Keysight U3810A Advanced IoT Teaching Solution

U3815/16A IoT Wireless Communications and Compliance

Lab 6: *Bluetooth*® Low Energy (LE) and Zigbee Modulation Analysis and Coexistence

Lab Sheet

KEYSIGHT TECHNOLOGIES

Notices

Copyright Notice

Trademark

Bluetooth® and the *Bluetooth*® logos are trademarks owned by *Bluetooth*® SIG, Inc., U.S.A. and licensed to Keysight Technologies, Inc.

Edition

Edition 1, June 2020

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.

Table of Contents

Objective

1. To analyze RF spectrum and modulation of *Bluetooth*® LE and Zigbee (802.15.4) using vector signal analyzer.
2. Understand some of the RF factors contributing to Wireless Device Coexistence.
3. To view interference between *Bluetooth*® and Zigbee signals

Pre-Lab Setup Instructions

1. Prepare the required items as listed in 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.
3. Make sure that the student has completed Lab 1 before attempting this lab.

Equipment Required

1. 1 x Keysight U3810A IoT development kit with new BeagleBone Wireless CPU with XBee3 installed in its XB socket
2. 1 x RF Signal Analyzer Keysight N9000B CXA with option 503 (option 507 preferred), 89600 VSA software
3. Mouse and keyboard for Keysight N9000B CXA RF Signal Analyzer
4. 1 x A laptop or desktop PC running on Windows with Internet access (Linux and macOS may work but are not presently on Keysight’s list of supported platforms)
5. (Optional - If no screen room is available) 1 x
Keysight U3830A WaveChamber or
Keysight IXIA IxVeriWave WCH2900, Small WaveChamber (980-2083-03)

Accessories Required

1. 1 x TI SensorTag
2. 2 × Micro-USB cables
3. 1 x SMA(f) to SMA(m) RF Coaxial Cable
4. 1 x N-type(m) to SMA(f) Adaptor
5. 2 x SMA-RP(m) to SMA-RP(m) cables
6. 1 x XBee3 module in either XB1 or XB2 Transceiver socket
7. 1 x XBee3 module in socket on U3810A XB Transceiver socket

Software Required

1. PuTTY (http://www.putty.org/)
2. XCTU (https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu)
3. WinSCP (https://winscp.net/eng/download.php)
4. iPerf version 2.0.9 for Keysight U3810A IoT Development Kit and PC (TCP, UDP, and SCTP speed test tool). (https://iperf.fr/iperf-download.php)

IMPORTANT

You must complete U3813, 14, 15 or 16 Lab 1 before you start this lab. There are basic operations that are covered in Lab 1 will not be covered in this lab.

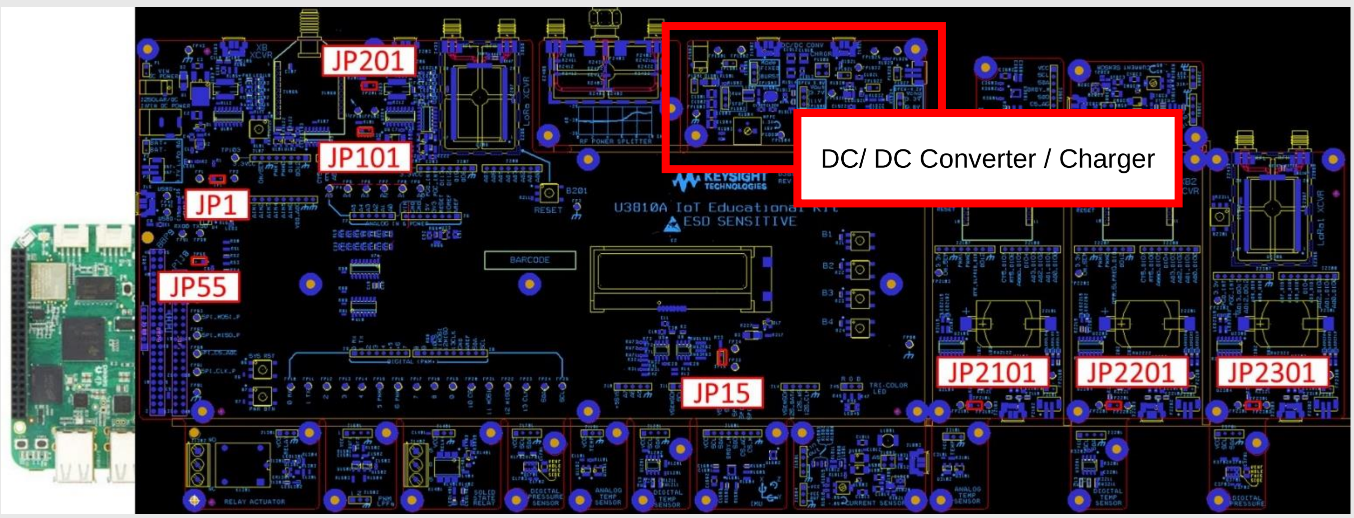
Task 1 - *Bluetooth*® LE Modulation Analysis

In the prior WLAN lab sheet, you learned about IQ modulation and WLAN modulation analysis. In the next two tasks, you will learn how to measure and analyze *Bluetooth*® LE and Zigbee (IEEE 802.15.4) modulation using the vector signal analyzer.

NOTE

Before you begin the experiment, configure the Keysight U3810A IoT Development Kit 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



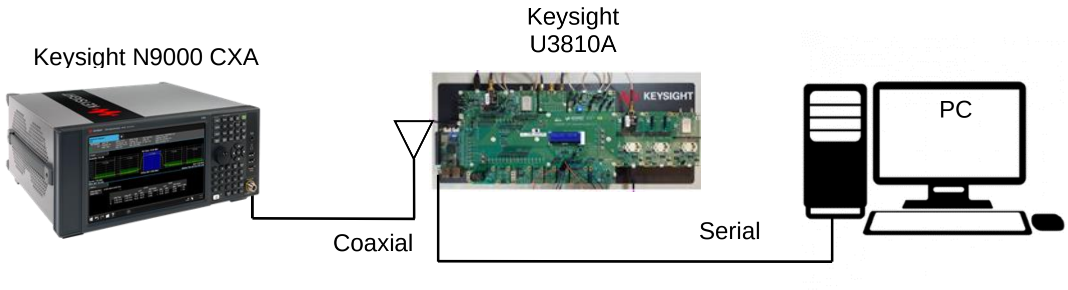
** DC/ DC Converter / Charger Board jumper positions are not relevant to this lab.

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

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.

The setup to carry out the measurement in this experiment is given in the figure below:



Task 1a - Initial Measurement Setup

1. Connect the N-type(m) to SMA(f) adaptor to the CXA signal analyzer RF input port. Connect the Antenna to the SMA(m) to SMA(f) RF coaxial cable and connect the other end of the cable to the CXA. Place the antenna close to the BeagleBone (which generates the *Bluetooth®* LE signal) on the U3810A board during measurements. This is to ensure the signal analyzer takes in only the *Bluetooth®* LE signal from the U3810A board. Connect a mouse and keyboard to the signal analyzer and turn it on.

2. Power up BeagleBone CPU and login in to BeagleBone via SSH login in Putty Terminal with following details.

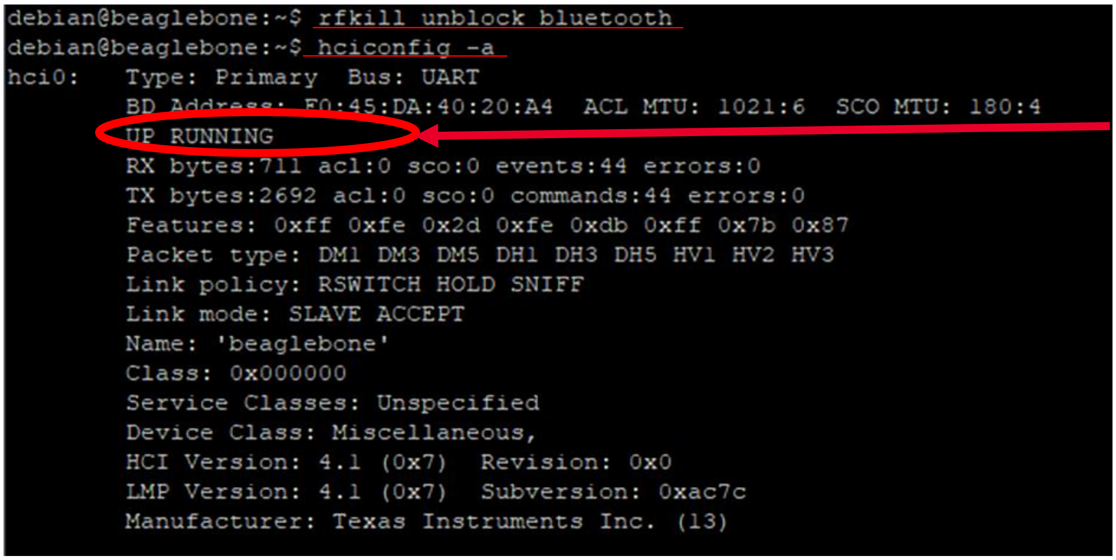
Host Name	192.168.7.2
Port	22
Connection type	SSH
Username	debian
Password	temppwd

3. The lab codes might have been modified in previous lab session and you need to reset the codes back to default. Run the following commands to reset the lab codes.

   `cd ~`

   `sh LabCodeReset.sh`

4. Enable *Bluetooth*® using **`rfkill unblock bluetooth`** and **`hciconfig -a`** to check for the status of the *Bluetooth*® interface on the BeagleBone (U3810A). A sample screenshot is shown below.



```
debian@beaglebone:~$ rfkill unblock bluetooth
debian@beaglebone:~$ hciconfig -a
hci0:   Type: Primary  Bus: UART
        BD Address: F0:45:DA:40:20:A4  ACL MTU: 1021:6  SCO MTU: 180:4
        UP RUNNING
        RX bytes:711 acl:0 sco:0 events:44 errors:0
        TX bytes:2692 acl:0 sco:0 commands:44 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: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Version: 4.1 (0x7)  Revision: 0x0
        LMP Version: 4.1 (0x7)  Subversion: 0xac7c
        Manufacturer: Texas Instruments Inc. (13)
```

Enable page scan, disable inquiry scan.

5. Enable *Bluetooth*® LE test mode using the following commands:

```
debian@BeagleBone: ~# sudo hcitool cmd 0x06 0x0003
debian@BeagleBone: ~# sudo hcitool cmd 0x03 0x0003
debian@BeagleBone: ~# sudo hcitool cmd 0x03 0x0059 0x00
```

   The first command puts the module into test-mode; the second command will reset the *Bluetooth*® LE interface. The third command is for setting the output power to 0 dBm.

Sending commands to the BeagleBone:

Upon power-up the BeagleBone will ask for login and password. The default login is “debian” and the default password is “temppwd”. Certain commands to the BeagleBone require “supervisor” privilege. To use these commands, you must prefix the commands with “sudo” which means “supervisor do” which permits execution of that privileged function. The first time using the supervisor privilege you will be required to give the supervisor password (by default it is “temppwd”). Periodically, you may be asked again to give the password when using “sudo”. This is normal operation.

6. Start the transmitter test by using the following command:

   `debian@BeagleBone: ~# sudo hcitool cmd 0x08 0x001E 0x00 0x25 0x02`

   The **`hcitool cmd 0x08 0x001E 0x00 0x25 0x02`** command transmits the test packets in channel 0 (2402 MHz) with a payload length of 37 bytes and the test pattern is the alternating bits '10101010'.

```
debian@beaglebone:~$ sudo hcitool cmd 0x06 0x0003
[sudo] password for debian:
< HCI Command: ogf 0x06, ocf 0x0003, plen 0
> HCI Event: 0x0e plen 4
  01 03 18 00
debian@beaglebone:~$ sudo hcitool cmd 0x03 0x0003
< HCI Command: ogf 0x03, ocf 0x0003, plen 0
> HCI Event: 0x0e plen 4
  01 03 0C 00
debian@beaglebone:~$ sudo hcitool cmd 0x03 0x0059 0x00
< HCI Command: ogf 0x03, ocf 0x0059, plen 1
  00
> HCI Event: 0x0e plen 4
  01 59 0C 00
debian@beaglebone:~$ sudo hcitool cmd 0x08 0x001E 0x00 0x25 0x02
< HCI Command: ogf 0x08, ocf 0x001e, plen 3
  00 25 02
> HCI Event: 0x0e plen 4
  01 1E 20 00
debian@beaglebone:~$
```

   The last response "`01 1E 20 00`" indicates that the command was received and the transmitter test has begun. The device in test mode will send one packet every 625 µs.

NOTE

The transmitter tests the command "**`hcitool cmd 0x08 0x001E 0xXX 0xYY 0xZZ`**", the channel, payload length and test packet payload type are determined by the XX, YY and ZZ respectively. The details are as below:

XX = the channel you want to transmit on, any value from 0x00 to 0x27

(*Bluetooth®* LE channels go from 0 to 39, where the first channel is 0 and the last channel is 39).

The channel labeling is different from those of the *Bluetooth®* LE standard.

YY = length of payload bytes in each test packet, which can be any value from 0x00 to 0x25

ZZ = code for the type of data in the packet payload (refer to the table above for test packet payload type).

7. On the signal analyzer, launch the 89600 VSA software.

NOTE

If you are using a new CXA for this lab, check your CXA hardware configuration. Go to **Utilities > Hardware > Configuration**, then choose the **Analyzer 2 > Keysight/Agilent X-Series Signal Analyzer**.

8. Preset the VSA by clicking "**File > Preset > Setup**" to reset the settings to default values.
9. Go to measurement setup ("**MeasSetup > Frequency**"). Change the "**Centre**" frequency to the operating frequency of the *Bluetooth®* LE connection. The operating channel used in this case is **2402 MHz**.
10. Go to "**MeasSetup > Measurement Type: > General Purpose > Digital Demod**" to set the VSA to demodulate the *Bluetooth®* LE test packets.

11. The easiest way to demodulate the *Bluetooth*® LE (GFSK) signal is to use the standard preset supported by the VSA. Go to "**MeasSetup > Digital Demod Properties > Preset to Standard…**" and select "**Wireless Networking > Bluetooth**". The default parameter settings for this standard are given in the table below (the key settings are highlighted in red circles).

Parameter	Bluetooth
Demod Format	FSK 2
Frequency Span	2.5 MHz
Symbol Rate	1 MHz
Filtering	
- Measured	Cleared
- Reference	Gaussian
Alpha / BT	0.5
Points / Symbol	2
Result Length	320 symbols
Search Length	1650 Syms
Pulse Search	Cleared
Constellation Sync Search	Cleared
Normalize IQ Traces	Selected
Clock Adjust	0.0
Equalization	—

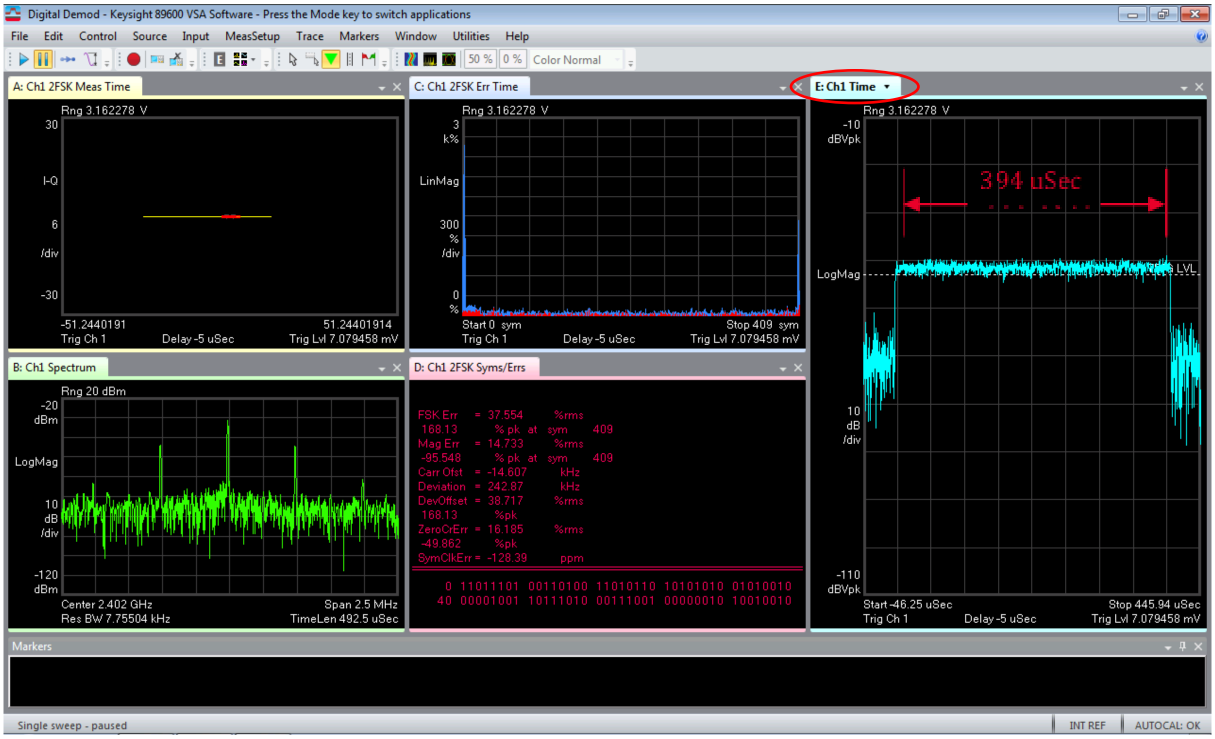
The "Frequency Span" used is 2.5 MHz which is about two times the bandwidth of the signal. The "Symbol Rate" (1 MHz) is the rate of the modulated signal. The bandwidth-bit period product is set to 0.5 (as per the *Bluetooth*® LE standard). "Pulse Search" is disabled.

12. Some fine adjustments to the default settings are needed. First, change the "**Points/Symbol**" to **20** ("**MeasSetup > Digital Demod Properties > Format > Points/Symbols**"). Then, change the "**Result Length**" to **410 symbols** ("**MeasSetup > Digital Demod Properties > Format > Result Length**") so that one complete *Bluetooth*® LE test packet is visible.

13. Go to "**Window > Trace Layout > Grid 2x3**" to set the trace layout. Set Trace E to "**Channel 1>Time**" and change the Y Scale to dBm ("**Trace > Y Scale**" and set Y unit to **Power**). Remove Trace F.

14. Go to "**Input >Trigger**", change the "**Style**" to "**IF Mag**" and set the "**Level**" to a power level above the noise floor (such as, -30 dBm). Set the "**Delay**" to **-5 µs**, so that the starting part of the captured packet can be seen. When the results are available on the display, stop the signal capturing.

    Right click to **Autoscale** the windows as necessary. A sample screenshot is shown below.



On the "Time" trace, right-click to use the markers to measure the packet duration. The upper left trace shows the constellation diagram for the GFSK modulation. The result is not as expected due to the noisy signals included in the measurement data (that is, the starting and ending parts of the packet). You may use the Small WaveChamber recommended by Keysight to capture the *Bluetooth*® LE test packet or perform this task inside a RF-shielded chamber for better result.

NOTE

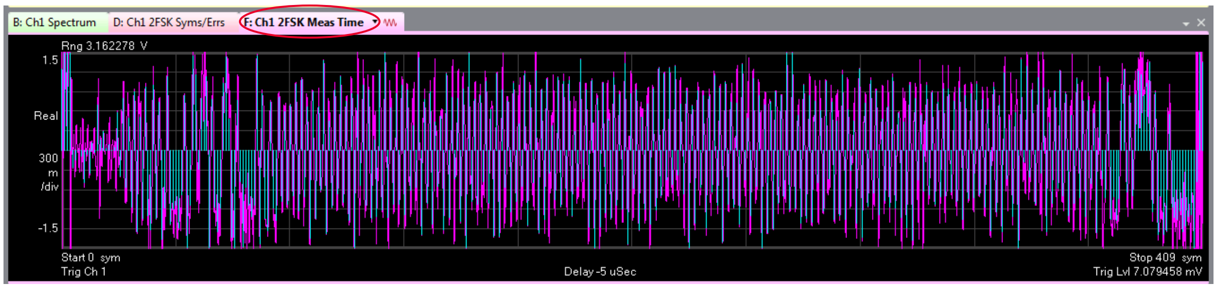
Use WinSCP to copy the M2_L6_T1_SETUP.setx from the BeagleBone /LabCode/M2-L6 folder to your computer. Copy it to your CXA using USB and use the setup file for this task. Go to "**File > Recall > Recall Setup…"** and select the setup file. Change the center frequency to the operating frequency of your Zigbee channel.

Task 1b - *Bluetooth*® LE Test Packet

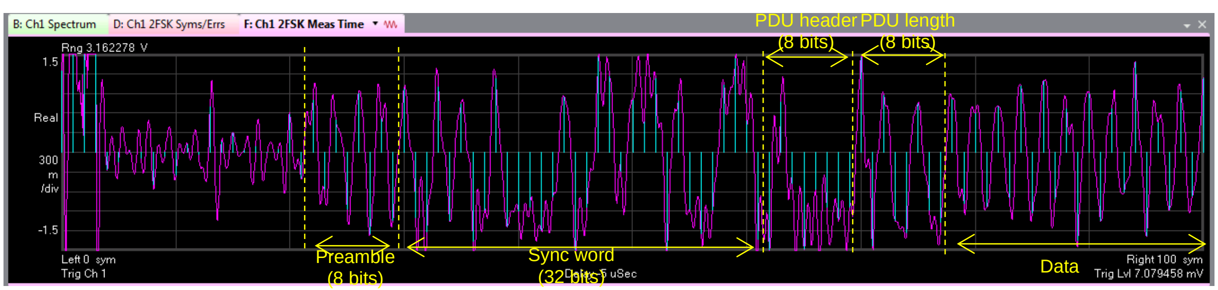
In this experiment, the format of the *Bluetooth*® LE test packet is verified by using the FSK demodulation function of the VSA. Follow the instructions below.

1. First, add a trace to display the demodulated waveform. Go to "**Trace > Add new trace**", then "**Data > Channel 1**" and select "**FSK Meas Time**". Go to "**Trace > Format**", change the "**Format**" to "**Real (I)**". The display result represents a frequency versus time plot. Change the Y per Division to 300 m at "**Trace > Y Scale**" and change display to dual layout ("**Window > Trace Layout > Stack 2**"). This is the output waveform of the FSK demodulator. When the measurement is paused, mouse-click the **Restart** button to continue. Traces A, C, and E may show no data.

   A sample screenshot is given below.



2. Zoom in to the starting part of the captured packet of **Trace F**. Go to "**Trace > X Scale…**", clear the "**Full Scale**" checkbox and change the "**Right reference**" to **100 symbols**. The following waveform will appear.



   A binary one is represented by a positive frequency deviation, and a binary zero is represented by a negative frequency deviation. As shown in the demodulated waveform above, the data payload has a repeating pattern of 10101010. This is expected since the transmitted test pattern is 0x02 or 10101010 (see Section 1.1 above). The *Bluetooth*® LE test packet starts with a preamble 01010101 follows by a sync word of length 32 bits. The first four bits of the PDU header indicates the payload type. In this case, it is 0100. Since the rightmost bit is the MSB, 0100 equals to 2. This confirms that the test pattern is of type 0x02 (or 10101010). The first six bits of the PDU length indicates the length of the payload. In this case, the demodulated data is 101001. Hence, the payload length is 37 bytes (or 296 bits).

3. In a similar way, zoom in to the ending part of the test packet and verify the length of the CRC section (see Exercise).

Exercise

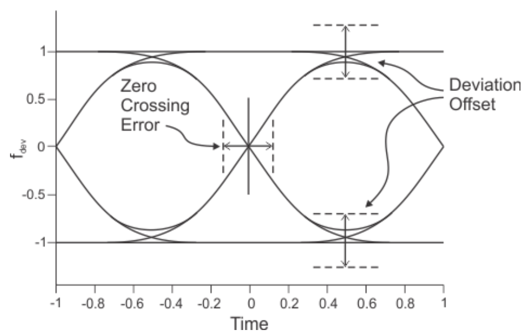
Display the ending portion of the demodulated *Bluetooth*® LE test packet and verify that the CRC section contains 24 bits.

Task 1c - Modulation Accuracy

Several performance measures are given in the “Syms/Errs” trace of the FSK demodulation function. The following table briefly describes the definitions of these measures.

Parameter	Description
FSK Err	FSK Error is computed by comparing the FSK reference signal with the FSK signal measured at the symbol locations
Mag Err	Carrier Magnitude Error shows the magnitude error of the carrier over all symbols
Carr Ofst	Carrier Offset shows the carrier frequency error relative to the VSA's center frequency
Deviation	Deviation is the average deviation of all symbols in the FSK Meas Time trace (determined by Result Length). A symbol's Deviation is the distance between the detected carrier frequency and the measured frequency.
DevOffset	Deviation Offset is the amount of variation in the frequency Deviation of the measured signal, relative to that of the reference signal.
ZeroCrErr	Zero Crossing Error is the variation of the zero crossing of the measured signal, relative to the reference signal.
SymClkErr	Symbol Clock Error is a measure of the accuracy of the 2-FSK modulation's symbol clock

The figure below illustrates the definitions for “Deviation Offset (DevOffset)” and “Zero Crossing Error (ZeroCrErr)”.



Procedure

1. Connect the N-type(m) to SMA(f) adaptor to the CXA signal analyzer RF input port. Connect the Antenna to the SMA(m) to SMA(f) cable and connect the other end of the cable to the CXA. Place the antenna close to the BeagleBone (which generates the *Bluetooth*® LE signal) on the U3810A board during measurements. This is to ensure the signal analyzer takes in only the *Bluetooth*® LE signal from the U3810A board. Connect a mouse and keyboard to the signal analyzer and turn it on.

2. Establish a serial SSH communication session with the U3810A board using PuTTY.

3. Enable *Bluetooth*® using `rfkill unblock bluetooth` and `hciconfig -a` to check for the status of the *Bluetooth*® interface on the BeagleBone (U3810A). A sample screenshot is given below:

```
debian@beaglebone:~$ rfkill unblock bluetooth
debian@beaglebone:~$ hciconfig -a
hci0:   Type: Primary  Bus: UART
        BD Address: F0:45:DA:40:20:A4  ACL MTU: 1021:6  SCO MTU: 180:4
        UP RUNNING
        RX bytes:711 acl:0 sco:0 events:44 errors:0
        TX bytes:2692 acl:0 sco:0 commands:44 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: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Version: 4.1 (0x7)  Revision: 0x0
        LMP Version: 4.1 (0x7)  Subversion: 0xac7c
        Manufacturer: Texas Instruments Inc. (13)
```

4. Enable *Bluetooth*® LE test mode by using the following commands:

```
debian@BeagleBone: ~# sudo hcitool cmd 0x06 0x0003

debian@BeagleBone: ~# sudo hcitool cmd 0x03 0x0003

debian@BeagleBone: ~# sudo hcitool cmd 0x03 0x0059 0x00
```

   The first command puts the module into test-mode; the second command will reset the *Bluetooth*® LE interface. The third command is for setting the output power to 0 dBm.

5. Start the transmitter test using the following command:

   `debian@BeagleBone: ~# sudo hcitool cmd 0x08 0x001E 0x00 0x25 0x00`

   The command **`hcitool cmd 0x08 0x001E 0x00 0x25 0x00`** transmits the test packets in channel 0 (2402 MHz) with a payload length of 37 bytes and the data payload of PRBS9. See the screenshot of the output on the Command Prompt Window below.

```
debian@beaglebone:~$ sudo hcitool cmd 0x06 0x0003
[sudo] password for debian:
< HCI Command: ogf 0x06, ocf 0x0003, plen 0
> HCI Event: 0x0e plen 4
  01 03 18 00
debian@beaglebone:~$ sudo hcitool cmd 0x03 0x0003
< HCI Command: ogf 0x03, ocf 0x0003, plen 0
> HCI Event: 0x0e plen 4
  01 03 0C 00
debian@beaglebone:~$ sudo hcitool cmd 0x03 0x0059 0x00
< HCI Command: ogf 0x03, ocf 0x0059, plen 1
  00
> HCI Event: 0x0e plen 4
  01 59 0C 00
debian@beaglebone:~$ sudo hcitool cmd 0x08 0x001E 0x00 0x25 0x00
< HCI Command: ogf 0x08, ocf 0x001e, plen 3
  00 25 00
> HCI Event: 0x0e plen 4
  01 1E 20 00
debian@beaglebone:~$
```

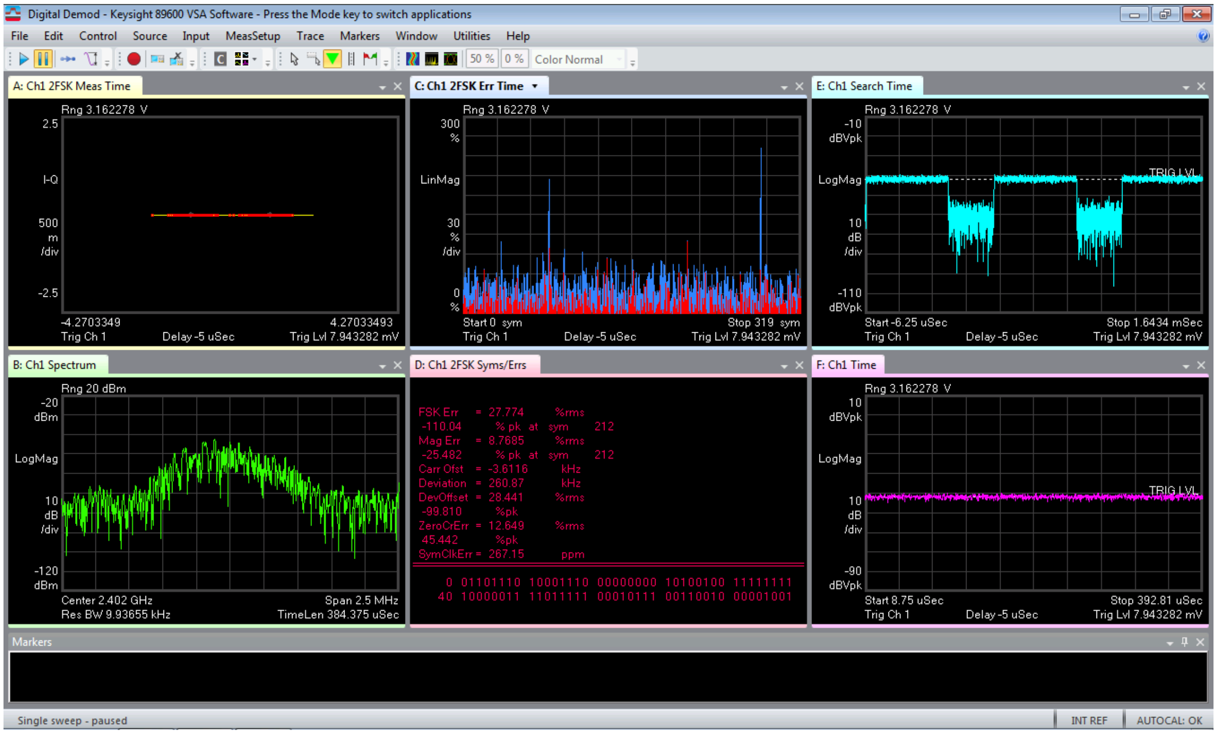
   For proper performance evaluation, only the data symbols in the test packet should be used for calculating the performance measures. Hence, “Pulse Search” needs to be enabled to capture the test packet and to position the “Result Length” window within the test packet. Only the symbols within the “Result Length” window will be used for calculation.

6. Go to “**MeasSetup > Digital Demod Properties > Search tab**” and enable “**Pulse Search**”.

7. Change the “**Result Length**” to **320 symbols** (see “**MeasSetup > Digital Demod Properties > Format tab**”).

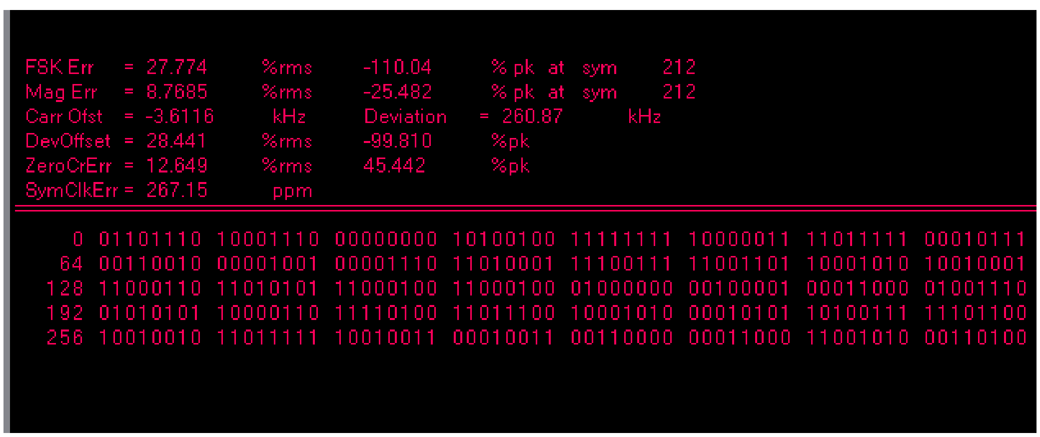
8. Add the “**Search Time**” trace and “**Time**” trace. The “**Search Time**” trace shows time-data before “**Pulse Search**” and digital demodulation. The “**Time**” trace shows the time record after “**Pulse Search**” which will be used for FSK demodulation. Set the **Y scale** of the **Time** to **Power**.

9. Start signal capturing. Repeat step 5 to restart the transmitter test if no data appears. Right click and "**Autoscale**" some of the traces for better view. A sample screenshot of the outputs is given below.



The upper left trace shows the 2-FSK constellation diagram. The demodulated symbols (red color dots) are very close to the ideal reference points. This indicates a high-quality modulated signal has been received.

10. The "**Syms/Errs**" trace shows the modulation performance measures. See the sample screenshot below.



11. Record the measured values for "**FSK Err**", "**Mag Err**", "**Carr Ofst**", "**Deviation**", "**DevOffset**", "**ZeroCrErr**", and "**SymClkErr**".

Exercise

Evaluate the performance of the demodulator when the bandwidth-bit period product is set to 0.3. Discuss your findings.

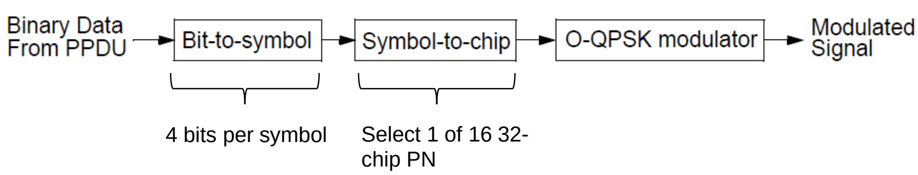
Task 2 - Zigbee (802.15.4) Modulation Analysis

NOTE

Before you begin the experiment, configure the Keysight U3810A IoT Development Kit as a “cape” on top of the BeagleBone CPU, and with the jumper configuration on Task 1 – *Bluetooth*® LE Modulation Analysis

WARNING Do not connect voltages higher 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.

The IEEE 802.15.4 offset-QPSK (O-QPSK) physical layer (PHY) is adopted by the 2.45 GHz Zigbee. In the PHY, the data payload from the upper layer is first appended with a PHY header (PHR) and a synchronization header (SHR) to create a PHY protocol data unit (PPDU). The PPDU is also known as a Zigbee frame. The PPDU is then converted to modulated signal using a combination of direct sequence spread spectrum and O-QPSK modulations. The reference modulator for O-QPSK PHY is shown in the figure below:



The data rate at the input of the modulator is 250 kbps. The following figure shows the format of a PPDU.

SHR		PHR		PHY payload
Preamble	SFD	Frame length	Reserved	PSDU
4 bytes	1 byte	7 bits	1 bit	Variable (specified in Frame Length)

All zeros

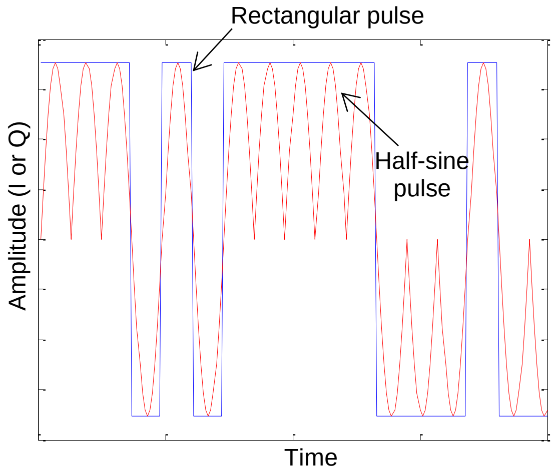
11100101 (Indicates end of SHR)

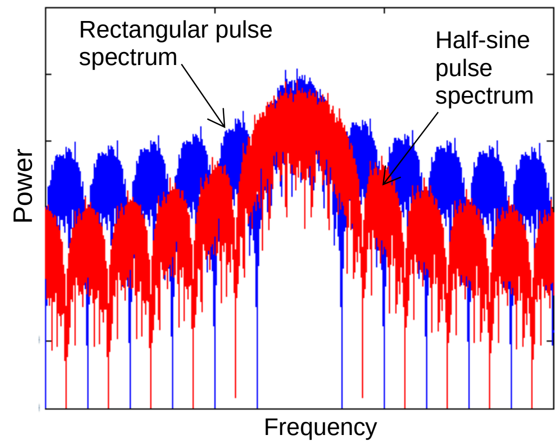
Max 127

Frame length values	Payload
0–4	Reserved
5	MPDU (Acknowledgment)
6–8	Reserved
9 to *aMaxPHYPacketSize*	MPDU

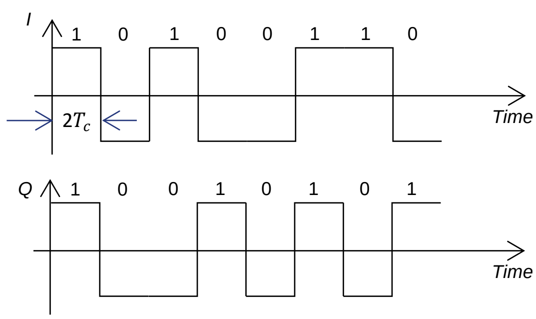
The binary bits in the PPDU are processed sequentially by the modulator. First, every four bits in the PPDU are grouped together to form data symbols. Each symbol is then mapped to a 32-chip pseudo-noise (PN) sequence. The symbol-to-chip mapping table is given in the Appendix A. The PN sequences for successive data symbols are concatenated, and the aggregate chip sequence is modulated onto the carrier using O-QPSK with half-sine pulse shaping.

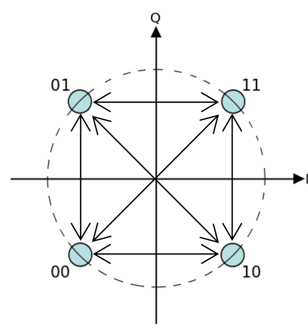
The purpose of pulse shaping is to limit the effective bandwidth of the transmitted signal. The figure below shows the simulated baseband waveforms and spectrums for signal with half-sine pulse shaping and without pulse shaping (rectangular pulse). The signal with half-sine pulse shaping clearly has lower side-lobes' magnitudes in its spectrum and hence smaller effective bandwidth.



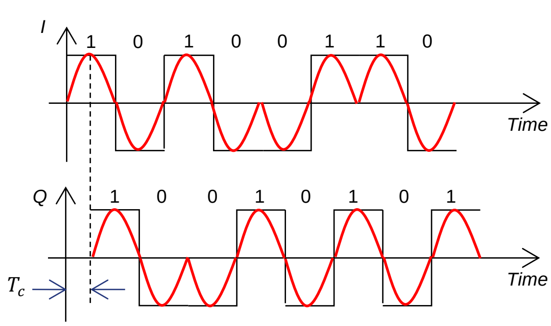


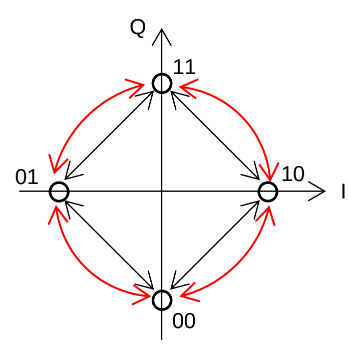
The canonical QPSK modulation has a shortcoming where phase change as much as 180° is possible at a time (such as, symbol 11 follows by 00). Sudden phase shift of 180° will make the signal envelope to go to zero momentarily. This causes large amplitude fluctuations in the signal which will reduce the efficiency of the power amplifier. The figure below shows the typical I and Q components of the QPSK signal and possible phase changes in the signal.



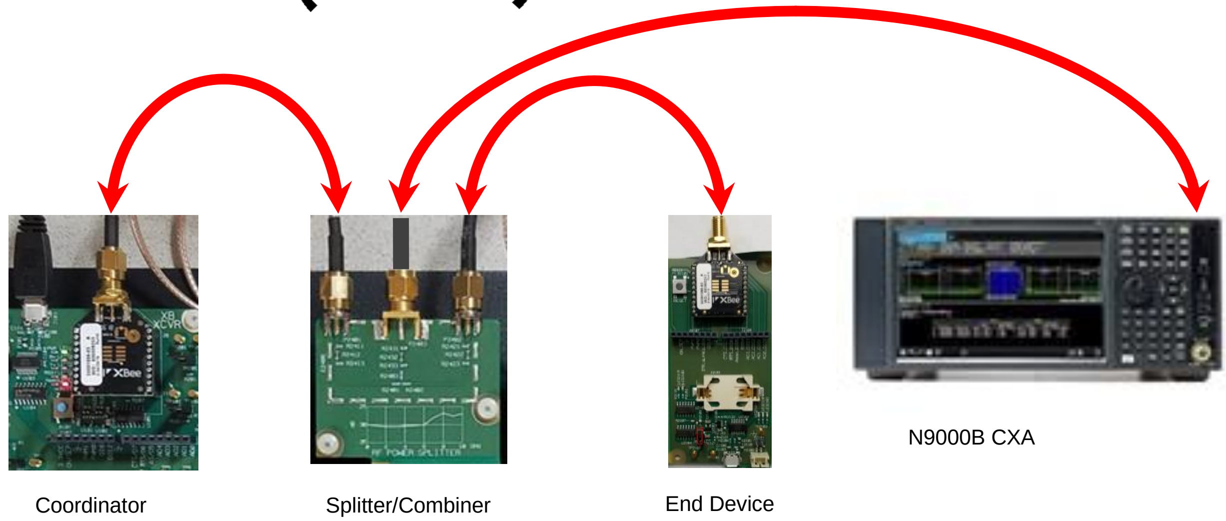


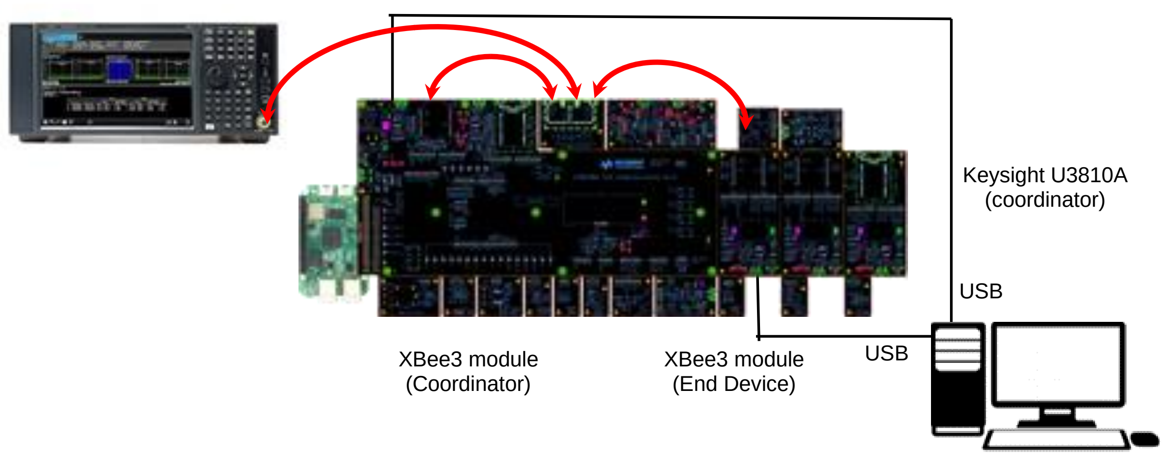
O-QPSK overcomes this problem by offsetting the Q component by half a symbol period. Now, I and Q components will not change at the same time. Hence, phase changes in the signal are now limited to up to 90° at a time. As shown in the figure below, zero crossing has been eliminated in the constellation diagram. Therefore, the O-QPSK signal has much lower amplitude fluctuations compared to QPSK. If half-sine pulse shaping is used (see red color waveform below), the transition paths in the constellation diagram have circular shape.





Task 2a - Initial Measurement Setup





NOTE

This task will use wired connections, not over-the-air, to obtain accurate power measurement at the device antenna connection. Also, the BeagleBone CPU will not be powered.

Task 2b - Configure the Zigbee Coordinator

First, set up the XB Transceiver on the U3810A assembly as the coordinator of a Zigbee PAN:

1. Connect the XB Transceiver on the U3810A assembly to the computer using a micro USB cable.
2. Start the XCTU software. Then, discover and add the Zigbee device on the U3810A.
3. Reset the firmware settings to their default values by clicking "**Default**". Then click "**Write**" to perform the write operation.
4. Configure the XB, XBee3 Zigbee module on the Keysight U3810A, according to the settings shown below. You may use the **Parameter** search field at the top right to locate each setting:

Parameter	Value	Description
CE	Formed Network [1]	Set the device as coordinator.
ZS	1	Set or read the Zigbee stack profile value. This must be the same on all devices that will join the same network.
NJ	FF	Set/Read the Node Join time. The value of NJ determines the time (in seconds) that the device will allow other devices to join to it. The coordinator will allow nodes to join without time limit.
NI	Coordinator	Define the node identifier, a human-friendly name for the module. The default NI value is a blank space. Make sure to delete the space when you change the value.
SC	800	Set the operating channel of the network. Refer to task for the channel selection. 800 = Channel 22 (Hex: 0x16)

5. To apply the changes, click **Write** and click **Read** to update the readings (You may need to click Read *twice* for the changes to be effective).

NOTE

Take note that the **Operating PAN ID (OP)** and **Operating Channel (CH)** are now updated. Record the new **OP** and **CH** values. Note that **MY=0** which is the 16-bit address of the coordinator and **AI=0** which indicates a successful startup.

Parameter	Value	Value (example)
Operating PAN ID (OP)		2020
Operating Channel (CH)		22

Task 2c - Configure the Zigbee End Device

Now, set up the XB1 or XB2 Transceiver as an End Device of your Zigbee PAN.

1. Connect the XB1 or XB2 Transceiver to the computer using a Micro-USB cable. Repeat the procedure above to add the XB Transceiver on the U3810 assembly into the XCTU. This module needs to be an End Device, so do not enable the CE setting.

2. Configure the XB1 (or XB2) to the settings shown below. You may use the **Parameter** search field at the top right to locate each setting.

Parameter	Value	Description
CE	Joined Network [0]	Sets the device as router or end device.
ID	2020	Defines the network that a radio will attach to. Refer to the Zigbee Coordinator's ID. This must be the same for all radios in your network.
ZS	1	Set or read the Zigbee stack profile value. This must be the same on all devices that will join the same network.
NI	End Device	Defines the node identifier, a human-friendly name for the module. The default NI value is a blank space. Make sure to delete the space when you change the value.
SC	800	Set the operating channel of the network. Refer to task for the channel selection. 800 = Channel 22 (Hex: 0x16)

3. Make the changes effective by clicking **Write** then clicking **Read** to update the readings (You may need to click Read twice for the changes to be effective).

NOTE

The **OP** and **CH** are now updated (same as the **OP** and **CH** of the coordinator, respectively). Also, take note that **MY** is updated with the 16-bit address of the end device and **AI=0** which indicates a successful join.

NOTE

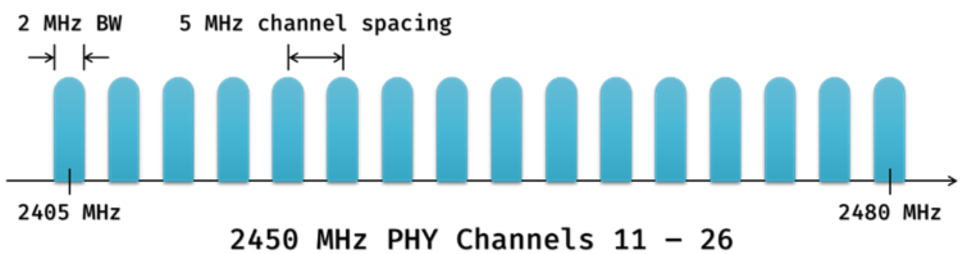
If you have not already saved these XBee3 configurations as Profiles, you may save time and assure accurate configurations by copying and extracting the files from the lab code directory in the BeagleBone to your computer. You may then use the **M2_L6_T2_COORDINATOR.xpro** and **M2_L6_T2_EndDevice.xpro** profiles for this Task.

Go to "**Profile** > **Create configuration profile**" and select the profile. For the gateway (U3810A XB) you will need to change PAN ID (ID) to match the OP sensor node (XB1 or XB2) and re-save the gateway profile you have customized to your network.

The **xpro** file saved by XCTU is a zip format file. You can use zip software to unzip the file to examine the contents. Do not use XCTU versions older than 6.3.10 as configuration files for those versions were saved in a format (.XML), which is incompatible with later releases.

Read the XCTU change log (XCTU > Help > Change Log) for more details.

Zigbee transmission will now take place in the "**Operating Channel (CH)**". In XCTU, the channel numbers are hexadecimal numbers from 0x0B (11) to 0x1A (26). The channel numbering is according to the 802.15.4 standard.



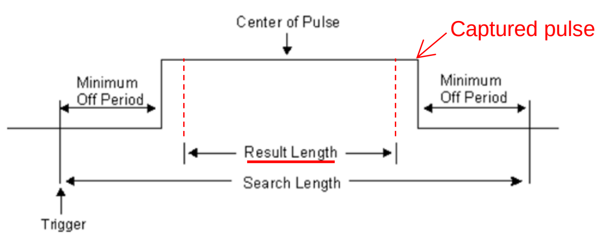
4. Now, the signal analyzer can be configured to capture the transmitted Zigbee frames for modulation analysis.
5. Attach a mouse and keyboard to the CXA vector signal analyzer. Turn on the signal analyzer. Then, launch the 89600 VSA software.
6. Preset the VSA by clicking on "**File > Preset > Setup**" to reset the settings to default values.
7. Go to measurement setup ("**MeasSetup > Frequency**"). Change the "**Centre**" frequency to the operating frequency of the Zigbee connection. For example, the operating channel used in this case is 22 (0x16) and the center frequency is **2460 MHz**.
8. Go to "**MeasSetup > Measurement Type > General Purpose > Digital Demod**" to set the VSA to demodulate digitally modulated signals which include O-QPSK.

9. The easiest way to demodulate the Zigbee signal is by using the standard preset supported by the VSA. Go to "**MeasSetup > Digital Demod Properties > Preset to Standard…**" and select "**Wireless Networking > Zigbee 2450 MHz**". The default parameter settings for this standard are given in the table below (the key settings are highlighted in red circles):

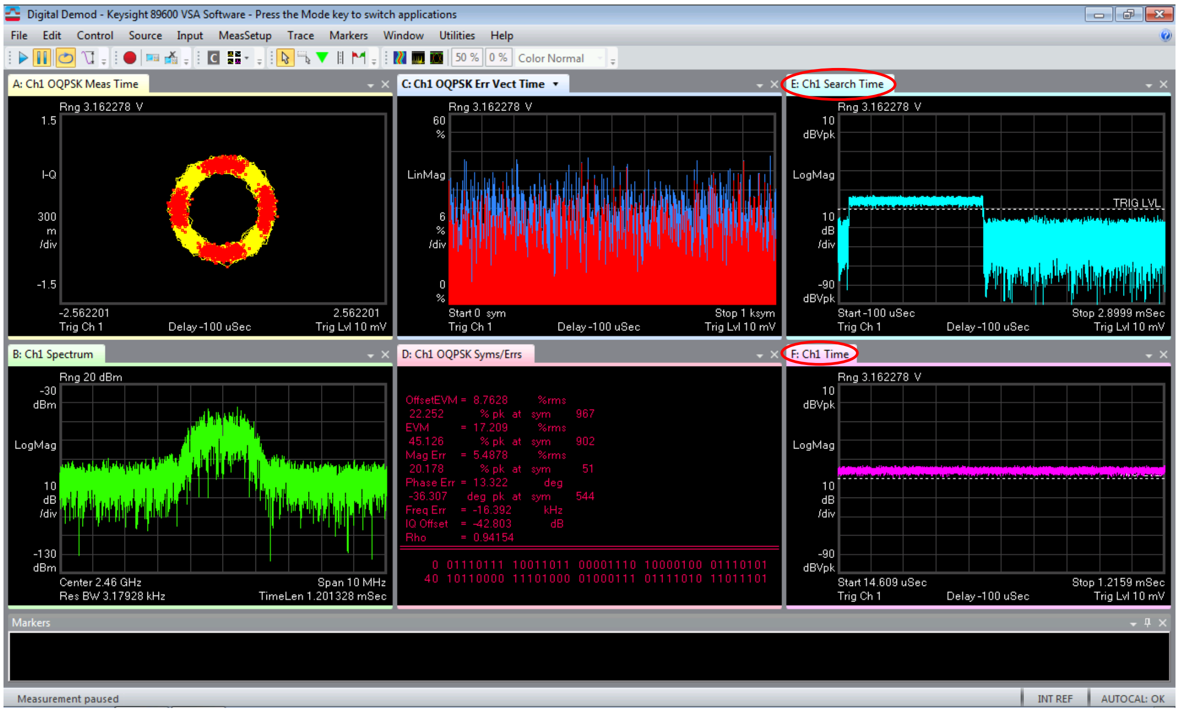
Preset-To-Standard Format

Parameter	ZigBee 868 MHz	ZigBee 915 MHz	ZigBee 2450 MHz
Demod Format	BPSK	BPSK	O-QPSK
Frequency Span	1 MHz	2 MHz	10 MHz
Symbol Rate	300 kHz	600 kHz	1 MHz
Filtering			
- Measured	Cleared	Cleared	Cleared
- Reference	raised cosine	raised cosine	Half Sine
Alpha / BT	1	1	---
Points / Symbol	4	4	10
Result Length	1000 symbols	1000 symbols	1001 symbols
Search Length	10 ms	5 ms	3 ms
Pulse Search	Selected	Selected	Selected
Constellation Sync Search	Cleared	Cleared	Cleared
Normalize IQ Traces	Selected	Selected	Selected
Clock Adjust	0.0	0.0	0.0

The "Frequency Span" used is 10 MHz which is about 5 times the bandwidth of the signal. The "Symbol Rate" (1 MHz) is the rate of the modulated signal (see Exercise 2.1). "Pulse Search" is enabled to demodulate pulsed (burst) transmissions. The demodulator searches within the defined "Search Length" to locate the first complete pulse, which can occur anywhere in the "Search Length". In the standard settings, the "Constellation Sync Search" is disabled. The "Result Length" (or the signal used for demodulation) is positioned around the center of the pulse, as shown below:



10. Go to “**Input > Trigger…**”. Change the “**Style**” to “**IF Mag**” and set the “**Level**” above the noise floor (such as, -25 dBm). The trigger level is shown in the “Search Time” trace. This will enable the VSA to capture any Zigbee-burst transmitted within the frequency span with a magnitude above the trigger level. Set the “**Delay**” to **-100 μs** so that the starting part of the captured pulse is visible.
11. You may need to wait up to 40 seconds to trigger on a received packet as shown below.
12. Several measurement traces will now appear on the screen. To display all the traces, go to “**Window > Trace Layout**” and select **2 x 3 layout**. Right click and “**Autoscale**” the windows as necessary for the best views. A sample screenshot is given below.



13. In addition to the default traces, it is often helpful to view the **Search Time** and **Time** trace data when setting up or troubleshooting pulsed measurements (see figure above). Click on **E** and **F** tabs and pull-down **Channel 1**, then select **Search Time** and **Time**. The **Search Time** trace data shows data before applying **Pulse Search** (under **MeasSetup > Digital Demod Properties… Search tab**); the “Time” trace data shows data after applying **Pulse Search**.

NOTE

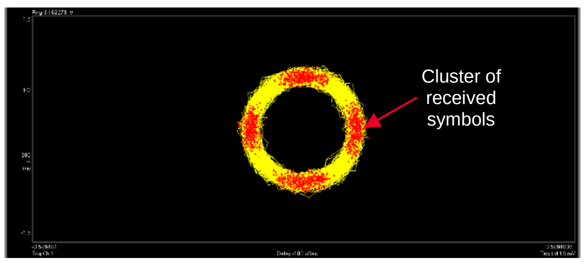
Copy the M2_L6_T2_SETUP.setx from the BeagleBone LabCode/M2-L6 folder to your computer using WinSCP. Copy it to your CXA using USB and use the setup file for this task. Go to “**File > Recall > Recall Setup…”** and select the setup file. Change the center frequency to the operating frequency of your Zigbee channel.

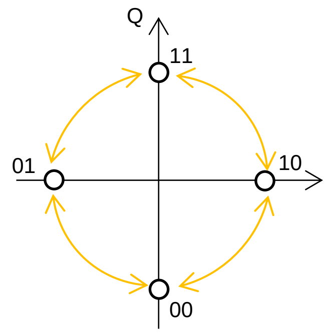
Task 2d - Modulation Accuracy

The modulation accuracy of a transmitter is reflected in several performance measures such as error vector magnitude (EVM), carrier frequency error, IQ origin offset, quadrature skew error and IQ gain imbalance. The 802.15.4 O-QPSK PHY requires that the EVM is less than 35% when measured for 1000 chips (or symbols) and the transmit center frequency tolerance shall be less than ±40 ppm.

Procedure

1. The most intuitive way of checking the modulation accuracy is by looking at the constellation diagram. The constellation diagram of the capture Zigbee symbols is shown in one of the default traces. A sample screenshot is given below:

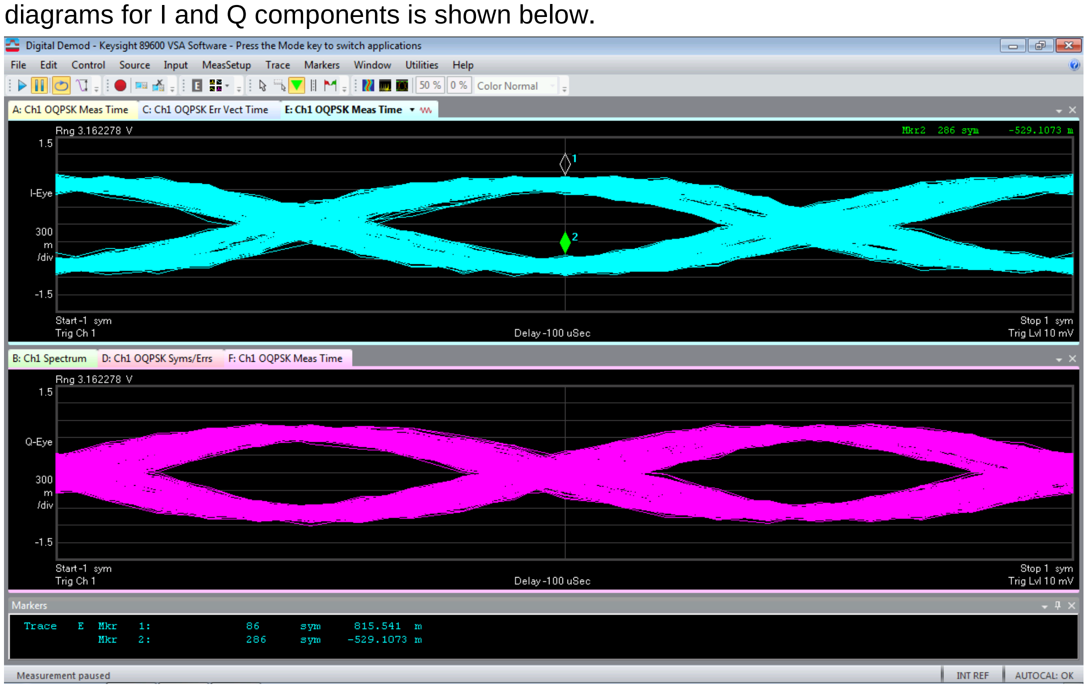




The received symbols are plotted as red color dots on the constellation diagram. The more concentrated the clusters of the received symbols around the ideal state, the better the modulation accuracy. The constellation diagram above also shows that when a Half-Sine reference filter is selected, the constellation changes from a square to a circle and the ideal state circles are moved to the I and Q axes.

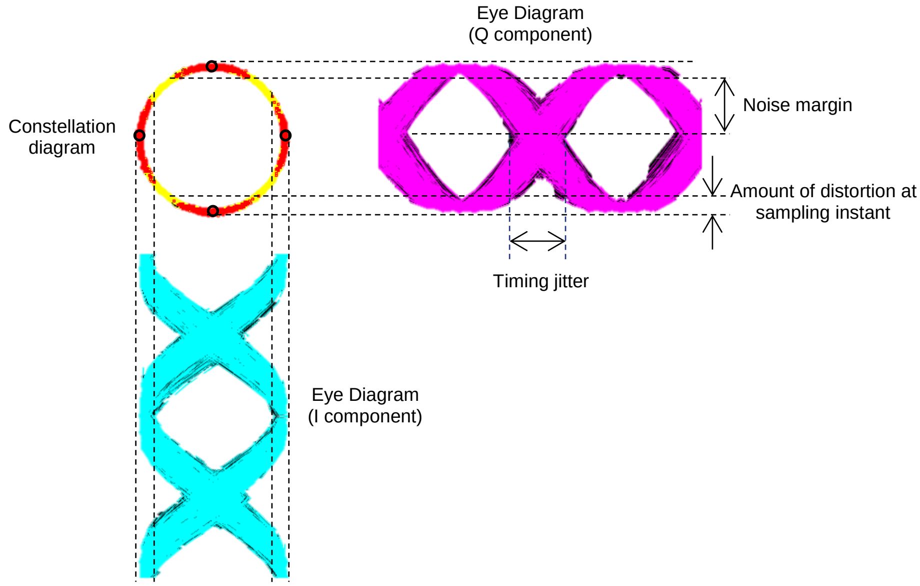
2. Besides constellation diagram, the eye diagram is another useful tool for determining the signal quality. Go to **Trace** and click on the “**+**” sign to add a new trace. Then, go to **Trace > Data > Channel 1: > IQ Meas Time**. Change the format by selecting **Trace > Format** and setting to **I-Eye**.

3. Repeat the step above to create the eye diagram for the Q component. A sample screenshot for the eye diagrams for I and Q components is shown below.



The eye diagram is obtained by overlapping many symbol periods.

4. The relationship between the constellation diagram and the eye diagrams is illustrated in the figure below.

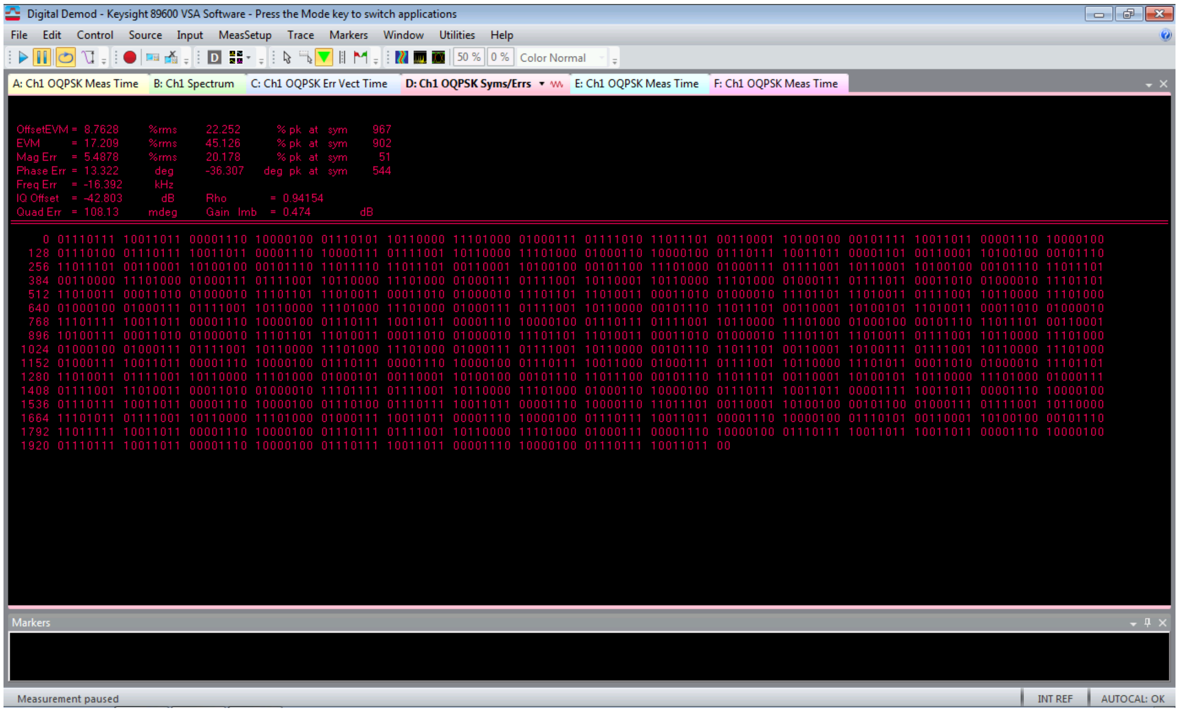


Use the markers to measure the noise margin, distortion at sampling instant, and time jitter.

5. The symbol table shows the error summary data and the symbol data (binary bits) for each symbol. A brief description for the results shown in the table is given below.

Parameter	Description
Offset EVM	Offset EVM is computed at one point per symbol, by combining the I value from the beginning of each symbol and the Q value from the middle of each symbol into a single complex value for EVM computations
EVM	For Offset QPSK, when the Half Sine Filter is selected, the OQPSK reference constellation points fall on a circle with a magnitude of sqrt(2)/2, but the EVM is still expressed as a percentage of the magnitude of a QPSK symbol point (magnitude = 1).
Mag Err	RMS-average of the IQ magnitude error over all symbol times and expressed as a percentage of the EVM Normalization Reference.
Phase Err	Phase difference between the I/Q reference signal and the I/Q measured signal, averaged over all symbol points
Freq Err	Carrier's frequency error relative to the VSA's center frequency
IQ Offset	Magnitude of the carrier feedthrough signal. When there is no carrier feedthrough, IQ offset is zero (-infinity dB).
Quad Err	Orthogonal error between the I and Q signals. Ideally, I and Q should be orthogonal (90 degrees apart).
Rho	Rho is computed by comparing the normalized correlated power between the measured signal and the reference signal and is designated as the waveform quality factor. The maximum value of Rho is 1.0 (which means the measured signal and reference signal are 100% identical).
Gain Imb	Ratio of gain of the I signal and the gain of the Q signal. Ideally, the ratio is one (or 0 dB).

A sample screenshot of the **Syms/Errs** table is shown below.



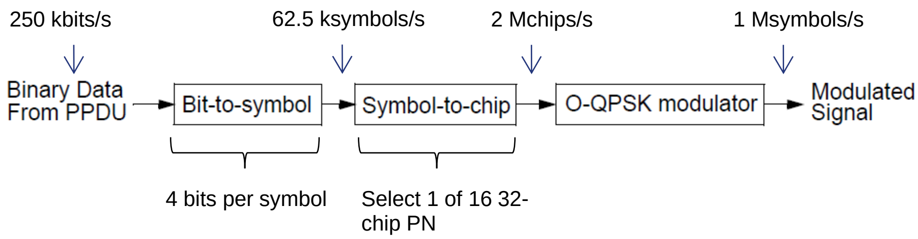
6. Record the measured data for "OffsetEVM" and "Freq Err" and compare them to the requirements of the 802.15.4 O-QPSK PHY standard.

Offset EVM: 8.7628 %rms

Freq Err: 17.209 %rms

Exercise

a. Show that the chip rate of the 802.15.4 O-QPSK PHY is 2 Mchips/s and the symbol rate of the O-QPSK output is 1 Msymbols/s.

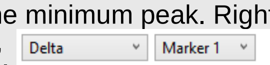


b. O-QPSK addresses the high-power fluctuations problem of the QPSK. In this exercise, you are required to measure and compare the peak power fluctuation for the O-QPSK and QPSK signals. Use the “Time” trace obtained in Task 2c above and the markers to measure the peak power fluctuation for O-QPSK. For QPSK, use the recorded signal in the VSA (go to “File>Recall>Recall Demo”, open “QPSK” folder, then select “QPSK”). Use the “Time” trace and markers to measure the peak power fluctuation for QPSK.

Compare and discuss your findings.

NOTE

Click the **Window** tab, select **Trace Layout > Single**. Then click the **F: Ch1 Time** trace to select the time trace. Change the Y scale of **Time** trace to power (dBm), select **Trace** tab, **Y Scale…** then change the Y unit to **Power**. Right-click on the **Time** trace and click “**+**” 

to add marker. Click the **Marker** tab, select **Search > Peak** to search for the maximum peak. Then add another marker, click the **Marker** tab, select **Search > Peak Minimum** to search for the minimum peak. Right-click the **Time** trace and change the marker from “**Normal**” to “**Delta**”, “**Marker 1**”. 

Introduction to Coexistence Testing

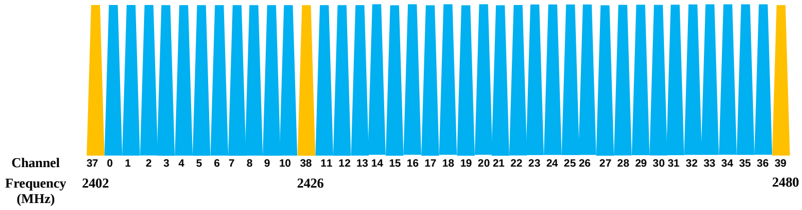
Wireless Coexistence is the ability of a device to perform its intended function in the presence of other types of RF signals. In shared spectrum (the ISM Bands) there may be many different signal modulations and protocols active simultaneously and a device may experience interference and degraded operation compared to operating in quiet wireless conditions. Several signal and device characteristics may determine how well a wireless device tolerates interference, including signal frequency, signal strength and signal timing (rate and duration of signal).

Wireless Coexistence Test places a device in a mixed signal environment and measures Key Performance Indicators (KPIs) in interference conditions. KPIs may be internal (packet error rate, for example) or external (effect of interference on device intended external functions such as gross data throughput, impairment of audio or video signals, or dropped connections).

In this lab sheet, you will set up an experiment to measure an external KPI (throughput) of a Zigbee device in the presence of a *Bluetooth*® LE modulated signal (as interferer), using the U3810A and vector signal analyzer. The U3810A uses a BeagleBone CPU module. It supports *Bluetooth*® LE which complies with *Bluetooth*® Core Specification Version 4.0 through the WL18SBMOD chip (a *Bluetooth*® /WLAN chip). The table below gives a brief overview of the *Bluetooth*® LE physical layer (PHY) specifications [1].

Technical Specification	***Bluetooth*® LE**
Radio frequency	2400 to 2483.5 MHz
Modulation technique	Frequency hopping Standard hop rate = 1600 hops/s (3200 hops/s in page mode)
Bandwidth bit period product (BT)	0.5
Modulation scheme/Index	GFSK/0.5
Number of channels	40
Range	~10 – 50 m
Data rate	1 Mbps
Maximum Output Power	10 dBm

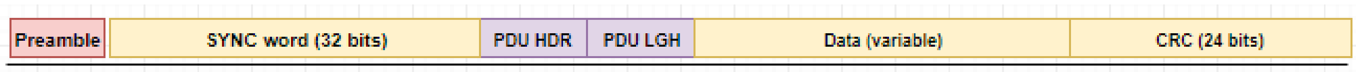
There are 40 channels for *Bluetooth*® *LE* transmission. The center frequencies for every channel are shown in the figure below. The first channel is 37 (2402 MHz) and the last channel is 39 (2480 MHz).



Channels 37 (2402 MHz), 38 (2426 MHz), and 39 (2480 MHz) are the advertising channels (for discovering devices, initiating connection and broadcasting data). The rest of the channels are used for data transmission (after a connection is established). Before a connection is established, an active *Bluetooth*® LE slave (or advertiser) will continuously send out advertising packets in all three advertising channels. After a connection is set up, data packets are transmitted in the data channels according to a pseudo-random frequency hopping pattern.

The U3810A supports *Bluetooth*® LE direct test mode via HCI communication. A special test packet is used for this purpose. The *Bluetooth*® LE test packet includes an 8-bit preamble, 32-bit sync word, 8-bit PDU header, and 8-bit PDU length, a payload that can vary between 0 and 296 bits and a 24-bit CRC. The test packets do not have an access address field.

The figure below shows the *Bluetooth*® LE test packet structure.

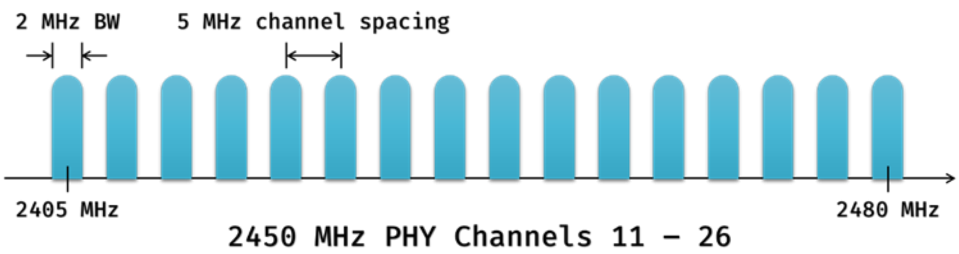


The *Bluetooth*® LE test packet 4-bit Payload Type defines the payload content. Payload Type is part of the PDU header. The table below lists the payloads for each payload type bit sequence.

Payload Type	Payload Description
0x00	PRBS9
0x01	Repeating ‘11110000’ sequence
0x02	Repeating ‘10101010’ sequence
0x03	PRBS15
0x04	Repeating ‘11111111’ sequence
0x05	Repeating ‘00000000’ sequence
0x06	Repeating ‘00001111’ sequence
0x07	Repeating ‘01010101’ sequence

The *Bluetooth*® test signal will be used as the unwanted/interferer signal to a Zigbee data link. The Zigbee link uses the Digi XBee modules controlled by the XCTU development API. You will configure the XBee modules into a Zigbee network and begin a data throughput test built into the XCTU program on your PC.

Once XCTU Throughput test is running, we create interference on the same frequency as the Zigbee channel (2.460 GHz). Note that while channel numbering and spacing are different between Zigbee and *Bluetooth*®, they do use the same radio frequencies for certain channels. Compare the *Bluetooth*® frequency/channel chart above to the Zigbee frequency/channel chart below.



For the *Bluetooth*® and the Zigbee standards use different modulation types and protocols, they cannot cooperatively use the radio channel and interference will result. This will cause the throughput test to show a decrease in throughput on the Zigbee link.

Three signal characteristics affect how signals interfere: Frequency, Power and Time. Since these transmissions are brief (*Bluetooth*® test packet is about 405 µSec long) they must be frequent to cause significant interference. The *Bluetooth*® LE transmitter test signal in BeagleBone has 405 µSec ON and 220 µSec OFF, yielding a total period of 625 uSec, and a duty cycle of about 65%. This means that 65% of the time, a Zigbee transmission may encounter the interfering signal, and experience transmission errors and lost packets. Also note that the Zigbee throughput test uses longer duration data packets, increasing probability of interference. The Zigbee radio protocol and encoding will allow it to recover packets with some lost parts.

Sending commands to the BeagleBone:

Upon power-up the BeagleBone will ask for login and password. The default login is "debian" and the default password is "temppwd". Certain commands to the BeagleBone require "supervisor" privilege. To use these commands, you must prefix the commands with "sudo" which means "supervisor do" which permits execution of that privileged function. The first time using the supervisor privilege you will be required to give the supervisor password (by default it is "temppwd"). Periodically, you may be asked again to give the password when using "sudo". This is normal operation.

Utility programs/commands used in this lab:

`rfkill` is a command line tool that provides an interface to enable or disable RF chips in the BeagleBone system. The chips can be activated using the following command: **`sudo rfkill unblock bluetooth`** Use the command **`sudo rfkill -h`** to see a list of rfkill commands.

HCI is the *Bluetooth*® Host Controller Interface that connects a host to a *Bluetooth*® chip. HCI commands enable vendor-specific actions in a RF chip to be controlled. This lab will use several HCI functions using the utilities **`hciconfig`** and **`hcitool`**.

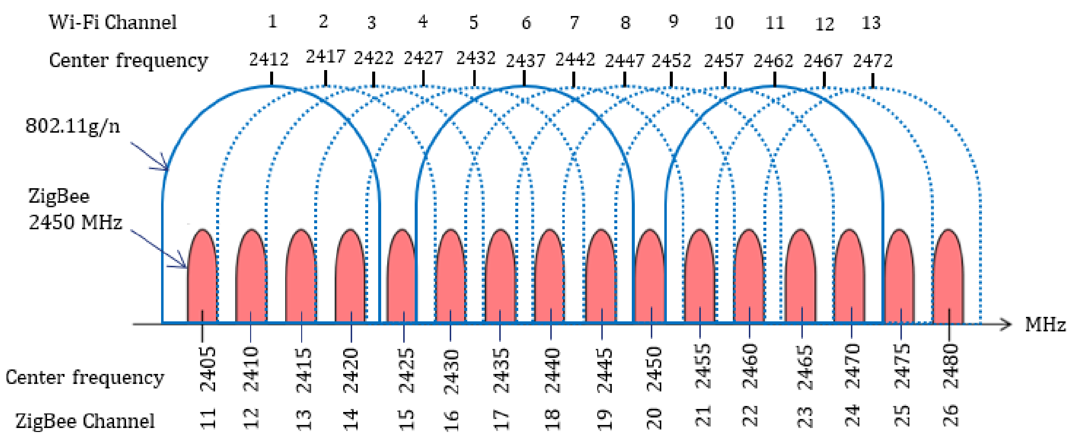
For example: **`hcitool cmd 0x06 0x0003`** sends a command code (hexadecimal) 6 with a parameter (hexadecimal) 3.

The command **`sudo hcitool -h`** in the Putty window will display commands available to the hcitool utility.

The command **`sudo hciconfig -h`** will display commands available to the hciconfig utility.

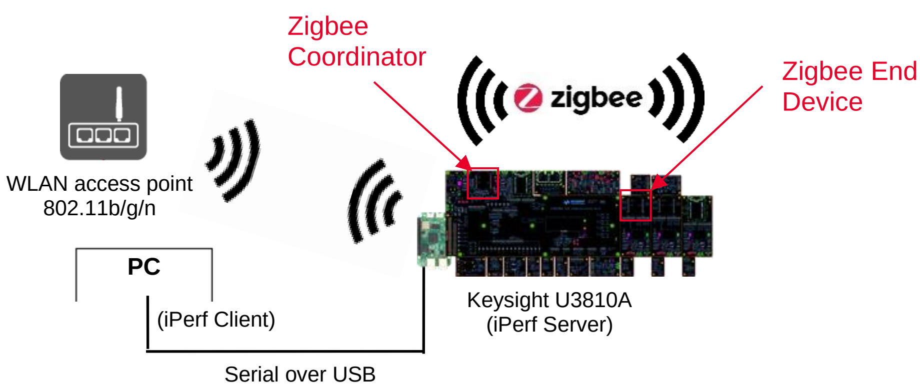
Task 3 - Coexistence of Zigbee and WLAN (Wireless Local Area Network)

Coexistence of Zigbee (2450 MHz) and WLAN (IEEE 802.11g/n) in the 2.4 GHz ISM band may cause interference problem. The channels and their corresponding center frequencies for 802.11g/n and Zigbee are shown in the figure below:



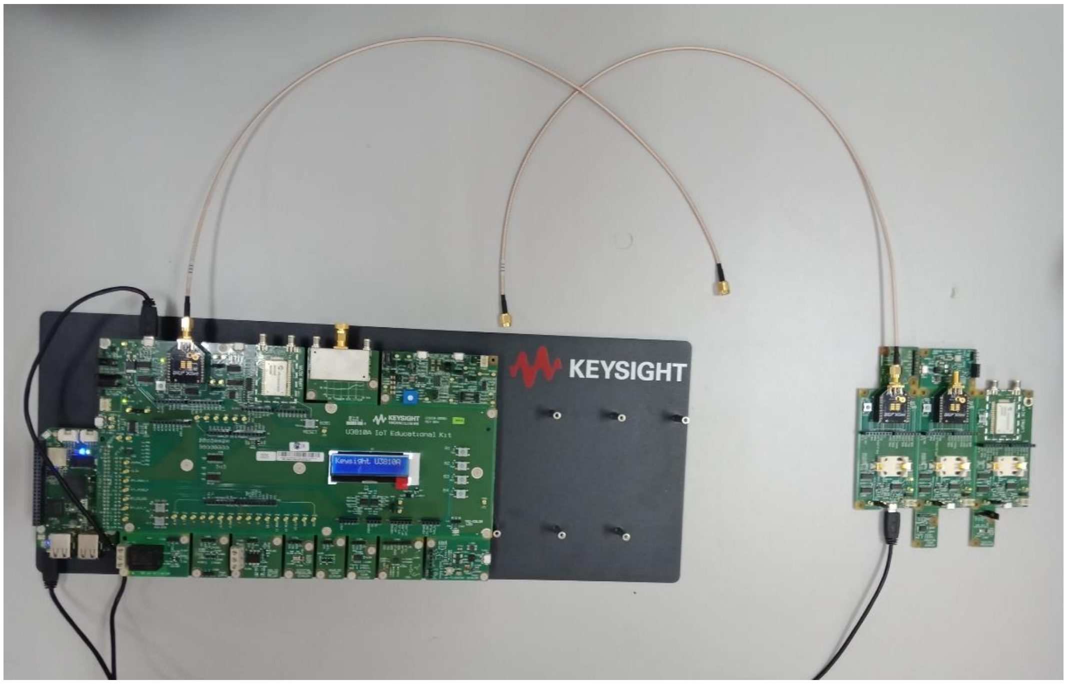
One WLAN channel may overlap with several Zigbee channels at the same time. Since Zigbee signals normally have lower power compared to WLAN signals, the impact of WLAN signals on Zigbee signals can be very severe.

In this experiment, the impact of WLAN interference on the Zigbee packet-error-rate (PER) will be evaluated. A possible setup for this experiment is shown in the figure below:



Using the U3810A platform, a weak RF link may be established between the Zigbee modules by arranging the coaxial cables near each other without making a wired link. This may yield a link with about -80 to -90 dBm signal strength. This arrangement (as seen in the image below) results in signal levels in the weak signal range and makes interference more aparrent in testing. The coaxial cables “leak” signal capacitvely though not actually connected.

A sample screenshot of the U3810A setup for the Coexistence of Zigbee and WLAN test is shown below.

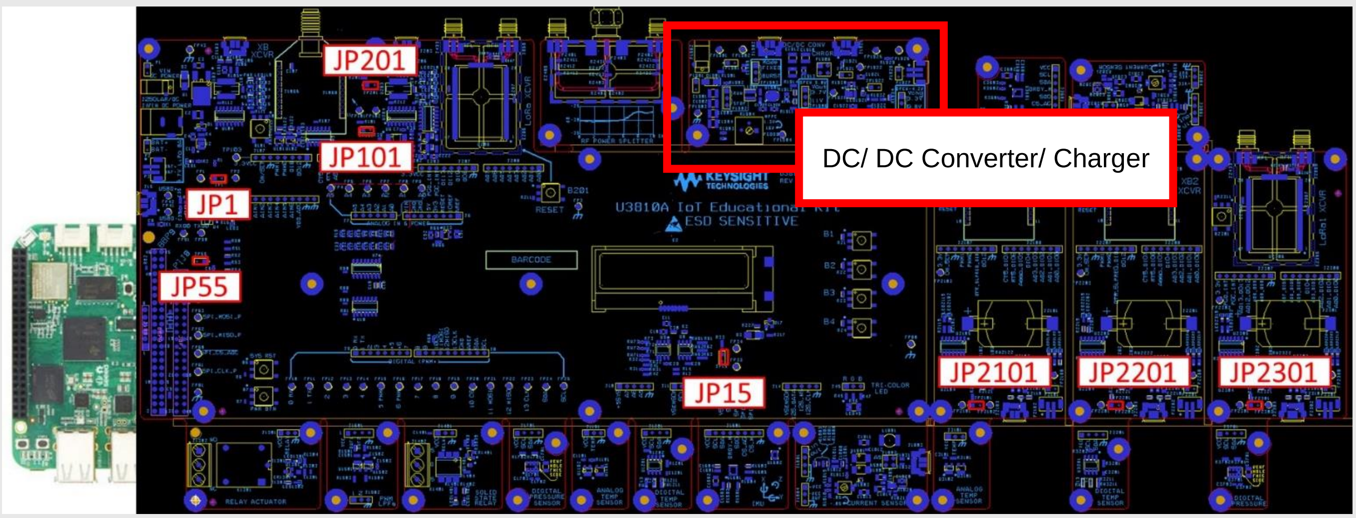


Task 3a - Initial Experiment Setup

NOTE

Before you begin the experiment, configure the Keysight U3810A IoT Development Kit as a “cape” on top of the BeagleBone CPU, and assure the same jumper configuration as in the previous task.

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

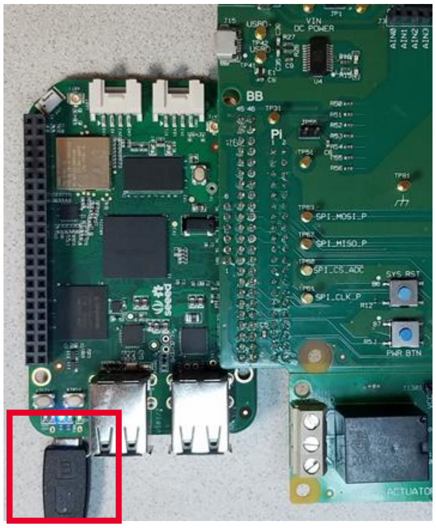


** DC/ DC Converter/ Charger Board jumper positions are not relevant to this lab.

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

WARNING Do not connect voltages higher 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. Connect the BeagleBone CPU to your computer using a micro-USB cable.



2. Login to BeagleBone via SSH Communication in PuTTY terminal with the following details. Refer to Set Up SSH connection for more information.

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

3. Set up a WLAN connection between the Keysight U3810A and a WLAN access point. Make sure the WLAN access point is operating with a known SSID and password. You may refer to Appendix D - Configure BeagleBone to connect to WLAN network to setup the WLAN connection.

4. When the connection is established, type **`ifconfig wlan0`** to check the IP address of the WLAN interface ("inet" or "inet addr"). In this case, the IP address is 192.168.1.3. Your setup will **most likely have different IP address**. Note this address as it will be needed later.

```
root@edison:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr [redacted]
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:703 errors:0 dropped:0 overruns:0 frame:0
          TX packets:633 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:59817 (58.4 KiB)  TX bytes:56314 (54.9 KiB)
```

5. Verify the settings of the WLAN connection using the command **`iwconfig`**. For this example, the center frequency of the WLAN connection is 2.447 GHz which is channel 8 (refer to WLAN channels figure above). The WLAN channel 8 overlaps with channels 18 to 21 of the Zigbee.

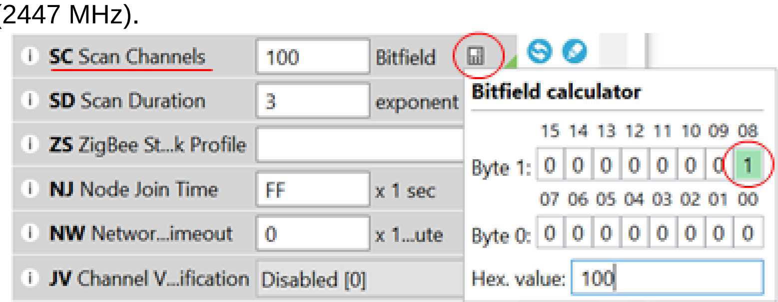
```
root@edison:~# iwconfig wlan0
wlan0     IEEE 802.11abgn  ESSID:[redacted]
          Mode:Managed  Frequency:2.447 GHz  Access Point: [redacted]
          Bit Rate=65 Mb/s   Tx-Power=31 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=70/70  Signal level=-22 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:26  Invalid misc:1   Missed beacon:0
```

Task 3b - Configure the Zigbee Coordinator

Configure the Zigbee connection between the Keysight U3810A and another Zigbee module. The Zigbee module on the Keysight U3810A will be configured as the Zigbee coordinator.

1. Connect the XB Transceiver on the U3810A assembly to the computer using a micro-USB cable.
2. Launch the XCTU software on the PC1. Then, **discover** and **add** the Zigbee module on the U3810A.
3. Reset the firmware settings to their default values by clicking "**Default**". Then click "**Write**" to perform the write operation.
4. Configure the XB, XBee3 Zigbee module on the Keysight U3810A, according to the settings shown below. You may use the **Parameter** search field at the top right to locate each setting.

Parameter	Value	Description
CE	Formed Network [1]	Sets the device as coordinator.
ZS	1	Set or read the Zigbee stack profile value. This must be the same on all devices that will join the same network.
NJ	FF	Set/read the Node Join time. The value of NJ determines the time (in seconds) that the device will allow other devices to join to it. The coordinator will allow nodes to join without time limit.
NI	Coordinator	Defines the node identifier, a human-friendly name for the module. The default NI value is a blank space. Make sure to delete the space when you change the value.
PL	0 – Minimum [0]	Transmitter output power
AP	API Mode Without Escapes [1]	Enable API
SC	100	Set the operating channel of the network. Use the **Bitfield calculator** of the **Scan Channels (SC) to** select a channel which overlaps with the WLAN channel setup previously. The Zigbee channels are labeled as follows: Bit 0 = Chan 11, …, Bit 15 = Chan 26. In this example, Bit 8 is selected which corresponds to Channel 19 (2445 MHz) of Zigbee. The center frequency of this selected channel is 2 MHz away from the center frequency of the operating WLAN channel (2447 MHz).

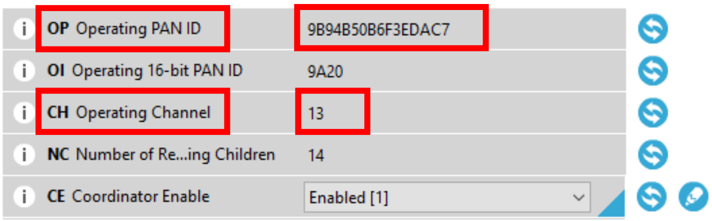


5. To apply the changes, click **Write** and click **Read** to update the readings (You may need to click Read *twice* for the changes to be effective).

> **NOTE**
>
> Take note that the **Operating PAN ID (OP)** and **Operating Channel (CH)** are now updated. Record the new **OP** and **CH** values. Note that **MY = 0** which is the 16-bit address of the coordinator and **AI = 0** which indicates a successful startup.

Parameter	Value (example)
Operating PAN ID (OP)	9894B50B6F3EDAC7
Operating Channel (CH)	19

A sample screenshot is given below:



In this case, the operating channel is 13 (in hexadecimal) or equivalent to Channel 19.

Task 3c - Configure the Zigbee End Device

1. Connect either Xbee3 XB1 or XB2 Zigbee Module to your computer and add this Zigbee device into the XCTU software. Reset the firmware settings to their default values by clicking **Default** as before. Then **Write** the settings to the XBee3.
2. Configure the XBee3 XB1 or XB2 Zigbee Module to the settings shown below. You may use the **Parameter** search field at the top right to locate each setting.

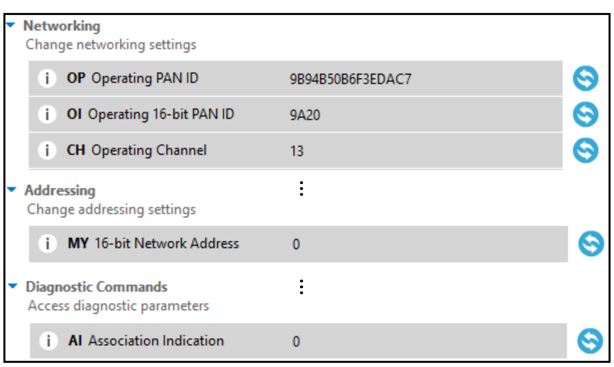
Parameter	Value	Description
CE	Joined Network [0]	Set the device as router or end device.
ID	9894B50B6F3EDAC7	Define the network that a radio will attach to. Refer to the Zigbee Coordinator's ID. This must be the same for all radios in your network.
ZS	1	Set or read the Zigbee stack profile value. This must be the same on all devices that will join the same network.
PL	0 – Minimum [0]	Transmitter output power
AP	API Mode Without Escapes [1]	Enable API
NI	End Device	Define the node identifier, a human-friendly name for the module. The default NI value is a blank space. Make sure to delete the space when you change the value.

3. Make the changes effective by clicking **Write** then clicking **Read** to update the readings (You may need to click Read twice for the changes to be effective).

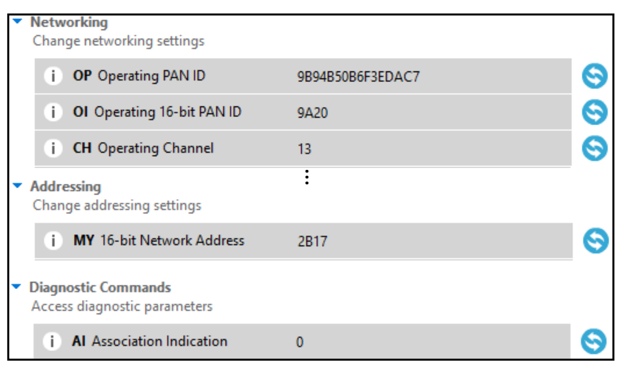
NOTE

The **OP** and **CH** are now updated (same as the **OP** and **CH** of the coordinator, respectively). Also take note that **MY** is updated with the 16-bit address of the end device and AI=0 which indicates a successful join.

A sample screenshot for Zigbee coordinator XCTU and Zigbee end device XCTU are shown below:



Zigbee Coordinator



Zigbee End Device

NOTE

If you have not already saved these XBee3 configurations as Profiles, you may save time and assure accurate configurations by copying and extracting the files from the lab code directory in the BeagleBone to your PC. You may then use the **M2_L6_T3_COORDINATOR.xpro** and **M2_L6_T3_EndDevice.xpro** profiles for this Task.

Go to "**Profile** > **Create configuration profile**" and select the profile. For the gateway (U3810A XB) you will need to change PAN ID (ID) to match the OP sensor node (XB1 or XB2) and re-save the gateway profile you have customized to your network.

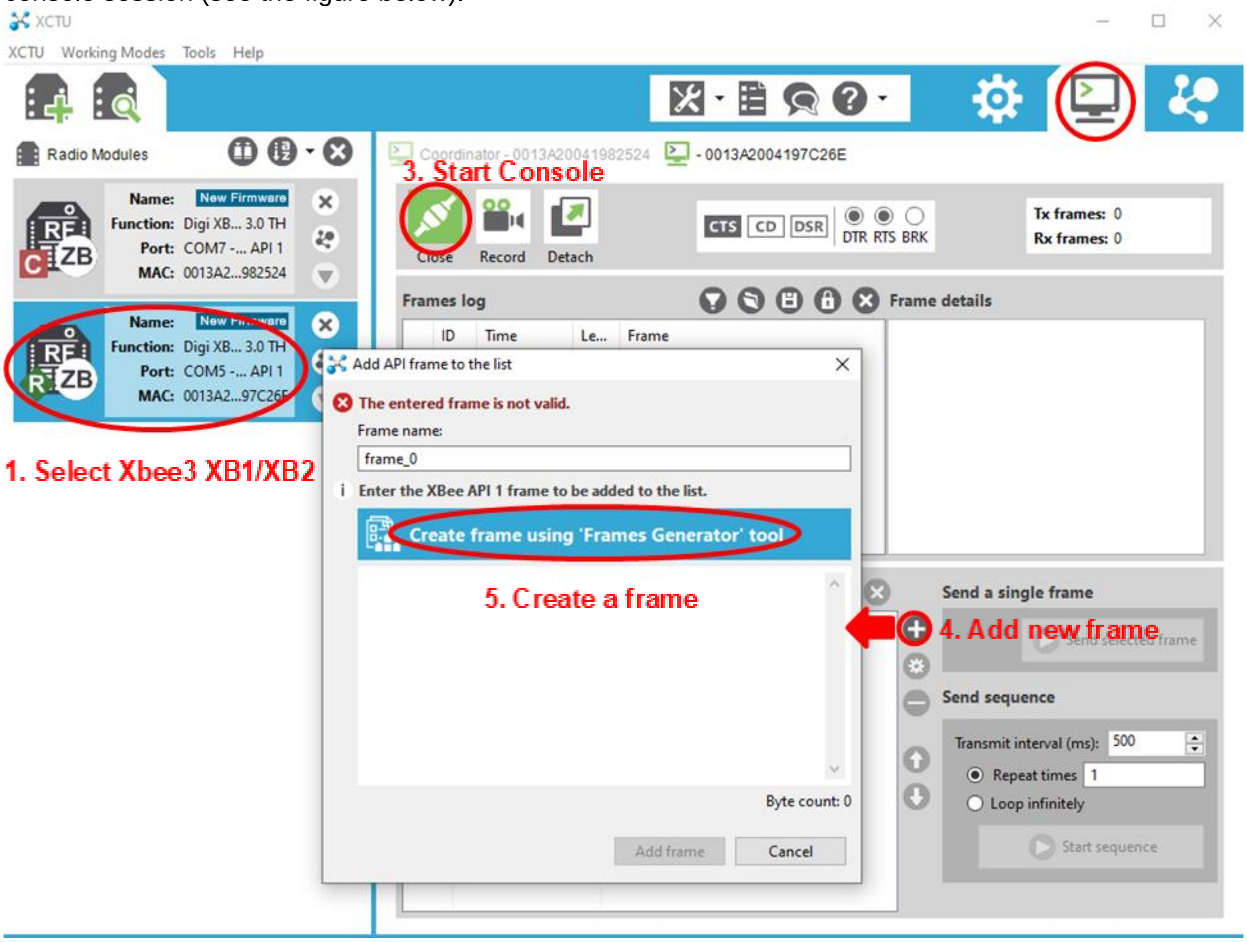
The **xpro** file saved by XCTU is a zip format file. You can use zip software to unzip the file to examine the contents. Do not use XCTU versions older than 6.3.10 as configuration files for those versions were saved in a format (.XML), which are incompatible with later releases.

Read the XCTU change log (XCTU > Help > Change Log) for more details.

4. To facilitate Zigbee packet analysis, the following key information (applicable only for this example) are recorded. You can discover, record and use your devices' different settings.

Parameter	Setting
Operating channel (CH)	13
Center frequency	2445 MHz
Operating 16-bit PAN ID (OI)	9A20
Coordinator MAC address	0013A200416736C2
Coordinator 16-bit network address (MY)	0000
End device MAC address	0013A200 4163096F
End device 16-bit network address (MY)	2B17

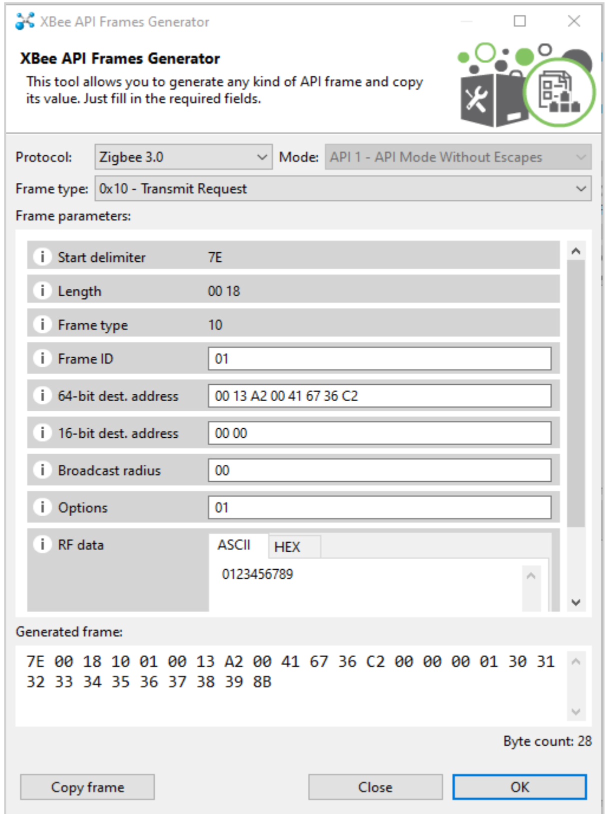
5. On the XCTU program, select the Zigbee End Device. Switch to console working mode and start the console session (see the figure below).



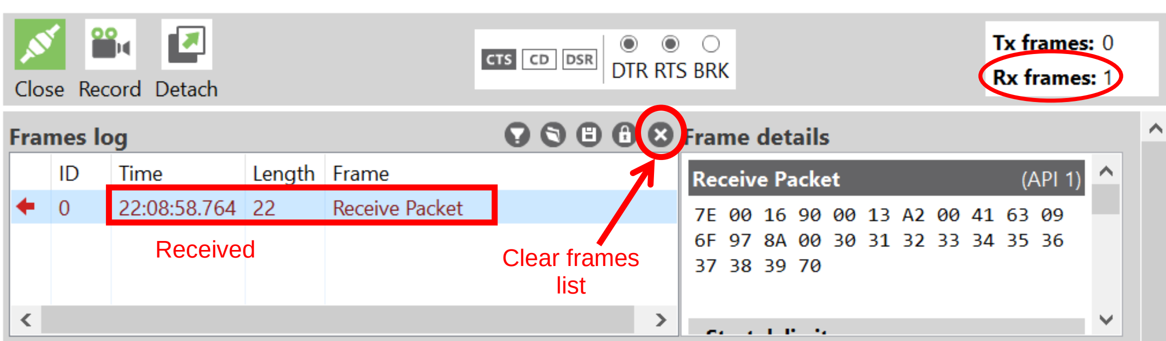
6. On the **XBee API Frames Generator** window, enter the following settings, using those you have recorded for your device, not the example settings shown here:

Parameter	Setting	Remarks
Protocol	Zigbee 3.0	-
Frame type	0x10 – Transmit Request	Unicast data transmission
64-bit dest. address	0013A200416736C2	Coordinator's MAC address
16-bit dest. address	0000	Coordinator's 16-bit network address
Broadcast radius	00	Max hops
Options	01	Disable acknowledgement
RF data	ASCII: 0123456789	Arbitrary 10-byte data

Sample screenshot of the **XBee API Frames Generator** window



7. Click **OK** to generate the XBee API frame then click **Add frame** to add the API frame to list.
8. Select the Zigbee Coordinator. Switch to console working mode and start the console session.
9. Then, send the selected frame from the Zigbee End Device to the Zigbee Coordinator and check whether it is correctly received or not. Send a single frame by clicking **Send selected frame**.
10. Go to the console of the Zigbee Coordinator. The received frame should be in the **Frames log**.
11. Click the Receive Packet to show the Frame details. A sample screenshot is given below:



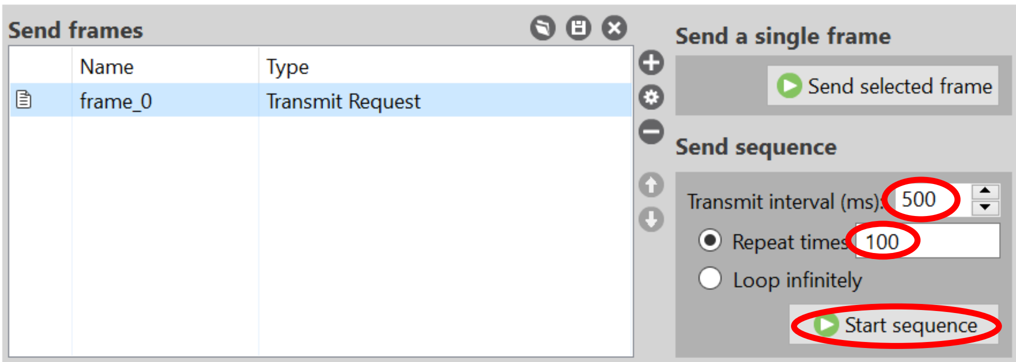
NOTE

The frame ID starts from 0.

Task 3d - PER Performance of Zigbee in the Presence of Idle WLAN Interference

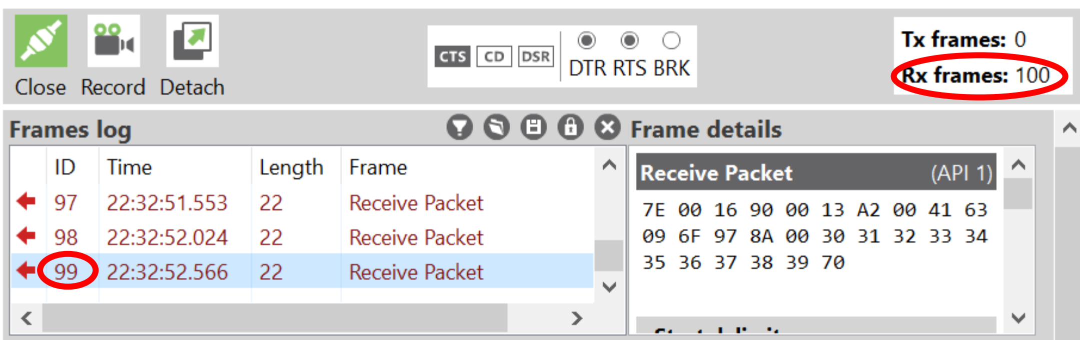
Perform the following procedure.

1. Clear the frames list in the **Frames log only** (not from the created frames) windows for both the coordinator and end device.
2. Navigate to the console of the End device. Set the **Transmit interval** and **Repeat times** to **500** and **100** respectively. A sample screenshot is shown below.



Click **Start sequence** to transmit 100 frames.

3. Go to the console of the Coordinator. The received frames are listed in the **Frames log**. A sample screenshot is given below.



As shown in example above, 100 frames have been received (the last frame ID is 99). The time log shows that the frames are received about every 500ms.

4. Since only frames received without errors are listed in the **Frames log**, the packet-error-rate (PER) in this case is zero when the WLAN interference is in idle state.
5. Repeat the procedure above several times to get more accurate result for the PER.

Task 3e - PER Performance of Zigbee in the Presence of Active WLAN Interference

Perform the following procedure.

1. In XCTU, clear the frames list in the **Frames log** windows for both the Zigbee coordinator and end device.
2. The following steps assume the iPerf software has been installed on both the BeagleBone module (on the Keysight U3810A) and PC.

   For this example, iPerf version 2.0.9 is used. This software is used for transmitting a near continuous stream of WLAN packets from the iPerf client to the server. This is to simulate an active WLAN interference for PER performance evaluation of the Zigbee transmission.
3. Navigate to the SSH window of the BeagleBone module and start the iPerf server using this command: **`debian@beaglebone:~$ iperf -s`** as shown below.

   The option **-s** will start the iPerf server. The Server is now awaiting data transmission from the Client.

```
debian@beaglebone:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
```

4. Open the command prompt in PC, go to the **iperf-2.0.9-win64** folder and start the iPerf client and WLAN transmission to the server (the BeagleBone address) using the following command.

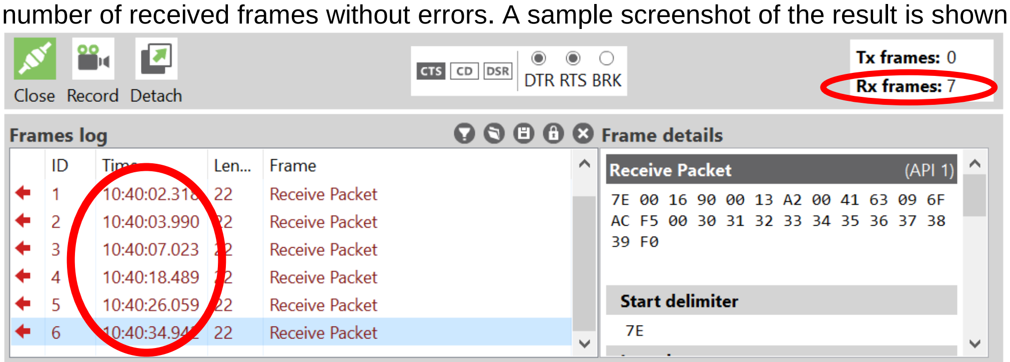
   `cd C:\iperf-2.0.9-win64`

   `iperf -c <server IP address> -t` **<transmit duration in s>**

   In this example, the iPerf server IP address is 192.168.0.146. Use the WLAN address you found in the setup procedure. A sample screenshot is shown below where the WLAN transmission will last for 120 s.

```
C:\Users\[redacted]>cd C:\iperf-2.0.9-win64

C:\iperf-2.0.9-win64>iperf -c 192.168.0.146 -t 120
------------------------------------------------------------
Client connecting to 192.168.0.146, TCP port 5001
TCP window size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.179 port 59005 connected with 192.168.0.146 port 5001
```

5. While the WLAN is transmitting, go to the console of the Zigbee end device. Start sending 100 frames to the coordinator. Make sure that the WLAN transmission is active throughout the duration of the Zigbee test.
6. When Zigbee frames transmission is done, go to the console of the Zigbee coordinator and check the number of received frames without errors. A sample screenshot of the result is shown below:



   As shown in the result above, only 7 frames are correctly received. The balance (93 frames) are either received with errors or lost. The time log shows varying time gaps between received frames. Based on this measurement result, the PER is a poor 0.93. It can be concluded that the performance of the Zigbee is significantly degraded by the active WLAN interference.
7. Repeat the procedure above several times to get more accurate result for the PER.

Exercise

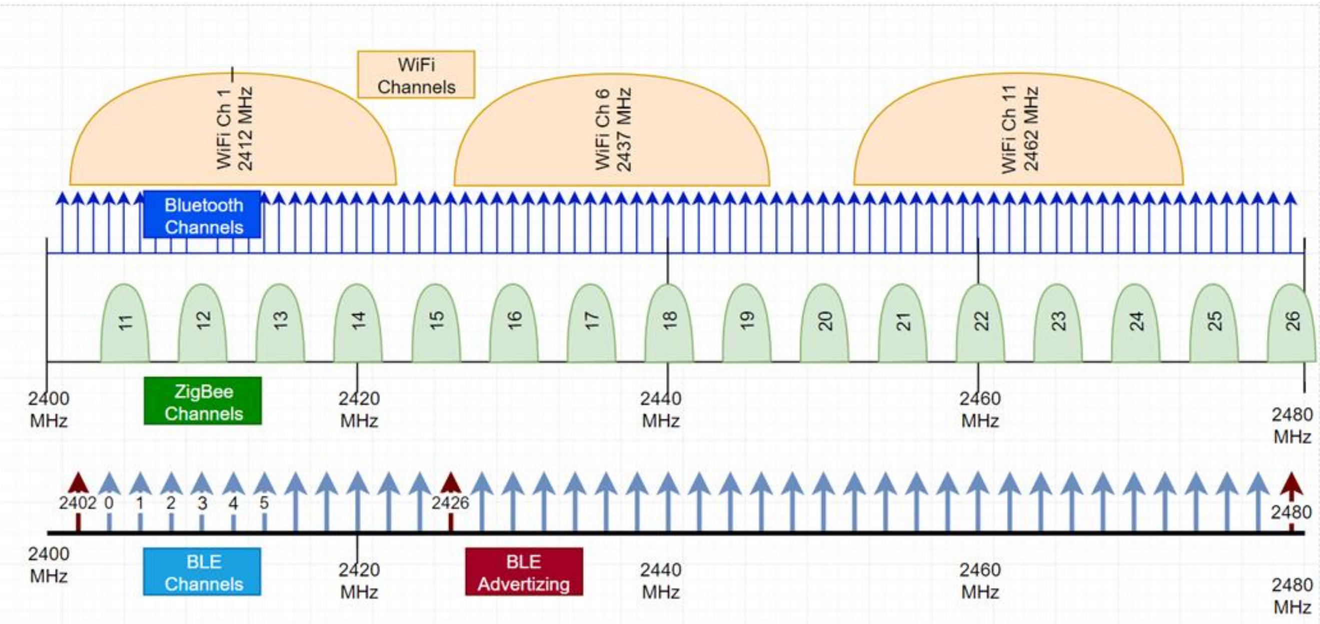
a. Repeat the procedure above to evaluate the PER performance of Zigbee in the presence of active WLAN interference for different frequency offsets between the operating frequencies of the Zigbee and WLAN connections. Note that each time you change the coordinator's network frequency, the Extended PAN ID will be changed. You must copy this new value to the router XBee and write it to the radio module.

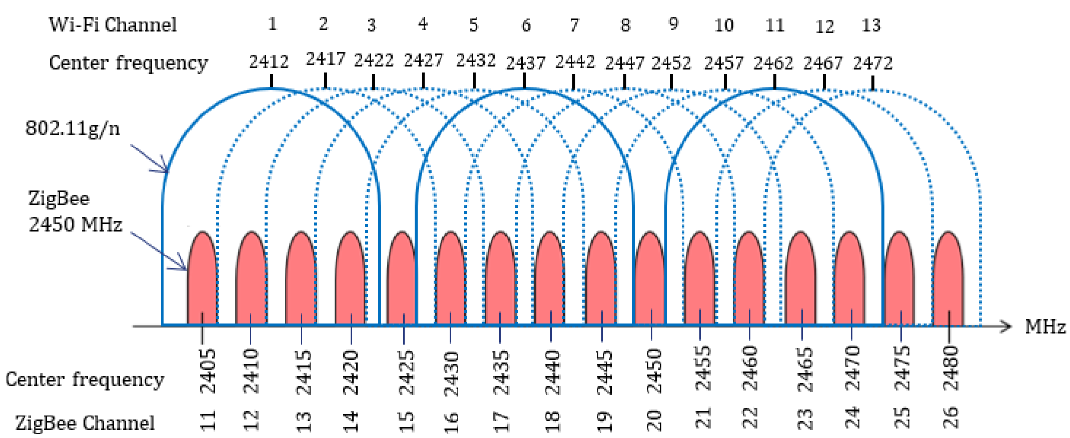
b. Plot the graph for PER versus frequency offset.

c. Discuss and analyze the results.

Task 4 - Coexistence of Zigbee and *Bluetooth*®

Coexistence of the 2.4 GHz Zigbee channels and *Bluetooth*® in the 2.4 GHz ISM band may cause interference problems. The *Bluetooth*® Classic, *Bluetooth*® LE, WLAN (802.11 b/g/n) and Zigbee channels and their corresponding center frequencies are shown in the figure below.





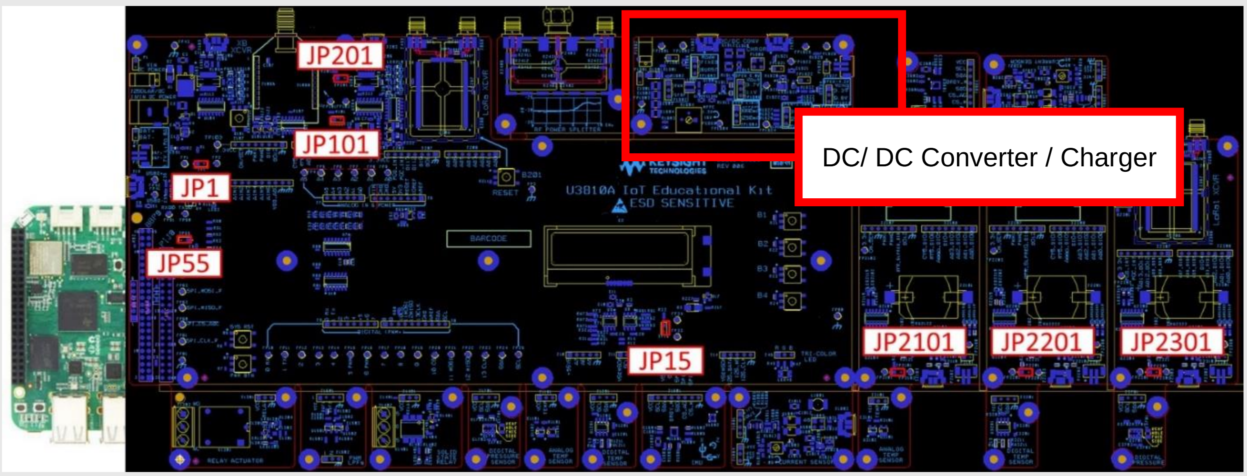
One or more *Bluetooth*® Classic (1 MHz wide) or BLE (2 MHz wide) channels may overlap with a Zigbee channel, which is 5 MHz wide. This can have a significant impact on a Zigbee network throughput. When a *Bluetooth*® transmission occurs within the channel bandwidth of a Zigbee network, the Zigbee receiver will have trouble demodulating the Zigbee transmission. Losing even part of the Zigbee packet will usually cause the whole transmission to be discarded with errors.

In this experiment, the impact of *Bluetooth*® interference on the Zigbee data throughput performance will be evaluated. A possible setup for this experiment is shown in the figure below:

NOTE

Before you begin the experiment, configure the Keysight U3810A IoT Development Kit as a "cape" on top of the BeagleBone CPU, and assure the same jumper configuration as the previous task.

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



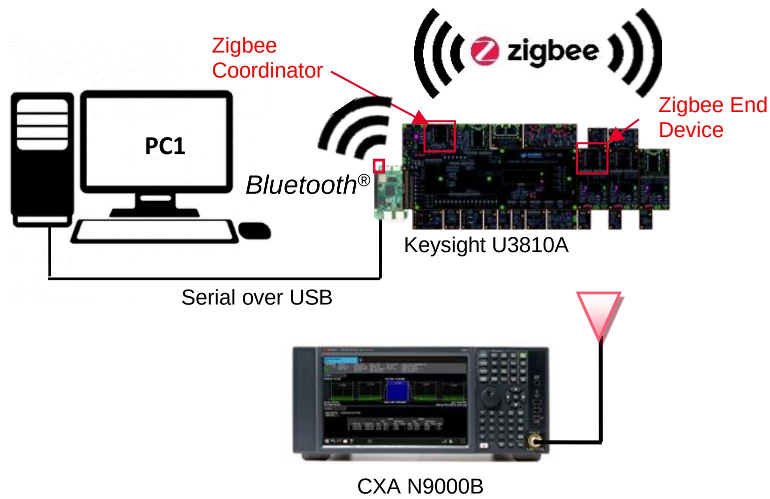
**DC/ DC Converter / Charger Board jumper positions are not relevant to this lab.

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

WARNING Do not connect voltages higher 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.

Zigbee signal modulation on the 2.4 GHz ISM band is different than the *Bluetooth*® signal modulation. (Zigbee modulation also varies depending upon what band is used, 900 MHz vs 2.4 GHz). Zigbee signals are analyzed more extensively in Task 2. For now, it is enough to know that Zigbee modulation is quite different than *Bluetooth*® LE modulation, so two devices of these two different types cannot communicate in a cooperative manner, leading to potential coexistence problems.

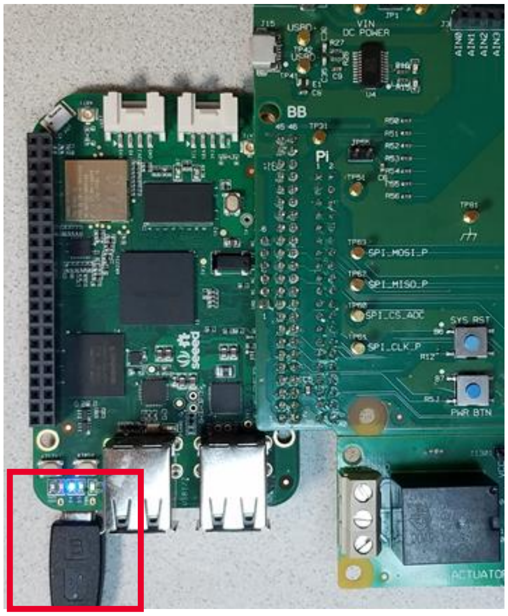
The setup for measurement in this task is shown in the figure below. An antenna is connected to the signal analyzer to receive the Zigbee signal transmitted by the U3810A board. A PC is used for controlling the XBee module on the U3810A and another XBee module via two serial connections.



Task 4a - Initial Measurement Setup

First, set up the U3810A board as the coordinator of a Zigbee PAN by doing the following.

1. Connect the BeagleBone CPU to your computer using a micro USB cable.



2. Login to the BeagleBone via SSH Communication in PuTTY terminal with the following details. For more information, refer to Set Up SSH connection.

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

3. Disable the WLAN transmission using the following command
   `rfkill block wlan`

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. For more information see Appendix.

Task 4b - Configure the Zigbee Coordinator

1. Connect the XB Transceiver on the U3810A assembly to the computer using a micro-USB cable.
2. Launch the XCTU software on the PC1. Then, **discover** and **add** the Zigbee module on the Keysight U3810A.
3. Reset the firmware settings to their default values by clicking “**Default**”. Then click “**Write**” to perform the write operation.
4. Configure the XB, XBee3 Zigbee module on the Keysight U3810A, according to the settings shown below. You may use the **Parameter** search field at the top right to locate each setting:

Parameter	Value	Description
CE	Formed Network [1]	Set the device as coordinator.
ZS	1	Set or read the Zigbee stack profile value. This must be the same on all devices that will join the same network.
NJ	FF	Set/Read the Node Join time. The value of NJ determines the time (in seconds) that the device will allow other devices to join to it. The coordinator will allow nodes to join without time limit.
NI	Gateway	Define the node identifier, a human-friendly name for the module. The default NI value is a blank space. Make sure to delete the space when you change the value.
SC	800	Set the operating channel of the network. Use the **Bitfield calculator** of the **Scan Channels (SC) to** select a channel which overlaps with the WLAN channel setup previously. The Zigbee channels are labeled as follows: Bit 0 = Chan 11, …, Bit 15 = Chan 26.
BD	38400 [5]	The serial interface baud rate for communication between the serial port of the module and the host.
AP	API Mode with Escapes [2]	Enable API

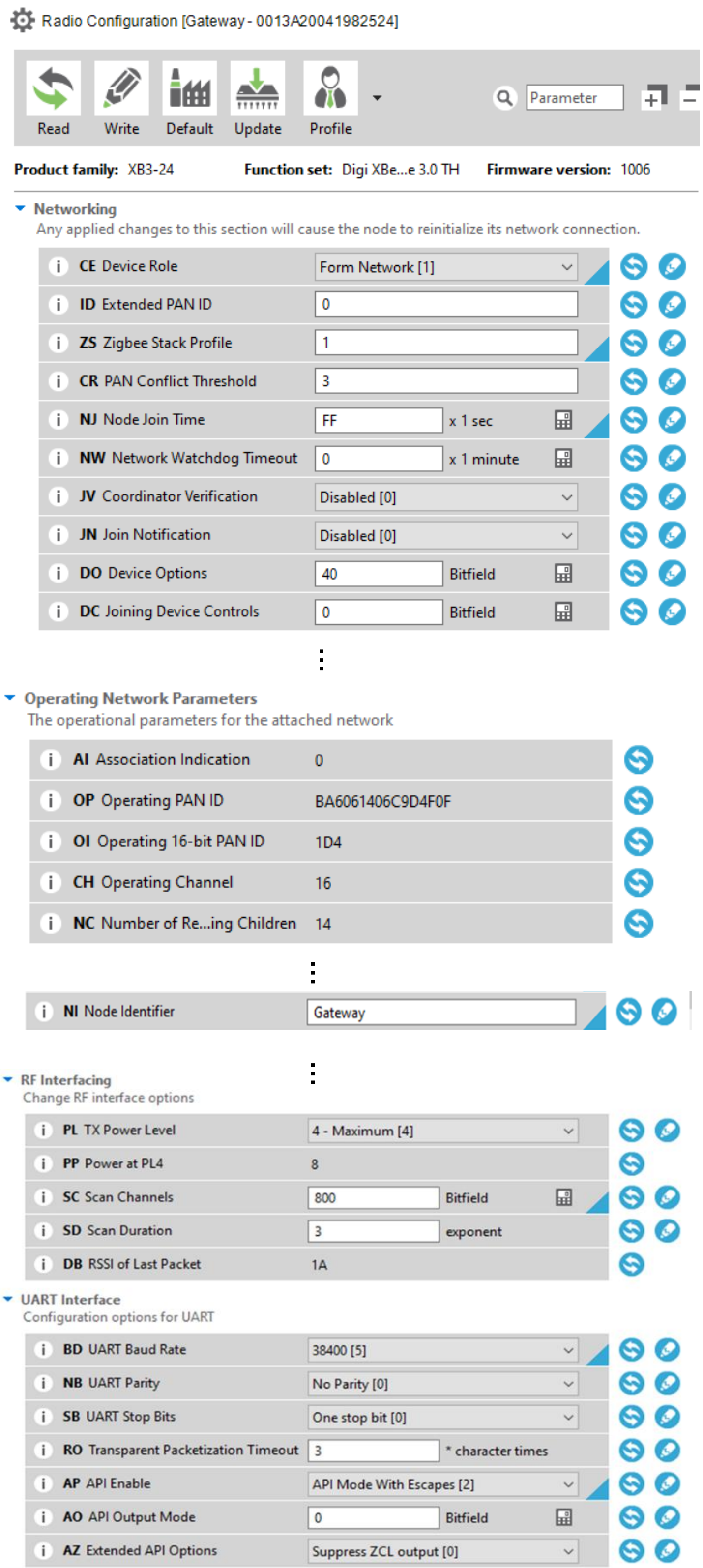
5. To apply the changes, click **Write** and click **Read** to update the readings (You may need to click Read *twice* for the changes to be effective).

NOTE

Take note that the **Operating PAN ID (OP)** and **Operating Channel (CH)** are now updated. Record the new **OP** and **CH** values. Note that **MY = 0** which is the 16-bit address of the coordinator and **AI = 0** which indicates a successful startup.

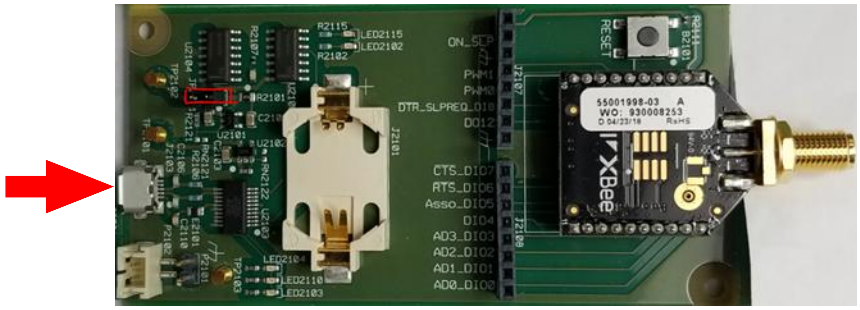
Parameter	Value	Value (example)
Operating PAN ID (OP)		BA6061406C9D4F0F
Operating Channel (CH)		0x16

6. The following images show a configured and operating Coordinator node:



Task 4c - Configure the Zigbee End Device

1. Connect the USB cable of the second XBee3 (XB1 or XB2) to the computer.



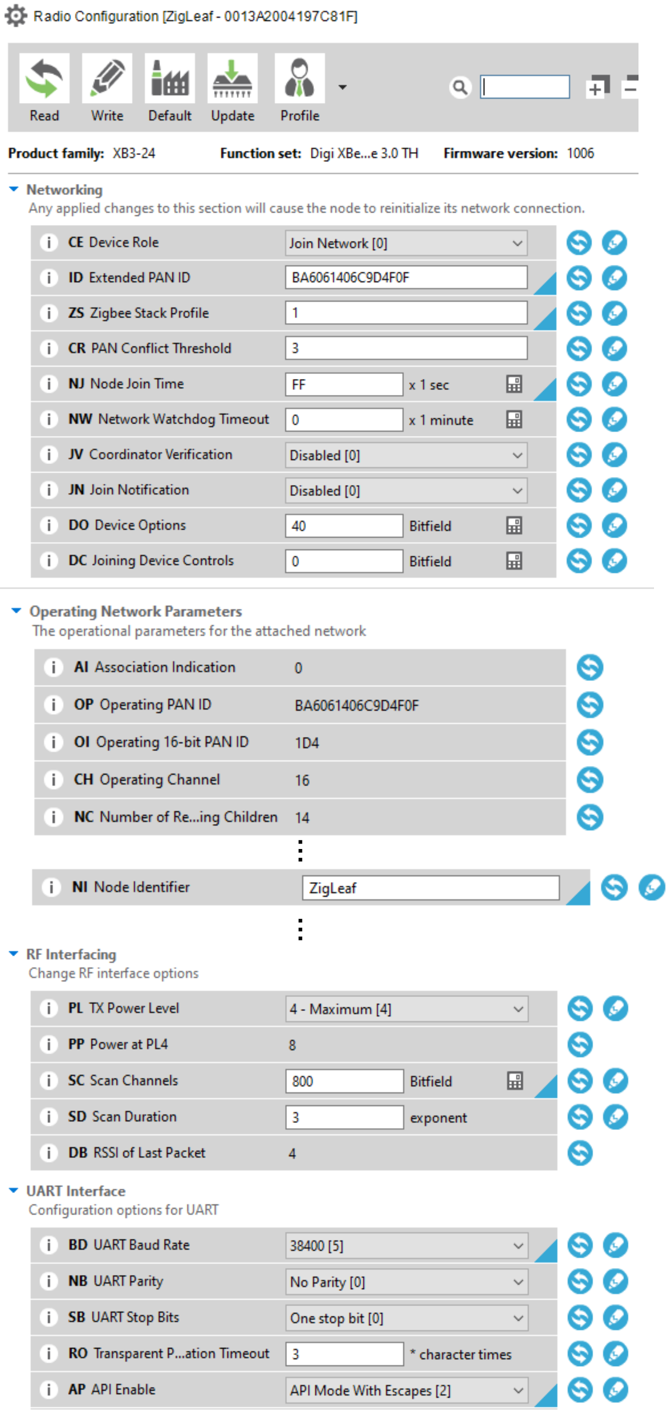
2. Launch the XCTU software on the PC1. Then, discover and add the Zigbee module on the Keysight U3810A.
3. Reset the firmware settings to their default values by clicking “Default”. Then click “Write” to perform the write operation.
4. Configure the XB, XBee3 Zigbee module on the Keysight U3810A, according to the settings shown below. You may use the **Parameter** search field at the top right to locate each setting:

Parameter	Value	Description
CE	Join Network [0]	Set the device as router or end device
ID	BA6061406C9D4F0F	Define the network that a radio will attach to. Refer to the Zigbee Coordinator’s ID. This must be the same for all radios in your network.
ZS	1	Set or read the Zigbee stack profile value. This must be the same on all devices that will join the same network.
NJ	FF	Set/Read the Node Join time. The value of NJ determines the time (in seconds) that the device will allow other devices to join to it. The coordinator will allow nodes to join without time limit.
NI	ZigLeaf	Define the node identifier, a human-friendly name for the module. The default NI value is a blank space. Make sure to delete the space when you change the value.
SC	800	Set the operating channel of the network. Use the **Bitfield calculator** of the **Scan Channels (SC) to** select a channel which overlaps with the WLAN channel setup previously. The Zigbee channels are labeled as follows: Bit 0 = Chan 11, …, Bit 15 = Chan 26.
BD	38400 [5]	The serial interface baud rate for communication between the serial port of the module and the host.
AP	API Mode with Escapes [2]	Enable API

5. To apply the changes, click **Write** and click **Read** to update the readings (You may need to click Read *twice* for the changes to be effective).

> **NOTE**
>
> Take note that the **Operating PAN ID (OP)** and **Operating Channel (CH)** are now updated. Record the new **OP** and **CH** values. Note that **MY** is updated with the 16-bit address of the end device and **AI = 0** which indicates a successful startup.

6. The Leaf Node should be configured as the following images indicate.


Radio Configuration [ZigLeaf - 0013A2004197C81F]
Read
Write
Default
Update
Profile
Product family: XB3-24
Function set: Digi XBe...e 3.0 TH
Firmware version: 1006
Networking
Any applied changes to this section will cause the node to reinitialize its network connection.
CE Device Role
Join Network [0]
ID Extended PAN ID
BA6061406C9D4F0F
ZS Zigbee Stack Profile
1
CR PAN Conflict Threshold
3
NJ Node Join Time
FF
x 1 sec
NW Network Watchdog Timeout
0
x 1 minute
JV Coordinator Verification
Disabled [0]
JN Join Notification
Disabled [0]
DO Device Options
40
Bitfield
DC Joining Device Controls
0
Bitfield
Operating Network Parameters
The operational parameters for the attached network
AI Association Indication
0
OP Operating PAN ID
BA6061406C9D4F0F
OI Operating 16-bit PAN ID
1D4
CH Operating Channel
16
NC Number of Re...ing Children
14
NI Node Identifier
ZigLeaf
RF Interfacing
Change RF interface options
PL TX Power Level
4 - Maximum [4]
PP Power at PL4
8
SC Scan Channels
800
Bitfield
SD Scan Duration
3
exponent
DB RSSI of Last Packet
4
UART Interface
Configuration options for UART
BD UART Baud Rate
38400 [5]
NB UART Parity
No Parity [0]
SB UART Stop Bits
One stop bit [0]
RO Transparent P...ation Timeout
3
* character times
AP API Enable
API Mode With Escapes [2]

NOTE

If you have not already saved these XBee3 configurations as Profiles, you may save time and assure accurate configurations by copying and extracting the files from the lab code directory in the BeagleBone to your computer. You may then use the **M2_L6_T4_COORDINATOR.xpro** and **M2_L6_T4_EndDevice.xpro** profiles for this Task.

Go to “**Profile** > **Create configuration profile**” and select the profile. For the gateway (U3810A XB) you will need to change PAN ID (ID) to match the OP sensor node (XB1 or XB2) and re-save the gateway profile you have customized to your network.

The **xpro** file saved by XCTU is a zip format file. You can use zip software to unzip the file to examine the contents. Do not use XCTU versions older than 6.3.10 as configuration files for those versions were saved in a format (.XML), which is incompatible with later releases.

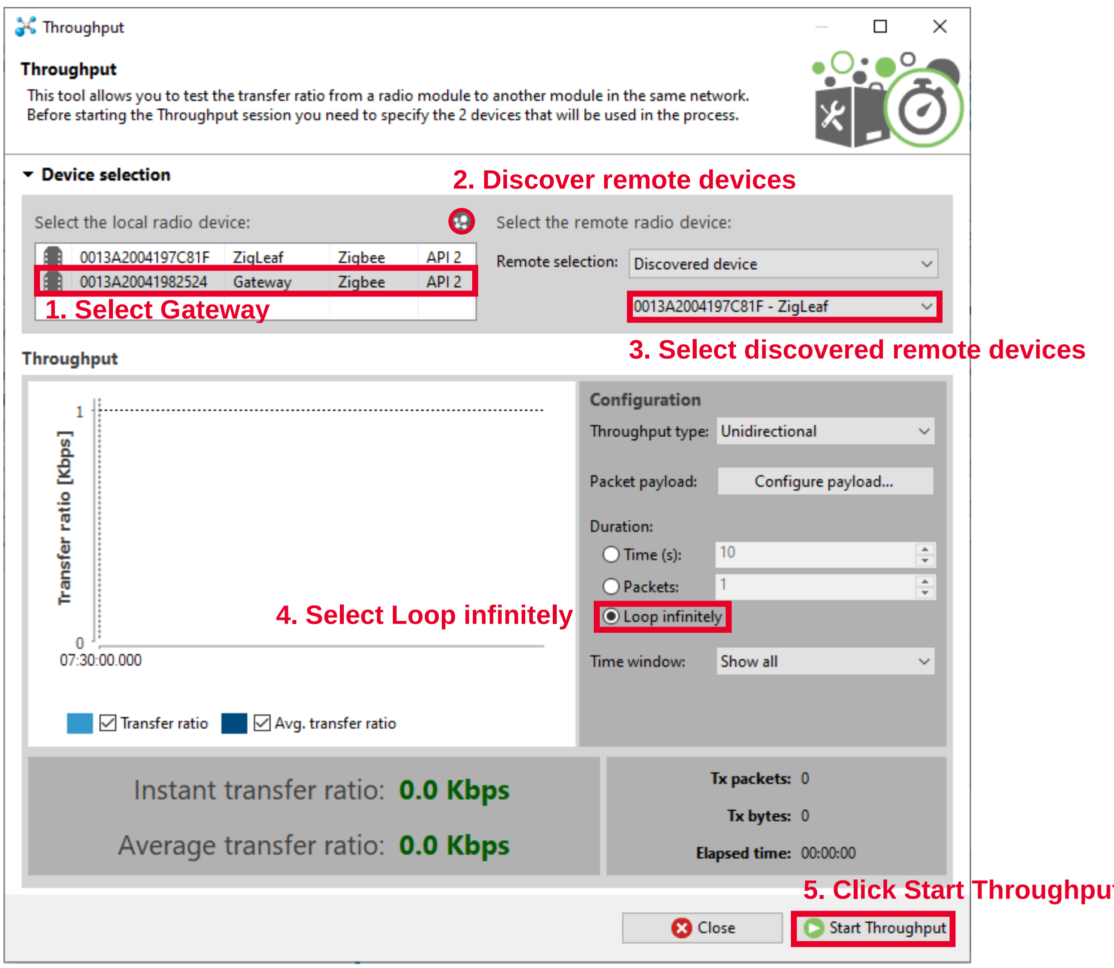
Read the XCTU change log (XCTU > Help > Change Log) for more details.

Task 4d - Throughput Performance of Zigbee in the Presence of BLE

Once the Zigbee link is established, we will use the XCTU "Throughput" test to evaluate one KPI (Key Performance Indicator) for the link This lab will use Throughput as the KPI for this test.

1. At the top of the right-hand window in XCTU (for the device named Gateway), click the tools icon and pulldown to "Throughput". This will start the throughput test tool.

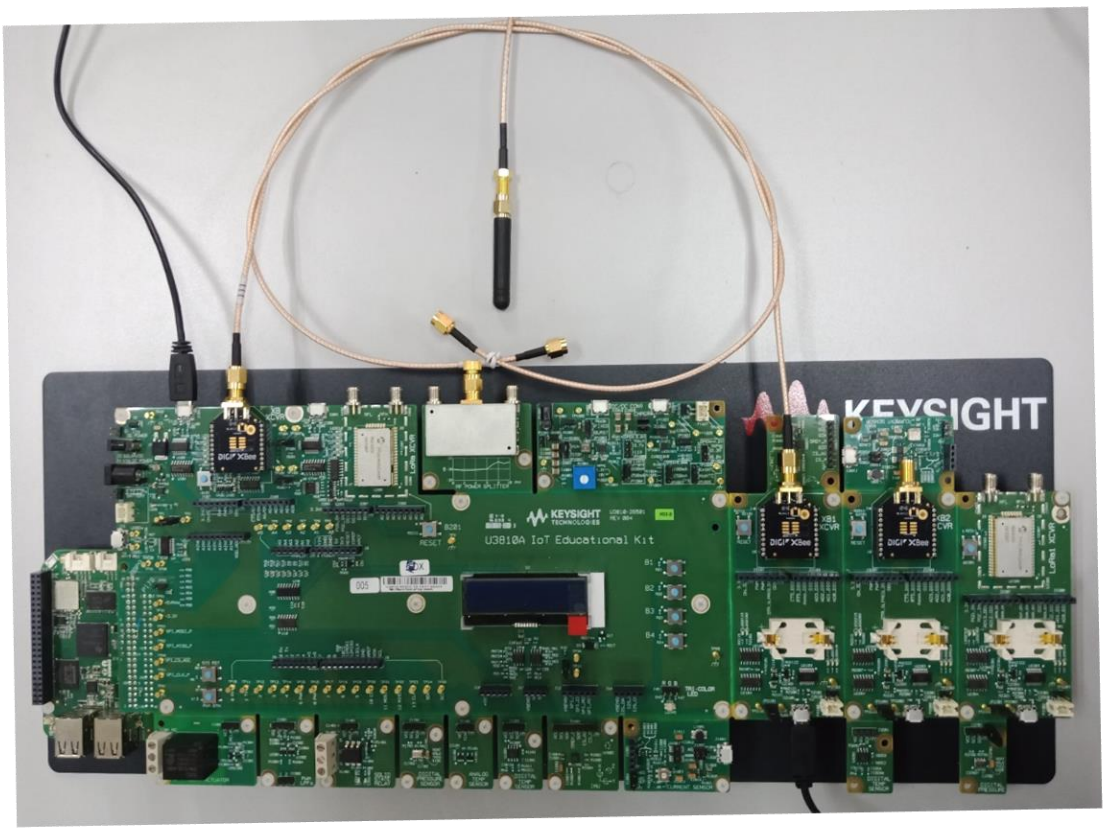
2. In the throughput tool, upper left, select the Gateway Zigbee device, and on the right, under "Discovered device" select ZigLeaf (or whatever you called your end node) also select "Loop infinitely" to allow a long duration test.



3. Click "**Start Throughput**" at the lower right to start the test. You should see the graph start updating with a rate of around 6.5 k bits/sec. You may want to adjust the position of antennas and coaxial cables to be close to the BeagleBone *Bluetooth*® bar antenna in the upper left portion of the BeagleBone board.

4. Place the antenna for the CXA spectrum analyzer in the vicinity of the Zigbee Unit and the BeagleBone bar antenna. Turn on the CXA and select Spectrum Analyzer application (Press **MODE/MEAS > Spectrum Analyzer > Swept SA > Normal**). Configure the CXA to the following settings.

Description	Setting	Value
Preset	Press **[MEAS SETUP]**	Meas Preset
Frequency	Press **[FREQ] > Center Frequency**	2.460 GHz
Span	Press **[FREQ]> Span**	5.0MHz
Reference Level	Press **[AMPTD > Ref Level**	0dBm
Scale	Press **[AMPTD] > Scale/Div**	10dB
Trace	Press **[Trace] > Trace Control >Trace Type**	Max Hold



5. You may need to adjust the amplitude later depending upon signal amplitudes your antennas pick up during the test. Keep the peak signals below the top of the screen.

6. Observe the signals on the screen of the CXA Spectrum Analyzer. You should be seeing the periodic Zigbee signals like this screen below.

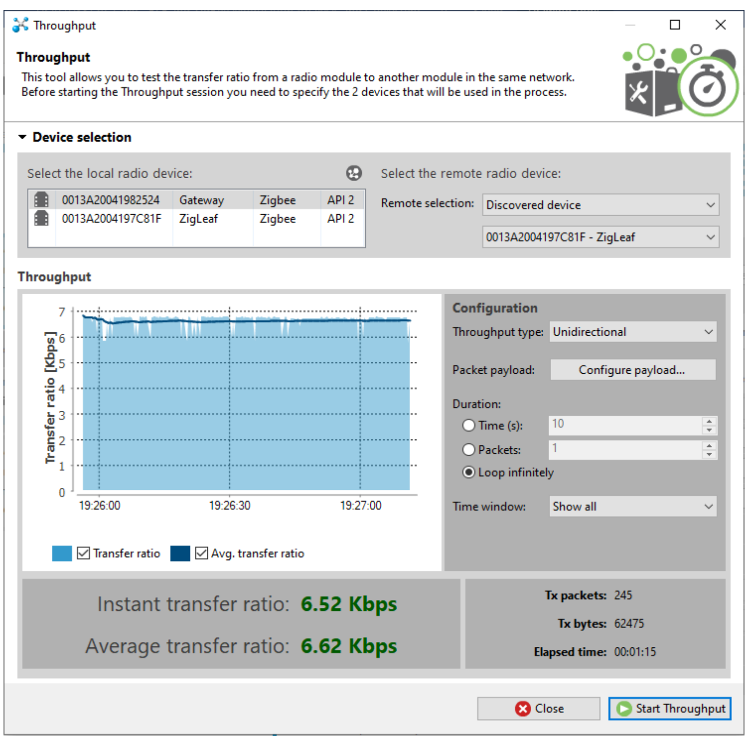
NOTE

You may need to move the antenna to capture the signal.

If the Zigbee signal is very low (<-50dBm), you may need to change the attenuation (Press **[AMPTD] > Attenuation > Atten**) to 0 dB to capture the signal.

7. Change the trace type back to normal: **TRACE > TraceType** > **Clear/Write**

8. Observe the XCTU Throughput window and notice the throughput is around 6.5 K bits/second. This is normal if the baud rates are set to 38400 and signals are strong enough for a reliable link.



9. After checking the throughput with no interferer, you will turn on the *Bluetooth*® LE transmitter as an interferer on the same RF channel. Throughput should drop to around 4K bits/sec if the signals are nearly equal amplitudes (*Bluetooth*® LE and Zigbee signals are nearly the same amplitude).

   Once XCTU Throughput test is running, you will need to create interference on the same or adjacent channels as the Zigbee Channel (2.460 GHz). You will use the BeagleBone *Bluetooth*® LE test transmissions to create an interfering signal. Since the *Bluetooth*® LE and the Zigbee use different modulation types and protocols, they cannot cooperatively use the radio channel and interference will result. This will cause the Throughput test to show a decrease in throughput on the Zigbee link.

   Wireless Coexistence is affected by three major signal characteristics affect how signals interfere: Frequency, Power and Time. Since these transmissions are brief (*Bluetooth*® LE test packet is about 405 uSec long) they must be frequent to cause significant interference. The BLE transmitter test has 405 uSec ON and 220 uSec OFF, yielding a total period of 625 uSec, and a duty cycle of about 65%. This means that 65% of the time, a Zigbee transmission may encounter the interfering BLE signal, and experience transmission errors and lost packets. The Zigbee protocol may be able to recover a damaged packet using the error correction built into the packet, or it may have to request a retry, causing the source node to retransmit. The retransmission will occupy more time, and the net throughput will decrease. Note that interference may damage data packets and may damage ACK (acknowledge) packets or even NAK (non-acknowledge) packets in addition, all slowing the link.

   The following image of the commands sent to the *Bluetooth*® module show the sequence of turning on the *Bluetooth*® LE transmitter test and then moving the *Bluetooth*® LE frequency from 2.402 GHz to 2.460 GHz. The 0x1D and 0x00 are the parameters which select the channel for the *Bluetooth*® LE transmissions.

   `rfkill unblock bluetooth` assures the *Bluetooth*® peripheral is active

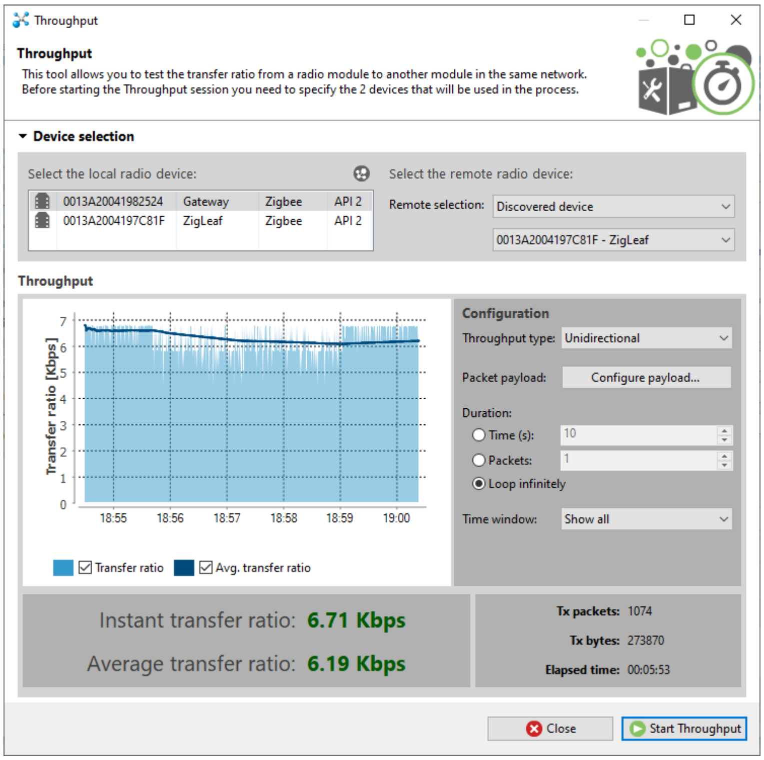
   `sudo hcitool cmd 0x08 0x001E 0x1D 0x25 0x02` will turn on the transmitter test on 2.460 GHz (same frequency as the Zigbee signal). At this point, note the decline in throughput. Then a few seconds later send the command:

   `sudo hcitool cmd 0x08 0x001E 0x00 0x25 0x02` – This will move the *Bluetooth*® LE test frequency to 2.402 GHz, and the throughput should recover to the approximately 6.5 Kbps range.

```
debian@beaglebone:~$ sudo hcitool cmd 0x08 0x001E 0x1D 0x25 0x02
< HCI Command: ogf 0x08, ocf 0x001e, plen 3
  1D 25 02
> HCI Event: 0x0e plen 4
  01 1E 20 00
debian@beaglebone:~$ sudo hcitool cmd 0x08 0x001E 0x00 0x25 0x02
< HCI Command: ogf 0x08, ocf 0x001e, plen 3
  00 25 02
> HCI Event: 0x0e plen 4
  01 1E 20 00
debian@beaglebone:~$
```

10. Start the throughput test without interference. Wait for around 30-60 seconds, execute the command **`sudo hcitool cmd 0x08 0x001E 0x1D 0x25 0x02`** to turn on the *Bluetooth*® LE transmitter test at 2.46 GHz. After some time, turn off the *Bluetooth*® LE transmitter test. Observed the throughput test graph at the XCTU software.

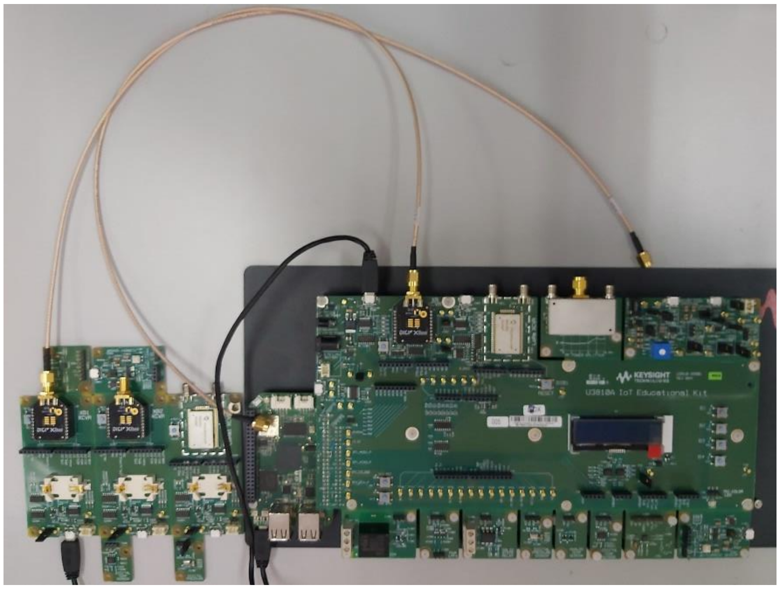
    The following image from XCTU shows the throughput test before interference (up to 18:55:40) is introduced, throughput in the presence of interference (18:55:40 to 18:59:00), and the after the interference is removed (18:59:00 to end). Your exact throughput values will differ, depending upon the relative signal strengths of the Zigbee link signals and the *Bluetooth*® LE interference signal.



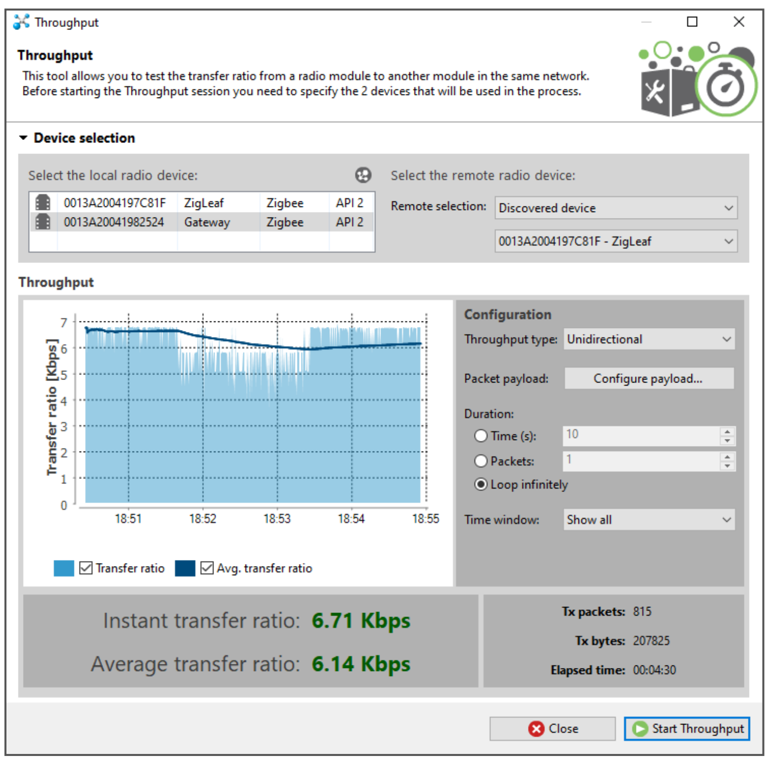
11. Stop the *Bluetooth*® LE transmitter by executing the following command:

```
debian@beaglebone:~$ sudo hcitool cmd 0x08 0x1f
```

12. Move the cable closer to the antenna on BeagleBone. Repeat previous step.



The following image from XCTU shows the throughput test before interference (up to 18:51:40) is introduced, throughput in the presence of interference (18:51:40 to 18:53:35), and the after the interference is removed (18:53:35 to end). After moving the cable close to the antenna, the relative signal strengths of the Zigbee link signals decrease while the *Bluetooth*® LE interference signal strength increase causing the throughput value to decrease.



Exercise

1. Turn on the *Bluetooth*® LE transmitter test at 2.462 GHz and start the throughput test. Wait for 30 to 60 seconds before you change the *Bluetooth*® LE transmitter test to 2.46 GHz. After some time, change the *Bluetooth*® LE transmitter test to 2.402 GHz. Observe the throughput test graph at the XCTU software. Explain:
    a. When you changed the *Bluetooth*® LE test frequency to 2.462 GHZ, 2 MHz higher than the Zigbee operating frequency of 2.460 GHz, what effect would this likely have on the Zigbee throughput?
    b. Why does moving the *Bluetooth*® LE test frequency back to 2.402 GHz change the throughput of the Zigbee link?
    c. What if you increased or decreased the RF Power Level (**PL** parameter in XCTU) of both Zigbee units?
2. Configure the CXA to capture the Zigbee throughput test signal and *Bluetooth*® LE transmitter test signal. Observed the timing different between the Zigbee throughput test signal and *Bluetooth*® LE transmitter test signal.
3. Re-enable the *Bluetooth*® transmission using the following command:

```
rfkill unblock bluetooth
```

WARNING It is important to always execute the command **`rfkill unblock bluetooth`** after completing the procedure in which you used the command **`rfkill block bluetooth`**. This must always be done before your BeagleBone CPU is shutdown or rebooted. For more information see Appendix.

Task 5 – Reset the Lab Code in BeagleBone

You have now at the end of this lab and any modifications on the lab codes need to be reset and ready for future lab sessions. Run the following commands to reset the lab code.

```
cd ~

sh LabCodeReset.sh
```

NOTE

Resetting the lab code will erase any changes made in the LabCode directory. If there are files that you want to keep, copy them out of the LabCode directory tree first.

References

[1] *Bluetooth®* SIG, "*Bluetooth®* Core Specification v4.0", 2010

[2] IEEE, "IEEE 802.15.4-2015: IEEE Standard for Low-Rate Wireless Networks", 2015

[3] CXA N9000B product page (www.keysight.com/find/n9000B)

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

Appendix A - 802.15.4 O-QPSK PHY Symbol-to-Chip Mapping

Data symbol	Chip values (c_0 c_1 … c_{30} c_{31})
0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
5	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
6	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
7	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
8	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
11	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
13	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
14	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
15	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

Appendix B - Establish Secure Shell (SSH) Communication between BeagleBone and PC

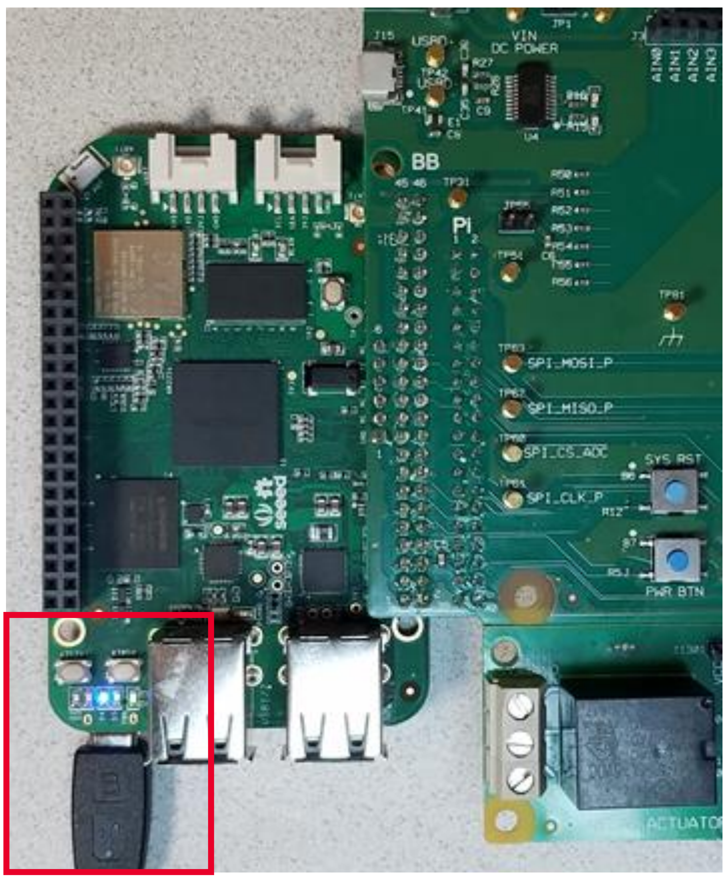
NOTE

The following procedures are part of Lab 1 and are provided here as a reference. The connection procedure is usually to be performed only once. Please no to repeat it unless necessary.

In this exercise, you will connect the PC (Host) to Beagle Bone via a USB cable and establish an 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 SSH 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.**

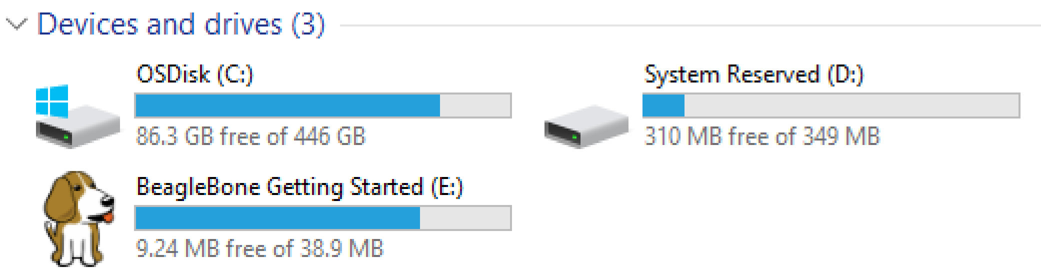
WARNING When JP1 is in place, do not connect a USB cable to both the BeagleBone and J15 at the same time, or anomalous behavior may result.

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 1 minute to complete the boot process:

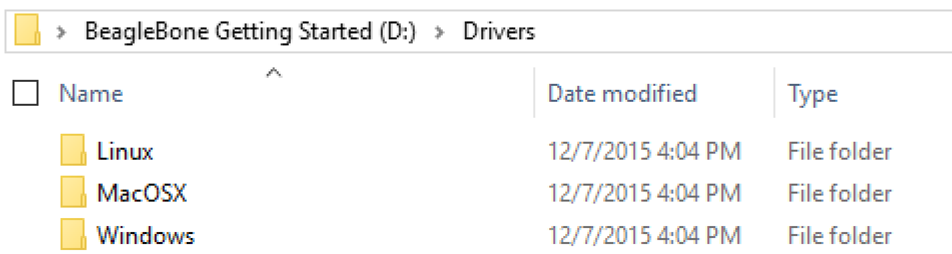


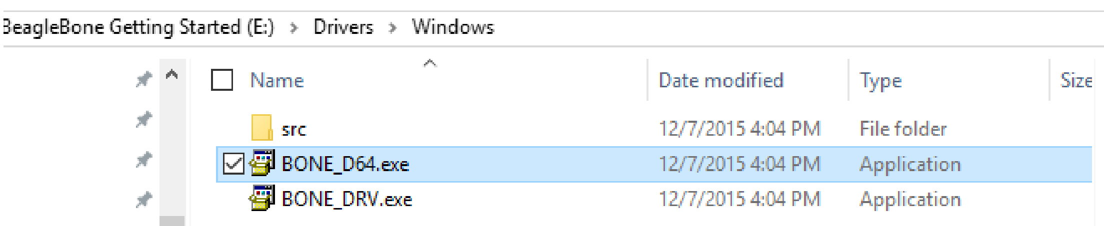
Install RNDIS drivers

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

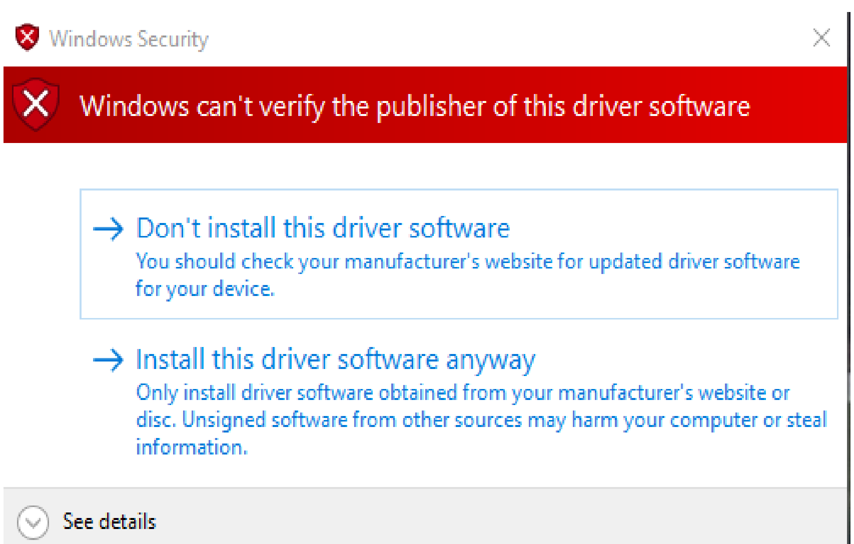


3. Select the driver for your OS from the Drivers folder and install the BONE_D64.exe file.

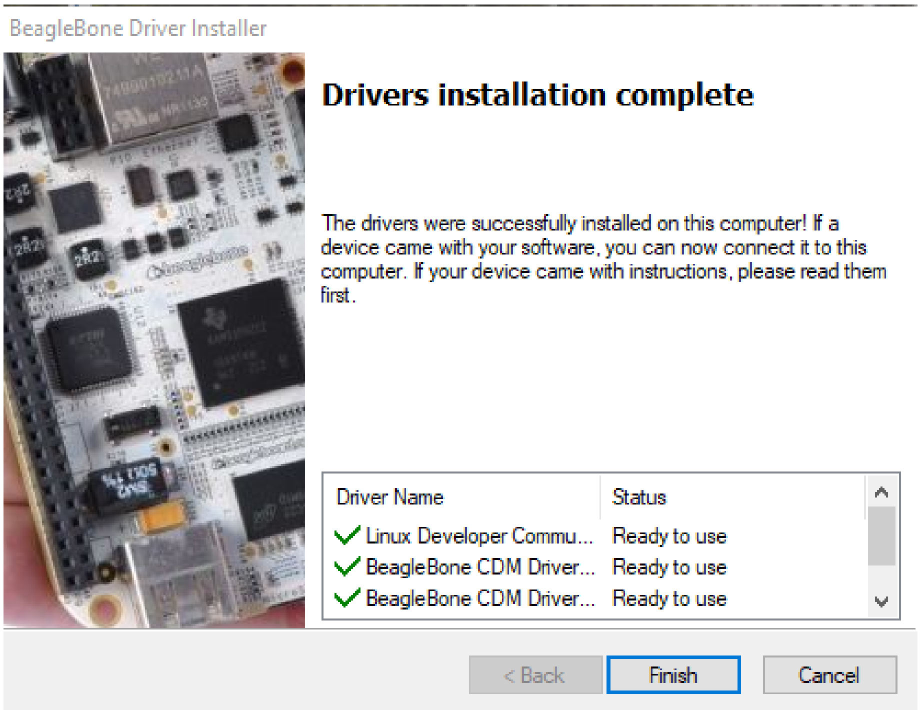




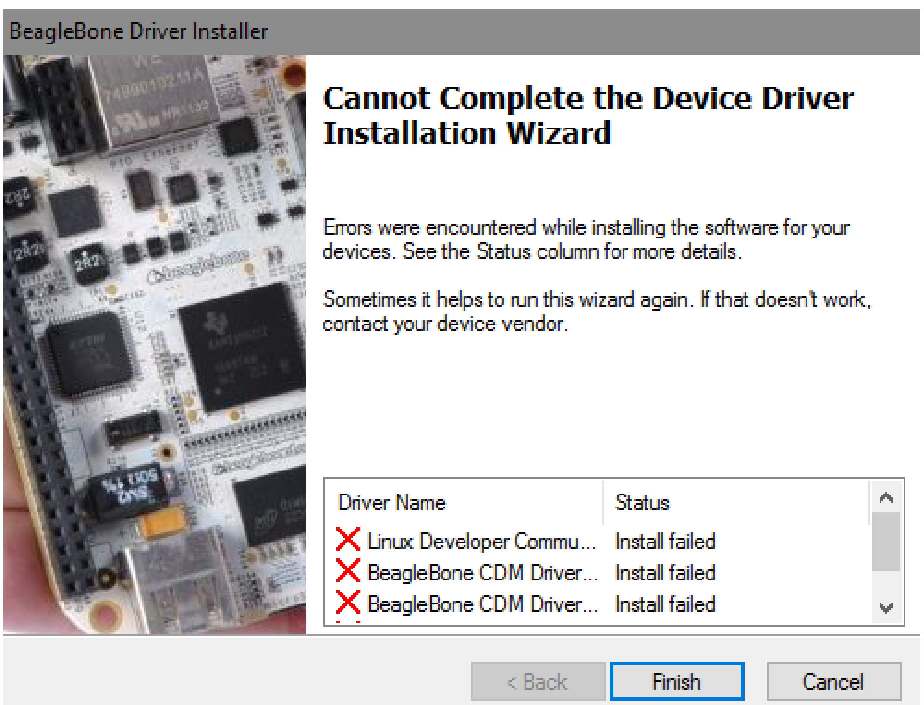
4. During the installation, Windows 10 users may see this message. Select the **Install this driver software anyway**.



Successful installation will show the message below.



Refer **Lab 1 Appendix C – Troubleshooting RNDIS Drivers installation** for more information if you receive the error below.

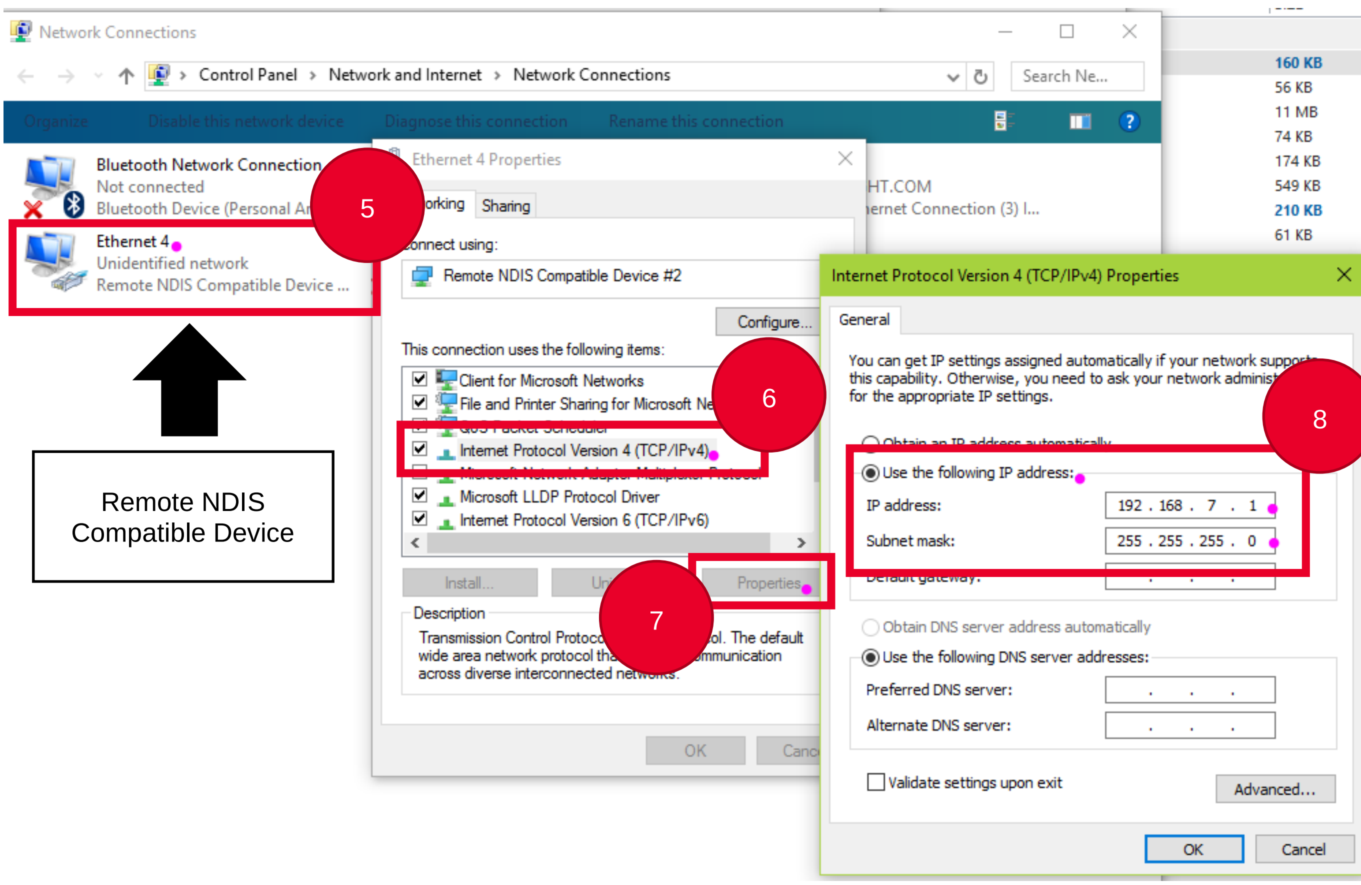


Configure RNDIS adapter

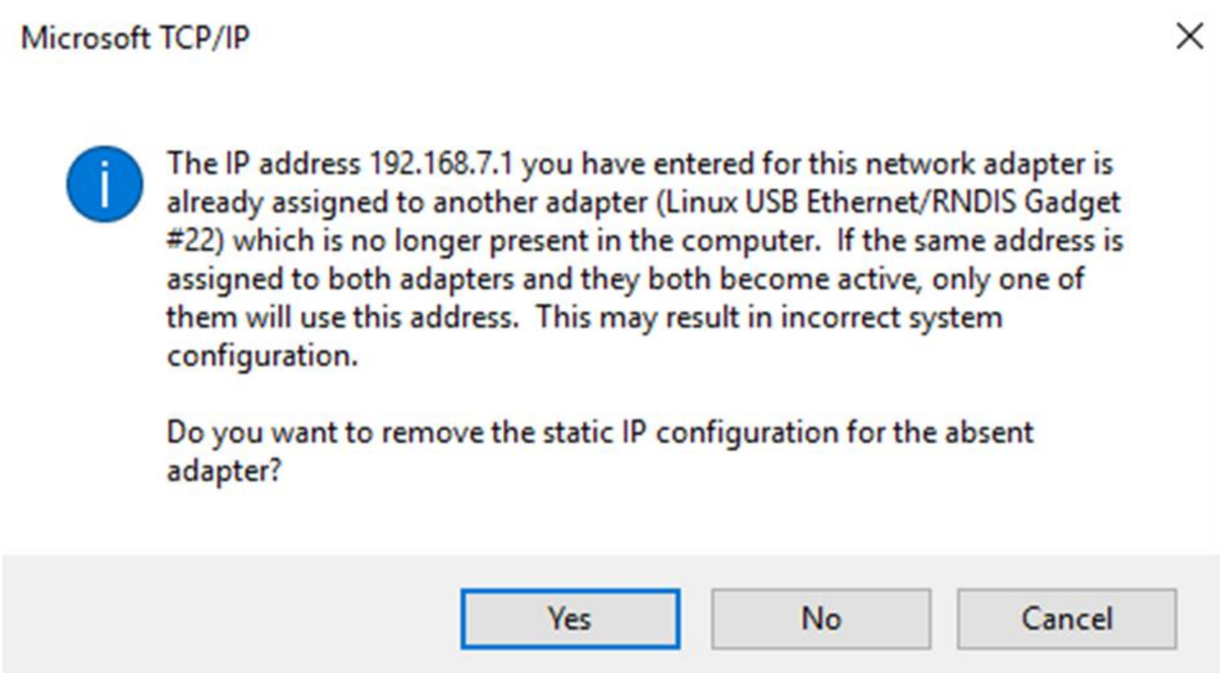
Your PC will need to be on the same subnet using the RNDIS connection. This does not have DHCP, so your PC address needs to be set to **192.168.7.1**.

> **NOTE**
>
> Each time a different BeagleBone is connected to a PC, this step may need to be performed.

5. Go to **Network Settings**, right-click Remote NDIS Compatible device, and select Properties.
6. Click **<your Remote NDIS Adapter>** and click **Internet Protocol Version 4 (TCP/IPv4)**.
7. Click **Properties**.
8. Set up the general settings for the IP address as shown below.

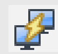


If you receive the message below, it could either mean that a BeagleBone or other device was using this address previously. Select **Yes** if any of the device are not in use or select **No** as long as both devices are not present.

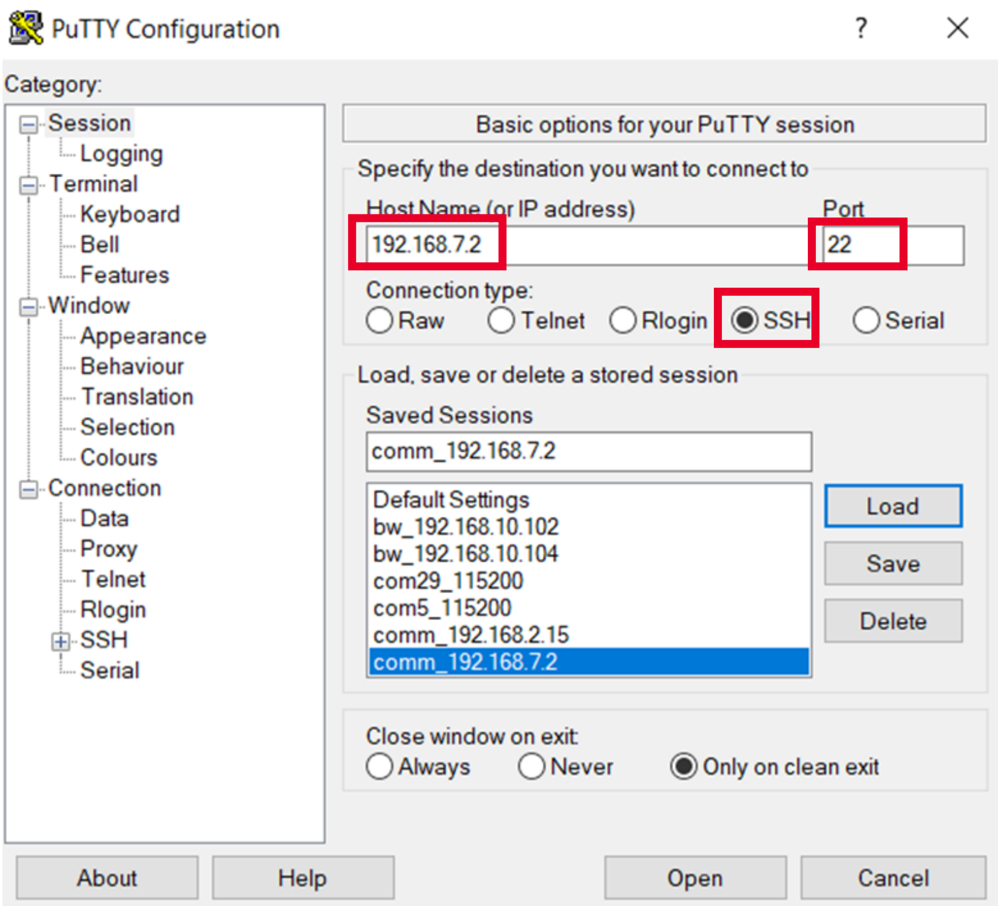


Set Up SSH connection

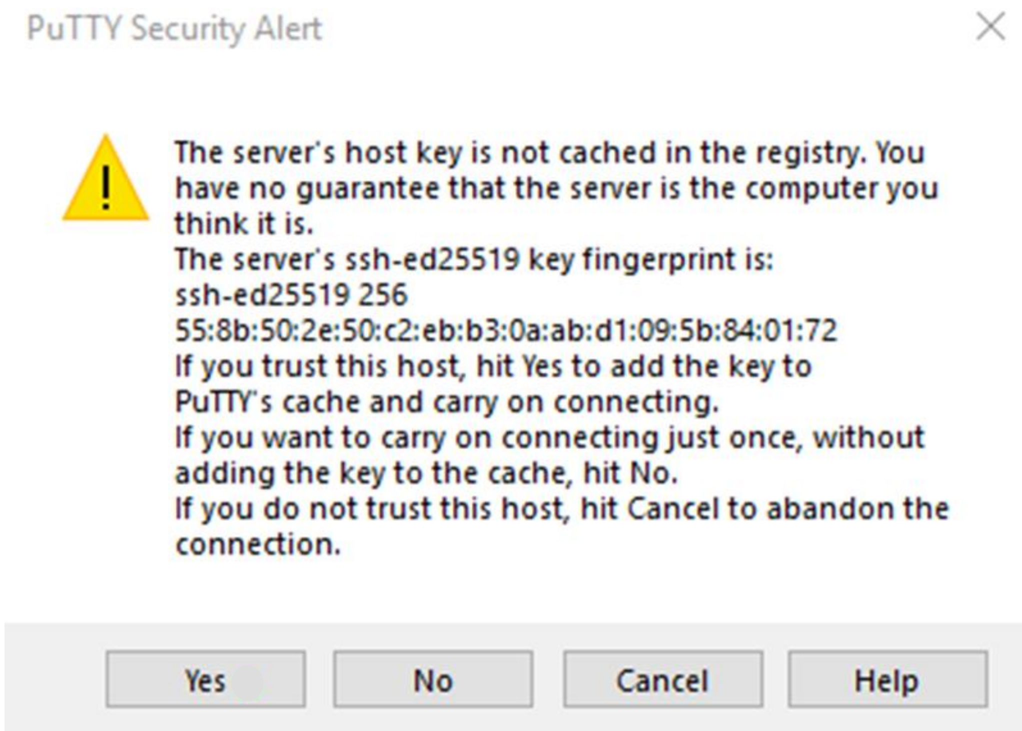
9. Once the ping command comes back with a reply and a response time, double-click PuTTY.exe to launch the PuTTY terminal program.

> **NOTE**
>
> PuTTY may also be launched from WinSCP by clicking  . When launched in this manner, the connection is completed automatically and the steps below are not required.

10. 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**.



11. 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**.



12. 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

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

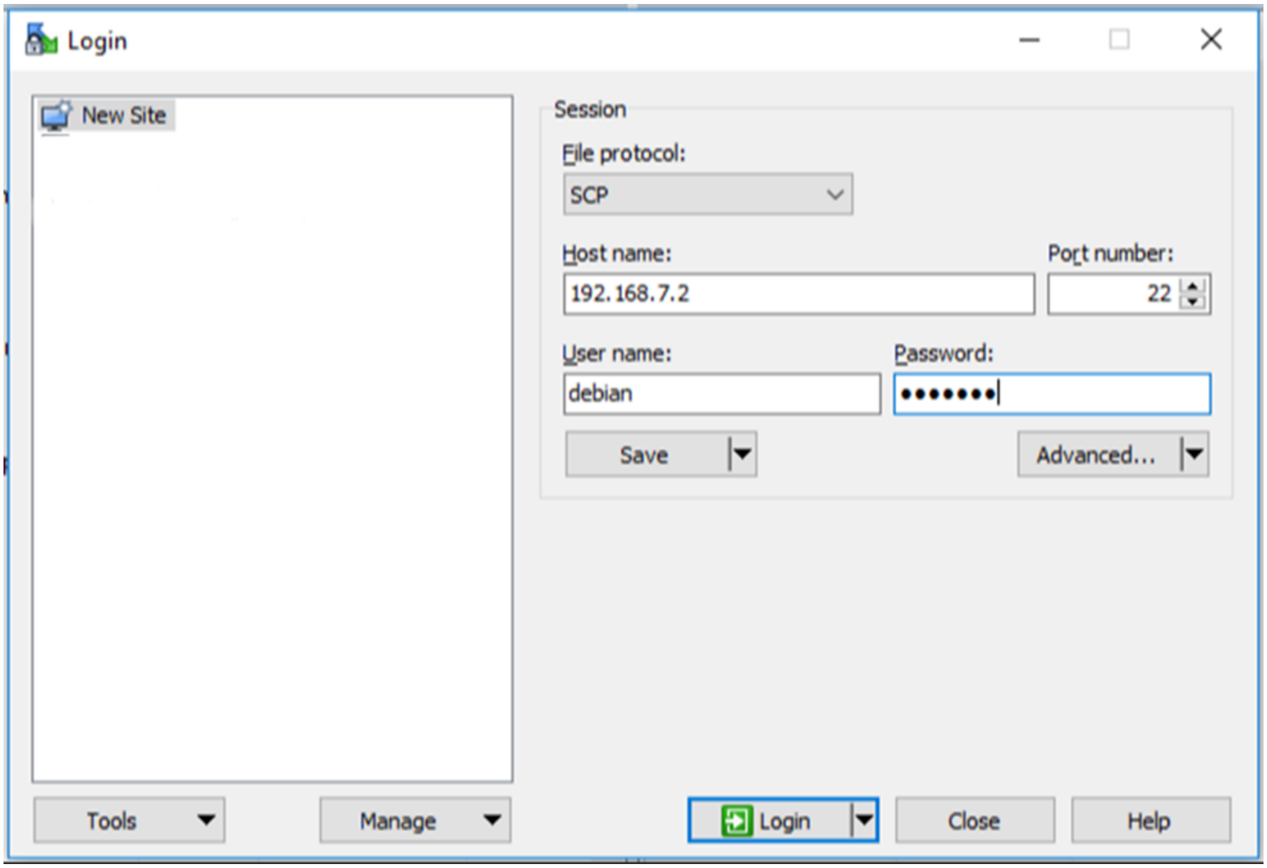
    Note that the password will appear as blank and unresponsive as you type.

```
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:~$
```

Appendix C - Transfer Files Using WinSCP

Set Up WinSCP

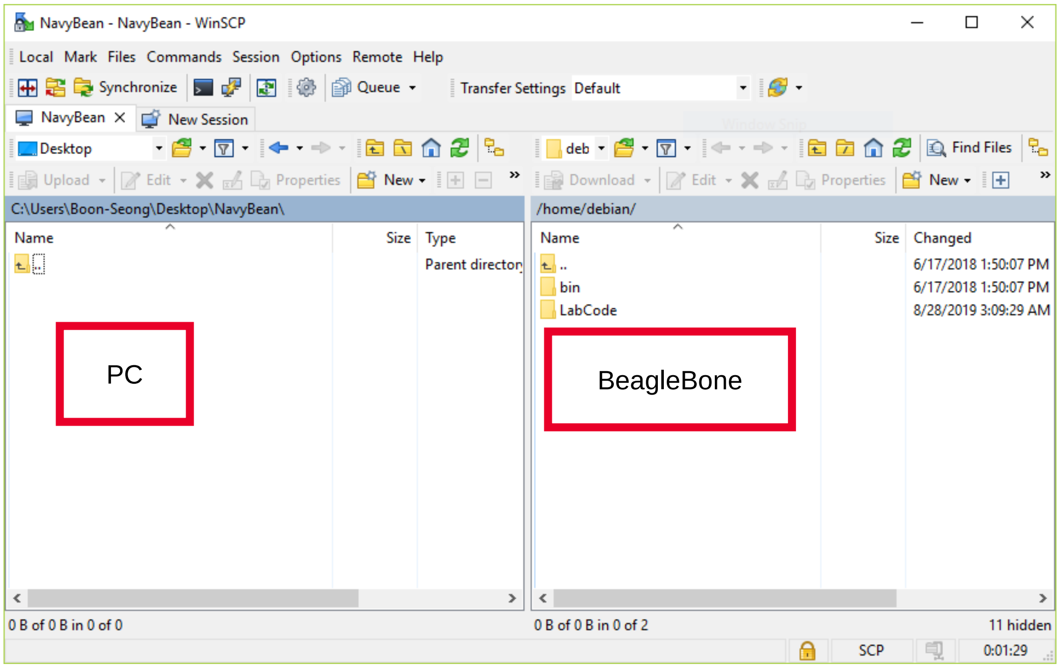
1. For Windows users, download and install a copy of WinSCP from https://winscp.net/eng/download.php
2. Launch WinSCP and click "**New Site**". Select **SCP**, enter **192.168.7.2** Port Number **22**, username **debian** and password **temppwd**



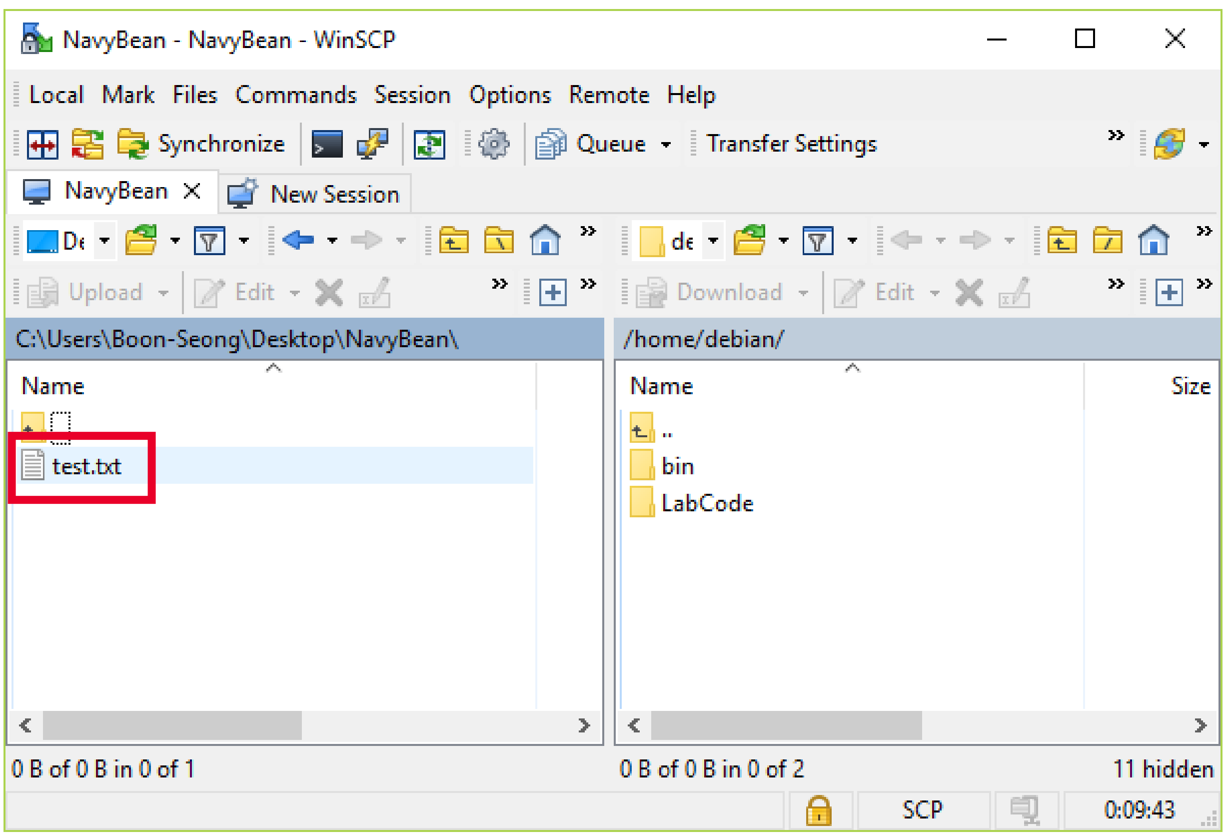
3. Answer Yes to the question about connecting to an unknown server if this is the first time the computer is connecting WinSCP to the Beagle Bone.

Copy Files with WinSCP

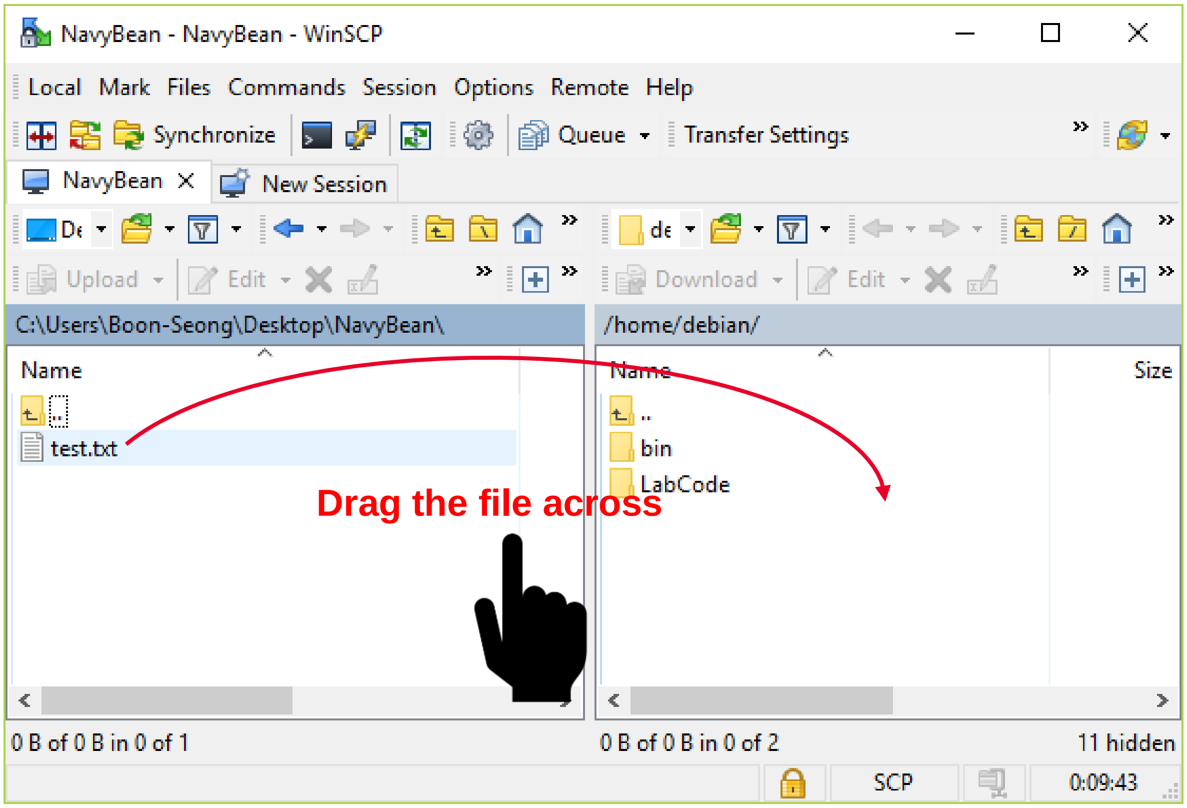
You should see the GUI below where you can drag files across, to transfer it from the PC to the BeagleBone and vice-versa.



4. Create a text file "test.txt" in your Desktop.



5. Copy the “test.txt” file over to the BeagleBone using WinSCP by dragging it across.

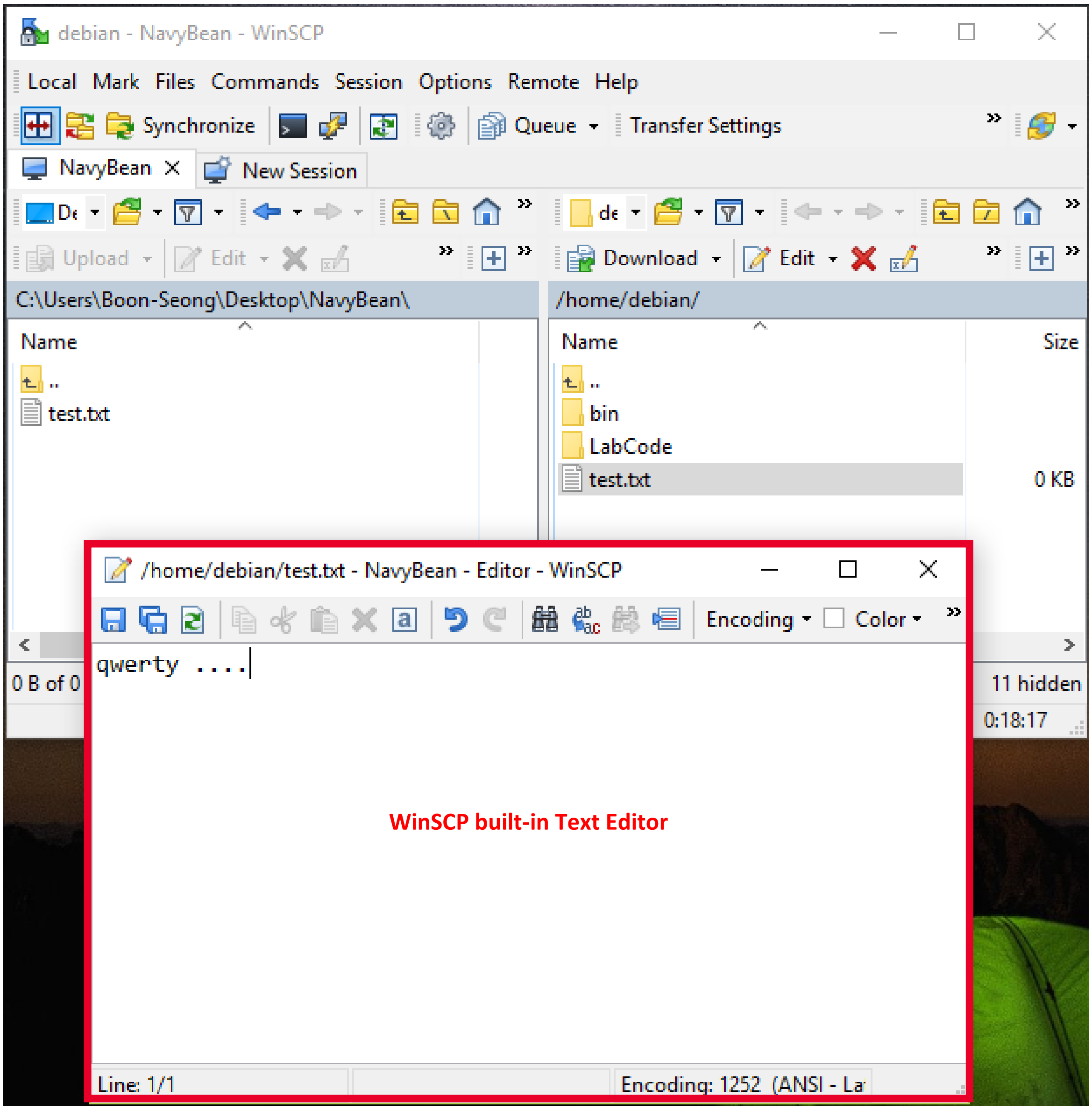


NOTE

For Linux based systems, copy the file using **scp M2-L1.zip debian@192.168.7.2** command.

Edit Files with WinSCP

6. After transferring the test.txt file to BeagleBone, right-click the test.txt file and select **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. After editing your text file, save your changes, and close the text editor.

> **NOTE**
>
> You may want to frequently save your changes while editing the file due to the risk of losing your changes if there are any disconnection between your PC and BeagleBone.

Appendix D - Configure BeagleBone to connect to WLAN network

From Lab 1 Task 1d – 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. It is 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>
```

> **NOTE**
>
> 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, then type in **`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 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/PuTTY system, right-click the mouse to paste, and on a Linux system middle-mouse-click.

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

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 you 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
```

Appendix E - Keysight U3810A Technical Documents

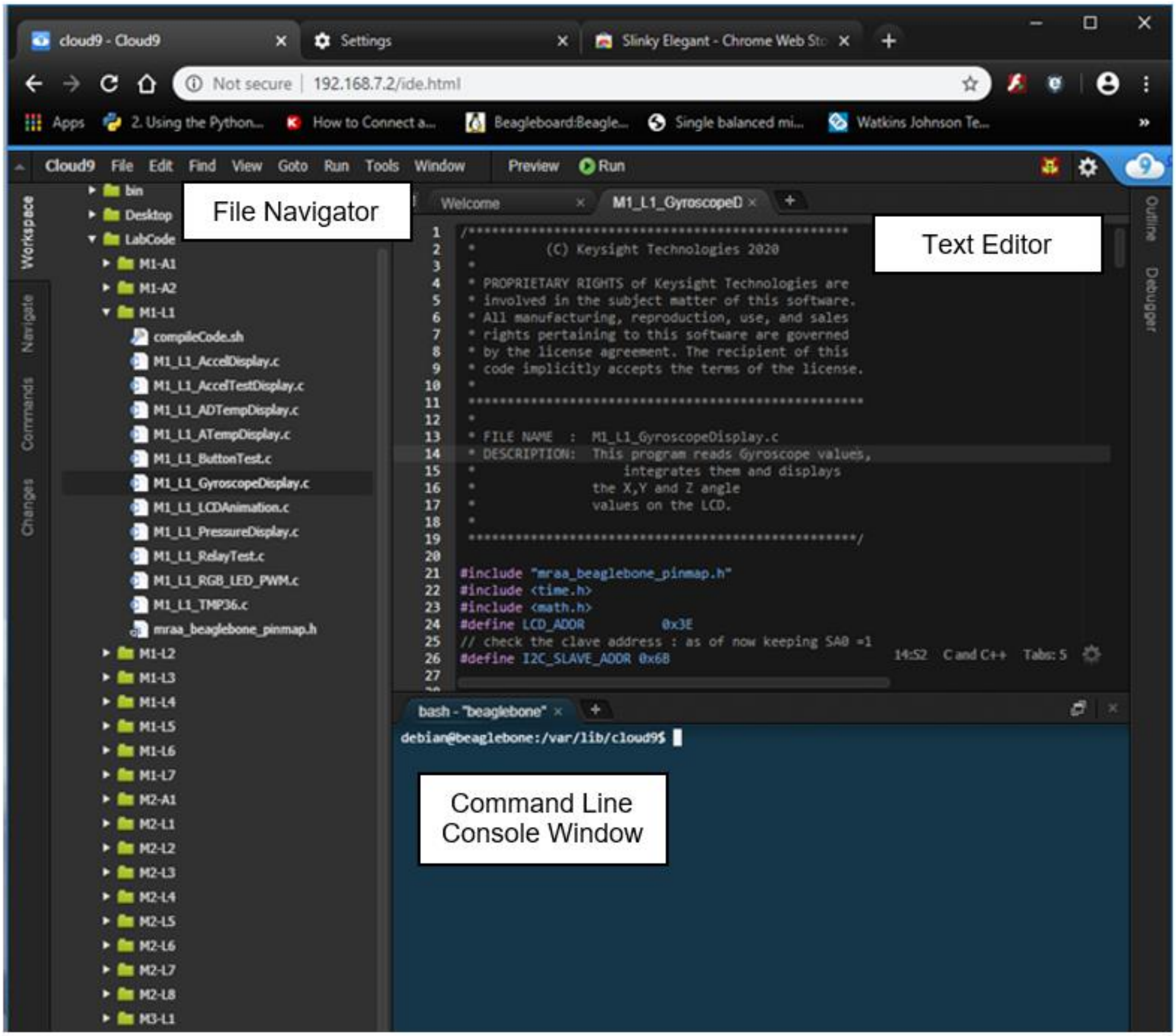
Board Component Locator and Schematic diagrams are available in the BeagleBone in this folder **…/LabCode**.

I. Board Component Locator diagram - /home/debian/LabCode/asm_U3810-66501.pdf
II. Schematic diagram - /home/debian/LabCode/U3810-66501.pdf

Transfer these to your computer using WinSCP for your use.

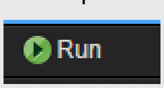
Appendix F – Cloud 9 IDE Usage

Over the RNDIS connection, the Cloud9 IDE (Integrated Development Environment) 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.

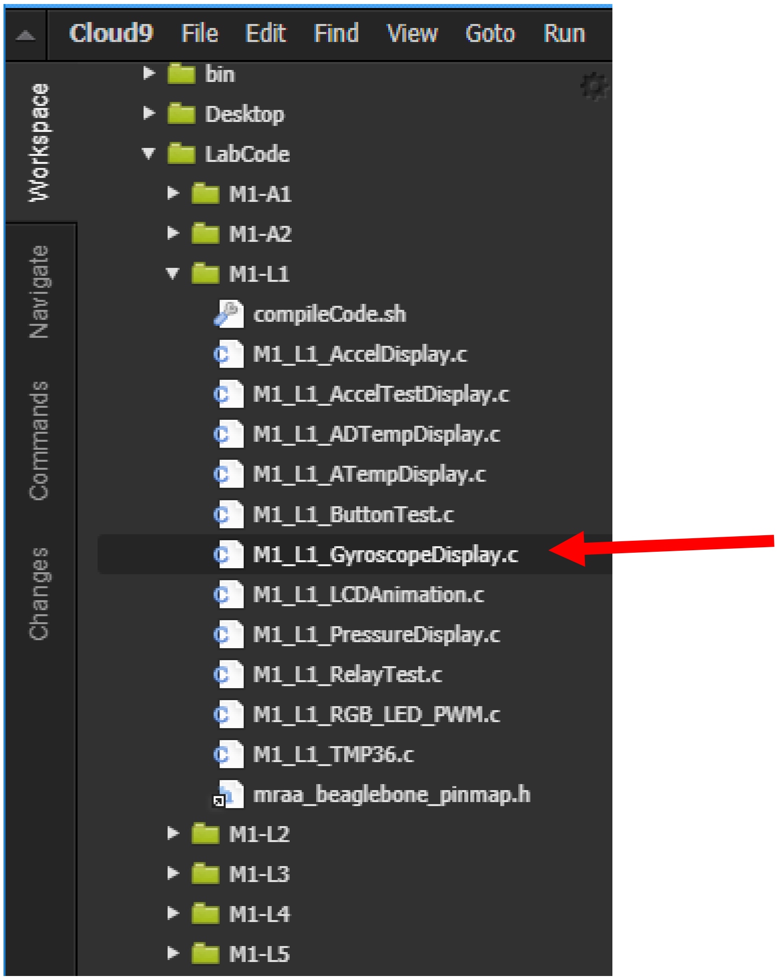


NOTE

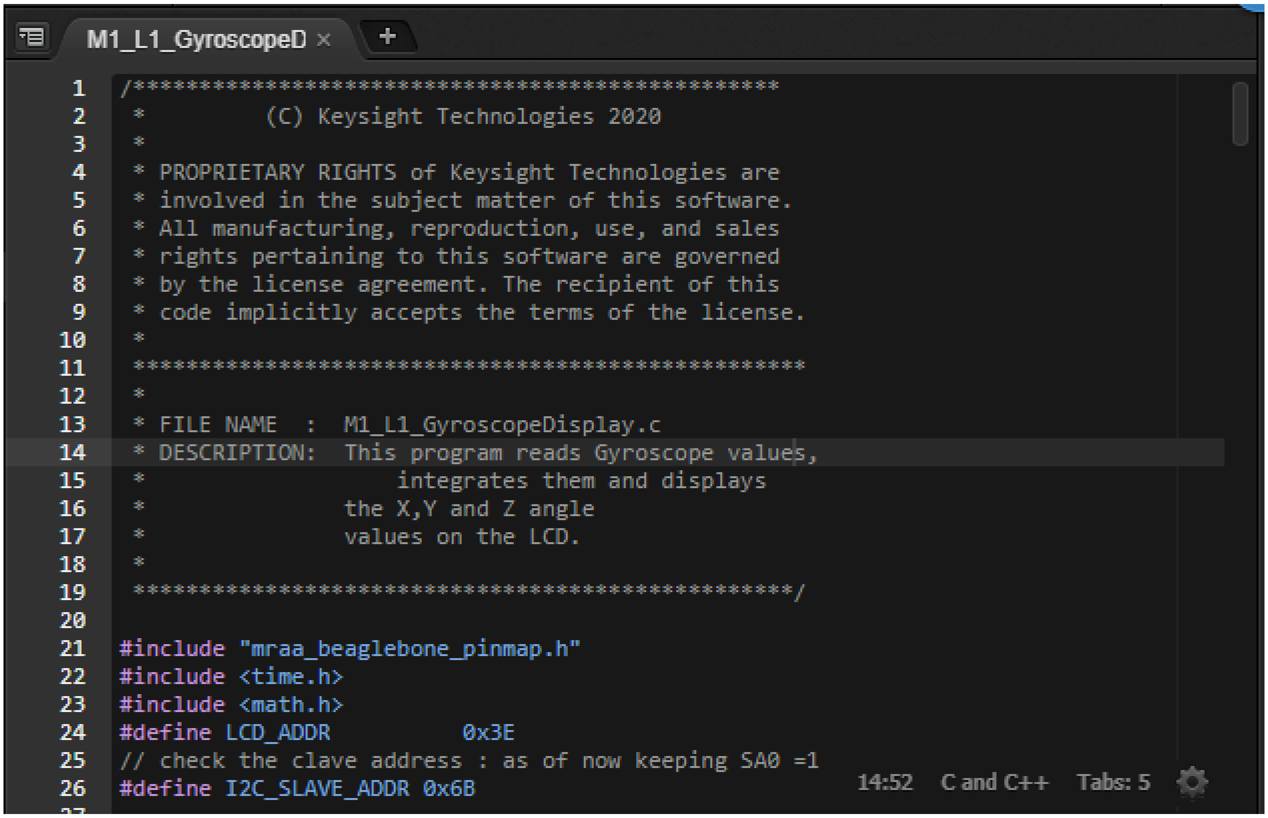
At the present time only .js and .php files run using the debugger mode.



1. Open a file in the editor and double-click the **M2_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.

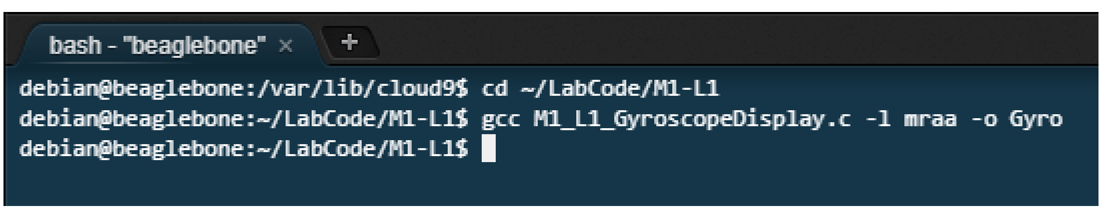


2. Run the following command in the console window to change to the directory that you are working in.

   `cd ~/LabCode/M2-L1`

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

   `gcc M2_L1_GyroscopeDisplay.c -l mraa -o Gyro`

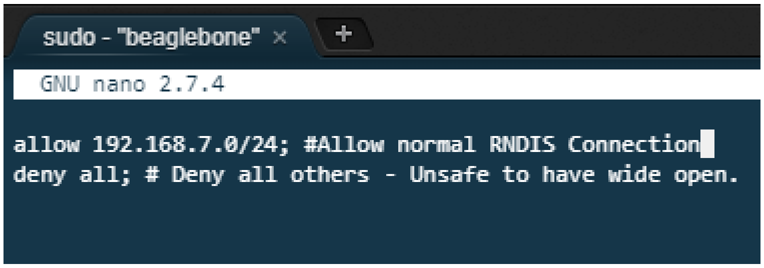


4. Enter **`./Gyro`** to run the code.

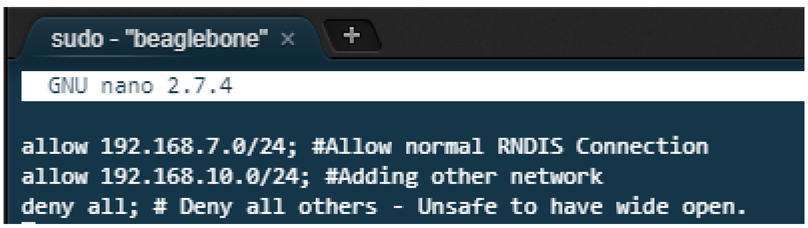
The Cloud9 IDE is secured so that it can only be accessed via the RNDIS port on 192.168.7.2.

5. In order to enable other network access, it can be opened using the **`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 elevated permissions.

       `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.

7. To see the available SSID's type the **`services`** command.

```
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
```

8. Select the SSID key that is needed, type in **`connect`** and the selected key. For example:

   `connect wifi_1234567890_managed_psk`

   Note on Windows select the key and right click. On Linux and MAC, you may use center click. Enter 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
```

NOTE

The WLAN network id can be copy and pasted by using the mouse to highlight the section. On a Windows/PuTTY system Right Click the mouse to paste, and on a Linux system center click.

9. Type **Ctrl + C** to exit **connmanctl** and verify your connection with **ping** by entering **`ping www.keysight.com`** in PuTTY. You may stop the ping process by pressing **Ctrl** + **C** on the keyboard.

```
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:/$
```

NOTE

In case you run into the following problem while setting up WLAN, e.g.

```
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 this:

```
connmanctl> tether wifi disable
Disabled tethering for wifi
connmanctl> enable wifi
Error wifi: Already enabled
connmanctl> scan wifi
Scan completed for wifi
```

Appendix G - Troubleshooting *Bluetooth*® and Wi-Fi

A way to enable or disable bluetooth or wifi 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 wifi and bluetooth are enabled. If you disable (**block**) you must re-enable (**unblock**) before powering down or rebooting Linux.

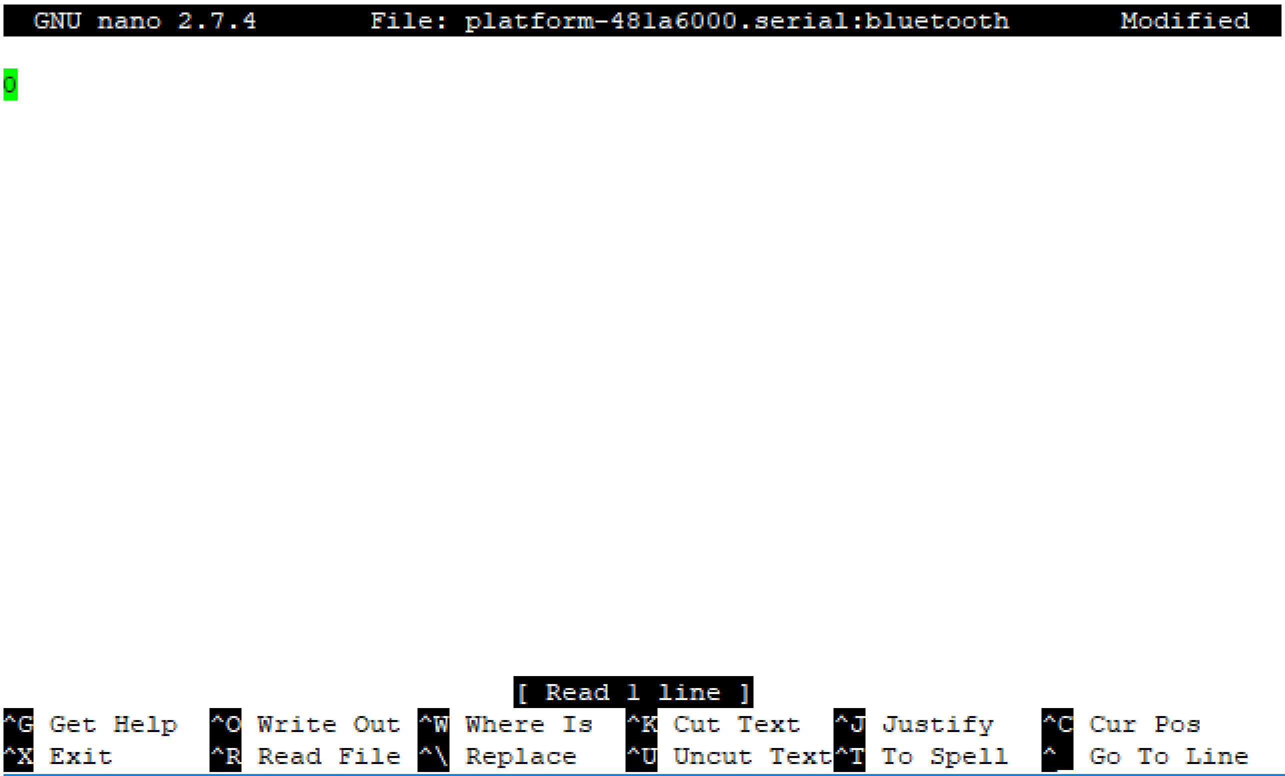
Bluetooth® Disabled

An improper sequence may disable bluetooth in such a way that **unblock** will not re-enable it. If bluetooth cannot be enabled.

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:~$
```

2. 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[4] 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.

3. 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
```

4. 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[4] 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**.

Edition 1, June 2020

Published in Malaysia

www.keysight.com