GAP–Peripheral Demo User’s Guide

GAP_Peripheral_Demo_User's_Guide

Contents

GAP_Peripheral_Demo_User's_Guide

1. Overview

This document explains how to setup a GAP-Peripheral demo using SAML21 Xplained PRO, BM71-XPro, BNO055 Xplained PRO board and Smart phone. This document briefly talks about setting up hardware, building application, programming configuration file, programming firmware, and running a demo.

This demo application showcases a Proof-of-Concept example of using **ble_host_sdk** to setup BM71 as GAP-Peripheral. This demo application uses custom GATT service to share accelerometer and gyroscope sensor data with remote device.

The following table provides the list of supported BLE services and Characteristics in this application.

Name	UUID	Properties	Size (bytes)
Device Orientation Service (Custom)	0xF05ABAC1393611E587A60002A5D5C51B	-	-
Accelerometer Position Characteristic	0x1BC5D5A50200A687E5113639D7BA5AF0	Notify, Read	6
Gyroscope Position Characteristic	0x1BC5D5A50200A687E5113639D4BA5AF0	Notify, Read	6

2. Configuring and flashing the service, characteristics, and descriptors to BM71 XPRO board

It is necessary to configure the BM71 XPRO board by making changes to the configuration file and flash the modified changes to BM71 XPRO board based on the peripheral application flow. The configuration changes demand the module to configure in Manual mode and add GATT service table.

By default, the BM71 XPro board is configured to operate in Auto mode. The Microchip Studio project, however, requires the module to set up in Manual mode.

The following instructions show how to setup and configure the module to add configuration changes.

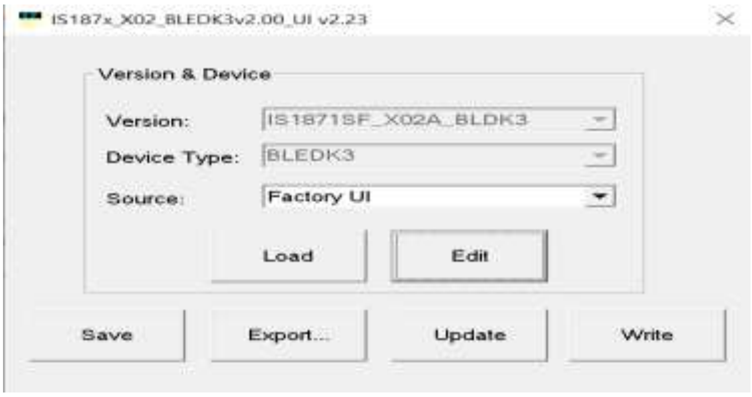
1. Connect the BM71 XPro directly to the PC using the MicroUSB on the board. The BM71 XPro board should enumerate a COM port. If not, check if the necessary MCP2200 drivers have been installed.

2. Set up the module to programming mode by configuring the Switch 1 in 3-pin DIP switch to ON state. The switch#1 sets the mode of operation on the module (between application mode and flash write mode). Refer to details on pin P2_0 in the BM70 datasheet for more details. The Blue LED (labeled BT_ACT, LD4) should be solid BLUE now. If not, check the following: a. Press 'Reset' button on the board.
NOTE: Make sure the jumper on J2 is set to USB.

3. Make sure that the BM71 module does have the correct BM71 firmware installed. By default, they should be. However, if you have programmed the module to be RN4871, change the firmware back to BM71. If the module has RN871 firmware, the module will not operate as expected and the Studio project will fail.

4. Open the UI tool for the BM70/71 modules. This tool is available for download from the BM70/71 webpage under the 'Software libraries/firmware' section.

5. Open the UI tool:



IS187x_X02_BLEDK3v2.00_UI v2.23

Version & Device

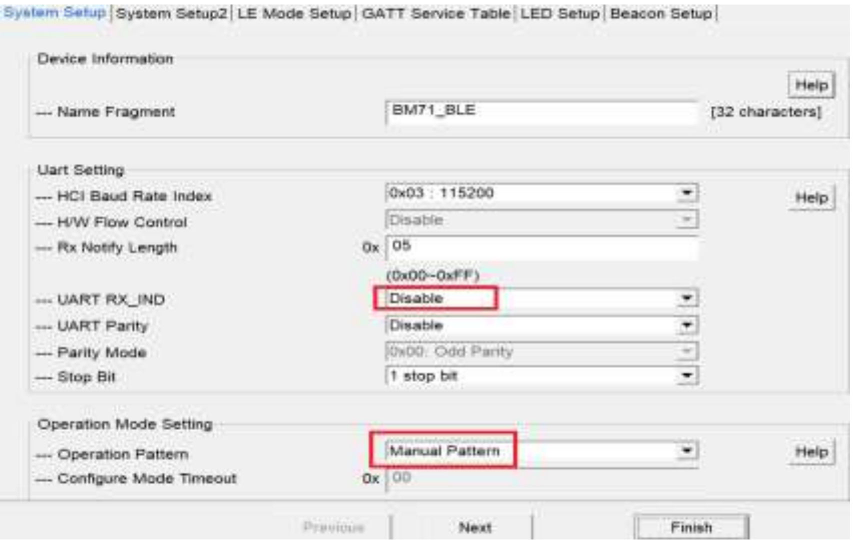
Version: IS1871SF_X02A_BLDK3

Device Type: BLEDK3

Source: Factory UI

a. In the example below, the 'BM71 default table' is being used as the base file. Click on 'Edit' to start editing the memory parameters.

b. The following changes are made in the first 'System Setup' tab:
   i. Disable the low power operation.
   ii. Change the operation mode to 'Manual pattern.'



Device Information
--- Name Fragment BM71_BLE [32 characters]

Uart Setting
--- HCI Baud Rate Index 0x03 : 115200
--- H/W Flow Control Disable
--- Rx Notify Length 0x 05
(0x00~0xFF)
--- UART RX_IND Disable
--- UART Parity Disable
--- Parity Mode 0x00: Odd Parity
--- Stop Bit 1 stop bit

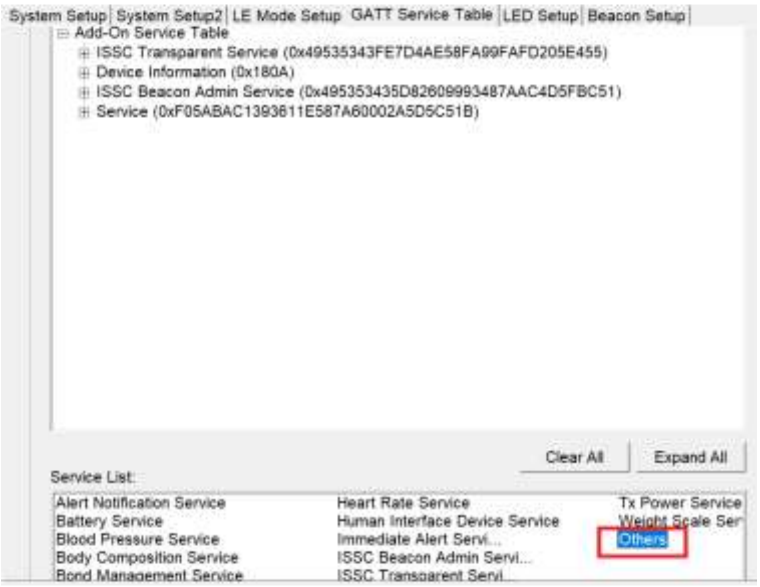
Operation Mode Setting
--- Operation Pattern Manual Pattern
--- Configure Mode Timeout 0x 00

c. In the 'GATT Service Table' tab, you will need to enter the private service/characteristics for the project.

The following table provides the list of supported BLE services and Characteristics in this application.

Name	UUID	Properties	Size (bytes)
Device Orientation Service (Custom)	0xF05ABAC1393611E587A60002A5D5C51B	-	-
Accelerometer Position Characteristic	0x1BC5D5A50200A687E5113639D7BA5AF0	Notify, Read	6
Gyroscope Position Characteristic	0x1BC5D5A50200A687E5113639D4BA5AF0	Notify, Read	6

To create the custom GATT table using the UI Tool, select the “others” option in the ‘Service List’.

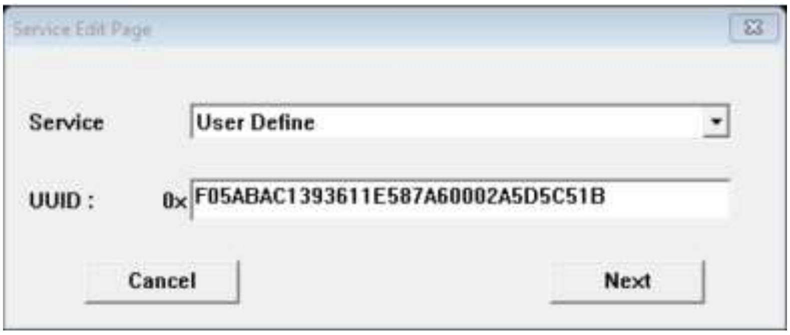


Add-On Service Table
- ISSC Transparent Service (0x49535343FE7D4AE58FA99FAFD205E455)
- Device Information (0x180A)
- ISSC Beacon Admin Service (0x495353435D82609993487AAC4D5FBC51)
- Service (0xF05ABAC1393611E587A60002A5D5C51B)

Service List:

Alert Notification Service
Battery Service
Blood Pressure Service
Body Composition Service
Bond Management Service
Heart Rate Service
Human Interface Device Service
Immediate Alert Servi...
ISSC Beacon Admin Servi...
ISSC Transparent Servi...
Tx Power Service
Weight Scale Ser
Others

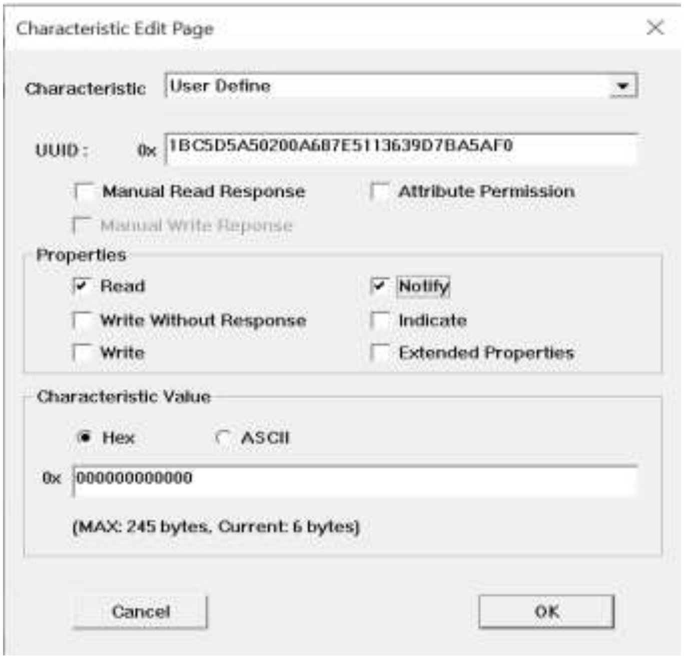
i. The ‘Others’ is the option that allows you to create a User-defined service and characteristics. **Drag this text into the ‘Add-On Service Table’ field** above. You will get a dialog box to create the private service. Enter the Private service UUID.



Service: User Define

UUID : 0x F05ABAC1393611E587A60002A5D5C51B

ii. Click 'Next' and then enter the private characteristic under this service.



Characteristic Edit Page

Characteristic: User Define

UUID : 0x 1BC5D5A50200A687E5113639D7BA5AF0

□ Manual Read Response □ Attribute Permission

□ Manual Write Reponse

Properties

☑ Read ☑ Notify

□ Write Without Response □ Indicate

□ Write □ Extended Properties

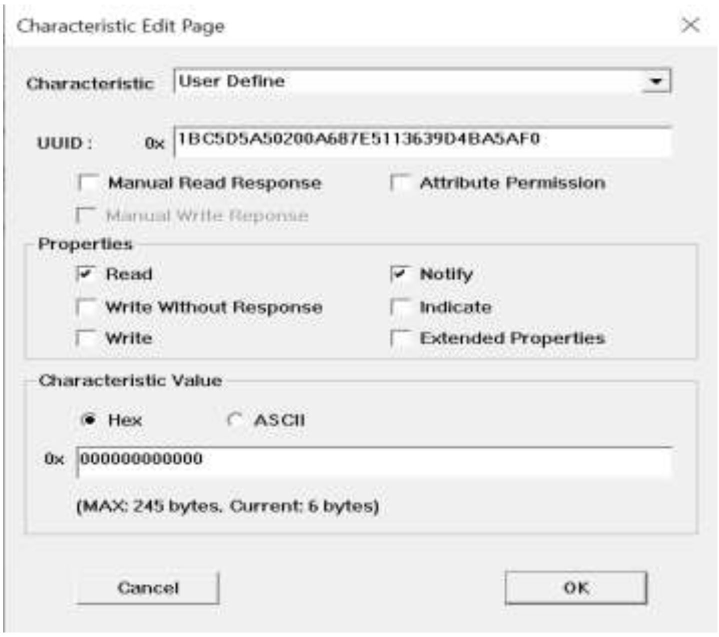
Characteristic Value

● Hex ○ ASCII

0x 000000000000

(MAX: 245 bytes, Current: 6 bytes)

iii. In the newly created private service in the 'Add-On service' field, right click on the private service to 'Add Characteristic' the second private characteristic. Enter the details of the second private characteristic.



Characteristic Edit Page

Characteristic: User Define

UUID : 0x 1BC5D5A50200A687E5113639D4BA5AF0

□ Manual Read Response □ Attribute Permission

□ Manual Write Reponse

Properties

☑ Read ☑ Notify

□ Write Without Response □ Indicate

□ Write □ Extended Properties

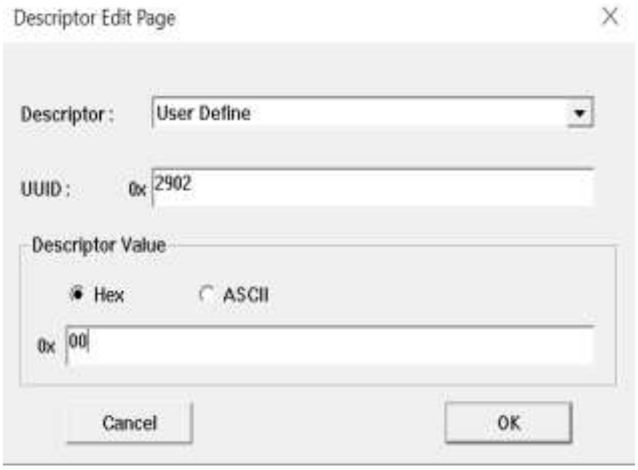
Characteristic Value

● Hex ○ ASCII

0x 000000000000

(MAX: 245 bytes, Current: 6 bytes)

iv. Right click on the first Characteristic and select 'Add Descriptor' and enter the UUID as 2902


Descriptor Edit Page

Descriptor : User Define

UUID : 0x 2902

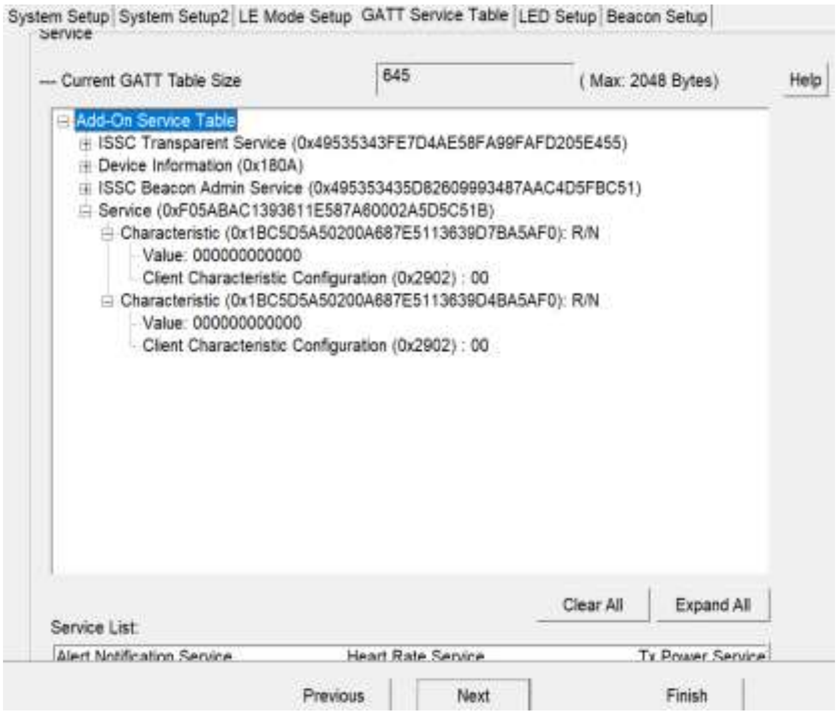
Descriptor Value

● Hex ○ ASCII

0x 00

v. Similarly add the descriptor for the second characteristic as well with the same UUID value(2902)

vi. The final 'Add-On Service Table' should look like below.



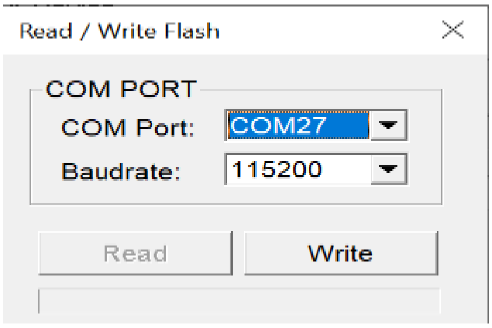
Current GATT Table Size 645 (Max: 2048 Bytes)

```
Add-On Service Table
  ISSC Transparent Service (0x49535343FE7D4AE58FA99FAFD205E455)
  Device Information (0x180A)
  ISSC Beacon Admin Service (0x495353435D82609993487AAC4D5FBC51)
  Service (0xF05ABAC1393611E587A60002A5D5C51B)
    Characteristic (0x1BC5D5A50200A687E5113639D7BA5AF0): R/N
      Value: 000000000000
      Client Characteristic Configuration (0x2902) : 00
    Characteristic (0x1BC5D5A50200A687E5113639D4BA5AF0): R/N
      Value: 000000000000
      Client Characteristic Configuration (0x2902) : 00
```

vii. Click on 'Finish'. All the parameter required are now set and ready to flash to device.

viii. Click on 'Write' option in UI tool.

ix. Select the COM port corresponding to the BM71 XPRO board. Make sure the Blue LED is still Solid ON to make sure the module is 'Write Flash' mode.



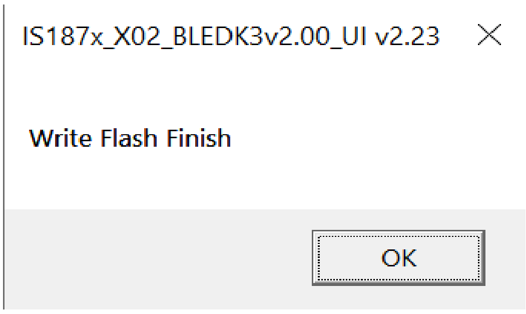
Read / Write Flash

COM PORT

COM Port: COM27

Baudrate: 115200

x. Click on 'Write'. After the confirmation acceptance the configuration file will be written into the device memory.



IS187x_X02_BLEDK3v2.00_UI v2.23

Write Flash Finish

xi. Now, that the Flash write operation is done, the module needs to be set back to 'Application mode': Set the Switch#1 in the DIP switch to OFF and change the J2 jumper.

3. Hardware Setup

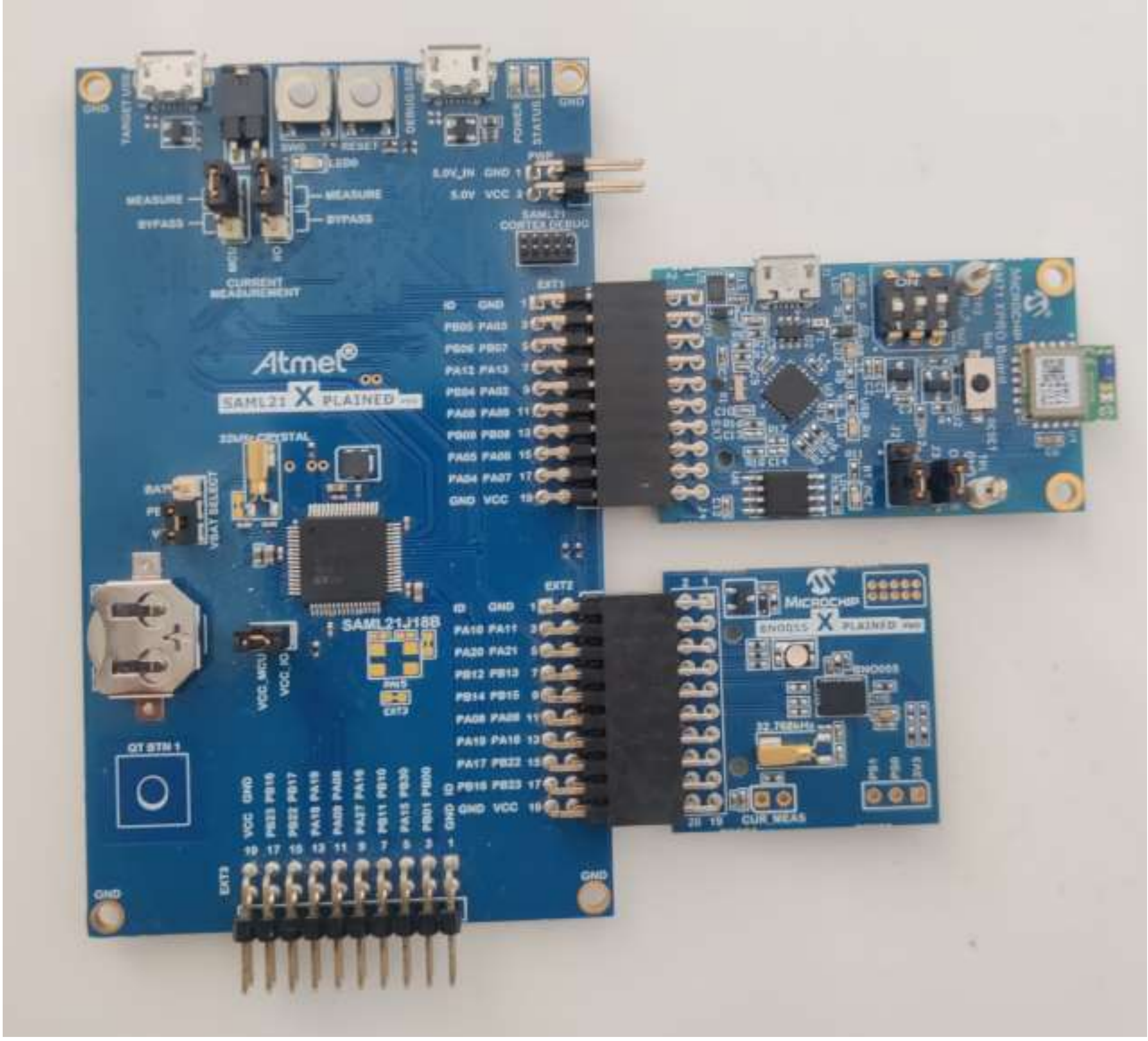


Figure 1: SAML21 Xplained Pro with BM71-XPRO and BNO055 Xplained PRO Board

1. Plugin the BM71-XPRO board into EXT1 of SAML21 Xplained Pro board as shown in Figure 1.
2. BNO055 Xplained PRO Board is used in this demo. Plug-in the BNO055 board into the EXT2 of SAML21 Xplained Pro as shown in Figure 1.
3. Connect the SAML21 Xplained Pro board to the host PC using micro USB cable.

3.1 Console

The GAP-Peripheral demo application uses the Universal Asynchronous Receiver/Transmitter (UART) interface on SAML21 Xplained Pro to send the status messages like Advertising, Connected, Disconnected … etc. Any serial application (ex: TeraTerm) can be used to interact with SAML21 Xplained Pro.

UART (COM port) settings,

Baud rate	115200
Data	8 bits
Parity	none
Stop	1 bit
Flow control	none

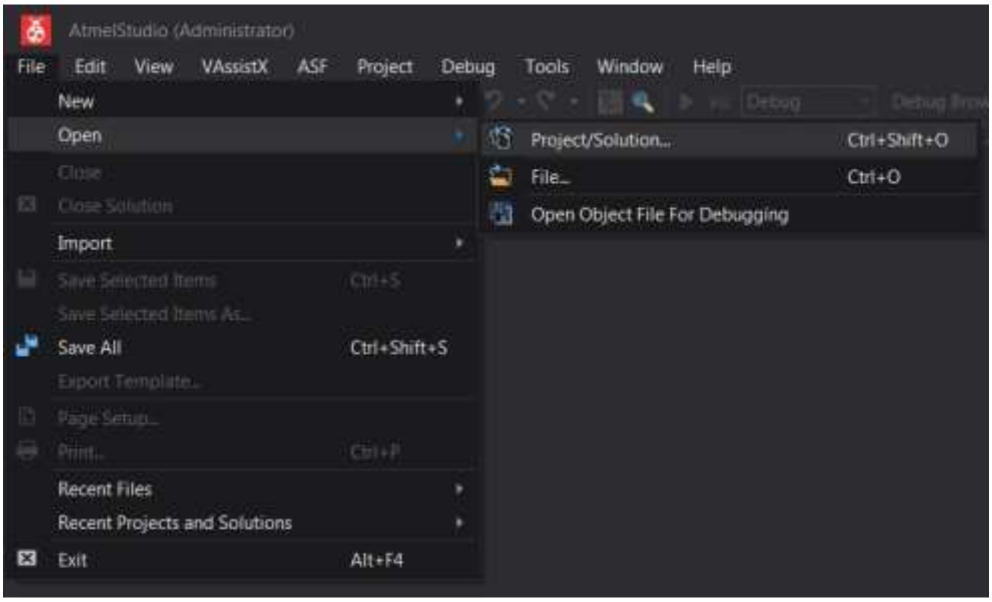
4. Build Procedure

This section describes build procedure of GAP-Peripheral demo application on Microhip Studio 7.

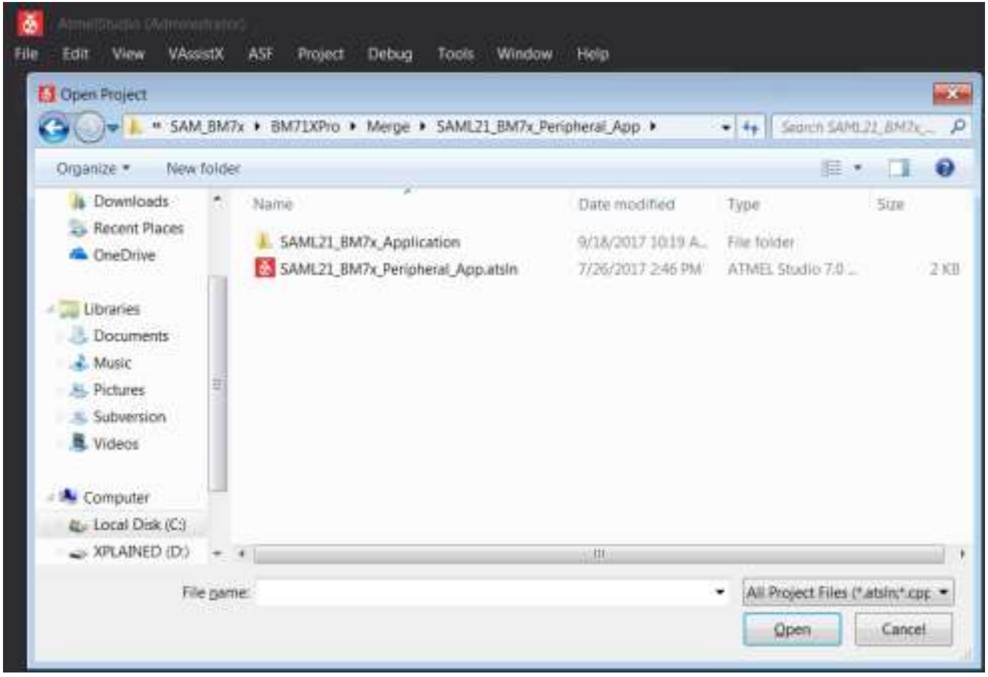
4.1. Open Microchip Studio 7

4.2. Open GAP-Peripheral Demo Application

1. Go to menu **File → Open → Project/Solution.**



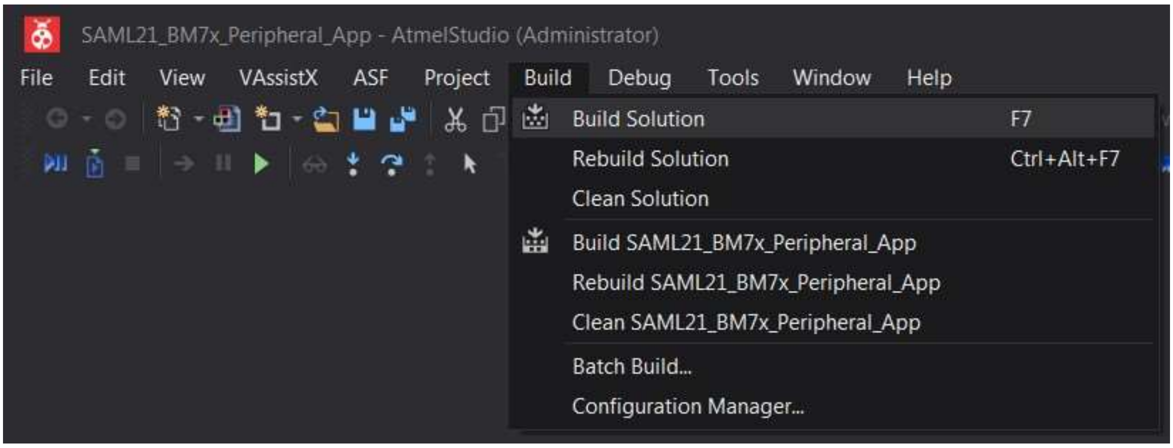
2. Select "SAML21_BM7x_Peripheral_App.atsln" and press **Open**.



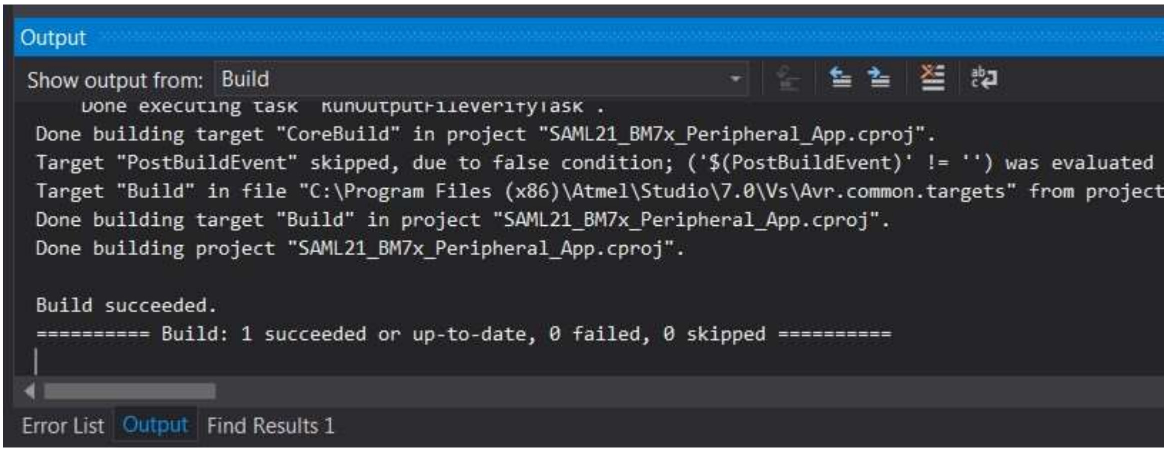
3. Once the project is opened, you can see the files attached to this project in Solution Explorer Window

4.3. Build GAP-Peripheral Demo Application

1. Go to menu **Build → Build Solution** or Press **F7**


SAML21_BM7x_Peripheral_App - AtmelStudio (Administrator)
File
Edit
View
VAssistX
ASF
Project
Build
Debug
Tools
Window
Help
Build Solution F7
Rebuild Solution Ctrl+Alt+F7
Clean Solution
Build SAML21_BM7x_Peripheral_App
Rebuild SAML21_BM7x_Peripheral_App
Clean SAML21_BM7x_Peripheral_App
Batch Build...
Configuration Manager...

2. Build status can be checked in **Output** window.



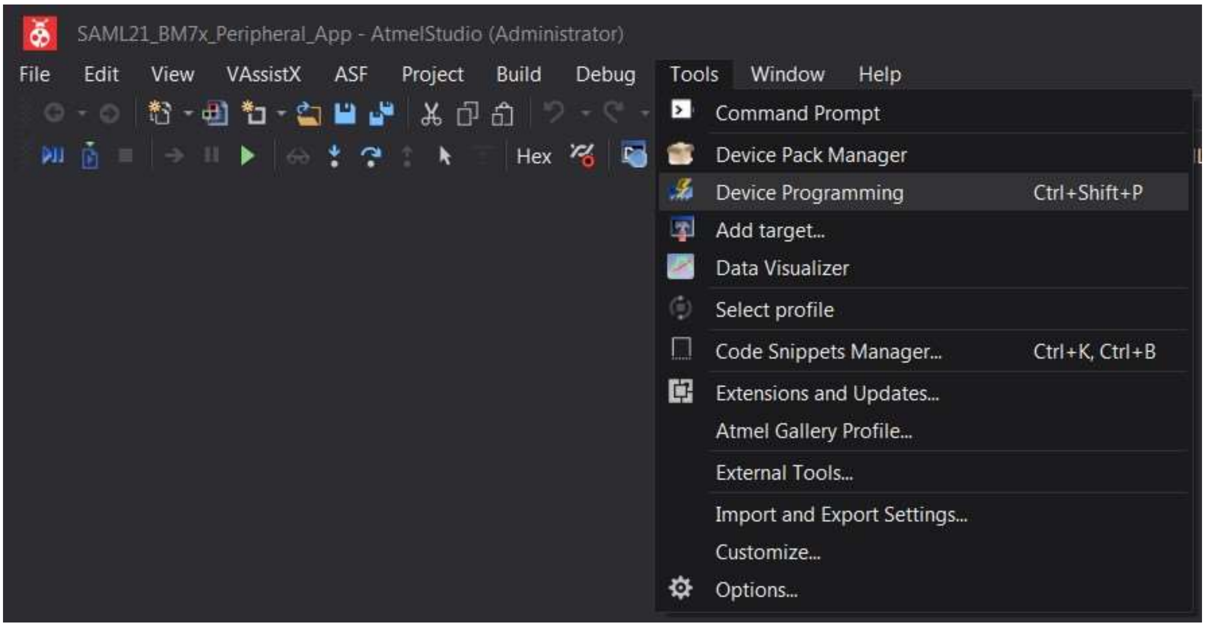
```
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "SAML21_BM7x_Peripheral_App.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project
Done building target "Build" in project "SAML21_BM7x_Peripheral_App.cproj".
Done building project "SAML21_BM7x_Peripheral_App.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

3. You can find the Hex images in "..\SAML21_BM7x_Application\Debug".

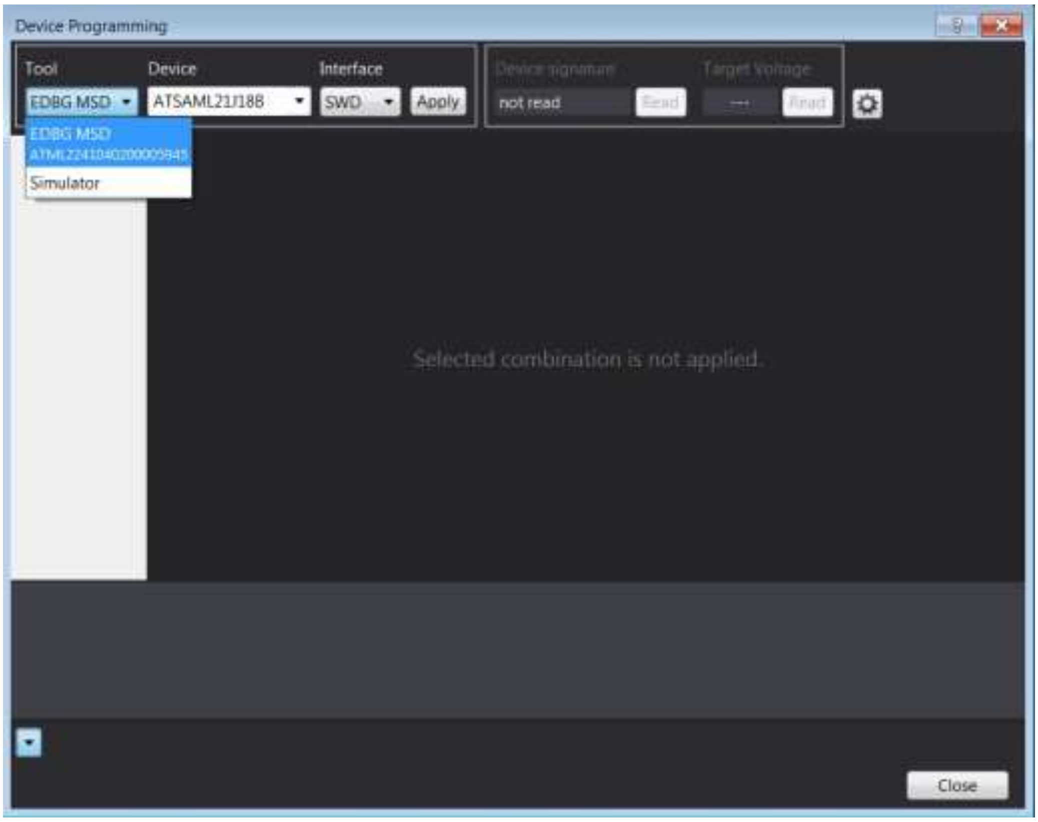
5. Programming Firmware

This section describes the procedure to program GAP-Peripheral demo firmware on SAML21 Xplained Pro board.

1. Connect the SAML21 Xplained Pro board to the host PC using micro-USB cable. Perform the following steps:
    a. Verify that the virtual COM port is enumerated on the host PC.
    b. Make sure that POWER LED (green) is solid ON.
2. To program the HEX files into the SAML21, go to menu **Tools → Device Programming** or Press **Ctrl + Shift + P**.



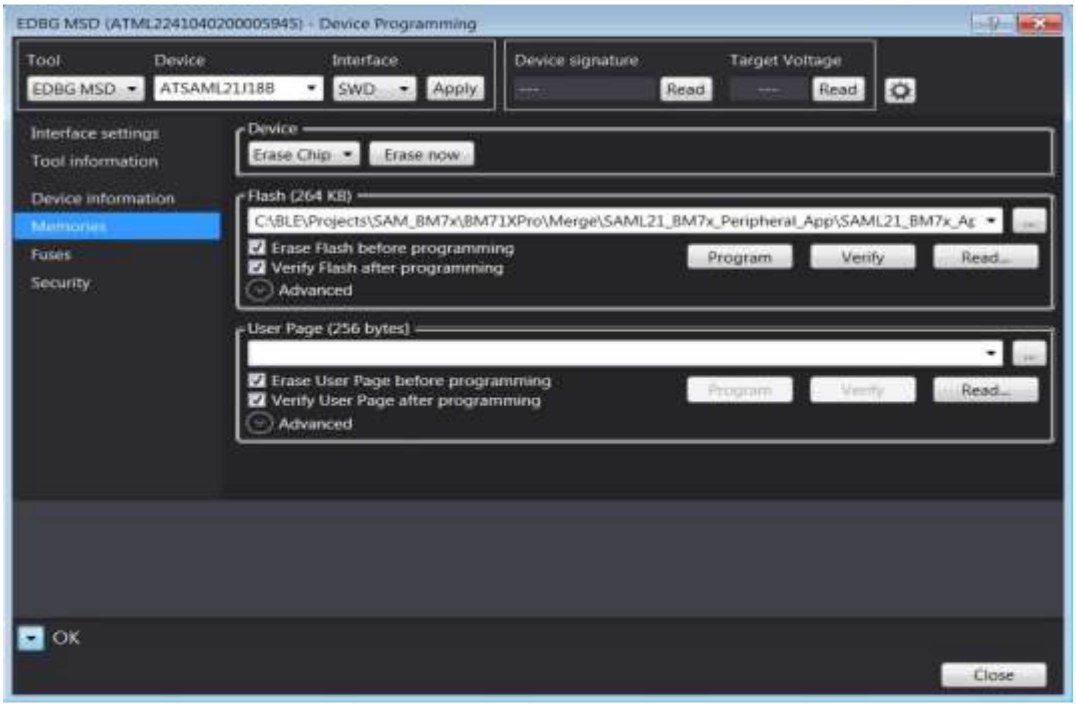
3. Select the corresponding **EDBG** and press **Apply**.



Tool: EDBG MSD
Device: ATSAML21J18B
Interface: SWD
Device signature: not read

Selected combination is not applied.

4. Go to **Memories** Tab and select Hex file.



EDBG MSD (ATML2241040200005945) - Device Programming

Tool: EDBG MSD
Device: ATSAML21J18B
Interface: SWD
Device signature: ----
Target Voltage: ---

Flash (264 KB): C:\BLE\Projects\SAM_BM7x\BM71XPro\Merge\SAML21_BM7x_Peripheral_App\SAML21_BM7x_Ap

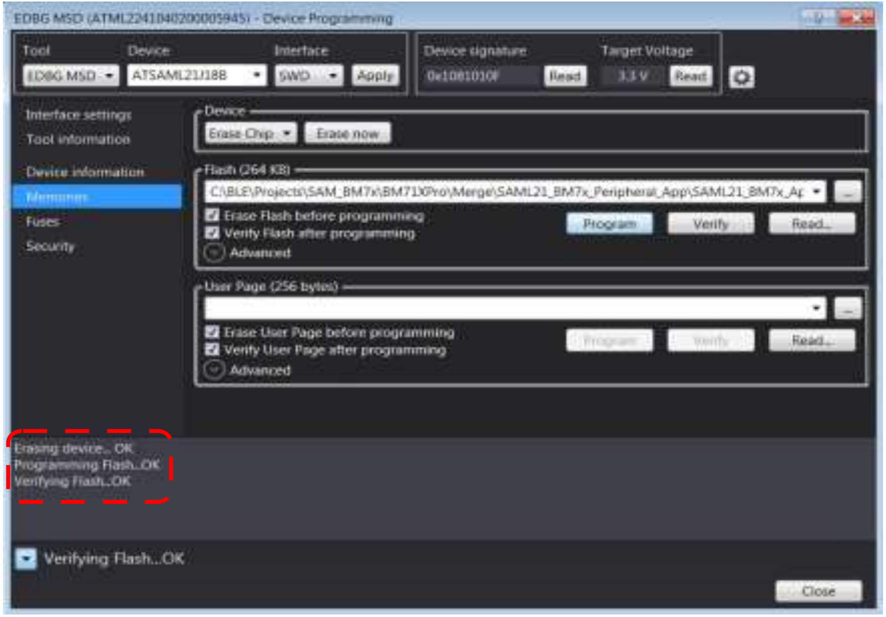
☑ Erase Flash before programming
☑ Verify Flash after programming

User Page (256 bytes)

☑ Erase User Page before programming
☑ Verify User Page after programming

OK

5. Press **Program**, the tool will program SAML21. You can check the status of programming.



EDBG MSD (ATML2241040200005945) - Device Programming

Tool: EDBG MSD
Device: ATSAML21J18B
Interface: SWD
Device signature: 0x1081010F
Target Voltage: 3.3 V

Flash (264 KB): C:\BLE\Projects\SAM_BM7x\BM71XPro\Merge\SAML21_BM7x_Peripheral_App\SAML21_BM7x_Ap

☑ Erase Flash before programming
☑ Verify Flash after programming

User Page (256 bytes)

☑ Erase User Page before programming
☑ Verify User Page after programming

```
Erasing device... OK
Programming Flash...OK
Verifying Flash...OK
```

Verifying Flash...OK

6. Once programming is done, close the Device Programming window.

6. Running GAP-Peripheral Demo with Microchip Bluetooth Data app

This section describes the GAP-Peripheral Demo procedures to work with Android Microchip Bluetooth Data app.

NOTE: There are differences in the response packet send by the BM71 firmware for the Read_Local_Specific_Primary_Service (0x3C) command in regards to different firmware version. Until the firmware Version 1.11 it follows same pattern and in V2.03 the response data packet structure changes. To make the application compactable with all the firmware version, the read back function is modified with a macro.

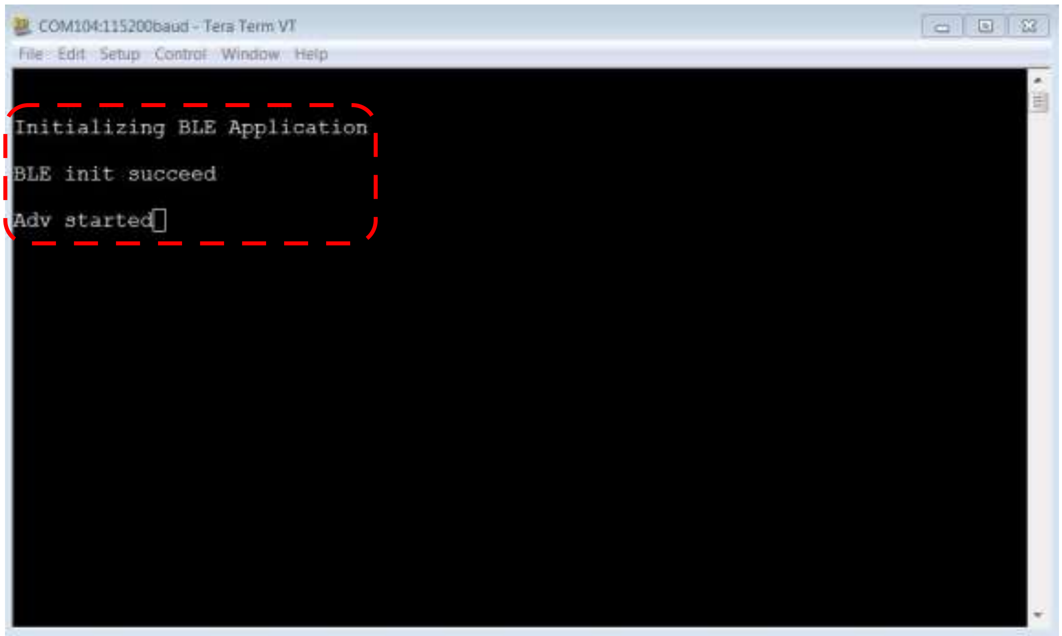
```
#define Older_Firmware_Versions
```

By default, this macro is commented in the code to make is work with the latest released firmware (2.03 at time of creation of this document). If the user is using an earlier firmware version, then it is recommended to uncomment this macro in the code.

1. Connect BM71-XPro on EXT1 of SAML21 Xplained Pro board (GAP-Peripheral device).
2. Connect the GAP-Peripheral device to the host PC using micro-USB cable. Perform the following steps:
    a. Verify that the virtual COM port is enumerated on the host PC.
    b. Open the enumerated COM port on a serial terminal application like TeraTerm with the following settings:

Baudrate	115200
Data	8 bits
Parity	none
Stop	1 bit
Flow control	none

    c. Make sure that POWER LED (green) on SAML21 Xplained Pro board is solid ON.
    d. Press Reset button on SAML21 Xplained Pro board and verify that LD4 (blue) on BM71-XPro is blinking at a regular interval.
3. Ensure that the GAP-Peripheral device is advertising by checking the TeraTerm window for a message "Adv started".



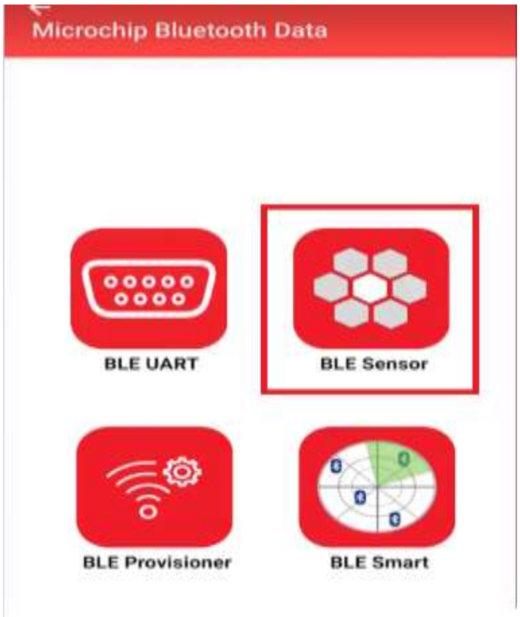
COM104:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Initializing BLE Application
BLE init succeed
Adv started
```

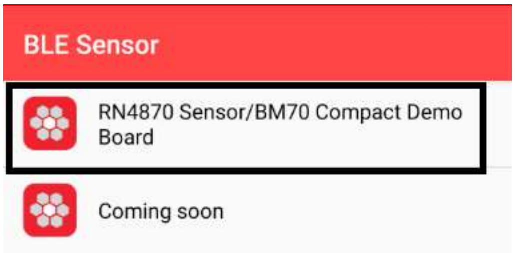
6.1. Smart Phone Application

i. Download the Microchip Bluetooth Data (MBD) mobile application from the play store.

ii. Select the BLE Sensor tab



iii. Select the RN4870 Sensor/BM70 Compact Demo Board option

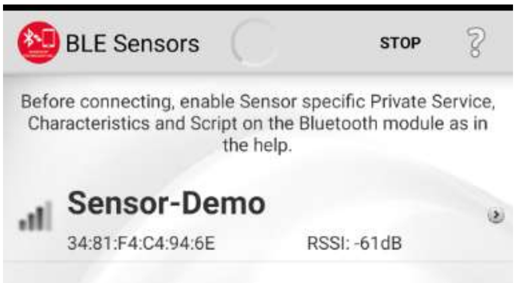


BLE Sensor

RN4870 Sensor/BM70 Compact Demo Board

Coming soon

iv. Press **SCAN** and click on Sensor-Demo for the scan list to connect to the BM71

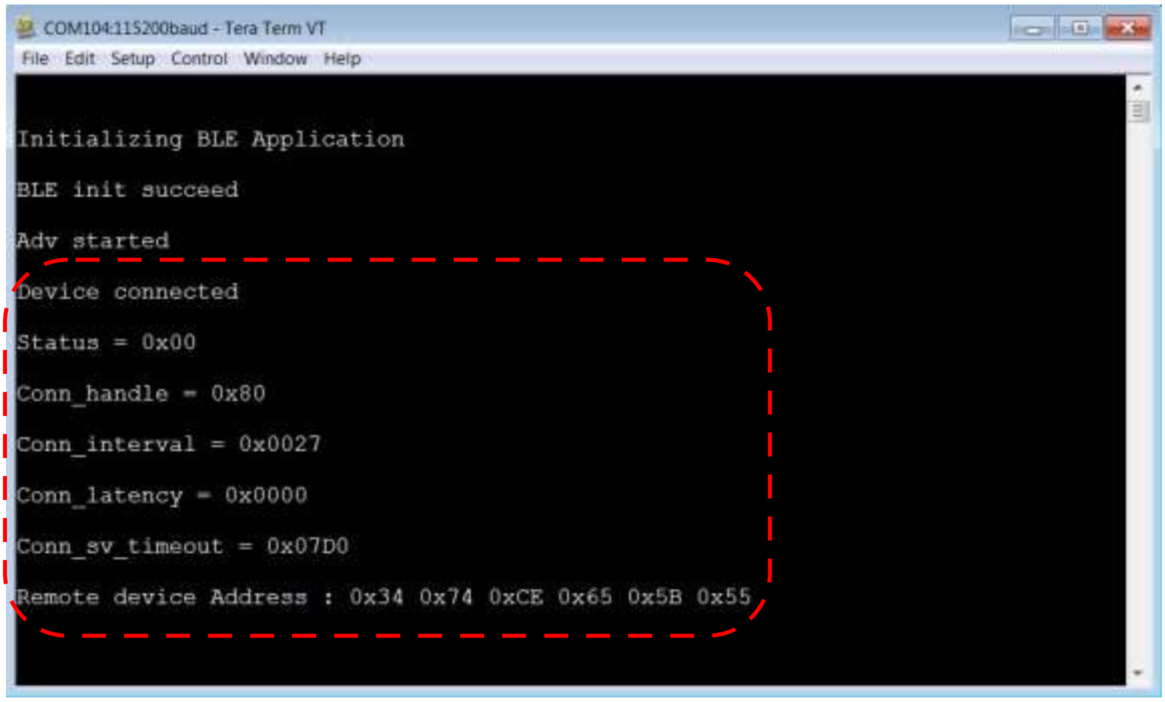


Before connecting, enable Sensor specific Private Service, Characteristics and Script on the Bluetooth module as in the help.

Sensor-Demo

34:81:F4:C4:94:6E RSSI: -61dB

v. After successful connection, GAP-Peripheral device prints the connection details in console.

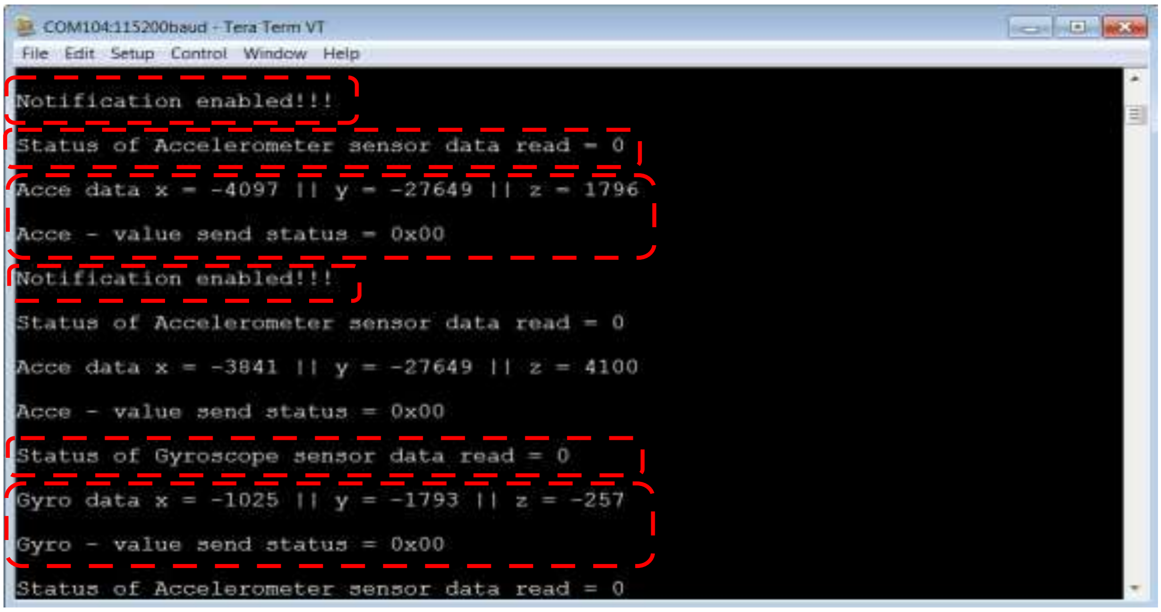


COM104:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Initializing BLE Application
BLE init succeed
Adv started
Device connected
Status = 0x00
Conn_handle = 0x80
Conn_interval = 0x0027
Conn_latency = 0x0000
Conn_sv_timeout = 0x07D0
Remote device Address : 0x34 0x74 0xCE 0x65 0x5B 0x55
```

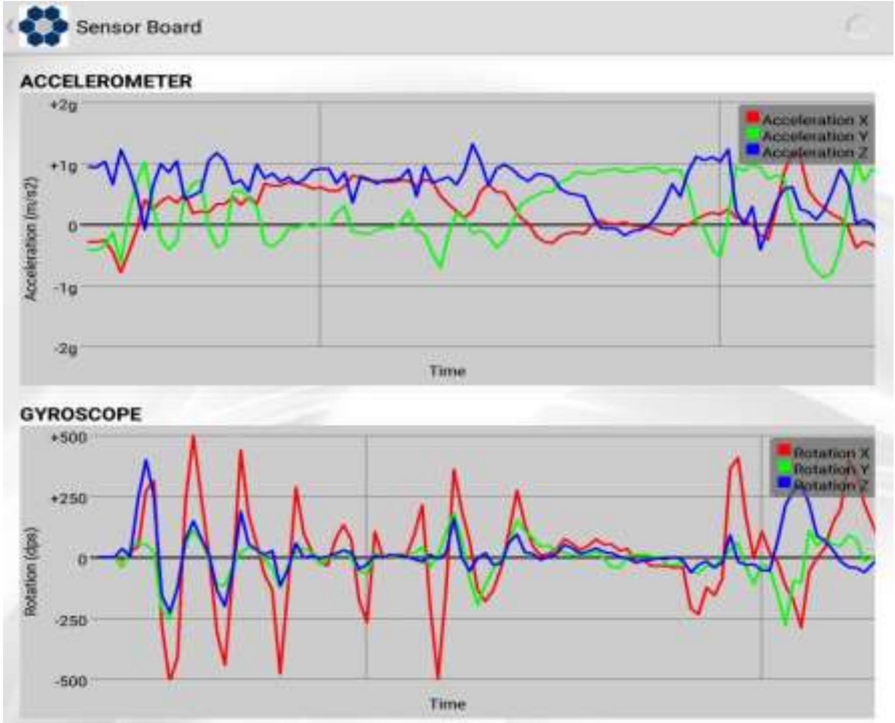
vi. Once connected, the MBD app discovers the accelerometer and gyroscope sensor characteristics and enable notifications.

vii. Once notification enabled by MBD app, GAP-Peripheral prints the accelerometer and gyroscope sensor read operation status and data on console.


COM104:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Notification enabled!!!
Status of Accelerometer sensor data read = 0
Acce data x = -4097 || y = -27649 || z = 1796
Acce - value send status = 0x00
Notification enabled!!!
Status of Accelerometer sensor data read = 0
Acce data x = -3841 || y = -27649 || z = 4100
Acce - value send status = 0x00
Status of Gyroscope sensor data read = 0
Gyro data x = -1025 || y = -1793 || z = -257
Gyro - value send status = 0x00
Status of Accelerometer sensor data read = 0
```

viii. Once connected with GAP-Peripheral (Sensor-Demo) and notifications are enabled, BLESensorApp shows the Accelerometer and Gyroscope sensor data in graph view.



7. Running GAP-Peripheral Demo with GAP-Central Device

This section describes the GAP-Peripheral Demo procedures to work with GAP-Central device.

1. Connect BM71-XPro on EXT1 of SAML21 Xplained Pro board (GAP-Peripheral device).
2. Connect the GAP-Peripheral device to the host PC using micro-USB cable. Perform the following steps:
    a. Verify that the virtual COM port is enumerated on the host PC.

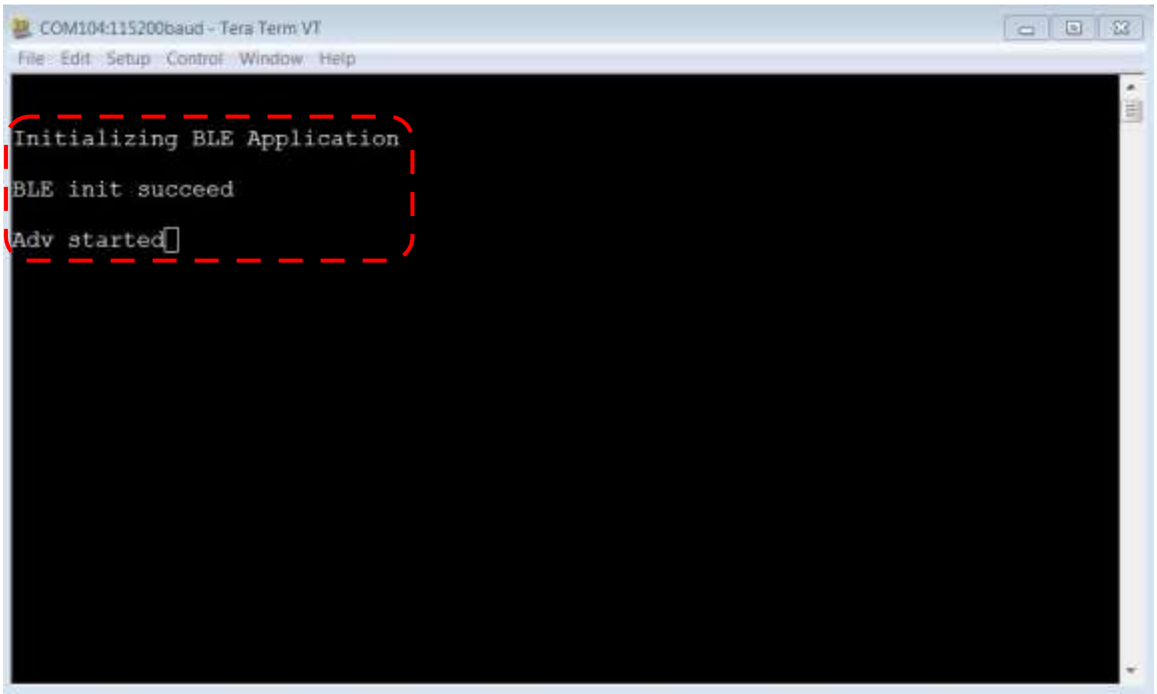
b. Open the enumerated COM port on a serial console application like TeraTerm with the following settings:

Baudrate	115200
Data	8 bits
Parity	none
Stop	1 bit
Flow control	none

c. Make sure that POWER LED (green) on SAML21 Xplained Pro board is solid ON.

d. Press Reset button on SAML21 Xplained Pro board and verify that LD4 (blue) on BM71-XPro is blinking at a regular interval.

3. Ensure that the GAP-Peripheral device is up and running and start advertising. Check status on a serial console application.


COM104:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Initializing BLE Application

BLE init succeed

Adv started
```

4. Connect BM71-XPro on EXT1 of SAML21 Xplained Pro board (GAP-Central device).

5. Connect the GAP-Central device to the host PC using micro USB cable. Perform the following steps:

   a. Verify that the virtual COM port is enumerated on the host PC.

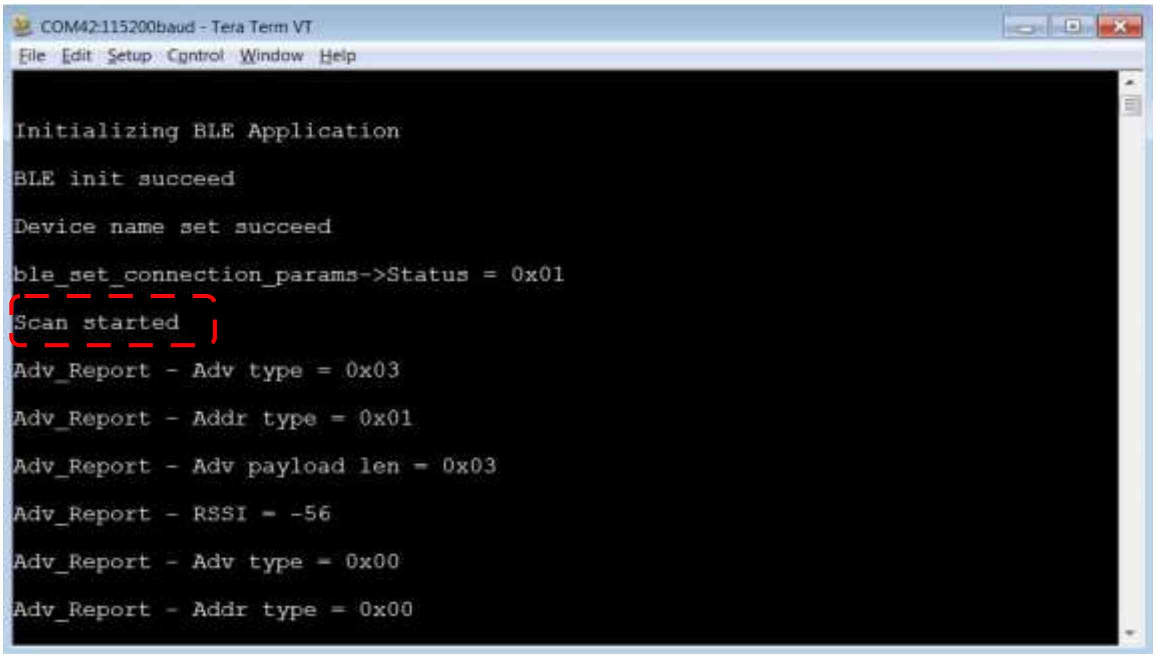
   b. Open the enumerated COM port on a serial console application like TeraTerm with the following settings:

Baudrate	115200
Data	8 bits
Parity	none
Stop	1 bit
Flow control	none

c. Make sure that POWER LED (green) on SAML21 Xplained Pro board is solid ON.

d. Press Reset button on SAML21 Xplained Pro board and verify that LD4 (blue) on BM71-XPro is blinking at a regular interval.

6. Ensure GAP-Central device is up and running and start scanning. Check status on a serial console application.

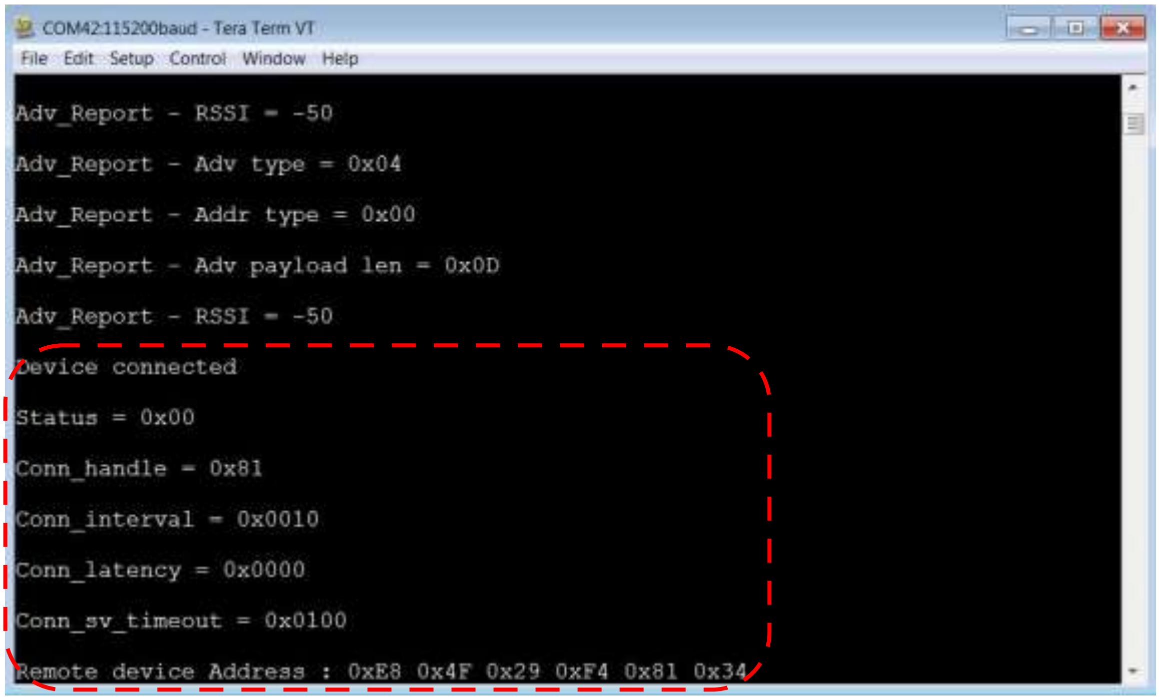

COM42:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Initializing BLE Application
BLE init succeed
Device name set succeed
ble_set_connection_params->Status = 0x01
Scan started
Adv_Report - Adv type = 0x03
Adv_Report - Addr type = 0x01
Adv_Report - Adv payload len = 0x03
Adv_Report - RSSI = -56
Adv_Report - Adv type = 0x00
Adv_Report - Addr type = 0x00
```

7. As GAP-Central device receives advertisements from devices in vicinity, it prints the advertisement report in serial console.

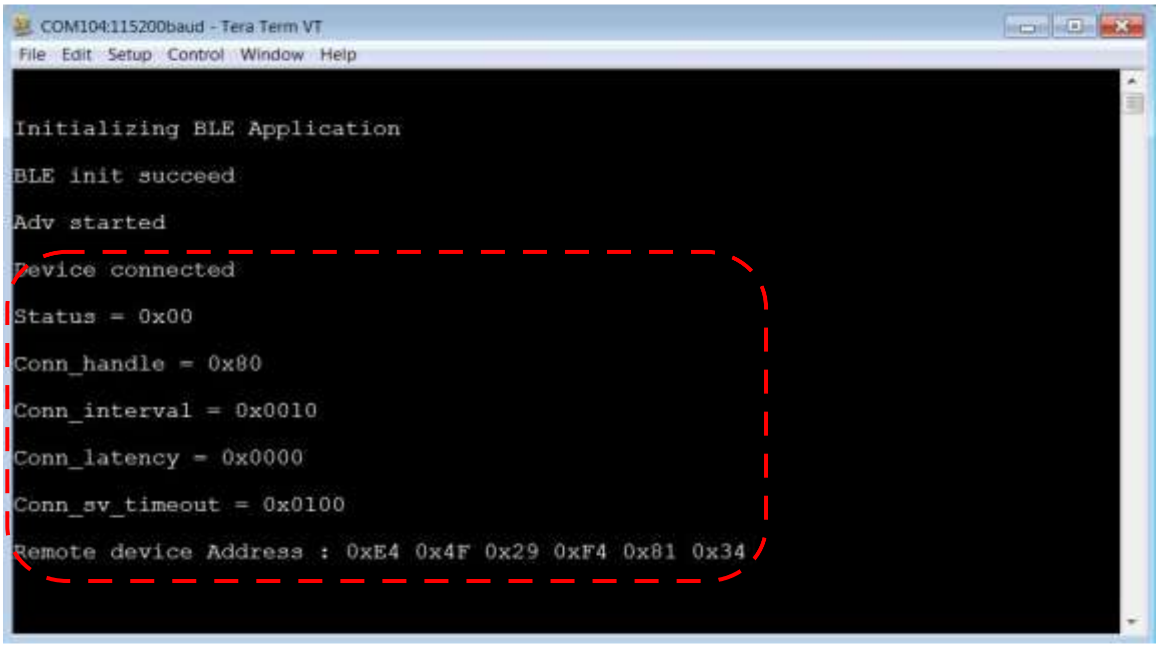
8. GAP-Central device finds the suitable GAP-Peripheral device based on the advertisement payload and initiates a connection.

9. Once connected the GAP-Central device prints the GAP-Peripheral device address and connection parameters in serial console.


COM42:115200baud - Tera Term VT
File Edit Setup Control Window Help

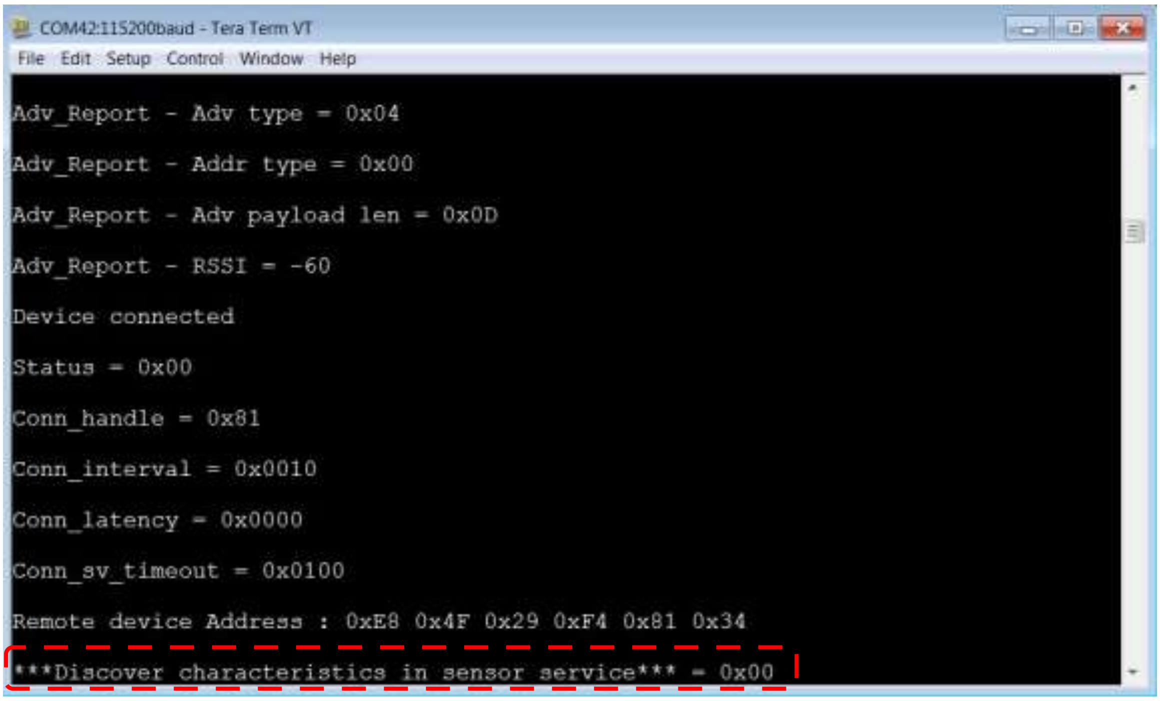
```
Adv_Report - RSSI = -50
Adv_Report - Adv type = 0x04
Adv_Report - Addr type = 0x00
Adv_Report - Adv payload len = 0x0D
Adv_Report - RSSI = -50
Device connected
Status = 0x00
Conn_handle = 0x81
Conn_interval = 0x0010
Conn_latency = 0x0000
Conn_sv_timeout = 0x0100
Remote device Address : 0xE8 0x4F 0x29 0xF4 0x81 0x34
```

10. Similarly, once connected GAP-Peripheral device prints the GAP-Central device address and connection parameters in serial console.


COM104:115200baud - Tera Term VT
File Edit Setup Control Window Help

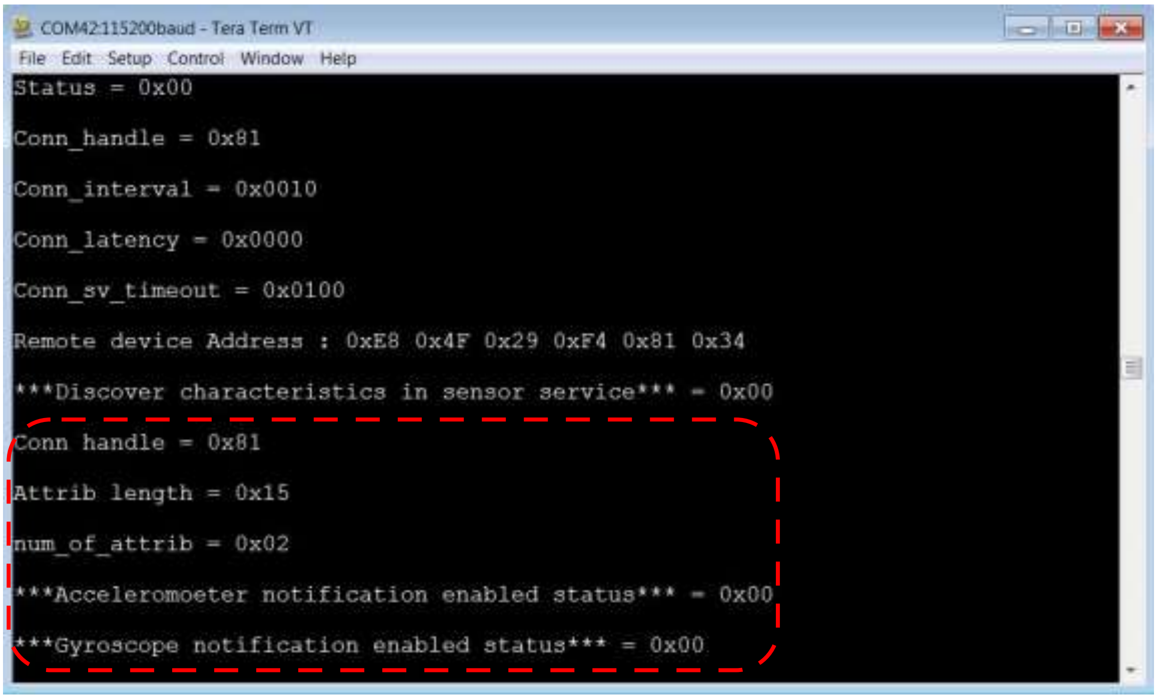
```
Initializing BLE Application
BLE init succeed
Adv started
Device connected
Status = 0x00
Conn_handle = 0x80
Conn_interval = 0x0010
Conn_latency = 0x0000
Conn_sv_timeout = 0x0100
Remote device Address : 0xE4 0x4F 0x29 0xF4 0x81 0x34
```

11. GAP-Central discovers the characteristics in remote device based on the service UUID.


COM42:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Adv_Report - Adv type = 0x04
Adv_Report - Addr type = 0x00
Adv_Report - Adv payload len = 0x0D
Adv_Report - RSSI = -60
Device connected
Status = 0x00
Conn_handle = 0x81
Conn_interval = 0x0010
Conn_latency = 0x0000
Conn_sv_timeout = 0x0100
Remote device Address : 0xE8 0x4F 0x29 0xF4 0x81 0x34
***Discover characteristics in sensor service*** = 0x00
```

12. It also discovers the Client Characteristic Configuration Descriptors (CCCD) of accelerometer and gyroscope sensor characteristics and enabled them to receive notifications.

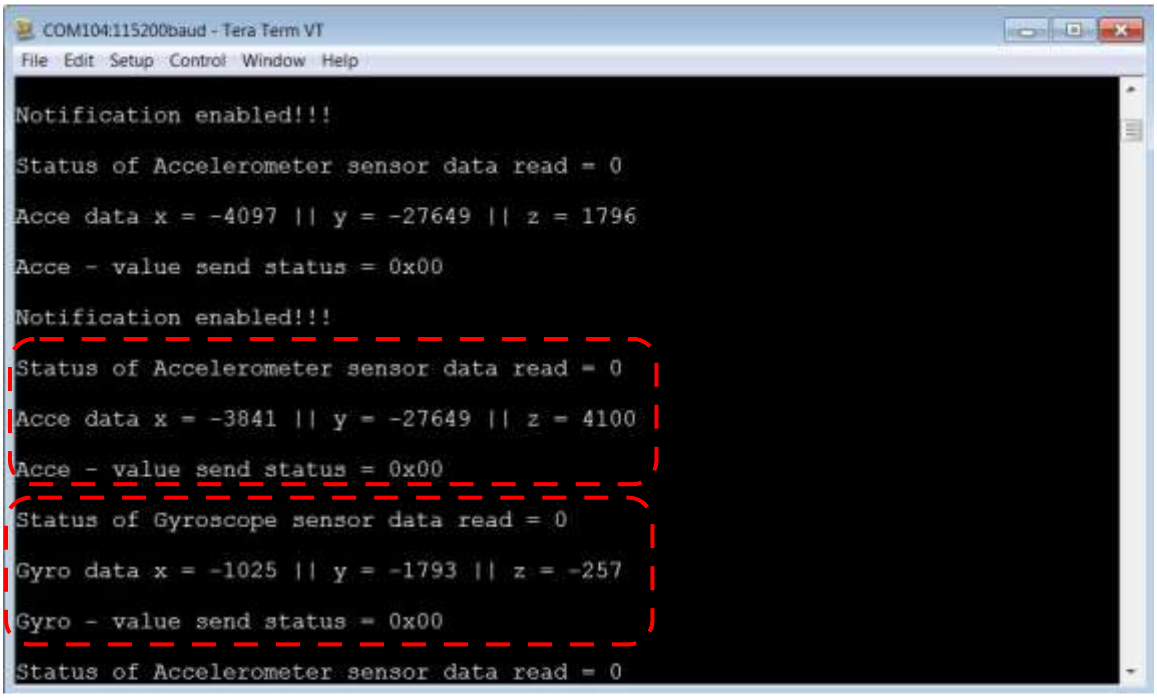

COM42:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Status = 0x00
Conn_handle = 0x81
Conn_interval = 0x0010
Conn_latency = 0x0000
Conn_sv_timeout = 0x0100
Remote device Address : 0xE8 0x4F 0x29 0xF4 0x81 0x34
***Discover characteristics in sensor service*** = 0x00
Conn handle = 0x81
Attrib length = 0x15
num_of_attrib = 0x02
***Accelerometer notification enabled status*** = 0x00
***Gyroscope notification enabled status*** = 0x00
```

13. Upon notification enabled by GAP-Central device, GAP-Peripheral device start reading accelerometer and gyroscope sensor data and notifies it to GAP-Central device.
14. GAP-Peripheral prints the following in serial console,
    a. Accelerometer and gyroscope sensor read operation status.

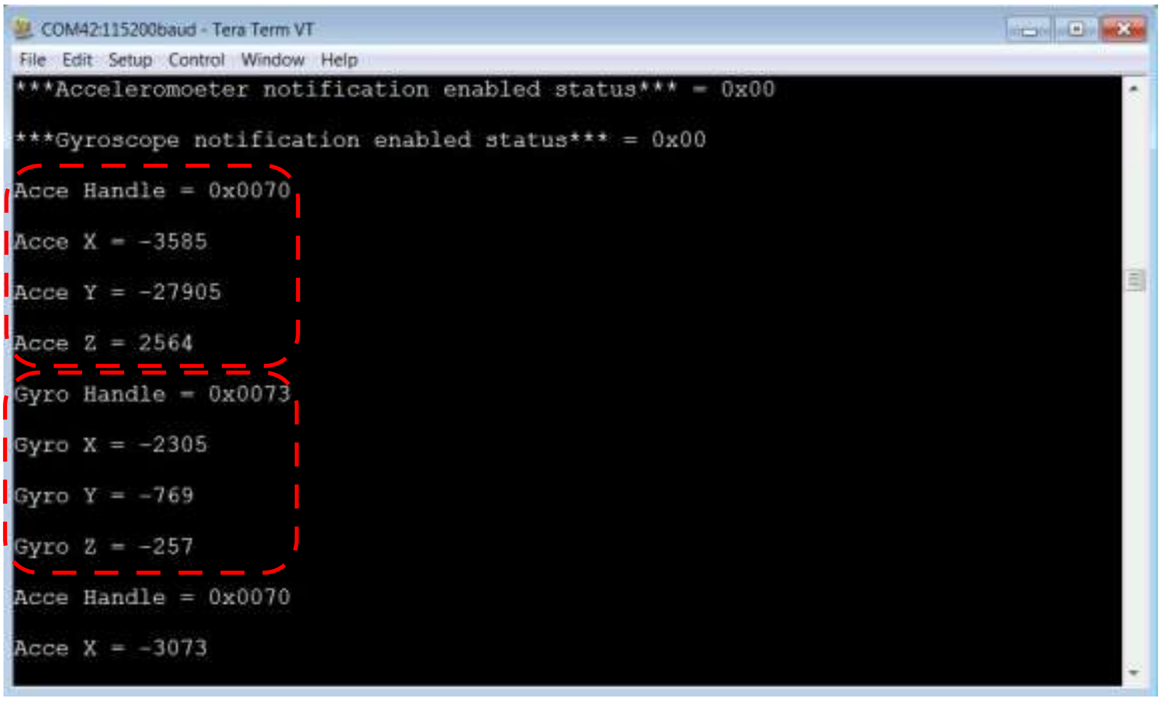
b. Accelerometer and gyroscope sensor data

c. Status of notification send to GAP-Central.


COM104:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Notification enabled!!!
Status of Accelerometer sensor data read = 0
Acce data x = -4097 || y = -27649 || z = 1796
Acce - value send status = 0x00
Notification enabled!!!
Status of Accelerometer sensor data read = 0
Acce data x = -3841 || y = -27649 || z = 4100
Acce - value send status = 0x00
Status of Gyroscope sensor data read = 0
Gyro data x = -1025 || y = -1793 || z = -257
Gyro - value send status = 0x00
Status of Accelerometer sensor data read = 0
```

15. GAP-Central devices receive sensor data notifications and print them on serial console.


COM42:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
***Accelerometer notification enabled status*** = 0x00
***Gyroscope notification enabled status*** = 0x00
Acce Handle = 0x0070
Acce X = -3585
Acce Y = -27905
Acce Z = 2564
Gyro Handle = 0x0073
Gyro X = -2305
Gyro Y = -769
Gyro Z = -257
Acce Handle = 0x0070
Acce X = -3073
```

Trademarks:

The Microchip name and logo, the Microchip logo, MPLAB, and PIC are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.