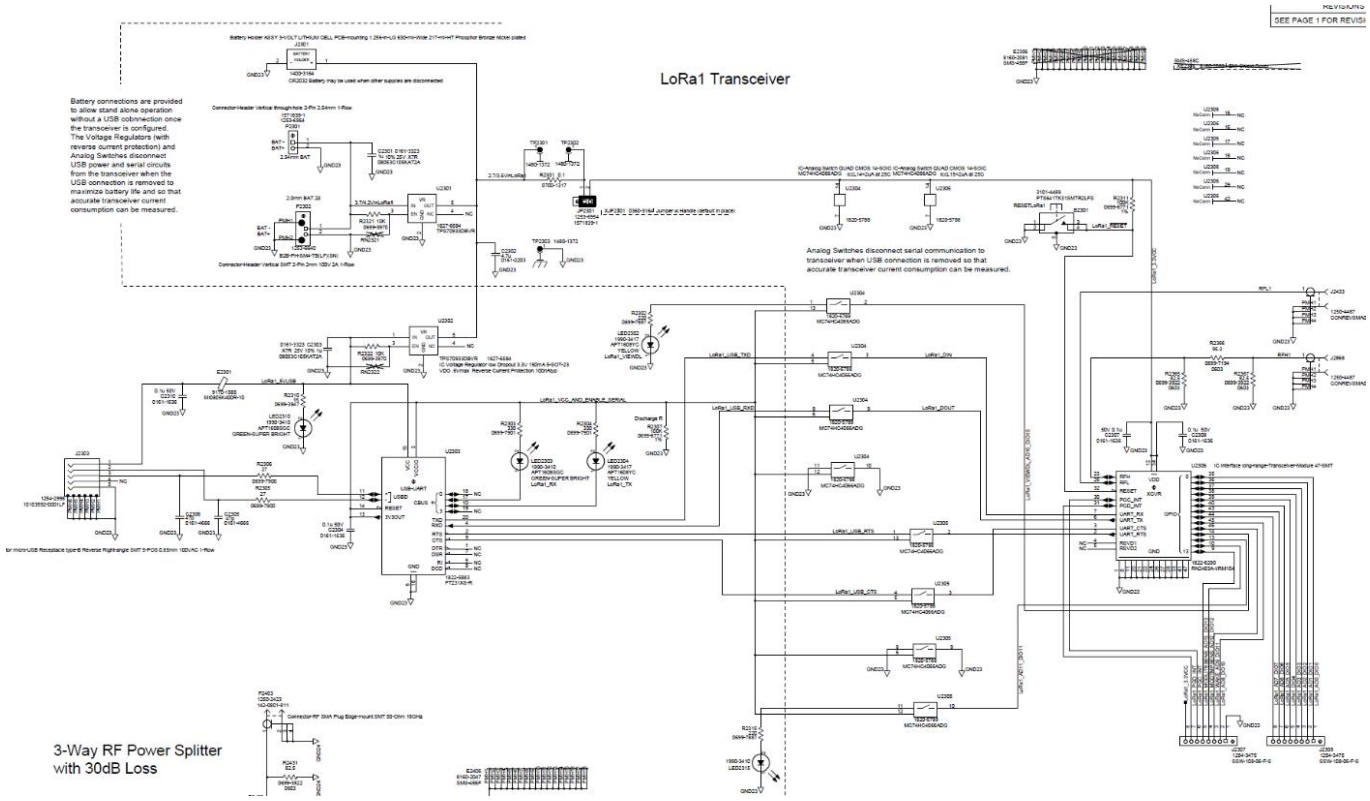
XB1 Transceiver



Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	2.1	3.6	3	Volts	CR2032 Coin Cell
BAT input	2.7	25		Volts	3.3V supply "OR" function with USB power
ICC Transmit **ICC Receive**			40 15	mA	+3.3 V, +8 dBm 135 mA @ +3.3 V, +19 dBm
Digital Comms			USB/U3810 CPU		Auto-selects if USB connected, else CPU
RF Connector			RP-SMA (female)		Reverse-Polarity SMA (female)
Operating frequency			ISM 2.4 – 2.4835	GHz	Cable-only, does not transmit OTA

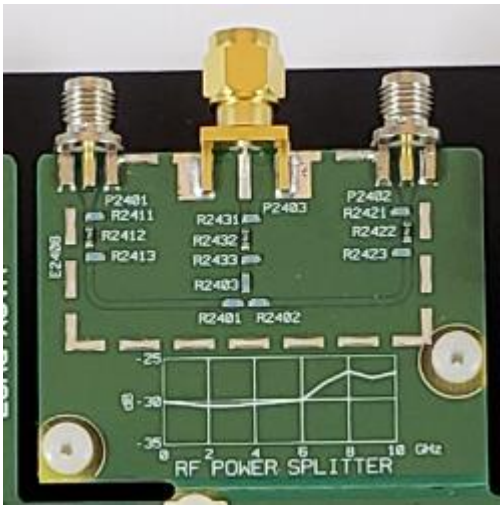
LoRa Radio Module

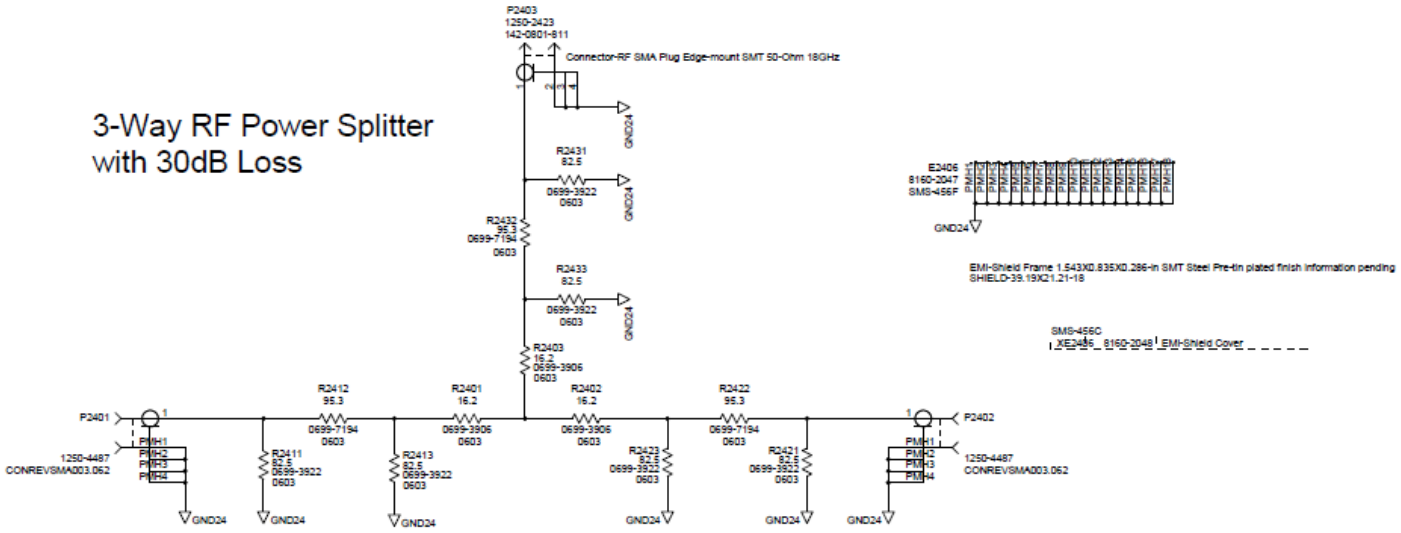




Parameter	Minimum	Maximum	Typical	Units	Condition
VCC	2.1	3.6	3	Volts	CR2032 Coin Cell
BAT input	2.7	25		Volts	3.3V supply "OR" function with USB power
ICC Transmit **ICC Idle**			38.9 2.8	mA	+3.3 V, +8 dBm 135 mA @ +3.3 V, +19 dBm
Digital Comms			USB/U3810 CPU		Auto-selects if USB connected, else CPU
RF Connector			RP-SMA(female)		Reverse-Polarity SMA (female)
Operating frequency	433.050 863.000	434.790 870.000		MHz	Selectable, cable-only, does not transmit OTA, 12dB attenuator in 863 band

Power Splitter



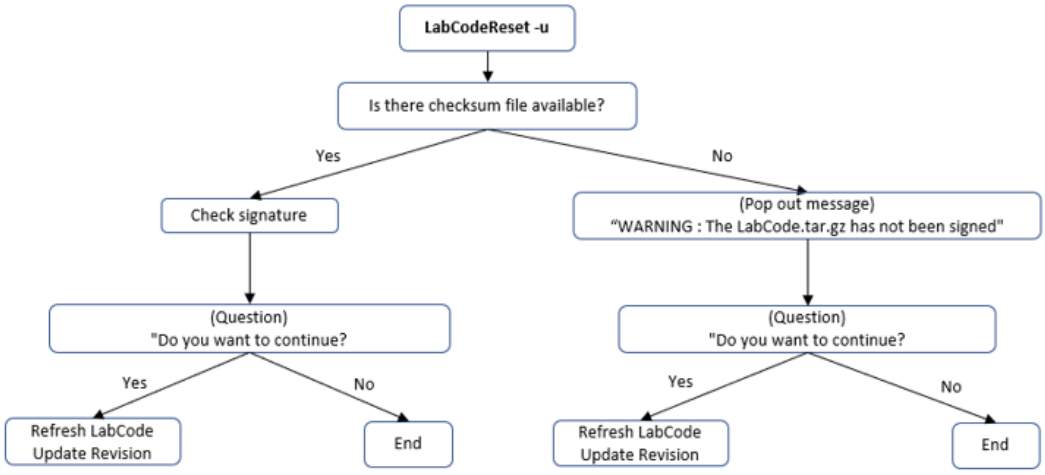


Parameter	Minimum	Maximum	Typical	Units	Condition
Impedance			50	Ω	
Power		.1		W	
Attenuation	27	31	30	dB	Any port to any other port
RF Connector			RP-SMA(female) SMA(male)		Ports 1 and 3 Port 2
Operating frequency	DC	10		GHz	Cable-only, does not transmit OTA

Updating Lab Code

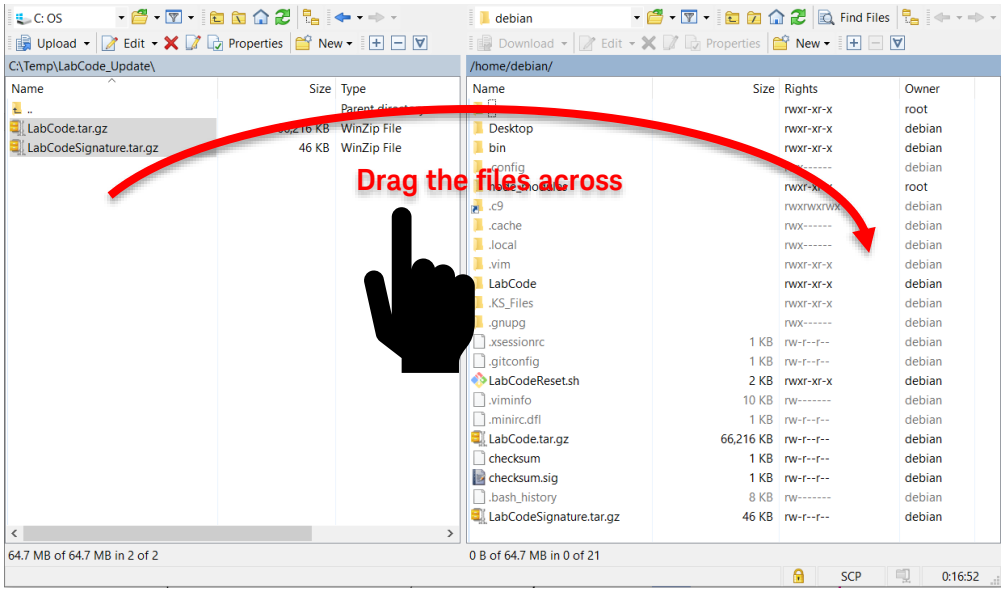
You may update to a newer version of Lab Code by updating the entire BeagleBone image as shown in Appendix A or you may update only the LabCode directories as shown here:

The **LabCode.tar.gz** file in the home directory of the Keysight BeagleBone contains all the programs, files, and configuration files for the U3810A Labware. This file can be used the refresh all the files in the **LabCode** directory. Also using the update feature, a new revision of the lab code and BeagleBone configuration. The **LabCode.tar.gz** file has a signature file, **checksum.sig**, associated with this to ensure authenticity of the code.



1 Download the Keysight Lab Code and Signature file: **LabCode.tar.gz** and **LabCodeSignature.tar.gz**.
This image is a component of the purchased courseware. You can download the image together with the courseware using this process:
   a. Use the existing Keysight Entitlement Certificate
   b. Login into Keysight Software Manager at: https://ksm.software.keysight.com/
   c. Follow the instructions to download the latest Lab Code files along with the courseware.

2 Using WinSCP Copy the **LabCode.tar.gz** and **LabCodeSignature.tar.gz** to the Debian Home directory.



3 Login to the BeagleBone via PuTTY using either the RNDIS or serial connection.

4 To extract out the signature files from LabCodeSignatute.tar.gz, use the command tar -xzf LabCodeSignature.tar.gz.

5 To update the Lab Code to the latest revision, use the command ./LabCodeReset.sh -u This will refresh the code, and set the environmental variables to reflect the new revision code.

6 To ensure the code is authentic, check the signature. This will use the public.key, checksum.sig and LabCode.tar.gz to verify authenticity.

7 For the update process the script will ask for the sudo password. This is to update the system parameters.

8 After the update is complete, a reboot is suggested. This will load in all the latest system parameters into the running sessions.

```
debian@beaglebone:~$ ./LabCodeReset.sh -u
The LabCode.tar.gz has been signed.
Checking on the signature
gpg: assuming signed data in 'checksum'
gpg: Signature made Mon 27 Jul 2020 01:24:21 PM UTC
gpg:                using RSA key 011F5B903A1BFE765E4EDC4F7C92DD7F05C4BDAE
gpg: Good signature from "Copyright Keysight Technologies 2020" [ultimate]
Checking on SHA256 hash ...
LabCode.tar.gz: OK
Do you want to proceed to refresh the code? (y/n)


•••

Updating Keysight Revisions
[sudo] password for debian:
Reboot is suggested
```

Troubleshooting

Does not Power Up (No Display)

1. Check the USB Cable.
2. Measure the voltage.
3. Try different port.

Display Lights Up but No Message

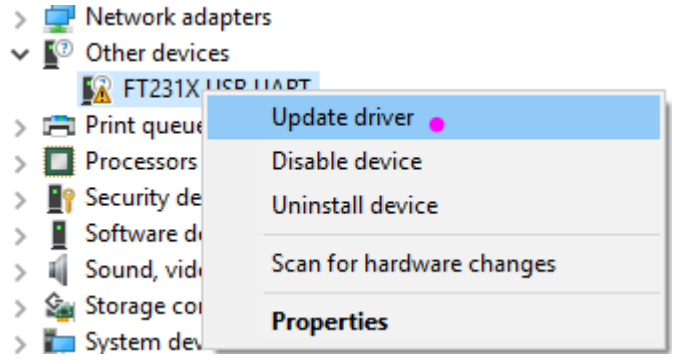
1. Verify Keysight Software is installed.
2. i2cdetect -r -y 2 should show the following. The display should show up as device 3e.

```
debian@beaglebone:~$ i2cdetect -r -y 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 3e --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- UU UU UU UU -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

3. Reload start up scripts.

USB Serial Driver Installation Problem

The warning below indicates that the driver was not automatically installed by Windows. Right-click the device and select **Update driver**.



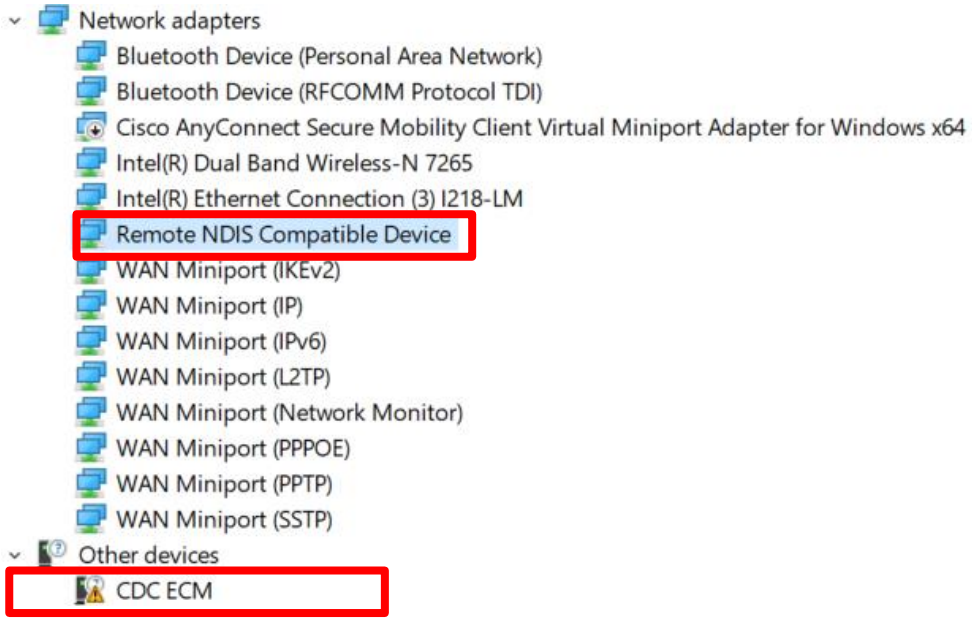
You can also download the driver from FTDI website (https://www.ftdichip.com/Drivers/D2XX.htm) and run the setup file "CDM21228_Setup.exe". The table below lists the currently Supported D2XX Drivers.

		Processor Architecture					
Operating System	**Release Date**	**x86 (32-bit)**	**x64 (64-bit)**	**ARM**	**MIPS**	**SH4**	**Comments**
Windows*	2017-08-30	2.12.28	2.12.28	-	-	-	WHQL Certified. Includes VCP and D2XX. Available as a setup executable Read the Release Notes and Installation Guides.
Linux	2018-06-22	1.4.8	1.4.8	1.4.8 ARMv5 soft-float 1.4.8 ARMv5 soft-float uClibc 1.4.8 ARMv6 hard-float(suits Raspberry Pi) 1.4.8 ARMv7 hard-float 1.4.8 ARMv8 hard-float	1.4.8 MIPS32 soft-float 1.4.8 MIPS32 hard-float 1.4.8 MIPS openwrt-uclibc		If unsure which ARM version to use, compare the output of readelf and file commands on a system binary with the content of release/build/libftd2xx.txt in each package. ReadMe NEW! Video Install Guide
Mac OS X 10.4 Tiger or later	2017-03-03	-	1.4.4	-	-	-	If using a device with standard FTDI vendor and product identifiers, install D2xxHelper to prevent OS X 10.11 (El Capitan) claiming the device as a serial port (locking out D2XX programs). ReadMe NEW! Video Install Guide

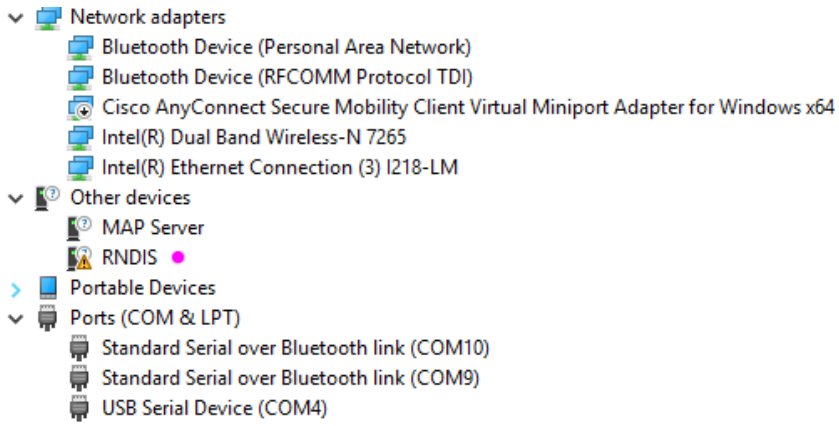
USB RNDIS Problems

Follow the steps below to troubleshoot this error message.

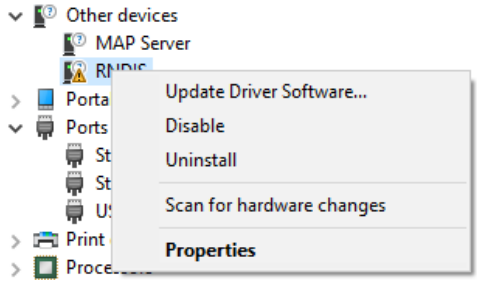
1 Go to Device Manager. If you see the Remote NDIS Compatible Device under Network Adapters, Windows has automatically detected the RNDIS Device and installed its adapter. Your computer is updated and ready for use. You may ignore or disable the CDC ECM (right-click and select disable).



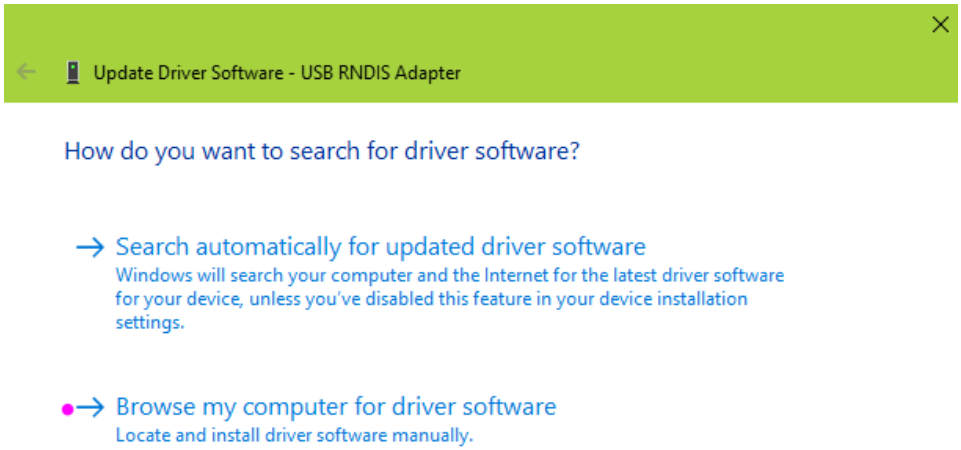
If instead you see the RNDIS listed in the Other devices category with the Property **Manufacturer: Unknown**, you will need to load the Microsoft version of the RNDIS Device Driver. You may also need to reload the driver if you change USB ports.



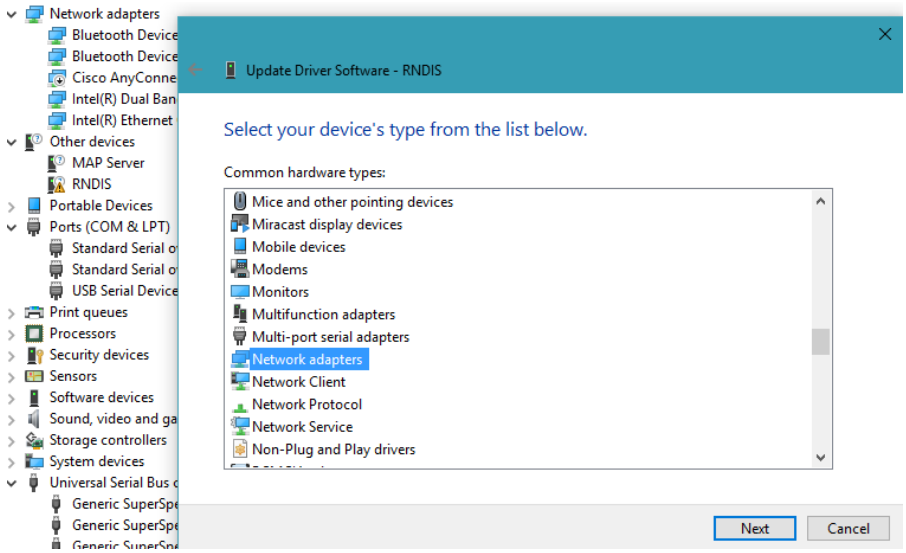
2 Right-click the RNDIS device and select **Update Driver Software**.



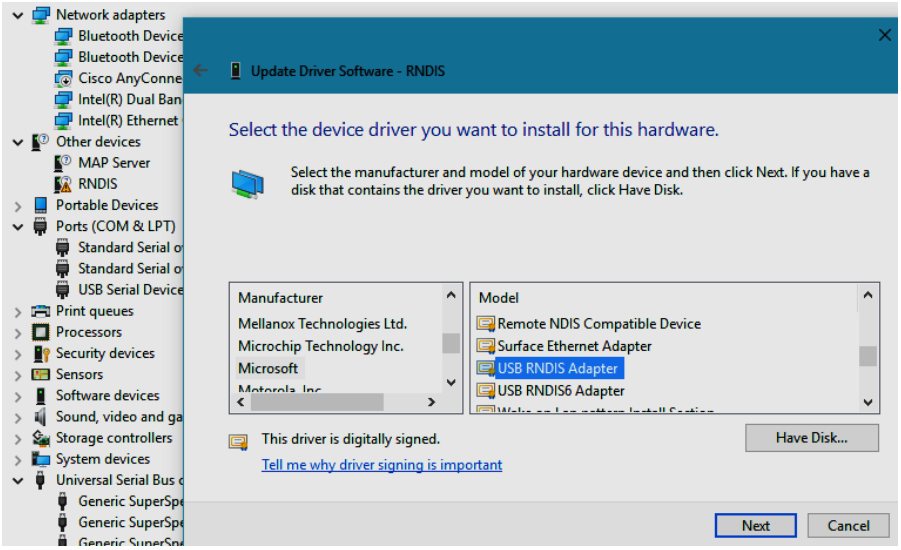
3 Select the **Browse my computer for driver software** option:



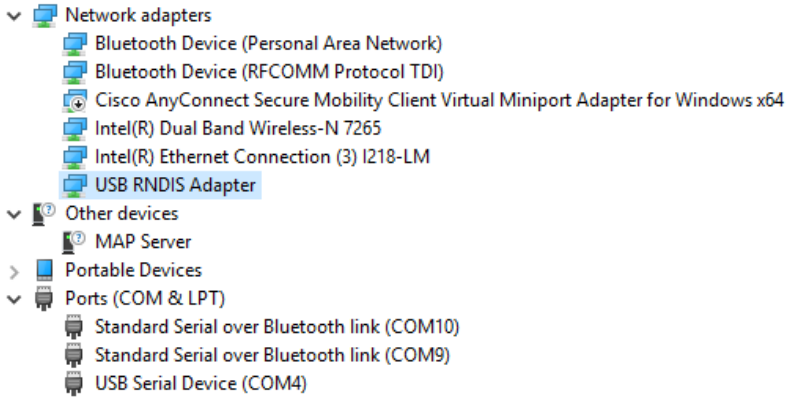
4 Select the **Network Adapters** device type.



5 Click **Microsoft > USB RNDIS Adapter** as shown below.



6 A warning might appear and click **Yes** to bypass. Upon the successful installation of the driver, you should see the new driver in the Device Manager tree.



Typical Boot Process Review

1. Connect a Micro USB to J15 of the U310A system.
2. Open a PuTTY terminal to the serial port found using the device manager.
3. Connect using 115200 Baud.
4. Pres the **Reset** button and observe the boot process on the PuTTY terminal. The output should look something like this:

The typical console and PuTTY monitor and reboot while observing the console.

```
:
U-Boot SPL 2018.03-00002-gac9cce7c6a (Apr 05 2018 - 13:07:46 -0500)
Trying to boot from MMC2
Loading Environment from EXT4... Card did not respond to voltage
select!
** Bad device mmc 0 **
Failed (-5)
```

Note the SD Card is not present, will cause this error message. This is the correct response.

```
U-Boot 2018.03-00002-gac9cce7c6a (Apr 05 2018 - 13:07:46 -0500), Build:
jenkins-github_Bootloader-Builder-47
CPU  : AM335X-GP rev 2.1
I2C:   ready
DRAM:  512 MiB
No match for driver 'omap_hsmmc'
No match for driver 'omap_hsmmc'
Some drivers were not found
Reset Source: Global external warm reset has occurred.
Reset Source: Power-on reset has occurred.
RTC 32KCLK Source: External.
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1
Loading Environment from EXT4... Card did not respond to voltage
select!
** Bad device mmc 0 **
Failed (-5)
Board: BeagleBone Black
<ethaddr> not set. Validating first E-fuse MAC
BeagleBone Black:
Model: SeeedStudio BeagleBone Green Wireless:
BeagleBone: cape eeprom: i2c_probe: 0x54:
BeagleBone: cape eeprom: i2c_probe: 0x55:
BeagleBone: cape eeprom: i2c_probe: 0x56:
BeagleBone: found invalid cape eeprom: i2c_probe: 0x57:
Net:   eth0: MII MODE
Could not get PHY for cpsw: addr 0
cpsw, usb_ether
Press SPACE to abort autoboot in 2 seconds
board_name=[A335BNLT] ...
board_rev=[GW1A] ...
Card did not respond to voltage select!
Card did not respond to voltage select!
```

```
Card did not respond to voltage select!
gpio: pin 56 (gpio 56) value is 0
gpio: pin 55 (gpio 55) value is 0
gpio: pin 54 (gpio 54) value is 0
gpio: pin 53 (gpio 53) value is 1
Card did not respond to voltage select!
Card did not respond to voltage select!
switch to partitions #0, OK
mmc1(part 0) is current device
Scanning mmc 1:1...
gpio: pin 56 (gpio 56) value is 0
gpio: pin 55 (gpio 55) value is 0
gpio: pin 54 (gpio 54) value is 0
gpio: pin 53 (gpio 53) value is 1
switch to partitions #0, OK
mmc1(part 0) is current device
gpio: pin 54 (gpio 54) value is 1
Checking for: /uEnv.txt ...
Checking for: /boot.scr ...
Checking for: /boot/boot.scr ...
Checking for: /boot/uEnv.txt ...
gpio: pin 55 (gpio 55) value is 1
2267 bytes read in 15 ms (147.5 KiB/s)
Loaded environment from /boot/uEnv.txt
Checking if uname_r is set in /boot/uEnv.txt...
gpio: pin 56 (gpio 56) value is 1
Running uname_boot ...
loading /boot/vmlinuz-4.14.49-ti-r54 ...
10453504 bytes read in 676 ms (14.7 MiB/s)
uboot_overlays: [uboot_base_dtb=am335x-bonegreen-wireless-uboot-
univ.dtb] ...
uboot_overlays: [uboot_base_dtb=am335x-boneblack-uboot.dtb] ...
uboot_overlays: Switching too: dtb=am335x-boneblack-uboot.dtb ...
loading /boot/dtbs/4.14.49-ti-r54/am335x-boneblack-uboot.dtb ...
57952 bytes read in 36 ms (1.5 MiB/s)
uboot_overlays: [fdt_buffer=0x60000] ...
uboot_overlays: loading /lib/firmware/BB-UART1-00A0.dtbo ...
1075 bytes read in 55 ms (18.6 KiB/s)
uboot_overlays: loading /lib/firmware/BB-UART2-00A0.dtbo ...
1075 bytes read in 240 ms (3.9 KiB/s)
uboot_overlays: loading /lib/firmware/BB-I2C1-00A0.dtbo ...
1152 bytes read in 251 ms (3.9 KiB/s)
uboot_overlays: loading /lib/firmware/BB-PWM1-00A0.dtbo ...
1409 bytes read in 228 ms (5.9 KiB/s)
uboot_overlays: loading /lib/firmware/BB-BONE-eMMC1-01-00A0.dtbo ...
1440 bytes read in 325 ms (3.9 KiB/s)
uboot_overlays: loading /lib/firmware/BB-BBGW-WL1835-00A0.dtbo ...
4839 bytes read in 115 ms (41 KiB/s)
uboot_overlays: loading /lib/firmware/BB-ADC-00A0.dtbo ...
711 bytes read in 121 ms (4.9 KiB/s)
uboot_overlays: loading /lib/firmware/AM335X-PRU-RPROC-4-14-TI-
00A0.dtbo ...
3513 bytes read in 314 ms (10.7 KiB/s)
uboot_overlays: add [enable_uboot_cape_universal=1] to /boot/uEnv.txt
to enable...
```

```
loading /boot/initrd.img-4.14.49-ti-r54 ...
4034122 bytes read in 271 ms (14.2 MiB/s)
debug: [console=ttyO0,115200n8 bone_capemgr.uboot_capemgr_enabled=1
root=/dev/mmcblk1p1 ro rootfstype=ext4 rootwait coherent_pool=1M
net.ifnames=0 quiet] ...
debug: [bootz 0x82000000 0x88080000:3d8e4a 88000000] ...
## Flattened Device Tree blob at 88000000
   Booting using the fdt blob at 0x88000000
   Loading Ramdisk to 8fc27000, end 8ffffe4a ... OK
   reserving fdt memory region: addr=88000000 size=70000
   Loading Device Tree to 8fbb4000, end 8fc26fff ... OK
Starting kernel ...
[    0.000814] timer_probe: no matching timers found
[    0.546264] dmi: Firmware registration failed.
[    1.035757] wkup_m3_ipc 44e11324.wkup_m3_ipc: could not get rproc
handle
[    1.334985] omap_voltage_late_init: Voltage driver support not added
Debian GNU/Linux 9 beaglebone ttyS0
BeagleBoard.org Debian Image 2018-06-17
U3810A Version 2.50
Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
default username:password is [debian:temppwd]
beaglebone login:
```

Debugging I²C Devices

1. **i2cdetect -r -y 1**

Debugging SPI and ADC

1. Measure and verify VSENSOR and +5SYS supplies.

Debugging GPIO

1. **mraa-gpio list**
2. look in **/sys/class/gpio**
3. Mention 1 second turnaround for status change.

Debugging PWM

1. /sys/class/pwm See if there is pwmchip0 and pwmchip2.
2. **$ echo '0' > /sys/class/pwm/pwmchip0** should result in seeing the **/sys/class/pwm/pwm-0:0 directory**
3. In that directory echoing various values will affect PWM output on GP5.

Managing Disk Usage

df -h will give the usage of the different mounted partitions. The '/' partition is the primary. This image has a high usage of the available space.

```
debian@beaglebone:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            215M     0  215M   0% /dev
tmpfs            49M  8.5M   40M  18% /run
/dev/mmcblk1p1  3.5G  2.7G  582M  83% /
tmpfs           242M     0  242M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           242M     0  242M   0% /sys/fs/cgroup
tmpfs            49M     0   49M   0% /run/user/1000
```

1 df -h . will give the space of the partition in the current directory.

```
debian@beaglebone:~$ df -h .
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk1p1  3.5G  2.7G  582M  83% /
```

2 **du -sh *** will give the disk usage of the files and directories in the current directory.

```
debian@beaglebone:~$ du -sh *
4.0K    bin
3.5M    LabCode
```

3 **sudo du -sh /*** will give the usage of all the root directories in the image. (ignore any **cannot access** files)

```
debian@beaglebone:~$ sudo du -sh /*
4.0K    /bbb-uEnv.txt
9.4M    /bin
26M     /boot
0       /dev
5.2M    /etc
3.6M    /home
4.0K    /ID.txt
215M    /lib
16K     /lost+found
4.0K    /media
4.0K    /mnt
4.0K    /nfs-uEnv.txt
207M    /opt
0       /proc
2.9M    /root
8.5M    /run
6.4M    /sbin
4.0K    /srv
0       /sys
48K     /tmp
```

Troubleshooting *Bluetooth*® and Wi-Fi

A way to enable or disable *Bluetooth*® or Wi-Fi is with **`rfkill`**:

Command Descriptions	Linux Commands
Disable Wi-Fi	`rfkill block wifi`
Enable Wi-Fi	`rfkill unblock wifi`
Disable *Bluetooth*®	`rfkill block bluetooth`
Enable *Bluetooth*®	`rfkill unblock bluetooth`

By default, both Wi-Fi and *Bluetooth*® are enabled. If you disable (**`block`**) you must re-enable (**`unblock`**) before powering down or rebooting Linux.

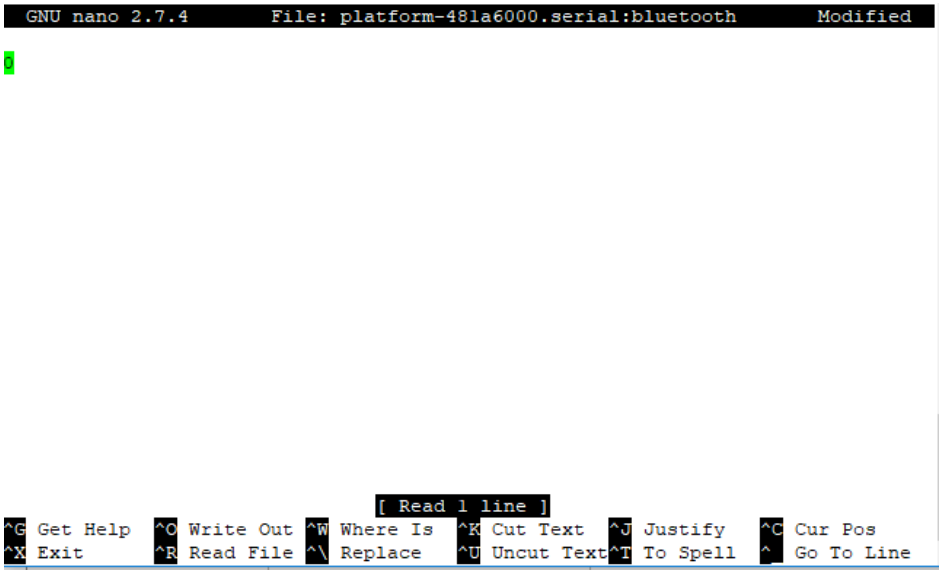
Bluetooth® Disabled

An improper sequence may disable *Bluetooth*® in such a way that **`unblock`** will not re-enable it. If *Bluetooth*® cannot be enabled by the normal process:

1 Run **`hciconfig -a`**. This should result in a listing like this.

```
debian@beaglebone:~$ hciconfig -a
hci0:   Type: Primary  Bus: UART
        BD Address: F0:45:DA:3B:6C:E0  ACL MTU: 1021:6  SCO MTU: 180:4
        UP RUNNING
        RX bytes:746 acl:0 sco:0 events:49 errors:0
        TX bytes:3441 acl:0 sco:0 commands:49 errors:0
        Features: 0xff 0xfe 0x2d 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF
        Link mode: SLAVE ACCEPT
        Name: 'beaglebone'
        Class: 0x480000
        Service Classes: Capturing, Telephony
        Device Class: Miscellaneous,
        HCI Version: 4.1 (0x7)  Revision: 0x0
        LMP Version: 4.1 (0x7)  Subversion: 0xac7c
        Manufacturer: Texas Instruments Inc. (13)

   debian@beaglebone:~$
```

If nothing is returned, there a known and common problem that the **rfkill** command blocking Bluetooth was run, and it was not **unblock**ed before reboot. The way to fix this state is to manually edit the **rfkill** file and enable it. Edit the file **/var/lib/systemd/rfkill/platform-481a6000.serial:bluetooth** with root permissions—this will require the use of the **nano** editor[1] since the file will not be visible on WinSCP. This file has one character in it make sure it is a "0" (Zero) A "1" (One) in this file will disable Bluetooth and cannot be enabled by the rfkill **unblock** command if the device is not active. A Reboot is required after changing this file.



2 If the **hciconfig -a** command shows the Bluetooth is down. Check the **rfkill** status using **rfkill list all**.

```
debian@beaglebone:~$ hciconfig -a
hci0:   Type: Primary  Bus: UART
        BD Address: F0:45:DA:3B:6C:E0  ACL MTU: 1021:6  SCO MTU: 180:4
        DOWN
        RX bytes:1037 acl:0 sco:0 events:53 errors:0
        TX bytes:3461 acl:0 sco:0 commands:53 errors:0
        Features: 0xff 0xfe 0x2d 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF
        Link mode: SLAVE ACCEPT

debian@beaglebone:~$ rfkill list all
0: hci0: Bluetooth
        Soft blocked: yes
        Hard blocked: no
1: phy0: Wireless LAN
        Soft blocked: no
        Hard blocked: no
```

3 To remove a blocked state, use the **rfkill unblock bluetooth** command.

Wi-Fi Disabled

An improper sequence may disable Wi-Fi in such a way that it looks permanently disabled.

```
debian@beaglebone:~/temp/LabCode$ connmanctl
Error /net/connman/technology/wifi: No carrier
connmanctl>exit

debian@beaglebone:~$ rfkill list all
0: hci0: Bluetooth
        Soft blocked: no
        Hard blocked: no
1: phy0: Wireless LAN
        Soft blocked: yes
        Hard blocked: no
```

1 If the Wi-Fi was blocked by the **rfkill** command without a reboot, **rfkill unblock wifi** will restore it. However, if there was a reboot or power down, after unblocking an additional reboot will be required.

2 If the wireless is turned off in the /boot/uEnv.txt, then it will not show up as a technology available in the connmanctl control system.

```
debian@beaglebone:~/temp$ connmanctl
Error getting VPN connections: The name net.connman.vpn was not provided
   by any connmanctl> scan wifi
Error /net/connman/technology/wifi: Method "Scan" with signature "" on
   interface "net.connman.Technology" doesn't exist

connmanctl> technologies
/net/connman/technology/ethernet
  Name = Wired
  Type = ethernet
  Powered = True
  Connected = False
  Tethering = False
connmanctl>
```

3 Use **cat** to examine /boot/uEnv.txt and make sure the disable wireless has a "#" in front of the line. This is the normal configuration for the disable section of the uEnv.txt file. If necessary, use the **nano** editor[1] to make the change.

```
###Disable auto loading of virtual capes (emmc/video/wireless/adc)
#disable_uboot_overlay_emmc=1
#disable_uboot_overlay_video=1
disable_uboot_overlay_audio=1
#disable_uboot_overlay_wireless=1
#disable_uboot_overlay_adc=1
###
```

WARNING

It is important to always execute the command **rfkill unblock wlan** after completing the procedure in which you used the command **rfkill block wlan.** This must always be done before your BeagleBone CPU is shutdown or rebooted.

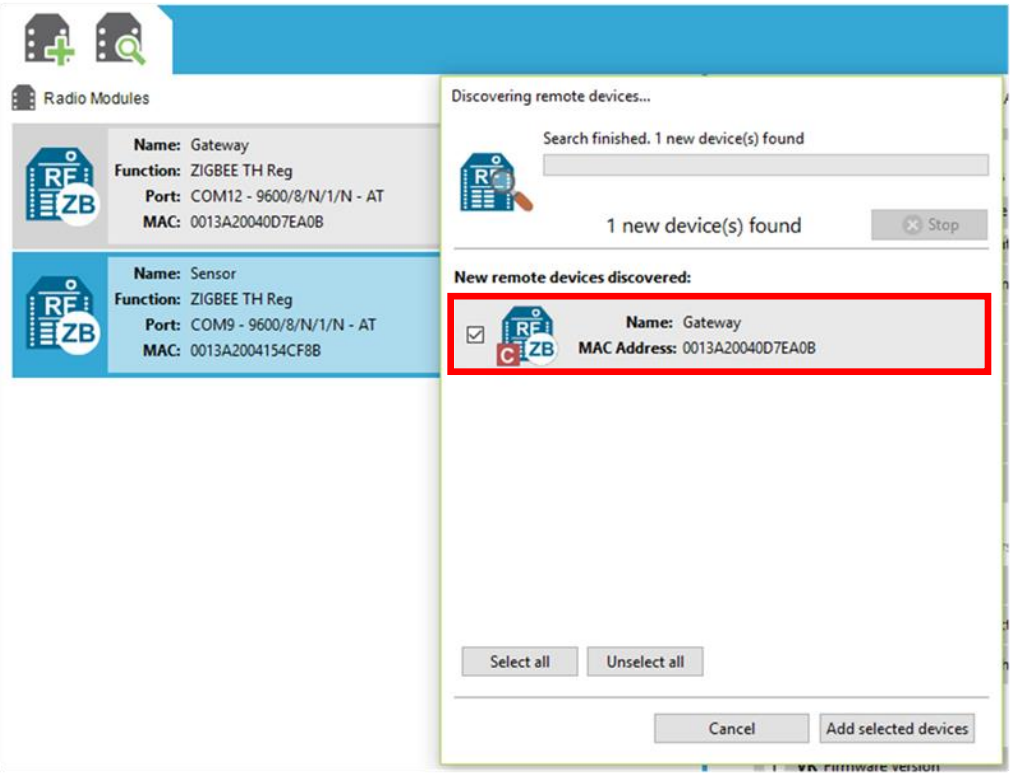
The same is true for **bluetooth**.

Debugging Zigbee Connections

1 To verify that two or more Zigbee modules are communicating with each other using the same wireless network, click **Discover radio nodes in the same network** button in the XCTU software.

NOTE Discover radio is available in all API configurations. (Network working mode is not available in Transparent Mode [0]).

2 A pop-up window titled **Discovering remote devices** will appear and list the Zigbee modules discovered in the same network. Make sure that the other Zigbee module is listed.



a On the “Discovered” Sensor node Zigbee module, verify that:
   i The MAC Address of the Gateway Zigbee module found matches your Gateway Zigbee module MAC address.
   ii The Gateway Zigbee module has the coordinator icon to indicate that it is a coordinator.

b Click the **Cancel** button after verification.

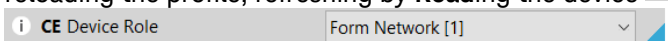
c On the “Discovered” Gateway Zigbee module, verify that:
   i. The MAC Address of the Sensor node Zigbee module found matches your Sensor node Zigbee module MAC address.
   ii. The Sensor node Zigbee module has the router icon to indicate that it is a router.

d Click the **Cancel** button after verification.

NOTE

Sometimes the C and R icons do not automatically refresh. If this happens, you can try reloading the profile, refreshing by **Read**ing the device 

or re**Write**ing the **CE** Role



.

Do not worry if the above does not work. Simply proceed. If the next steps work, then there is no problem.

Rarely after previously successful connection is established, the Zigbee connection may not reconnect. If this is the case, try temporarily reversing the CE Device Role of the two devices in place, complete the connection, and then reversing again.

---

3 Click the sensor node Zigbee.

4 Change to the **Consoles working mode** (top-right corner).

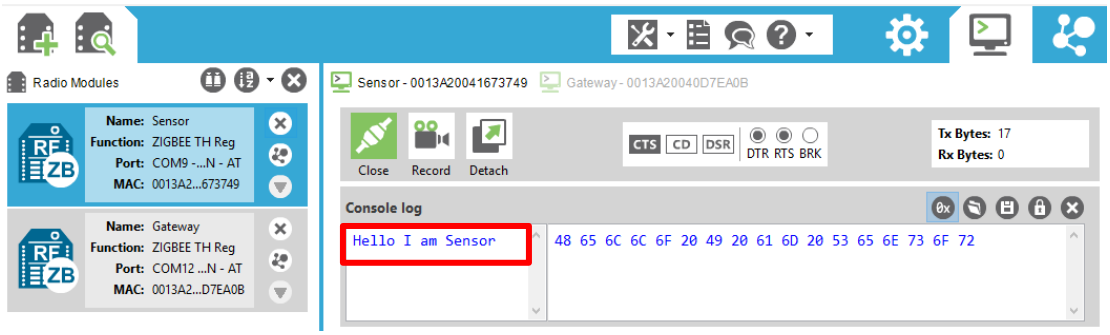
5 Click **Open** to establish communication with the sensor node Zigbee.

6 Click the gateway Zigbee which should be in Console Mode and click Open to establish communication. Both Zigbee modules can now communicate directly.

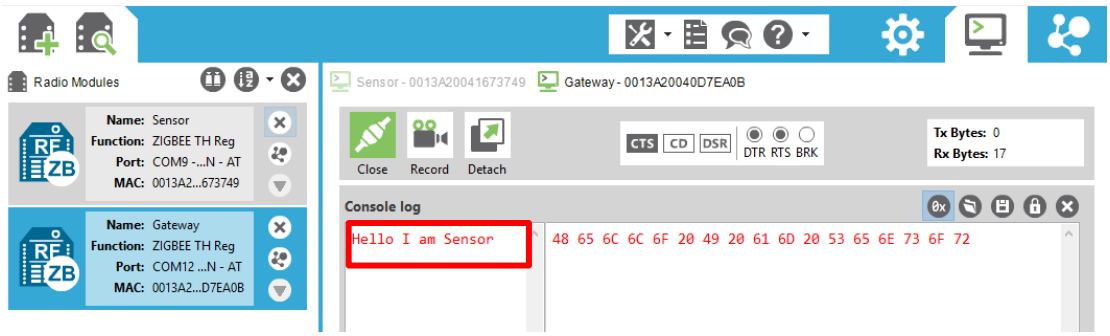
7 Type something in the **Console log** of either the gateway or sensor node Zigbee, and then do it again with the other module. You should see the same message appear in the other Zigbee module. This verifies that both Zigbee modules are communicating with each other.

HINT: No **Console Log**? Ensure AP API Enable parameter is set to Transparent Mode [0] on both XBees.

Sensor node Zigbee console:



Gateway Zigbee console:



NOTE

If the Zigbee modules are not able to communicate to each other, assure that:

- All the settings are set correctly as per the instructions.
- The value for Zigbee Stack Profile (ZS) is the same for both Zigbee modules.
- Assure that the PAN ID (ID) and Scan Channel (SC) of the sensor node Zigbee is set to the OP and SC of the gateway Zigbee.
- Rarely after previously successful connection is established, the Zigbee connection may not reconnect. If this is the case, try temporarily reversing the CE Device Role of the two devices in place, complete the connection, and then reversing again.

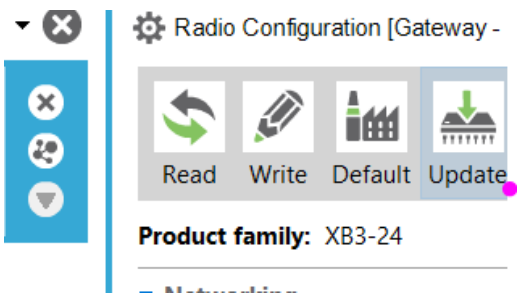
If the above steps do not work, the XBee firmware may need to be updated and made consistent for all Zigbee transceivers.

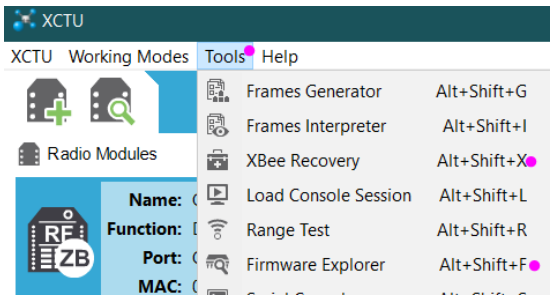
8 In XCTU click each Zigbee module (XBee3 + XB1 or XB2 Transceiver board) and the Configuration working mode icon.

9 At the top of the **Configuration working mode** window, assure the latest firmware revision, 1005 or newer:

- **Product family:** XB3-24
- **Function set:** Digi XBee3 Zigbee 3.0 TH
- **Firmware version:** 1005

10 If necessary, update all XBee3s to this or a newer revision using **Update** or **Tools** if Recovery is required.





11 Begin again at the top of this section.

Service and Repair

Preparing the Unit for Repair or Replacement

If the Troubleshooting steps above do not resolve the issue, either the BeagleBone CPU or the U3810A main board (or a snap-off accessory) is defective and should be repaired or replaced. The BeagleBone CPU and the U3810A main board must be separated and repaired or replaced separately. If you are not certain which unit is defective, first try replacing your BeagleBone CPU with a known good unit. Refer to Appendix F – Assembly and Disassembly.

For repair or replacement of any U3810A or any BeagleBone CPU provided by Keysight (label includes U3811A) see Obtaining Repair Service below.

For BeagleBone CPUs not supplied by Keysight, please see https://beagleboard.org/support/

Types of Service Available

If your product fails during the warranty period, Keysight Technologies will repair or replace it under the terms of your warranty. After your warranty expires, Keysight offers repair services at competitive prices.

Obtaining Repair Service (Worldwide)

To obtain service for your product, contact your nearest Keysight Technologies Service Center at www.keysight.com/find/contactus. They will arrange to have your unit repaired or replaced and can provide warranty or repair–cost information where applicable. Ask the Keysight Technologies Service Center for shipping instructions. Keysight recommends that you retain the original shipping carton for return shipments.

Repackaging for Shipment

To ship the unit to Keysight for service or repair:

- Separate the U3810A main board and the BeagleBone CPU.
- Send only the defective U3810A or the U3811A (Keysight-provided BeagleBone CPU) unit. If both units need repair, prepare each unit separately.
- Attach a tag to the unit identifying the owner and indicating the required service or repair. Include the model number and full serial number.
- Place the unit in its original container with appropriate packaging material.
- Secure the container with strong tape or metal bands.
- If the original shipping container is unavailable, use a container that will ensure at least 10 cm (4 in.) of compressible packaging material around the entire product. Use static-free packaging materials.

Keysight suggests that you always insure shipments.

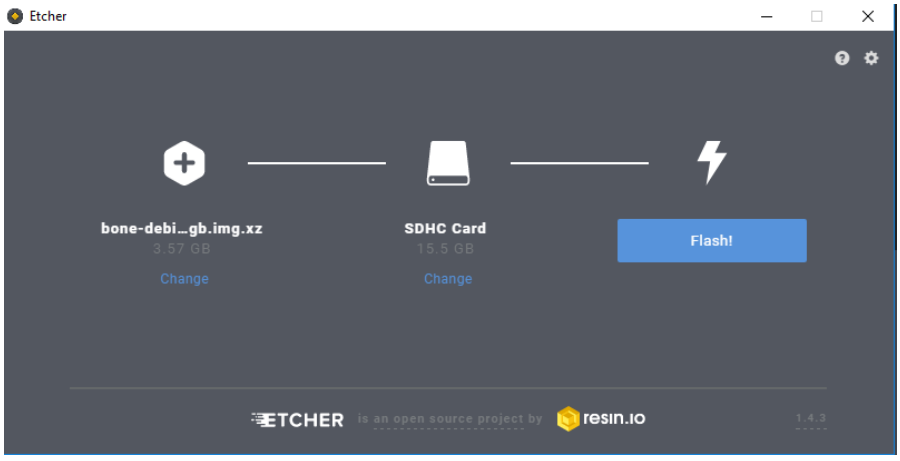
Appendix A – Initialize BeagleBone with Keysight U3810A Image

In this section, a new or used BeagleBone CPU will be placed in or returned to a known state (you may wish to retain your last state if you are continuing using the same BeagleBone from a previous course). The Embedded Multimedia Card (eMMC) will be flashed for the latest update and configured for the overlay style required by U3810A, and for disabling itself after the boot and overlay style loaders run.

1. Your BeagleBone will need to be connected to the U3810A board (a USB to TTL serial cable may also be used).
2. Download the (Optional) Keysight BeagleBone Initialization Image:
   `Keysight_BB_image.img.xz`
   This image is a component of the purchased courseware. You can download the image together with the courseware using this process:
   a. Obtain the Keysight Entitlement Certificate
   b. Login into Keysight Software Manager at: https://ksm.software.keysight.com/
   c. Follow the instructions to redeem the Keysight Entitlement Certificate and download the image along with the courseware.
3. This file contains the image needed to initialize the BeagleBone to a known state. You can use balenaEtcher on a Windows based PC to do this. On Mac or Linux, you can use **dd** (Disk Duplicate) **command** to put the image on the SD Card. With this SD card, it will copy the contents to the BeagleBone eMMC flash.

   This executable image contains:

   - Debian Linux
   - the Keysight-developed U3810A configuration
   - Keysight developed lab code
4. For Windows-based systems; download and install Etcher (might be balenaEtcher) from https://etcher.io/, then use Etcher to copy the downloaded image to a micro SD Card.



5 For Linux or Mac based systems; start a command window and run the following commands line by line:

```
df -h                                   <- Find the SD device
sudo umount /dev/sd(SDCARD)*
tar xf bone-debian…imgxz                <- Extract the img file if it has a .xz extension
sudo dd if = done-debian…img of = /dev/sd(SDCARD) bs = 1M
```

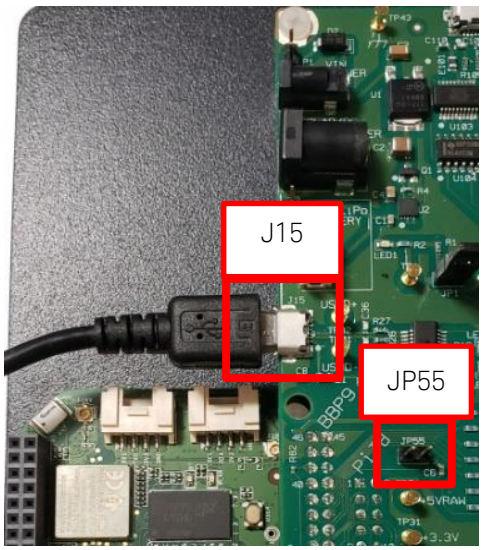
WARNING This SD Card will erase all the contents and programs on the BeagleBone eMMC and write a fresh image used for these labs. Copy any contents off the BeagleBone before inserting the SD card and powering up the BeagleBone.

6 Insert SD card with image into BeagleBone card slot.

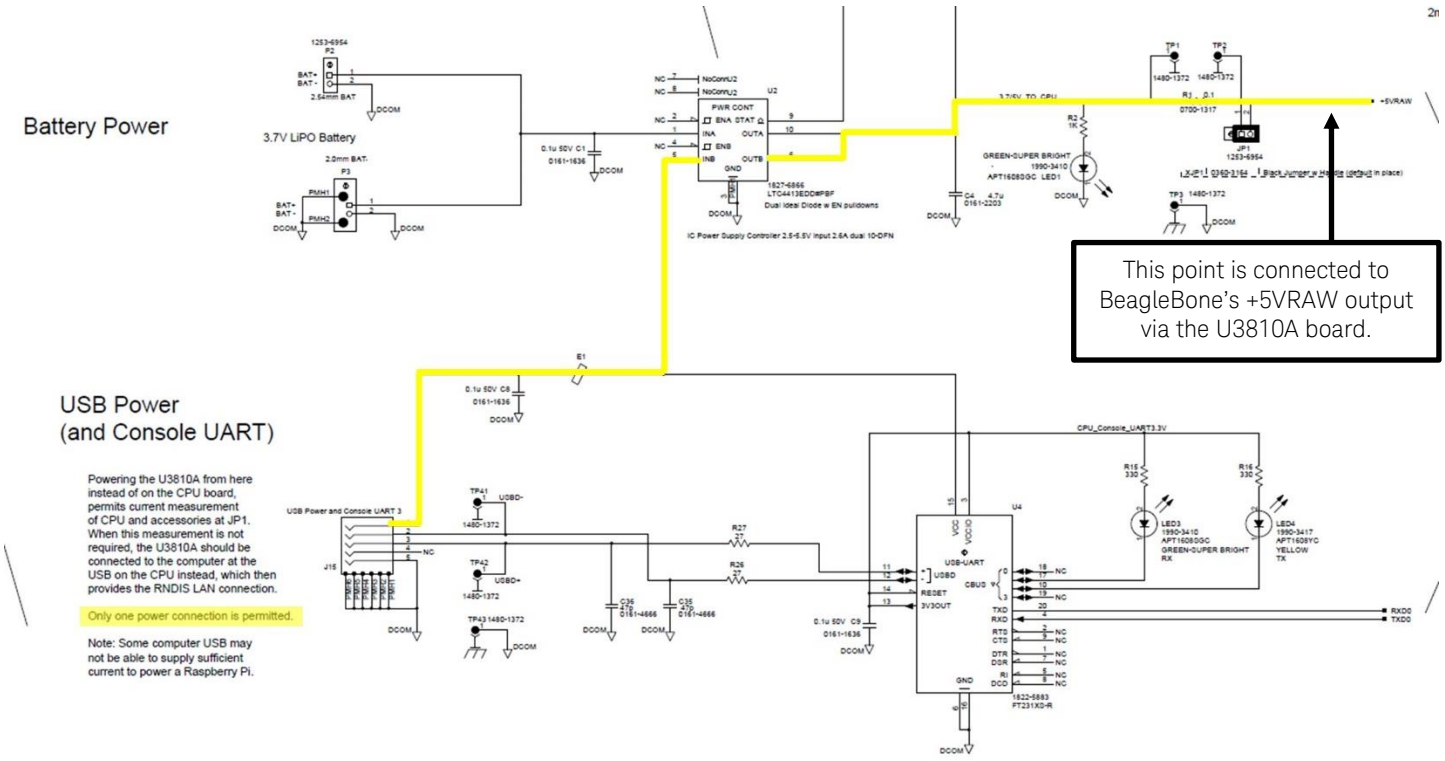
SD Card Correctly Inserted

SD Card Not Incorrectly Inserted

7 Connect a USB cable from your PC to J15 of the U3810A. When the JP55 is in place, you may need to press PWR BTN B7 to power on BeagleBone CPU.



WARNING When JP1 is in place, do not connect the USB cables to both the BeagleBone and J15 at the same time. This will interconnect the two USB power supply source and may cause contention. Anomalous behavior may result.



8 Observe that the four LEDs on BeagleBone (USR0, USR1, USR2, and USR3) will be in running state (scanning) during the firmware initialization. Once done (approximately 10 minutes), they will all be turned OFF.

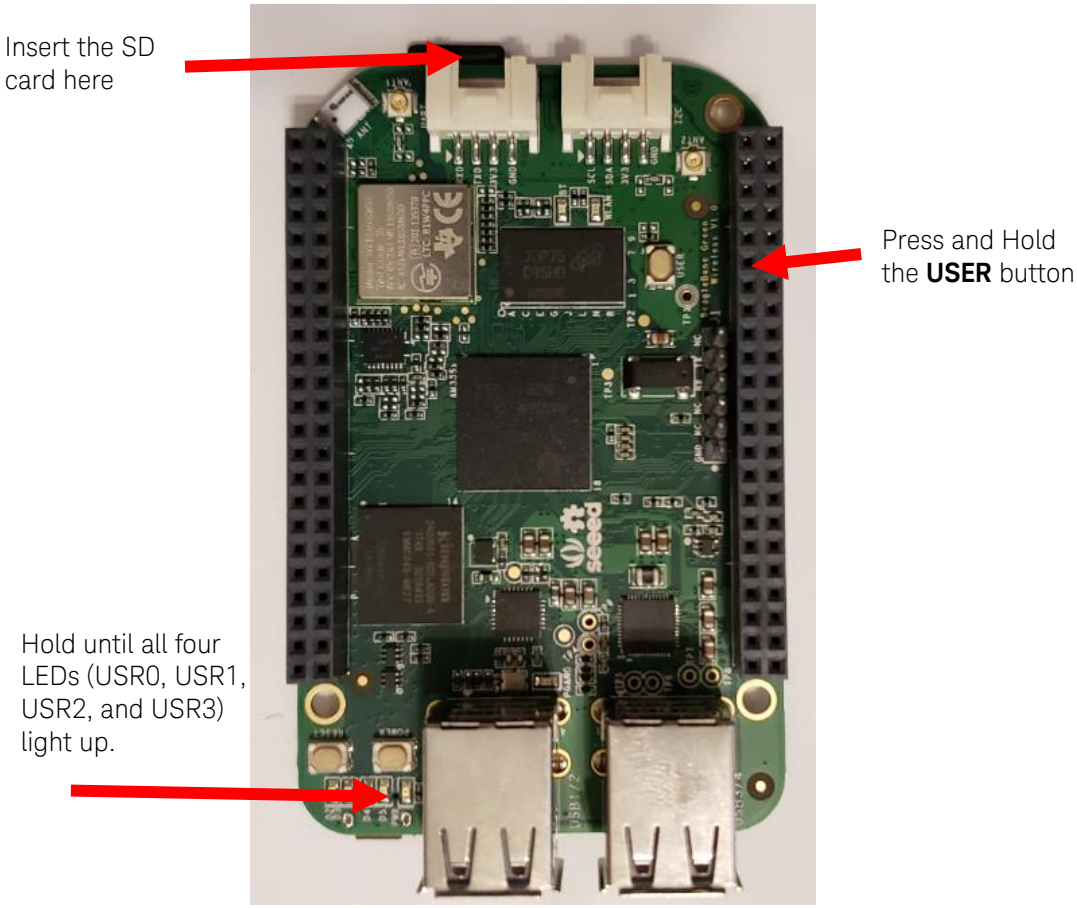
WARNING Do not power off the BeagleBone while the update is running. It may corrupt the eMMC and possibly render the BeagleBone useless. If the SD Card is left in and powered on, it will start a new initialization cycle. Wait for any power cycle to finish before disconnecting power.

Uninitialized BeagleBone devices

You will only need to perform this when the BeagleBone LEDs do not scan and turn off after 10 minutes (only for Un-initialized BeagleBones and older Debian images).

For some un-initialized BeagleBones or BeagleBones loaded with older Debian images, the **USER** button on the BeagleBone will need to be pressed and held until the four blue status lights all illuminate to install the image.

With the SDcard installed, press and hold the **USER** button as power is applied to the BeagleBone by connected the USB cable to your computer. Release the button when the four LEDs illuminate.

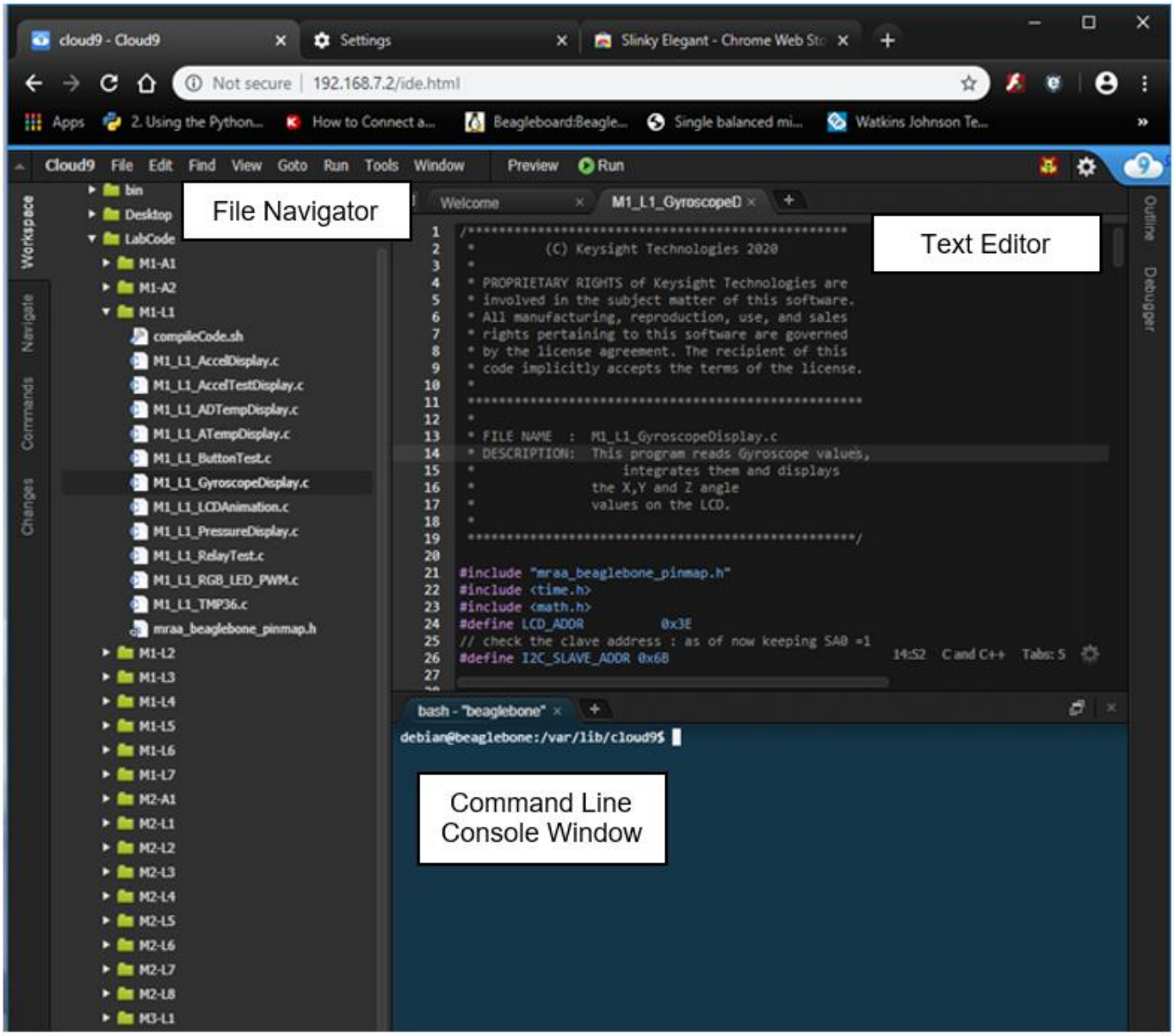


IMPORTANT - Remove the SD card while the BeagleBone is powered off. No LEDs should be active on the BeagleBone board. The SD card can be set aside and no longer needed for these labs.

9 Press the **PWR BTN** to power up the BeagleBone and verify the new image by connecting with PuTTY and observing the revision in the login notes.

Appendix B – Cloud 9 IDE Usage

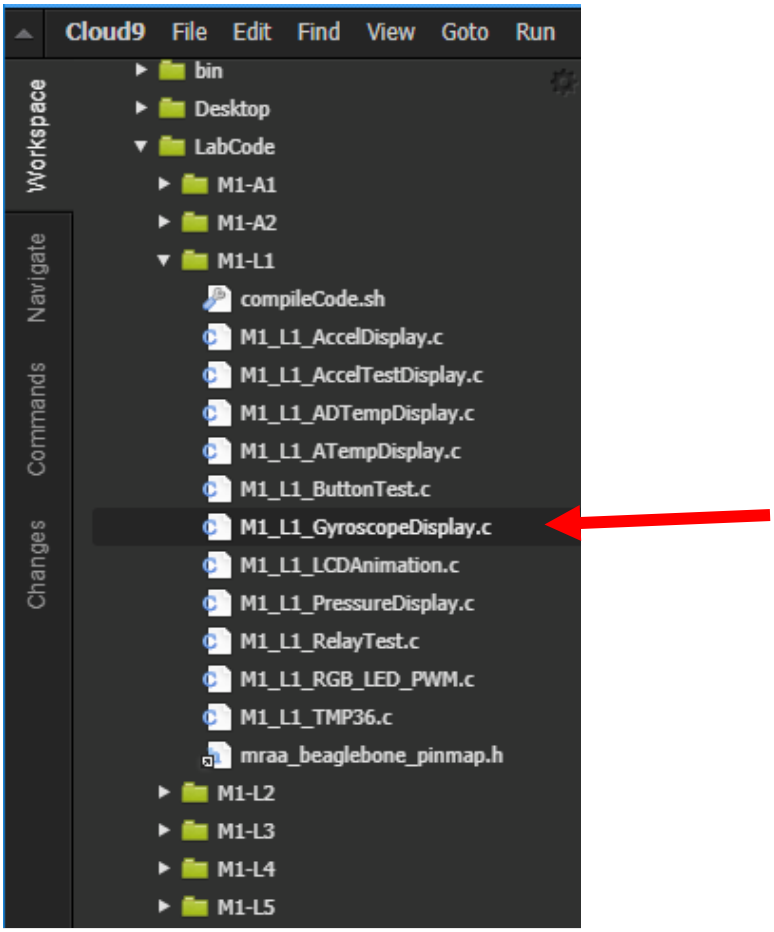
Over the RNDIS connection, the Cloud9 IDE can be seen by opening a web browser to http://192.168.7.2. The default page or the last saved state for the IDE should come up. The page has three major sections which are the file navigator, text editor, and the console window.



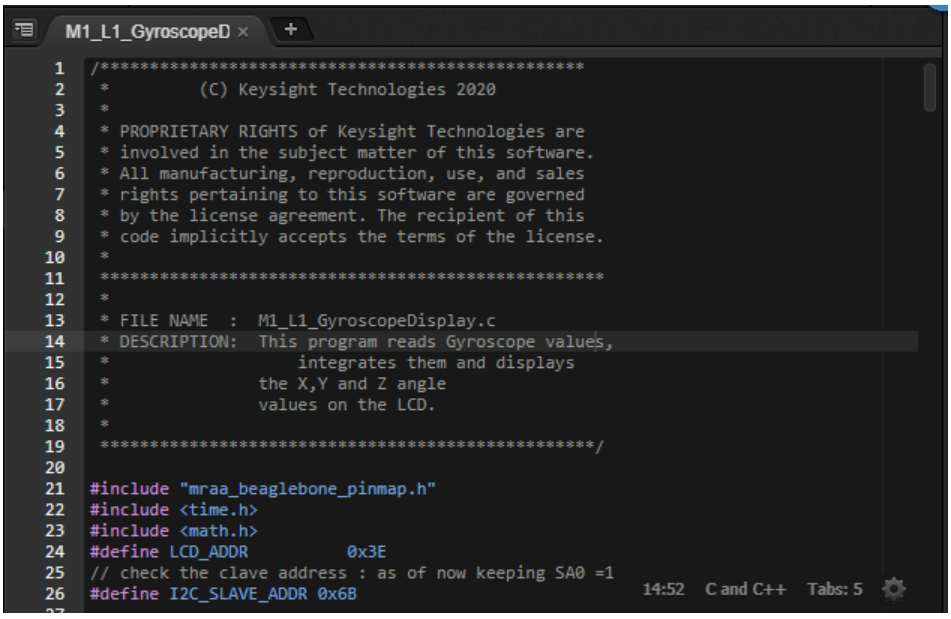
NOTE At the present time, only .js and .php files run using the debugger mode.

Run

1 Open a file in the editor and double-click **M1_L1_GyroscopeDisplay.c** file in the File Navigator.



The Editor window should now show the file below. This is a very intuitive text editor that uses the conventional Ctrl + C to copy and Ctrl + V to paste. Once the file has been modified, go to the console window to compile it.

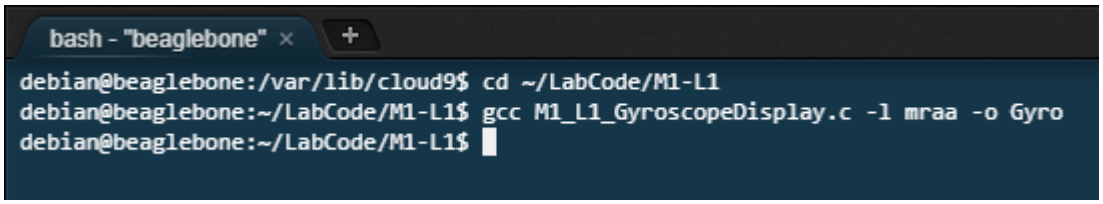


2 Run the following command in the console window to change to the directory that you are working in.

cd ~/LabCode/M1-L1

3 Run the following command in the console window to compile the C code.

gcc M1_L1_GyroscopeDisplay.c -l mraa -o Gyro



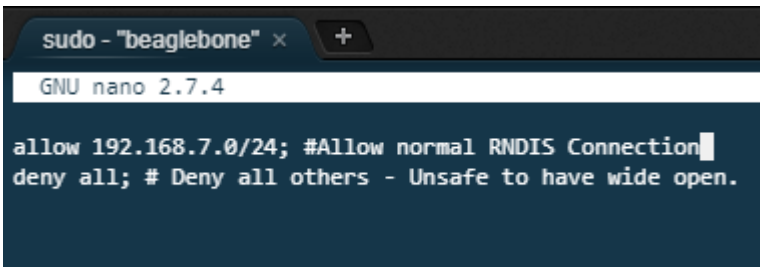
4 Enter **./Gyro** to run the code.

The Cloud9 IDE is secured so that it can only be accessed via the RNDIS port on 192.168.7.2.

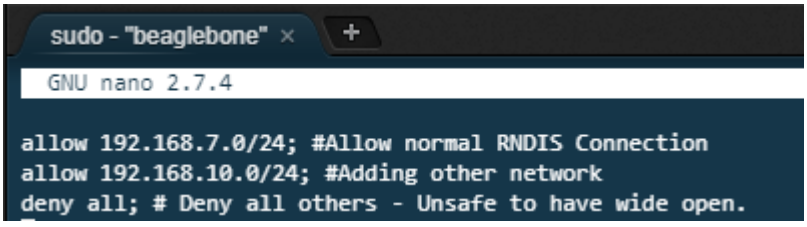
5 In order to enable other network access, you will need to edit this file with the sudo command **nginx server.blacklist.** This file is located at /etc/nginx/server.blacklist.

a To add other networks, add the networks to the allow section. This file will need to be edited with sudo command.

sudo nano /etc/nginx/server.blacklist



Once the file has been edited, write it out and exit the editor.



b To allow the new network access the nginx service will need to be restarted. To do this, run the command **sudo service nginx restart**. Access from other web browsers on the specified network can be made. That is, as long as the BeagleBone is connected to that network. Web browsers form different network locations will all see the same Cloud9 IDE. That is the information entered and display is the same. This works well for collaboration on problems. An instructor can open a browser window on a student IDE and help debug the problem.

WARNING It is strongly discouraged to enable all networks access to the Could9 IDE. It bypasses the login credentials.

Appendix C – Keysight U3810A Technical Documents

The most up-to-date copies of the following two documents are on your BeagleBone image in the LabCode folder.

Board Diagram (Searchable PDF)

This is a searchable PDF of the U3810A board diagram. Use this document to locate jumpers, connectors and components on the board.

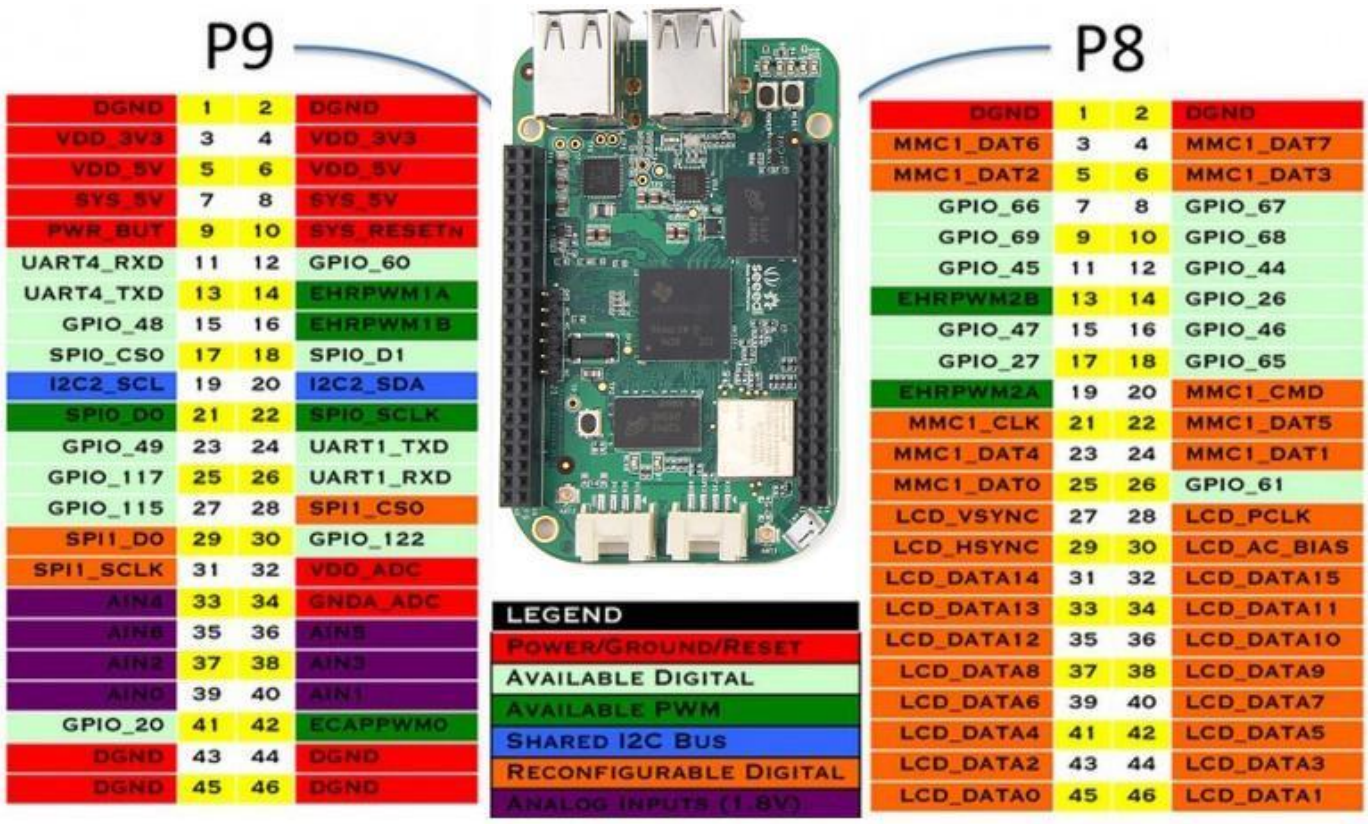
U3810-66501 Board Diagram.pdf

Schematic (Searchable PDF)

This is a searchable PDF of the U3810A schematic. Use this document to understand the electrical connections of the components and parts on the board.

U3810-66501 Schematic.pdf

Appendix D – BeagleBone Pinouts



Source: https://seeeddoc.github.io/Beaglebone_green_wireless/

Appendix E – Update, Upgrade, and Download Linux Packages

The Debian system can be used to add packages or update packages to a newer revision.

NOTE Your CPU will need to be connected to the internet via WLAN in order to perform the following downloads. Refer to Configure BeagleBone to connect to WLAN network for the instructions.

---

1 If you are continuing this lab from previous lab session;

   a. Make sure your U3810A jumper settings are set up according to Set Up the U3810A System.
   b. Connect your PC to the BeagleBone with a USB cable. Refer to Set Up Secure Shell (SSH) Communication.
   c. Establish a secure shell communication with the BeagleBone. Refer to Set Up SSH connection.

2 Run the following commands to update the Debian Linux Package Caches in the BeagleBone.

cd ~

sudo apt update

```
debian@beaglebone:~$ sudo apt update
[sudo] password for debian:
Ign:1 http://deb.debian.org/debian stretch InRelease
Get:2 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:3 http://deb.debian.org/debian-security stretch/updates InRelease
[94.3 kB]
.
.
.
Get:12 http://repos.rcn-ee.com/debian stretch/main armhf Packages [1,037
kB]
Get:13 http://deb.debian.org/debian stretch/non-free armhf Packages [59.7
kB]
Fetched 8,863 kB in 41s (215 kB/s)
Reading package lists... Done
```

3 Run the following command to see if there is a package with “Network Mapper” words.

apt-cache search Network Mapper

```
debian@beaglebone:~$ sudo apt-cache search Network Mapper
ndiff - The Network Mapper - result compare utility
nmap - The Network Mapper
zenmap - The Network Mapper Front End
```

4 Run the following command to install the Network Mapper program. You will need this for Module 1 Lab 7.

sudo apt install nmap

```
debian@beaglebone:~$ sudo apt install nmap
Reading package lists... Done
Building dependency tree
Reading state information... Done
nmap is already the newest version (7.40-1).
0 upgraded, 0 newly installed, 0 to remove and 202 not upgraded.
```

It is possible to upgrade the packages and operating system to the latest available revision. However, this can cause unexpected behavior of the system. New packages may not have the same functionality as the older packages. If a particular package needs to be upgraded, use the command **sudo apt upgrade <package>.** For example:

```
debian@beaglebone:~$ sudo apt upgrade nmap
[sudo] password for debian:
Reading package lists... Done
Building dependency tree
Reading state information... Done
nmap is already the newest version (7.40-1).
```

5 For a full upgrade of all packages, **sudo apt upgrade** , with no package specified. This download may require a significant amount of time and can require extra space. Look carefully at the upgrade messages before starting the process.

```
debian@beaglebone:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  bb-cape-overlays bb-customizations bone101 doc-beaglebone-getting-
started
  e2fslibs e2fsprogs file git git-core git-man libarchive13 libcomerr2
  libcpupower1 libexpat1 libexpat1-dev libmagic-mgc libmagic1 libsasl2-2
  libsasl2-modules-db libss2 libssl1.0.2 libssl1.1 linux-cpupower
  linux-libc-dev openssl sudo tzdata
27 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 262 MB of archives.
After this operation, 9,734 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Appendix F – Cloning a repository from GitHub to Install MRAA

In this section how to update and install Debian packages needed for MRAA installation. Then cloning an existing repository from GitHub to a local directory. Repeat all steps to Connect to WLAN if the system is not connected to the internet through Wi-Fi.

1 Enter the following line to download the required packages for the system.

sudo apt-get install build-essential python-dev cmake automake libpcre3 libpcre3-dev byacc flex swig3.0

2 Execute the following command to clone the entire mraa GitHub repository into the U3810A system.

git clone https://github.com/eclipse/mraa.git

```
debian@beaglebone:~$ git clone https://github.com/eclipse/mraa.git
Cloning into 'mraa'...
remote: Counting objects: 11859, done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 11859 (delta 50), reused 73 (delta 41), pack-reused 11757
Receiving objects: 100% (11859/11859), 3.25 MiB | 589.00 KiB/s, done.
Resolving deltas: 100% (8306/8306), done.
debian@beaglebone:~$
```

3 Execute the following command to check whether a mraa repository has been successfully downloaded.

cd mraa
ls

```
debian@beaglebone:~$ cd mraa
debian@beaglebone:~/mraa$ ls
api              docker-compose.yaml  DoxygenLayout.xml  jsstub     tools
cmake            docs                 doxyport           README.md
CMakeLists.txt   Doxyfile.in          examples           scripts
CONTRIBUTING.md  Doxyfile.java.in     imraa              src
COPYING          doxygen2jsdoc        include            tests
debian@beaglebone:~/mraa$
```

4 Once the packages are loaded, create a build area and descent into it.

```
debian@beaglebone:~$ cd mraa
debian@beaglebone:~/mraa$ mkdir build
debian@beaglebone:~/mraa$ cd build
```

5 Type **cmake -D CMAKE_INSTALL_PREFIX=/usr .. ..** to create the files to compile

```
debian@beaglebone:~/mraa/build$ cmake -D CMAKE_INSTALL_PREFIX=/usr ..
-- The C compiler identification is GNU 6.3.0
-- The CXX compiler identification is GNU 6.3.0
-- Check for working C compiler: /usr/bin/cc


  ●       ●       ●

-- Configuring done
-- Generating done
-- Build files have been written to: /home/debian/mraa/build
debian@beaglebone:~/mraa/build$
```

6 If there are no errors, it is ready for the next step which is to compile the files. There may be some warning messages during the compile, ignore them. To compile, type **make** and it may take a few minutes to compile.

```
debian@beaglebone:~/mraa/build$ make
Scanning dependencies of target mraa
[  1%] Building C object src/CMakeFiles/mraa.dir/mraa.c.o
[  2%] Building C object src/CMakeFiles/mraa.dir/gpio/gpio.c.o
[  3%] Building C object src/CMakeFiles/mraa.dir/gpio/gpio_chardev.c.o
[  4%] Building C object src/CMakeFiles/mraa.dir/i2c/i2c.c.o
[  5%] Building C object src/CMakeFiles/mraa.dir/pwm/pwm.c.o



  ●       ●       ●

[ 97%] Built target mraa-i2c
Scanning dependencies of target mraa-uart
[ 98%] Building C object tools/CMakeFiles/mraa-uart.dir/mraa-uart.c.o
[100%] Linking C executable mraa-uart
[100%] Built target mraa-uart
debian@beaglebone:~/mraa/build$
```

NOTE Do not continue if there are errors in the **make**. Warnings are allowed.

---

7 To install what was built the older revision of the code will need to be removed. To do this it is an apt-get uninstall of the mraa files done in Task 1e Step 2. Type **sudo apt-get remove libmraa1 libmraa-dev mraa-tools**.

8 Install the build with elevated privileges **sudo make install**. This will place all the executables and link libraries in the correct locations.

9 Check the build by typing **mraa-gpio list**. The result should look something like this.

```
debian@beaglebone:~$ mraa-gpio list
01         GND:
02         GND:
03     MMC1_D6:
04     MMC1_D7:
05     MMC1_D2:
06     MMC1_D3:
07      GPIO66: GPIO
08      GPIO67: GPIO
09      GPIO69: GPIO
10      GPIO68: GPIO
11      GPIO45: GPIO
12      GPIO44: GPIO
13      GPIO23: GPIO PWM
14      GPIO26: GPIO
15      GPIO47: GPIO
16      GPIO46: GPIO
17      GPIO27: GPIO
18      GPIO65: GPIO
19      GPIO22: GPIO PWM
20    MMC1_CMD:
21    MMC1_CLK:
22     MMC1_D5:
23      MMC_D4:
24      MMC_D1:
25     MMC1_D0:
26      GPIO61: GPIO
27      GPIO86: GPIO
28      GPIO88: GPIO
29      GPIO87: GPIO
30      GPIO89: GPIO
31      GPIO10: GPIO
32      GPIO11: GPIO
33       GPIO9: GPIO
34      GPIO81: GPIO PWM
35       GPIO8: GPIO
36      GPIO80: GPIO PWM
37      GPIO78: GPIO UART
38      GPIO79: GPIO UART
39      GPIO76: GPIO
40      GPIO77: GPIO
41      GPIO74: GPIO
42      GPIO75: GPIO
43      GPIO72: GPIO
44      GPIO73: GPIO
45      GPIO70: GPIO PWM
46      GPIO71: GPIO PWM
47         GND:
48         GND:
49        3.3V:
50        3.3V:
51          5V:
52          5V:
```

```
53          5V:
54          5V:
55         PWR:
56       RESET:
57      GPIO30: GPIO UART
58      GPIO60: GPIO
59      GPIO31: GPIO UART
60      GPIO50: GPIO PWM
61      GPIO48: GPIO
62      GPIO51: GPIO PWM
63     I2C1SCL: I2C  SPI
64     I2C1SDA: I2C  SPI
65     I2C2SCL: I2C
66     I2C2SDA: I2C
67       GPIO3: GPIO SPI  PWM  UART
68    EHRPWM0A: PWM
69      GPIO49: GPIO
70      GPIO15: GPIO UART
71     GPIO117: GPIO
72      GPIO14: GPIO UART
73     GPIO115: GPIO
74     GPIO113: GPIO SPI
75     GPIO111: GPIO SPI
76     GPIO112: GPIO SPI
77     GPIO110: GPIO SPI
78     VDD_ADC:
79        AIN4: AIO
80     GND_ADC:
81        AIN6: AIO
82        AIN5: AIO
83        AIN2: AIO
84        AIN3: AIO
85        AIN0: AIO
86        AIN1: AIO
87      GPIO20: GPIO
88       GPIO7: GPIO
89         GND:
90         GND:
91         GND:
92         GND:
debian@beaglebone:~$
```

Appendix G – Restore U3810A startup files

1. If you are continuing this lab from previous lab session;
   - **a** Make sure your U3810A jumper settings are set up according to Set Up the U3810A System.
   - **b** Connect your PC to the BeagleBone with a USB cable. Refer to Set Up Secure Shell (SSH) Communication.
   - **c** Establish a secure shell communication with the BeagleBone. Refer to Set Up SSH connection.
2. Run the following command to unzip the Startup Scripts from its ZIP archive.

   ```
   cd ~/LabCode
   tar -xzf startup_scripts.gz
   ```

3. Execute the following command to change your working directory to the startup directory.

   ```
   cd startup_scripts
   ```

4. Execute the following command to run the autodisplay.sh script. This will reset your U3810A startup to the factory state.

   ```
   sudo ./autodisplay.sh
   ```

Appendix H – U3810A Image Build from Scratch

1. Download the latest IoT image from http://beagleboard.org/latest-images. The LXQT image may not have enough space to hold all the packages and software required.
2. Burn the image onto a SD card using Belena Etcher as described in Appendix A – Initialize BeagleBone with Keysight U3810A Image. In this case, use the downloaded image from the BeagleBone.org site.
3. For a new BeagleBone, the eMMC needs to be initialized with newer firmware. Use the procedure in Uninitialized BeagleBone devices.
4. This will boot from the SD card login as debian
5. Connect to the internet using Wi-Fi or bridge PC connection via RNDiS.
6. Turn off the tether mode. (Tether mode leaves a wide open access point. This should be disabled)

   Edit /etc/default/bb-wl18xx Change TEATHER_ENABLED=no and USE_CONNMAN_TEATHER=no

```
# TETHER_ENABLED: Whether or not to run the /usr/bin/bb-wl18xx-tether
daemon; set to no to disable.
TETHER_ENABLED=no

# USE_CONNMAN_TETHER: Whether or not to just use connman tether inteface;
set to no to disable.
USE_CONNMAN_TETHER=no
```

7. Download the required packages for the system, enter the following line.

   sudo apt install build-essential python-dev cmake automake libpcre3 libpcre3-dev byacc flex python-pip nmap paho-mqtt gspread oauth2client bluepy debsums mosquitto mosquitto-clients libglib2.0

8. Install the required Python Packages required for the system. **sudo pip install paho-mqtt gspread oauth2client bluepy**
9. There is one NPM package that needs to be installed: **sudo npm install node-rest-client**.
10. Execute the following command to clone the entire mraa GitHub repository into the U3810A system.

    git clone https://github.com/eclipse/mraa.git

```
debian@beaglebone:~$ git clone https://github.com/eclipse/mraa.git
Cloning into 'mraa'...
remote: Counting objects: 11859, done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 11859 (delta 50), reused 73 (delta 41), pack-reused 11757
Receiving objects: 100% (11859/11859), 3.25 MiB | 589.00 KiB/s, done.
Resolving deltas: 100% (8306/8306), done.
debian@beaglebone:~$
```

11 Execute the following commands to check whether a mraa repository has been successfully downloaded.

cd mraa

ls

```
debian@beaglebone:~$ cd mraa
debian@beaglebone:~/mraa$ ls
api              docker-compose.yaml  DoxygenLayout.xml  jsstub     tools
cmake            docs                 doxyport           README.md
CMakeLists.txt   Doxyfile.in          examples           scripts
CONTRIBUTING.md  Doxyfile.java.in     imraa              src
COPYING          doxygen2jsdoc        include            tests
debian@beaglebone:~/mraa$
```

Once the packages are loaded, create a build area and descent into it.

```
debian@beaglebone:~$ cd mraa
debian@beaglebone:~/mraa$ mkdir build
debian@beaglebone:~/mraa$ cd build
```

12 Create the files for compilation by typing **cmake -D CMAKE_INSTALL_PREFIX=/usr .. ..**

```
debian@beaglebone:~/mraa/build$ cmake -D CMAKE_INSTALL_PREFIX=/usr ..
-- The C compiler identification is GNU 6.3.0
-- The CXX compiler identification is GNU 6.3.0
-- Check for working C compiler: /usr/bin/cc
. . .
-- Configuring done
-- Generating done
-- Build files have been written to: /home/debian/mraa/build
debian@beaglebone:~/mraa/build$
```

If there are no errors, it is ready for the next step which is to compile the files. There may be some warning messages during the compile, ignore them.

13 Type **make** and it may take a few minutes to compile.

```
debian@beaglebone:~/mraa/build$ make
Scanning dependencies of target mraa
[  1%] Building C object src/CMakeFiles/mraa.dir/mraa.c.o
[  2%] Building C object src/CMakeFiles/mraa.dir/gpio/gpio.c.o
[  3%] Building C object src/CMakeFiles/mraa.dir/gpio/gpio_chardev.c.o
[  4%] Building C object src/CMakeFiles/mraa.dir/i2c/i2c.c.o
[  5%] Building C object src/CMakeFiles/mraa.dir/pwm/pwm.c.o
. . .
[ 97%] Built target mraa-i2c
Scanning dependencies of target mraa-uart
[ 98%] Building C object tools/CMakeFiles/mraa-uart.dir/mraa-uart.c.o
[100%] Linking C executable mraa-uart
[100%] Built target mraa-uart
debian@beaglebone:~/mraa/build$
```

WARNING

Do not continue if there are errors in the make. Warnings are allowed.

If there is an install from an apt install process, the older revision of the code will need to be removed. To do this it is an apt remove of the mraa files done. Type in **sudo apt remove libmraa1 libmraa-dev mraa-tools**

14 Type **sudo make install** to install the mraa system into the BeagleBone.

15 Type **mraa-gpio list** to check the build.

```
debian@beaglebone:~$ mraa-gpio list
01         GND:
02         GND:
03     MMC1_D6:
04     MMC1_D7:
05     MMC1_D2:
06     MMC1_D3:
07      GPIO66: GPIO
08      GPIO67: GPIO
09      GPIO69: GPIO
10      GPIO68: GPIO
11      GPIO45: GPIO
12      GPIO44: GPIO
13      GPIO23: GPIO PWM
14      GPIO26: GPIO
15      GPIO47: GPIO
16      GPIO46: GPIO
17      GPIO27: GPIO
18      GPIO65: GPIO
19      GPIO22: GPIO PWM
20    MMC1_CMD:
21    MMC1_CLK:
22     MMC1_D5:
23      MMC_D4:
24      MMC_D1:
25     MMC1_D0:
26      GPIO61: GPIO
27      GPIO86: GPIO
28      GPIO88: GPIO
29      GPIO87: GPIO
30      GPIO89: GPIO
31      GPIO10: GPIO
32      GPIO11: GPIO
33       GPIO9: GPIO
34      GPIO81: GPIO PWM
35       GPIO8: GPIO
36      GPIO80: GPIO PWM
37      GPIO78: GPIO UART
38      GPIO79: GPIO UART
39      GPIO76: GPIO
40      GPIO77: GPIO
41      GPIO74: GPIO
42      GPIO75: GPIO
43      GPIO72: GPIO
44      GPIO73: GPIO
45      GPIO70: GPIO PWM
46      GPIO71: GPIO PWM
47         GND:
48         GND:
49        3.3V:
50        3.3V:
51          5V:
```

```
52          5V:
53          5V:
54          5V:
55         PWR:
56       RESET:
57      GPIO30: GPIO UART
58      GPIO60: GPIO
59      GPIO31: GPIO UART
60      GPIO50: GPIO PWM
61      GPIO48: GPIO
62      GPIO51: GPIO PWM
63     I2C1SCL: I2C  SPI
64     I2C1SDA: I2C  SPI
65     I2C2SCL: I2C
66     I2C2SDA: I2C
67       GPIO3: GPIO SPI  PWM  UART
68    EHRPWM0A: PWM
69      GPIO49: GPIO
70      GPIO15: GPIO UART
71     GPIO117: GPIO
72      GPIO14: GPIO UART
73     GPIO115: GPIO
74     GPIO113: GPIO SPI
75     GPIO111: GPIO SPI
76     GPIO112: GPIO SPI
77     GPIO110: GPIO SPI
78     VDD_ADC:
79        AIN4: AIO
80     GND_ADC:
81        AIN6: AIO
82        AIN5: AIO
83        AIN2: AIO
84        AIN3: AIO
85        AIN0: AIO
86        AIN1: AIO
87      GPIO20: GPIO
88       GPIO7: GPIO
89         GND:
90         GND:
91         GND:
92         GND:
debian@beaglebone:~$
```

16 Edit the BeagleBone Overlay file in /boot/uEnv.txt. The key overlays changes are highlighted. Do not change the uname_r as that is linked to an image.

```
# Docs: http://elinux.org/Beagleboard:U-boot_partitioning_layout_2.0
# Keysight U3810A Image Version 3.64 Feb 3rd 2020
#This version of uEnv.txt enables UART1, UART2, I2C1, PWM1, PWM2 and
allows
#MRAA Pins 74 - 77 to be used as SPI

uname_r=4.14.108-ti-r115
#uuid=
#dtb=
```

```
###U-Boot Overlays###
###Documentation: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian#U-Boot_Overlays
###Master Enable
enable_uboot_overlays=1
###
###Overide capes with eeprom
uboot_overlay_addr0=/lib/firmware/BB-UART1-00A0.dtbo
uboot_overlay_addr1=/lib/firmware/BB-UART2-00A0.dtbo
uboot_overlay_addr2=/lib/firmware/BB-I2C1-00A0.dtbo
uboot_overlay_addr3=/lib/firmware/BB-PWM1-00A0.dtbo
uboot_overlay_addr4=/lib/firmware/BB-PWM2-00A0.dtbo
###
###Additional custom capes
#uboot_overlay_addr4=/lib/firmware/<file4>.dtbo
#uboot_overlay_addr5=/lib/firmware/<file5>.dtbo
#uboot_overlay_addr6=/lib/firmware/<file6>.dtbo
#uboot_overlay_addr7=/lib/firmware/<file7>.dtbo
###
###Custom Cape
#dtb_overlay=/lib/firmware/<file8>.dtbo
###
###Disable auto loading of virtual capes (emmc/video/wireless/adc)
#disable_uboot_overlay_emmc=1
#disable_uboot_overlay_video=1
disable_uboot_overlay_audio=1
#disable_uboot_overlay_wireless=1
#disable_uboot_overlay_adc=1
###
###PRUSS OPTIONS
###pru_rproc (4.4.x-ti kernel)
#uboot_overlay_pru=/lib/firmware/AM335X-PRU-RPROC-4-4-TI-00A0.dtbo
###pru_rproc (4.14.x-ti kernel)
uboot_overlay_pru=/lib/firmware/AM335X-PRU-RPROC-4-14-TI-00A0.dtbo
###pru_rproc (4.19.x-ti kernel)
#uboot_overlay_pru=/lib/firmware/AM335X-PRU-RPROC-4-19-TI-00A0.dtbo
###pru_uio (4.4.x-ti, 4.14.x-ti, 4.19.x-ti & mainline/bone kernel)
#uboot_overlay_pru=/lib/firmware/AM335X-PRU-UIO-00A0.dtbo
###
###Cape Universal Enable
enable_uboot_cape_universala=1
###
###Debug: disable uboot autoload of Cape
#disable_uboot_overlay_addr1=1
#disable_uboot_overlay_addr2=1
#disable_uboot_overlay_addr3=1
###
###U-Boot fdt tweaks... (60000 = 384KB)
#uboot_fdt_buffer=0x60000
###U-Boot Overlays###

cmdline=coherent_pool=1M net.ifnames=0 rng_core.default_quality=100 quiet

#In the event of edid real failures, uncomment this next line:
```

```
#cmdline=coherent_pool=1M net.ifnames=0 rng_core.default_quality=100
quiet video=HDMI-A-1:1024x768@60e

#Use an overlayfs on top of a read-only root filesystem:
#cmdline=coherent_pool=1M net.ifnames=0 rng_core.default_quality=100
quiet overlayroot=tmpfs

##enable Generic eMMC Flasher:
##make sure, these tools are installed: dosfstools rsync
#cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
```

17 A reboot is required to test the overlays. If there is an older version of the operating system, the USR button as describe in the Uninitialized BeagleBone devices procedure.

18 On reboot examine the devices in /dev. There should be three I²C devices, a pwm device, ttyS0, ttyS1, ttyS2 and no spi devices.

```
debian@beaglebone:~$ ls /dev
apm_bios          kmem                snapshot  tty27  tty50   ttyS3
autofs            kmsg                snd       tty28  tty51   ttyS4
bc_example        log                 stderr    tty29  tty52   ttyS5
block             loop-control        stdin     tty3   tty53   ubi_ctrl
btrfs-control     mapper              stdout    tty30  tty54   uhid
bus               mem                 tty       tty31  tty55   uinput
char              memory_bandwidth    tty0      tty32  tty56   urandom
console           mmcblk1             tty1      tty33  tty57   vcs
cpu_dma_latency   mmcblk1boot0        tty10     tty34  tty58   vcs1
cuse              mmcblk1boot1        tty11     tty35  tty59   vcs2
disk              mmcblk1p1           tty12     tty36  tty6    vcs3
dri               mmcblk1rpmb         tty13     tty37  tty60   vcs4
fd                mqueue              tty14     tty38  tty61   vcs5
full              net                 tty15     tty39  tty62   vcs6
fuse              network_latency     tty16     tty4   tty63   vcsa
gpiochip0         network_throughput  tty17     tty40  tty7    vcsa1
gpiochip1         null                tty18     tty41  tty8    vcsa2
gpiochip2         ppp                 tty19     tty42  tty9    vcsa3
gpiochip3         ptmx                tty2      tty43  ttyGS0  vcsa4
hwrng             pts                 tty20     tty44  ttyO0   vcsa5
i2c-0             pwm                 tty21     tty45  ttyO1   vcsa6
i2c-1             random              tty22     tty46  ttyO2   vhci
i2c-2             rfkill              tty23     tty47  ttyO3   watchdog
iio:device0       rtc                 tty24     tty48  ttyS0   watchdog0
initctl           rtc0                tty25     tty49  ttyS1   watchdog1
input             shm                 tty26     tty5   ttyS2   zero
```

In /sys/class/pwm, there should be pwmchip0 and pwmchip2.

```
debian@beaglebone:~$ ls /sys/class/pwm
pwm-0:0  pwm-0:1  pwm-2:0  pwm-2:1  pwmchip0  pwmchip2
```

19 To communicate with the serial devices minicom needs to be installed and set up. With the Beaglebone connected to the internet, install minicom by using the command: **`sudo apt install minicom`**

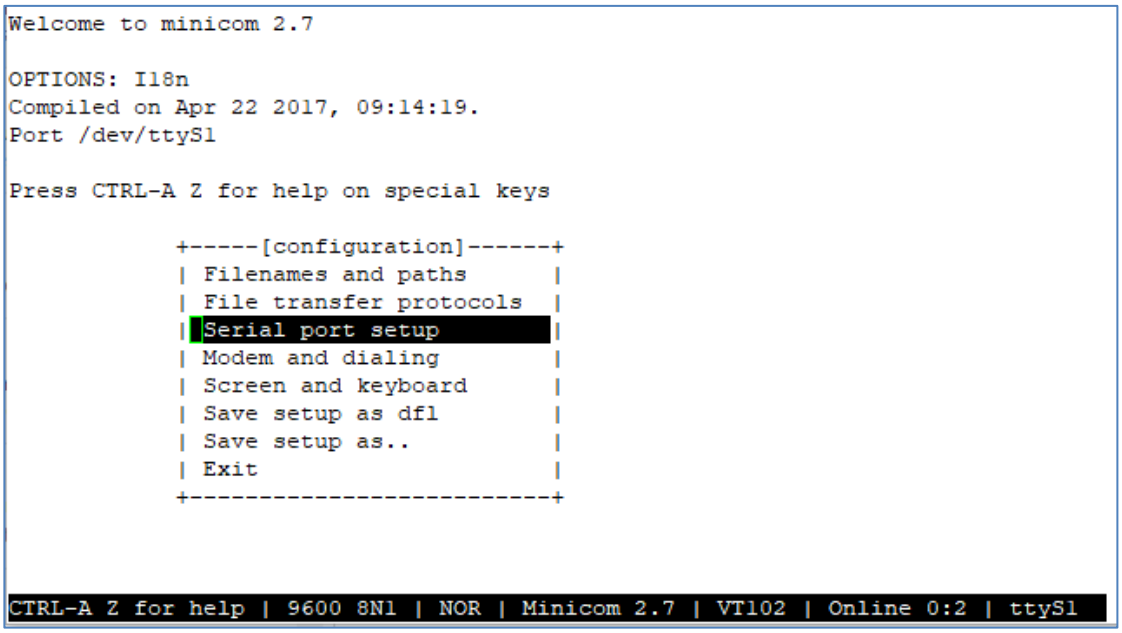
20 Now that minicom has been installed, there are two set up items. The default has hardware control and no echo. Both of these features need to be enabled.

21 To enable these features, a terminal needs to be open. For this ttyS1 can be used. To open minicom run the command: minicom -D /dev/ttyS1 -b 9600

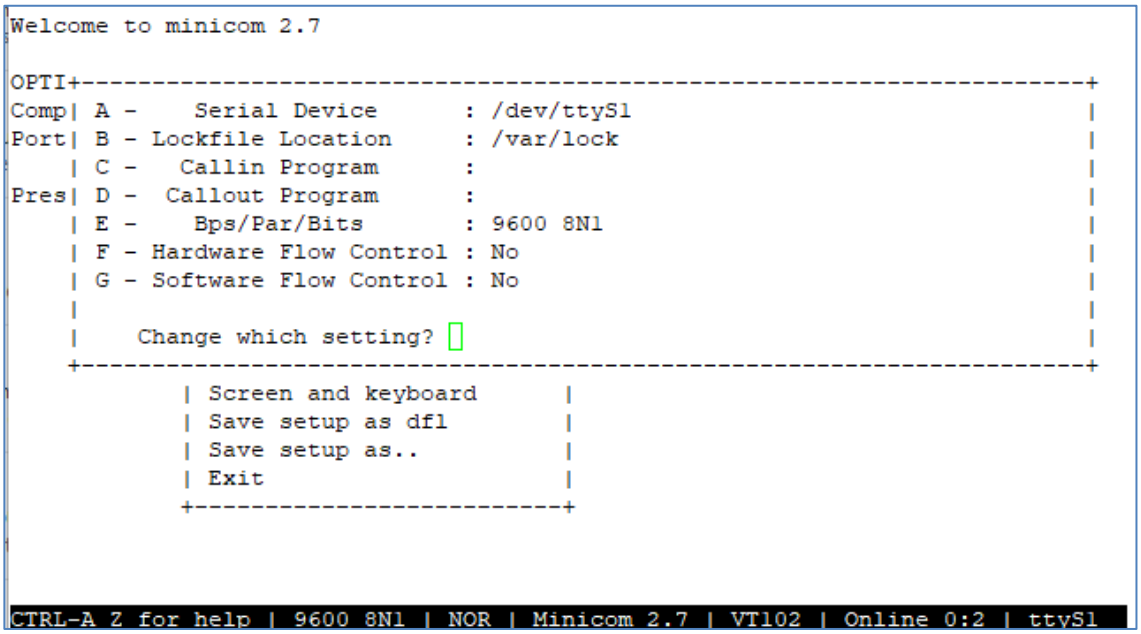
A terminal screen should come up like the figure below:

```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
Port /dev/ttyS1

Press CTRL-A Z for help on special keys
```

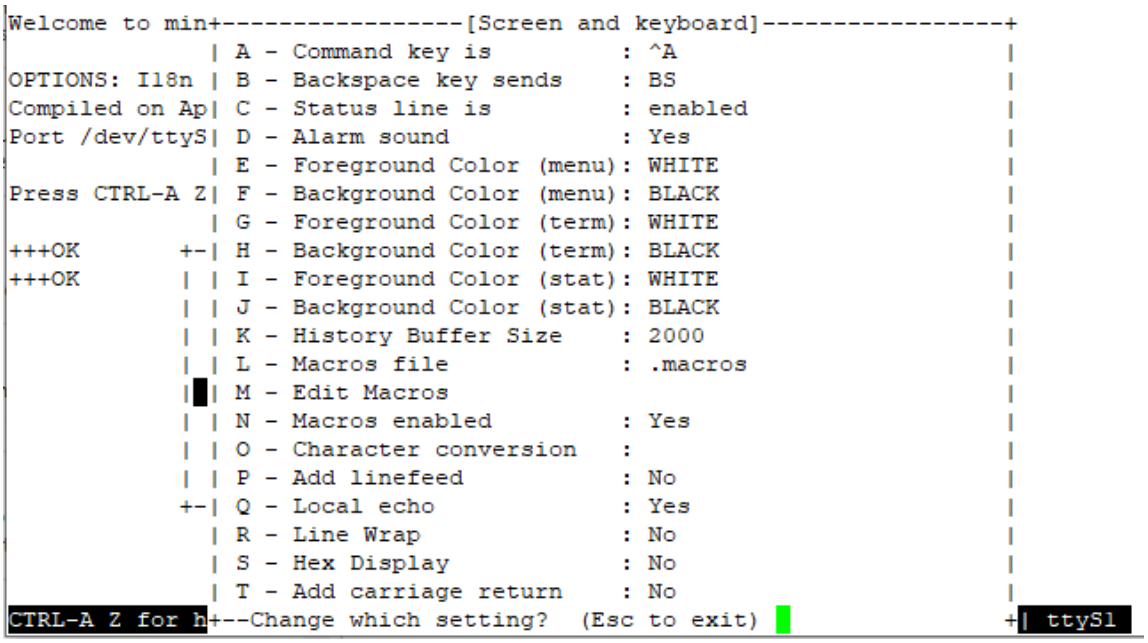
22 To get into the minicom options menu mode, press Ctrl + A and then o. Use the arrow keys to move to the Serial port setup selection.



23 Press the **Enter**, a new menu will come up. Press the **F key** to turn off hardware control.



24 Press the **Enter key** to exit the Serial port set up menu. Use the arrow key to select **Screen and keyboard**. For the ttyS1 and ttyS2 devices do not echo back the characters typed in, enable the local echo by using the **q key** to enable echo.



25 Use **Enter key** to leave the **Screen and keyboard** menu. To save the settings use the arrow keys to select the **save setup ad dft**. Press the **Enter key** and the defaults settings will be the setting from this session.

26 If the devices all show up it is time to install the U3810A code. Acquire the latest LabCode.tar.gz and LabCodeSignature.tar.gz from Keysight. Download this as well as the LabCodeReset.gz file to the Debian home directory.

LabCodeReset.gz

27 Extract the LabCodeReset.sh file by using the **`tar -xzf LabCodeReset.gz`** command and also extract the signature files by using the **`tar -xzf LabCodeSignature.tar.gz`** command.

28 Execute the file by entering **`./LabCodeReset.sh -u`** This will ask for the root password for it to install the latest revision of the LabCode.

```
debian@beaglebone:~/Desktop$ ./LabCodeReset.sh -u
The LabCode.tar.gz has been signed.
Checking on the signature
gpg: assuming signed data in 'checksum'
gpg: Signature made Mon 27 Jul 2020 01:24:21 PM UTC
gpg:                using RSA key E89C4532A5DB38EBE14CF510F55535C5FA4EB16E
gpg: Good signature from "Copyright Keysight Technologies 2020" [ultimate]
Checking on SHA256 hash ...
LabCode.tar.gz: OK
Do you want to continue? (y/n) y
Erasing LabCode
Refreshing LabCode
LabCode/
...

.KS_Files/boot/uEnv.txt
Updating Keysight Revisions
[sudo] password for debian:
Reboot is suggested
```

29 Before rebooting, follow the directions in Appendix D. Restoring U3810A startup files for setting up the startup scripts.

30 Reboot the system one more time. Once again If there is an older version of the operating system, the USR button as describe in the Uninitialized BeagleBone devices procedure.

31 On reboot test the LabCode and hardware interaction. Follow the Hardware Verification for U3810A section to test the system.

32 If everything is working, any entries into the connmanctl Wi-Fi Access Point table should be deleted. Connmanctl stores this information in plain text and can be read by anyone receiving a copy of the image. To delete the access point information sudo rm -Rf /var/lib/connman/wifi*

33 To clean up any compiled files or extra files execute **./LabCodeReset.sh** from the debian home directory. Select the "y" option to delete the code before refreshing.

```
debian@beaglebone:~/Desktop$ ./LabCodeReset.sh
The LabCode.tar.gz has been signed.
Checking on the signature
gpg: assuming signed data in 'checksum'
gpg: Signature made Mon 27 Jul 2020 01:24:21 PM UTC
gpg:                using RSA key E89C4532A5DB38EBE14CF510F55535C5FA4EB16E
gpg: Good signature from "Copyright Keysight Technologies 2020" [ultimate]
Checking on SHA256 hash ...
LabCode.tar.gz: OK
Do you want to continue? (y/n) y
Erasing LabCode
Refreshing LabCode
LabCode/
LabCode/M2-L1b/
LabCode/M2-L1b/M2-L1-E7_Led.c
LabCode/M2-L1b/Xbee Profiles/
LabCode/M2-L1b/Xbee Profiles/M2_L1_T7_END_DEVICE.xpro
LabCode/M2-L1b/Xbee Profiles/M2_L1_T7_ROUTER.xpro
LabCode/M2-L1b/Xbee Profiles/M2_L1_T7_COORDINATOR.xpro
LabCode/M2-L1b/mraa_beaglebone_pinmap.h
…
LabCode/M3-L7/M3_L7_Temp_Post_MQTT.py
LabCode/M3-L7/M3_L7_T4_sleepwake.c
LabCode/M3-L7/M3_L7_LCD_Fun.py
.KS_Files/
.KS_Files/etc/
.KS_Files/etc/issue
.KS_Files/etc/environment
.KS_Files/etc/motd
.KS_Files/Version.txt
.KS_Files/boot/
.KS_Files/boot/.uEnv.txt.swp
.KS_Files/boot/uEnv.txt
debian@beaglebone:~$
```

34 To turn the SD card into a version that will flash the code to the eMMC, change the last line of /boot/uEnv.txt to remove the "#".

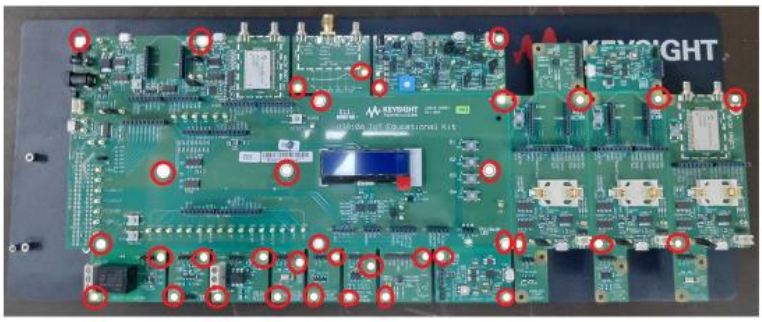
```
##enable Generic eMMC Flasher:
##make sure, these tools are installed: dosfstools rsync
#cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh

# Change to line without comment symbol

cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
```

35 The SD card is now ready to test by burring the image to a BeagleBone. Follow the directions in Appendix A – Initialize BeagleBone with Keysight U3810A Image with this SD card in place of the Keysight image. Start at step 4.

Appendix I – Assembly and Disassembly

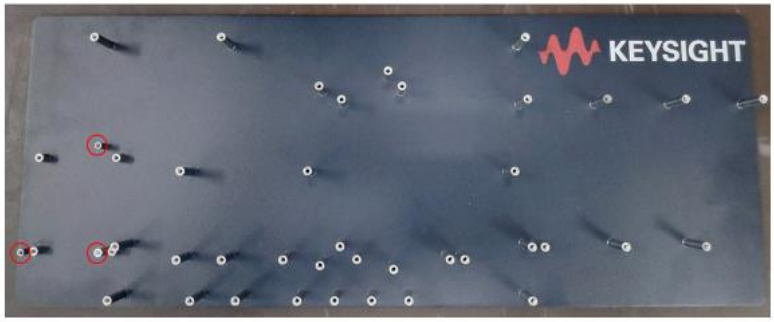
Assembly of BeagleBone Green Wireless (BBGW)

BBGW Assembly (For U3810A Shipped without BBGW)



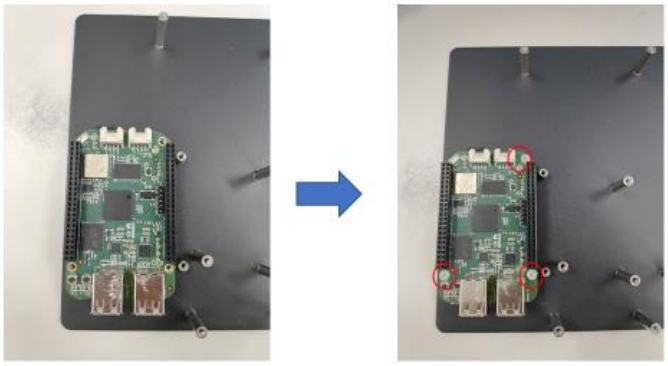
- Remove the screws marked in red using a Philips or Posidrive screwdriver

BBGW Assembly (For U3810A Shipped without BBGW)



- Remove the U3810A board and put it aside
- The BBGW will be placed on the standoffs which are marked in red

BBGW Assembly (For U3810A Shipped without BBGW)



- Place BBGW on the standoffs
- Install the plastic screws provided at the holes marked in red using a Philips or Posidrive screwdriver

BBGW Assembly (For U3810A Shipped without BBGW)

- Align the 46pin connector and the 6pin connector on the U3810A board with the ones on the BBGW
- Slowly press down on the areas marked in red to engage the connectors

BBGW Assembly (For U3810A Shipped without BBGW)

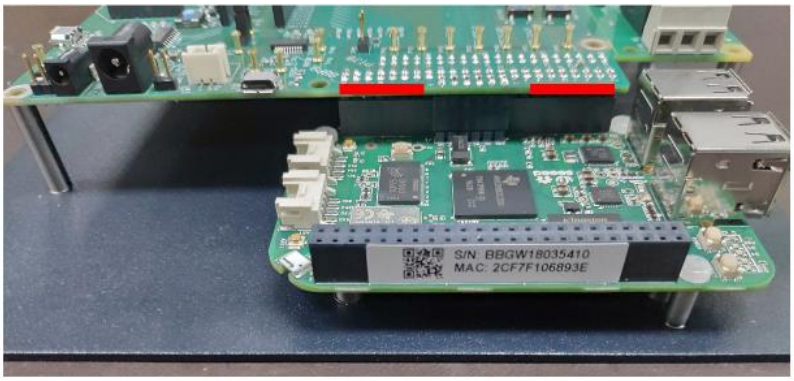
- Re-install all the plastic screws at the areas marked in red using a Philips or Posidrive screwdriver

Disassembly

BBGW Dis-assembly

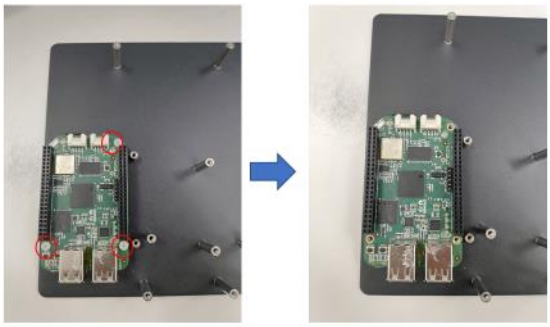
- Remove the screws marked in red using a Philips or Posidrive screwdriver

BBGW Dis-assembly



- Lift the U3810A board up slowly at the two areas highlighted in red. (Note: Lifting only on one side may damage the connector pins)

BBGW Dis-assembly



- Remove the U3810A board and put it aside.
- Remove the screws marked in red using a Philips or Posidrive screwdriver
- The BBGW can now be replaced

References

[1] How to Use Nano, the Linux Command Line Text Editor
https://linuxize.com/post/how-to-use-nano-text-editor/

Edition 2, May 2021
Printed in Malaysia

U3810-90002
www.keysight.com