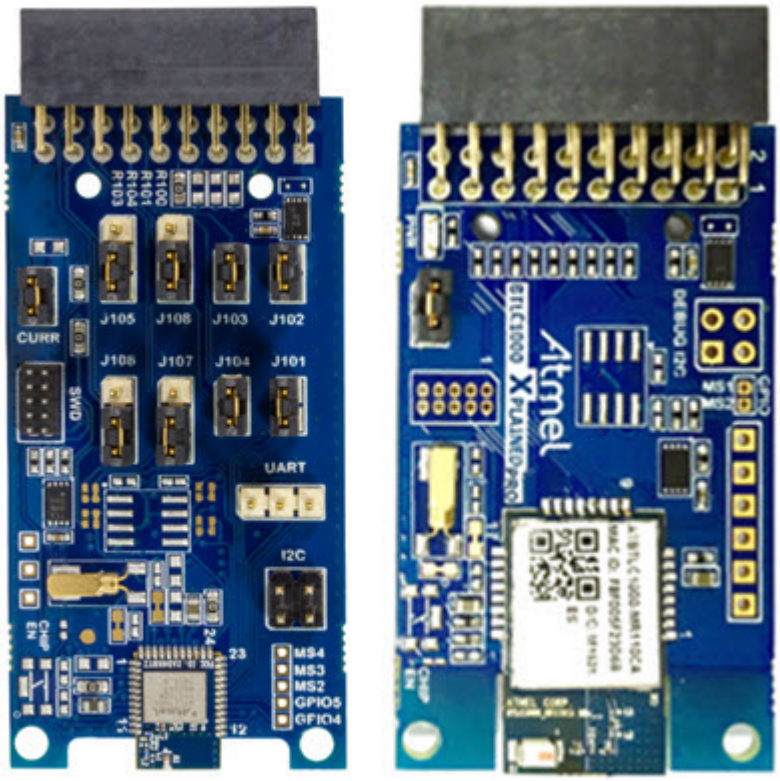
ATBTLC1000 BluSDK Example Profiles Application User's Guide

Introduction

This document describes how to set the ATBTLC1000-MR/ZR evaluation boards for various example applications supported by the Advanced Software Framework (ASF). This also provides the list of supported hardware platforms and IDEs to be used in conjunction with the ATBTLC1000-MR/ZR evaluation boards (see Table 2-1). The part number of the BTLC1000-ZR Xplained Pro (XPro) board is ATBTLC1000ZR-XPRO, and the part number of the BTLC1000-MR Xplained Pro board is ATBTLC1000MR-XPRO.

Note: All the example applications are included in BluSDK software package.

Figure 1. ATBTLC1000 Extension Boards



ATBTLC1000-ZR ATBTLC1000-MR

Features

- Observer application
- Proximity Monitor Application:
  - Device discovery and disconnection
  - Services and characteristics discovery
  - Services – Link Loss service, Immediate Alert service, and Tx Power service
  - Setting up Path Loss and Link Loss
  - Received Signal Strength Indicator (RSSI) sampling
- Proximity Reporter Application:
  - Advertisement
  - Pairing/bonding
  - Services – Link Loss service (mandatory), Immediate Alert service, and Tx Power service
- Apple® Notification Center Service (ANCS) Application
- Scan Parameters Service Application
- Time Information Profile Application:
  - Device discovery and disconnection
  - Pairing/bonding
  - BLE time client
- HID Mouse Device and HID Keyboard Device Applications:
  - Advertisement
  - Pairing
  - Services: HID service and Device Information service
  - Report mode (mouse)/Report mode (keyboard)
- Battery Service Application
- Simple Broadcaster Application
- Device Information Service
- Custom Serial Chat (CSC) Profile Application:
  - Device discovery and disconnection
  - Pairing/bonding
  - Send and receive messages
- Heart rate Profile Application:
  - Advertisement
  - Pairing/bonding
  - Heart rate sensor measurements
  - Console display
- Blood Pressure Profile Application:
  - Advertisement
  - Pairing/bonding
  - Blood pressure measurements
- Find Me Profile Application:
  - Advertisement

  - Pairing/bonding
  - Find Me alerts
- Phone Alert Status Profile Application:
  - Advertisement
  - Pairing/bonding
  - Phone alert status
- Alert Notification Profile Application:
  - Device discovery and disconnection
  - Pairing/bonding
  - Alert notification service
  - Alert on incoming call
- Multi-Role Peripheral Multi-Connect Application:
  - Supports eight connections
- L2CAP Throughput Application:
  - L2CAP central
  - L2CAP peripheral
  - Used LE L2CAP connection oriented channel for data communication
- Health Thermometer Profile (HTP) Application:
  - Advertisement
  - Pairing/bonding
  - RSSI sampling
  - Health thermometer service
  - Health thermometer profile app for iOS/Android
- iBeacon Application:
  - RSSI sampling
  - Beacon advertising
  - iBeacon demo app for iOS/Android
- AltBeacon Application:
  - AltBeacon advertising
  - AltBeacon demo app for iOS/Android
- Eddystone Beacon Application:
  - Eddystone UID, URL, and TLM frame types
  - URL configuration service with optional lock
  - Beacon demo application for Android and iOS
- Direct Test Mode (DTM) Application:
  - DTM setup procedure
  - Downloading DTM firmware

Table of Contents

1. Functional Overview

This chapter describes the functional overview of all the applications that are pre-defined in Atmel Studio.

1.1 Observer Application

The Observer application is used for continuously listening to the advertisement data over the air. This application supports the following advertisement data types:

- Incomplete list of 16-bit service class UUIDs
- Complete list of 16-bit service class UUIDs
- Incomplete list of 32-bit service class UUIDs
- Complete list of 32-bit service class UUIDs
- Incomplete list of 128-bit service class UUIDs
- Complete list of 128-bit service class UUIDs
- Shortened local name
- Complete local name
- Appearance
- Manufacturer specific data
- Tx Power
- Advertisement interval

1.2 Proximity Reporter Application

The Proximity profile is defined by the Bluetooth SIG to enable monitoring the proximity between two Bluetooth Low Energy (BLE) devices. The Proximity Monitor (a Generic Attribute (GATT) client) configures the behavior of the peer Proximity Reporter device (a GATT server) based on the link conditions. The configuration includes setting the alert level, which triggers on the Link Loss or based on different threshold of the Path Loss. The Path Loss determines the quality of the connection and it is derived out of the Received Signal Strength Indicator (RSSI) and transmits the power. The Proximity Monitor continuously evaluates the Path Loss and creates an immediate alert in the Proximity Reporter device when the Path Loss crosses threshold values.

On-board LED Status

The on-board LED is configured to notify the user about the alerts received. The different alerts for the Link Loss and Immediate Alert service are explained in the subsections.

Link Loss

On the Link Loss, the LED blinks according to the alert level set by the Proximity Monitor. The alert levels are:

- NO_ALERT for No alert level
- MILD_ALERT for Mild alert level
- HIGH_ALERT for High alert level

Based on the alert level configuration set by the Proximity Monitor, the LED blinks at different rates:

- If the alert level is “HIGH_ALERT” then the LED blinks faster (1 second interval)
- If the alert level is “MILD_ALERT” then the LED blinks moderately (2 second interval)
- If the alert level is “NO_ALERT” the LED must be off

Alert on Path Loss (Immediate Alert)

This alert is applicable when the “Immediate Alert” service is implemented. The example application relies on the Path Loss configuration done by the Proximity Monitor and notifies accordingly. The alert levels are:

- NO_ALERT for No alert level
- MILD_ALERT for Mild alert level
- HIGH_ALERT for High alert level

Based on the alert level configuration set by the Proximity Monitor, the LED blinks at different rates:

- If the alert level is “HIGH_ALERT” then the LED blinks faster (3 second interval)
- If the alert level is “MILD_ALERT” then the LED blinks moderately (5 second interval)
- If the alert level is “NO_ALERT” the LED must be off

1.3 Proximity Monitor Application

The Proximity profile is defined by the Bluetooth SIG to enable proximity monitoring between two BLE devices. The Proximity Monitor (a GATT client) configures the behavior of a peer Proximity Reporter device (a GATT server) based on the link conditions. The Proximity Monitor configures the desired behavior of the peer device through setting alert levels on the Link Loss and the Path Loss. In addition, it also maintains the connection with the Proximity Reporter and monitors the link quality of the connection based on RSSI reporting from the peer device.

1.4 ANCS Profile Application

The Apple® Notification Center Service (ANCS) is used to enable a device to access notifications from an iOS device that exposes ANCS.

The ANCS profile defines the following roles:

- Notification Provider (NP) is a device that provides the iOS notification
- Notification Consumer (NC) is a device that receives the iOS notifications and notification related data from Notification Provider

Incoming Call Notification

The programmed SAM L21 or the other supported hardware platforms (Notification Consumer) must be paired with an iPhone® to display the received incoming call notification on a console.

The Bluetooth® SIG defined Alert Notification profile provides similar functionality for Android devices. ANCS is a variant of the Alert Notification profile customized by Apple. For more details on Alert Notification, refer to the Alert Notification Profile Application.

1.5 Scan Parameters Service Application

The Scan Parameter service is an example application that demonstrates how to retrieve scan interval window information from a peer device. The Scan Parameter service must be implemented on a peer device to retrieve scan interval information. This application implements a GATT server role. This application can be used for obtaining the updated scan interval window value by configuring the scan refresh characteristic for notification.

1.6 Time Information Profile Application

The Time Information Profile is an example application for implementing the BLE time service, such as current time, date, and day, from a compatible Android/iPhone device and displays it on the console.

The profile defines the following roles:

- Time client, a device in a peripheral role to read the time, date, and day information
- Time server, a device to provide the time related information

Note: This application is supported in iOS 7.0 and above or a BLE compatible Android device in which the Microchip SmartConnect mobile application is installed.

1.7 HID Mouse Device or HID Keyboard Device Application

The HID Over GATT Profile (HOGP) is defined by the Bluetooth SIG to enable HID services support over a BLE protocol stack using the GATT profile. This allows devices like a keyboard or mouse to implement HOGP and to connect with a compatible HOGP/BLE host device, such as mobile phone, tablet, TV, and so on.

The HID Mouse device or HID Keyboard device application supports the following characteristics:

- Protocol mode (mouse/keyboard)
- Report (mouse/keyboard)
- Report map (mouse/keyboard)
- HID information (mouse/keyboard)
- HID control point (mouse/keyboard)
- Boot mouse input report (mouse only)
- Boot keyboard input report (keyboard only)
- Boot keyboard output report (keyboard only)

This example application simulates a function of a mouse or keyboard. Once the connection procedure is implemented between a mobile phone and the ATBTLC1000-MR/ZR board, the board can act as a mouse or a keyboard.

In the case of a HID Mouse device application, a mouse cursor, visible in the mobile screen, can be moved as per the predefined pattern by pressing the SW0 button on the board.

In the case of a HID keyboard device application, the predetermined text is sent to the mobile phone by pressing the SW0 button on the board. This can be viewed in any standard text editor in the mobile phone.

1.8 Battery Service Application

The Battery Service application is used for reporting the battery level of the device using the battery characteristics. Any application discovering the database can access the battery service instance during discovery services. This example application simulates the device battery level from 0% to 100%, with the step of 1% every second.

1.9 Simple Broadcaster Application

The Simple Broadcaster application is used for continuously broadcasting the advertisement data over the air. This application supports the following advertisement data types:

- Incomplete list of 16-bit service class UUIDs
- Complete list of 16-bit service class UUIDs
- Incomplete list of 32-bit service class UUIDs
- Complete list of 32-bit service class UUIDs
- Incomplete list of 128-bit service class UUIDs
- Complete list of 128-bit service class UUIDs
- Shortened local name
- Complete local name
- Appearance
- Manufacturer specific data

1.10 Device Information Service Application

The Device Information Service (DIS) application is used for providing a setup to the user to define and use the BLE DIS service. Any application discovering the database can access the DIS service instance during discovery services. This application supports the following characteristics:

- Manufacturer name string
- Model number string
- Serial number string
- Hardware revision string
- Firmware revision string
- Software revision string
- System ID
- IEEE® 11073-20601 regulatory certification data list
- PnP ID

1.11 Custom Serial Chat Profile Application

The Custom Serial Chat application is used for sending and receiving data between the boards (see Table 2-1) and the Microchip SmartConnect mobile application. This is a custom profile example application implemented over GATT. The user can send the information to the mobile phone using the console terminal that is configured with the board and vice versa.

Note: For more information on Custom Serial Chat service, refer to Custom Serial Chat Service Specification.

1.12 Heart Rate Profile Application

The Heart Rate Profile application is used for enabling the collector device (GATT client) to connect and interact with a Heart Rate sensor (GATT server) to be used in the fitness applications. The Heart Rate sensor sends the heart rate measurement in bpm (beats per minute), energy expended in kJ (kilojoules), and R-R intervals in seconds. In addition to the Heart Rate service, this profile also implements the Device Information Service, which provides information about the Heart Rate sensor device.

The Heart Rate profile provided by Bluetooth SIG defines three characteristics for the exchange of heart rate parameters between the sensor and monitor. The characteristics of the profile are used to transfer heart rate parameters like bpm, R-R interval measurements, and other parameters like body sensor

location and energy expended values. The optional “Heart Rate Control Point characteristic” is used by the Heart Rate monitor to reset the energy expended in the Heart Rate sensor.

The Heart Rate sensor, which is the GATT server, holds the characteristics and sends the measurement values to the Heart Rate monitor.

- The Heart Rate, R-R interval, and energy expended are sent using the Heart Rate measurement characteristics
- The Heart Rate measurements are sent to the monitor on a value change if the monitor has enabled the notifications
- The body sensor location is read by the monitor by its body sensor location characteristic. The energy expended sent in the heart rate measurement can be reset by the monitor by writing to the Heart Rate control point characteristic

Note: The example application simulates the sensor measurements and sends them to the Heart Rate collector.

1.13 Blood Pressure Profile Application

The Blood Pressure Profile (BLP) application is used for connecting to and interacting with a device with a blood pressure sensor device to be used in consumer and professional health care applications. This application enables the device to obtain blood pressure measurement and other data from a non-invasive blood pressure sensor that exposes the Blood Pressure service. For example, a nurse or doctor could use a non-invasive blood pressure sensor on a patient that sends blood pressure measurements to a laptop or other hand held device.

Blood Pressure Measurements

The blood pressure measurement characteristic can be used to send blood pressure measurements.

- Flags field (containing units of blood pressure and used to show the presence of optional fields)
- Blood pressure measurement compound value field and, depending upon the contents of the Flags field
- Timestamp (time of the measurement)
- Pulse Rate
- User ID
- Measurement status fields

The intermediate cuff pressure characteristic may be sent frequently during the course of a measurement, so that a receiving device can effectively update the display on its user interface during the measurement process.

When the client characteristic configuration descriptor is configured for indications and a blood pressure measurement is available, this characteristic is indicated while in a connection. When the client characteristic configuration descriptor is configured for indications and a blood pressure measurement is available, this characteristic is indicated while in a connection.

- The blood pressure measurement characteristic is used to send blood pressure measurements
- The intermediate cuff pressure characteristic is used to send current cuff pressure values to a device to display, while the measurement is in progress
- The blood pressure feature characteristic is used to describe the supported features of the blood pressure sensor

The ATBTLC1000-MR/ZR together with the host MCU simulates a blood pressure sensor (GATT server role) and sends simulated values to the blood pressure monitor (Microchip SmartConnect mobile application).

1.14 Find Me Profile Application

The Find Me Profile (FMP) application is used to define the device to create an alert signal behavior when a button is pressed on one device to cause an alerting signal on a peer device.

Find Me Target

The FMP defines the behavior when a button is pressed on a device to cause an immediate alert on a peer device. This can be used to allow users to find devices that have been misplaced.

The Find Me Target application, which is the GATT server, holds the alert level characteristics and waits for the Find Me locators alert and performs the following alert level characteristic:

- When the Find Me locator device wishes to cause an alert on the Find Me Target device, it writes the specific alert level (High, Mild and No alert) in the alert level characteristic.

1.15 Phone Alert Status Profile Application

The Phone Alert Status (PAS) profile is used to obtain the phone alert status exposed by the phone alert status service on a mobile phone. The alert status and ringer setting information of a mobile phone can be received and modified by the phone alert status service. The device can also use this profile to configure the ringer status on the mobile phone.

Phone Alert Status Notifications

This profile defines two roles:

- Phone alert server – device that originates the alerts
- Phone alert client – device that receives the alerts and alerts the user

The phone alert client (a GATT client) configuration is implemented on the ATBTLC1000-MR/ZR along with a few other supported hardware platforms and IDEs (see Table 2-1). The example application utilizes the SW0 button on the supported hardware platform to demonstrate the notification use-cases. A BLE compatible Android device that contains the Microchip SmartConnect mobile application provides the phone alert server functionality in this example. On the application, once the service is discovered and the user can click on the PAS service to enable the notifications.

1. After connecting with the mobile phone, press the SW0 button once to set the PAS server to “Silent” mode.
2. In the second SW0 button press, the device is set to “Mute” mode.
3. In the third SW0 button press, the device is returns to “Normal” mode.
4. In the fourth SW0 button press, a “Read Characteristic” request is issued that reads the characteristics of “Alert Status”, “Ringer Settings”, and “Ringer Control Point”.

Note: The PAS profile application is not supported in iOS devices. This example works only with BLE compatible Android devices that contain the Microchip SmartConnect mobile application.

1.16 Alert Notification Profile Application

The Alert Notification Profile allows a device to obtain information from a mobile phone about incoming calls, missed calls, and SMS/MMS messages. The information includes the caller ID for an incoming call

or the sender ID for an email/SMS/MMS, but not the message text. This profile also enables the client device to get information about the number of unread messages on the server device.

Note: This example application only works with BLE compatible Android devices that contain the Microchip SmartConnect mobile application.

The Microchip SmartConnect mobile application is used for implementing the Alert Notification service and can be used for demonstrating an example application. This example application supports missed call alert notification and SMS alert notification.

The device implements the GATT client, which reads (or notifies) about the characteristic values received from the GATT server (the mobile phone). The device must be paired with an Android phone. A missed call or SMS alert notifications can be enabled/disabled, once connection is established. The Microchip SmartConnect application notifies a missed call or SMS alert, which are then displayed on the terminal console on the device side.

The “SW0” user button on the supported platform is programmed in such a way that each successive button press either enables or disables the notifications.

Note: The SAM L21 or supported platforms (see Table 2-1) + ATBTLC1000-MR/ZR is referred as “device”.

1.17 Multi-Role Peripheral Multi-Connect Application

The Multi-Role Peripheral Multi-Connect application demonstrates the ATBTLC1000-MR/ZR to have eight simultaneous active connections. The ATBTLC1000-MR/ZR supports multiple roles such as GAP peripheral device with battery service and GAP central device with a Find Me locator profile at the same time. It also supports multiple connection such as a GAP peripheral device with battery service that can connect with seven GAP central devices simultaneously.

The Multi-Role Peripheral Multi-Connect application initially starts advertising using connectable advertisement packets as a GAP peripheral and if any device sends a connection request, the application gets connected to the remote device and exchanges the data on the link established. If the connection request from the device is not sent within a minute, then the application scans the devices and initiates a connection to the peripheral device, which advertises using connectable advertisement packets. The ATBTLC1000-MR/ZR is exchanging the data as a GAP central once the link is established. Again, the Multi-Role application is started to advertise using connectable advertisement packets as a GAP peripheral and gets connected to the remote device, which sends a connection request, and exchanges the data on the new link established. The process continues until the Multi-Connection application reaches eight connections.

1.18 L2CAP Throughput Application

The L2CAP Throughput example application supports the L2CAP central feature and the L2CAP peripheral feature.

1.19 Health Thermometer Profile Application

The Health Thermometer Profile (HTP) enables the data collection device to obtain data from a thermometer sensor that exposes the health thermometer service. The profile defines following roles:

- Thermometer – Device to measure temperature
- Collector – Device to receive temperature measurement and other data from a thermometer

The thermometer implements only one Health Thermometer service in addition to the Device Information Service to display the information about the thermometer device. The current HTP application implements the following characteristics:

- Temperature measurement
- Intermediate temperature
- Measurement interval

1.20 iBeacon Application

The iBeacon application is used to advertise iBeacon specific packets that include UUID, major and minor numbers. Any beacon scanner application can be used for finding the beacon device. The iOS Microchip SmartConnect app can be used to find the beacon devices in the vicinity.

This profile defines the following roles:

- Monitor – Device (iOS/Android) to search for beacon packets
- Reporter – Device that continuously advertises the beacon packet as a part of advertisement data

1.21 AltBeacon Application

The AltBeacon application advertises packets that includes MFG ID, Beacon code, Beacon ID, Reference RSSI, and MFG reserved value. Any AltBeacon scanner application can be used to find the AltBeacon device based on the beacon code. The supplied iOS demo app can be used to find the AltBeacon devices in the vicinity. The profile defines the following roles:

- Monitor – Device (iOS/Android) to search for AltBeacon packets
- Reporter – Device that continuously advertises the AltBeacon packet as part of advertisement data

1.22 Eddystone Beacon Application

The Eddystone™ is an open Bluetooth Smart beacon format from Google that works across Android and iOS devices. The Microchip SmartConnect BLE BluSDK software solution provides full support for this beacon format on the ATBTLC1000-MR/ZR devices.

The Eddystone beacon application supports UID, URL, and TLM frame types. The application can be configured as follows using the `APP_TYPE` define:

- Set `APP_TYPE` to `EDDYSTONE_UID_APP` to send UID and TLM beacon frames at regular beacon intervals
- Set `APP_TYPE` to `EDDYSTONE_URL_APP` to send URL and TLM frames. This also supports the URL configuration service that enables the beacon to be configured dynamically from a mobile application

The Eddystone application is completely configurable using the `conf_eddystone.h` file. The `#defines` present in the `conf_eddystone.h` file are supplied with default values, which can be changed by the user to meet the requirements. In addition to this compile time configuration, the frame fields like the UID value, URL, transmit power at 0 meters, and so on can be changed using the APIs provided in `eddystone.h` file.

1.23 Direct Test Mode Application

The Direct Test Mode (DTM) application is used to establish and test the Direct Test mode between two ATBTLC1000-MR/ZR modules. Windows-based ATBTLC1000-MR/ZR characterization software is used

at both ends. The ATBTLC1000-MR/ZR Xplained Pro extensions are connected to a compatible MCU host device such as the SAM L21, SAM D21, SAM G55 or SAM 4S. The Performance Analyzer (PC tool) communicates with the ATBTLC1000 using a serial bridge application running on the MCU.

Serial Bridge Application

Sends the DTM commands to the ATBTLC1000-MR/ZR to enable the DTM performance analyzer application. The supported hardware platforms (see Table 2-1) can act as a serial bridge between the ATBTLC1000-MR/ZR and Atmel Studio performance analyzer tool. Once the SAM L21 (or other supported hardware platforms) is powered on or Reset, it initializes the Wake-up and Chip Enable to download the patch file into the ATBTLC1000-MR/ZR and completes the initialization procedure of the BLE module. After the ATBTLC1000-MR/ZR initialization, the application initializes the SAM L21 to act as a serial bridge.

2. Supported Hardware Platforms and IDEs

The following table provides the supported hardware platforms and IDEs for the ATBTLC1000-MR/ZR.

Table 2-1. BluSDK – Supported Hardware and IDEs

Platform	MCU	Supported BLE Device	Supported Evaluation Kits	Supported IDEs
SAM L21	ATSAML21J18B	ATBTLC1000-MR, ATBTLC1000-ZR	ATBTLC1000-XSTK (ATSAML21-XPRO-B + ATBTLC1000-XPRO) or ATBTLC1000ZR-XSTK (ATSAML21-XPRO-B + ATBTLC1000ZR-XPRO)	Atmel Studio v7.0 and IAR
SAM L21	ATSAML21J18A	ATBTLC1000-MR, ATBTLC1000-ZR	ATSAML21-XPRO + ATBTLC1000-XPRO or ATSAML21-XPRO + ATBTLC1000ZR-XPRO	Atmel Studio v7.0 and IAR
SAM D21	ATSAMD21J18A	ATBTLC1000-MR, ATBTLC1000-ZR	ATSAMD21-XPRO + ATBTLC1000-XPRO or ATSAMD21-XPRO + ATBTLC1000ZR-XPRO	Atmel Studio v7.0 and IAR
SAM G55	ATSAMG55J19	ATBTLC1000-MR, ATBTLC1000-ZR	ATSAMG55-XPRO + ATBTLC1000-XPRO or ATSAMG55-XPRO + ATBTLC1000ZR-XPRO	Atmel Studio v7.0 and IAR
SAM 4S	ATSAM4SD32C	ATBTLC1000-MR, ATBTLC1000-ZR	ATSAM4S-XPRO + ATBTLC1000-XPRO or ATSAM4S-XPRO + ATBTLC1000ZR-XPRO	Atmel Studio v7.0 and IAR

3. Hardware Setup

3.1 ATBTLC1000 Board Types

The ATBTLC1000 supports the following extension boards:

1. ATBTLC1000-MR
2. ATBTLC1000-ZR
   - Supports from BluSDK 6.0 release and later

The following figures illustrate samples of ATBTLC1000-MR and ATBTLC1000-ZR kit details displayed in the Atmel Studio.

Figure 3-1. ATBTLC1000-MR

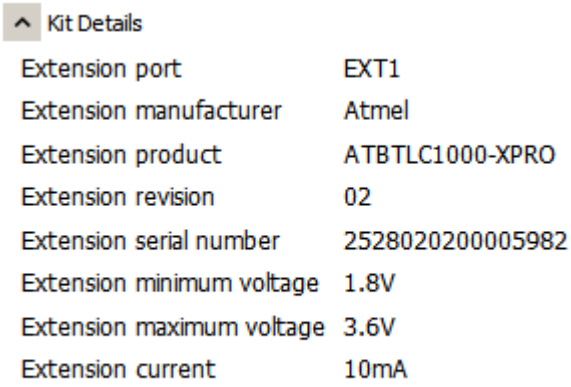
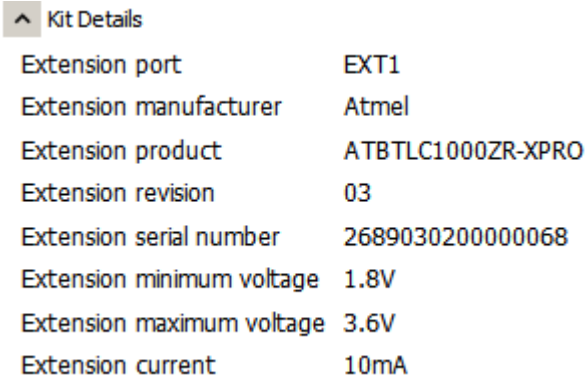


Figure 3-2. ATBTLC1000-ZR



3.2 SAM L21 Xplained Pro Setup

The following figure illustrates the connection between the ATBTLC1000-MR Xplained Pro Extension Board connected to the SAM L21 Xplained Pro Board.

Figure 3-3. ATBTLC1000-MR Xplained Pro Extension Connected to the SAM L21 Xplained Pro

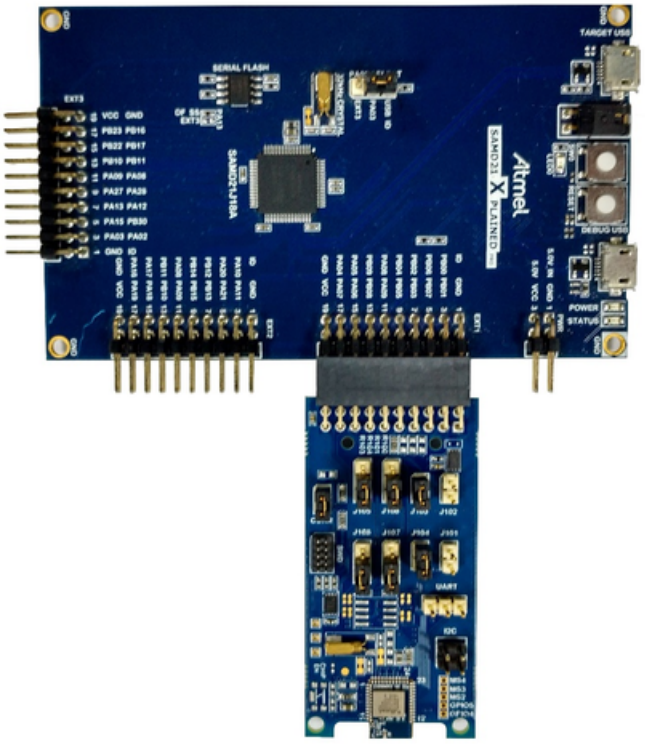
Note: Refer to the following section for the ATBTLC1000-ZR Xplained Pro Extension Board configuration.

3.3 SAM D21 Xplained Pro Setup

The following figures illustrate the connection between the ATBTLC1000-MR and ATBTLC1000-ZR Xplained Pro Extension Boards connected to the SAM D21 Xplained Pro.

Figure 3-4. ATBTLC1000-MR Xplained Pro Extension Connected to the SAM D21 Xplained Pro

Figure 3-5. ATBTLC1000-ZR Xplained Pro Extension Connected to the SAM D21 Xplained Pro

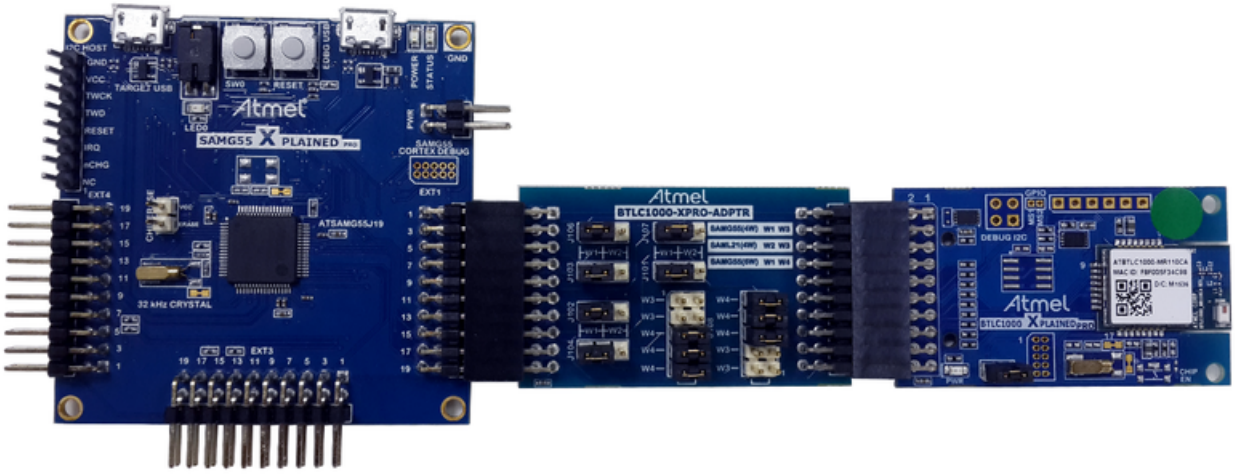


Note: The connection for the SAM L21 Xplained Pro with the ATBTLC1000-ZR Xplained Pro Extension Board is similar to the preceding figure.

3.4 SAM G55 Xplained Pro Setup

The following figures illustrate the connection between the ATBTLC1000-MR and ATBTLC1000-ZR Xplained Pro Extension Boards connected to the SAM G55 Xplained Pro Board.

Figure 3-6. ATBTLC1000-MR Xplained Pro Extension Connected to the SAM G55 Xplained Pro

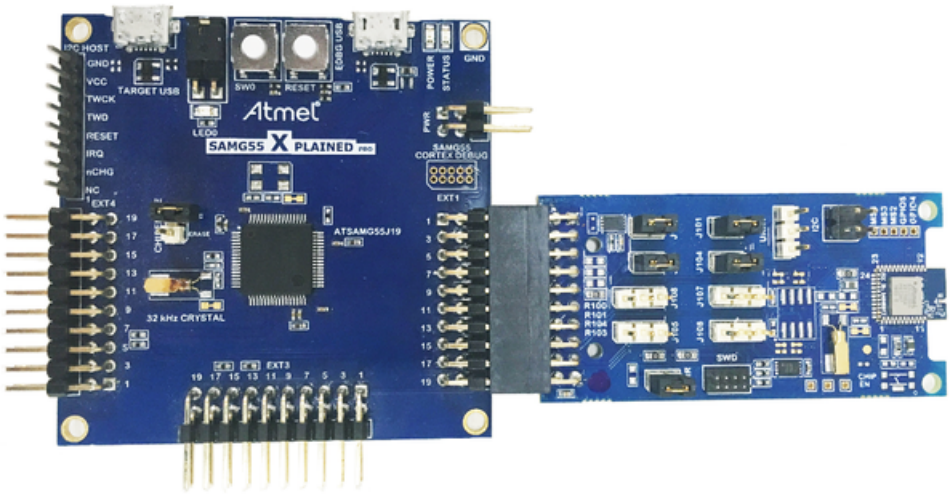


Note:

1. The SAM G55 Xplained Pro is connected to the ATBTLC1000-MR Xplained Pro through the ATBTLC1000 XPRO adapter board.

2. For a jumper connection on 4-wire and 6-wire connections, refer to the ATBTLC1000-XPRO-ADAPTER marking label.

Figure 3-7. ATBTLC1000-ZR Xplained Pro Extension Connected to the SAM G55 Xplained Pro

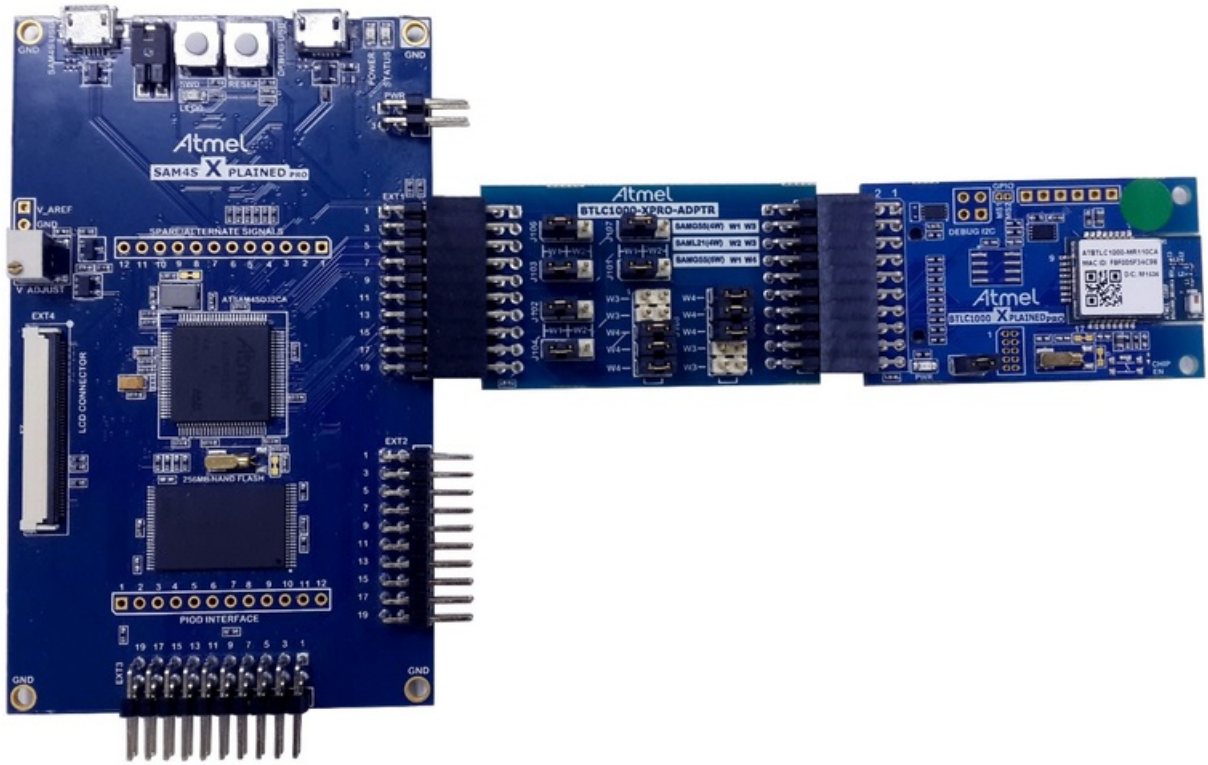


Note: The connection for the SAM 4S Xplained Pro with the ATBTLC1000-ZR Xplained Pro Extension Board is similar to the preceding figure.

3.5 SAM 4S Xplained Pro Setup

The following figure illustrates the connection between the ATBTLC1000-MR Xplained Pro Extension Board and the SAM 4S Xplained Pro Board. These two devices are connected using the ATBTLC1000-XPRO-ADAPTER Board.

Figure 3-8. ATBTLC1000-MR Xplained Pro Extension Connected to the SAM 4S Xplained Pro



Note: For a jumper connection on 4-wire and 6-wire connections, refer to the ATBTLC1000-XPRO-ADAPTER marking label.

4. Software Setup

4.1 Installation Steps

1. Download and install the Atmel Studio software.
2. Install the standalone Advanced Software Framework (ASF) package.
3. Download and install the Microchip SmartConnect App on the mobile phone, available in the Apple Store® for iPhone and in the Google Play™ Store for Android.

Note: Atmel Studio offers predefined example projects for the SAM L21, SAM D21, SAM G55 and SAM 4S extension boards.

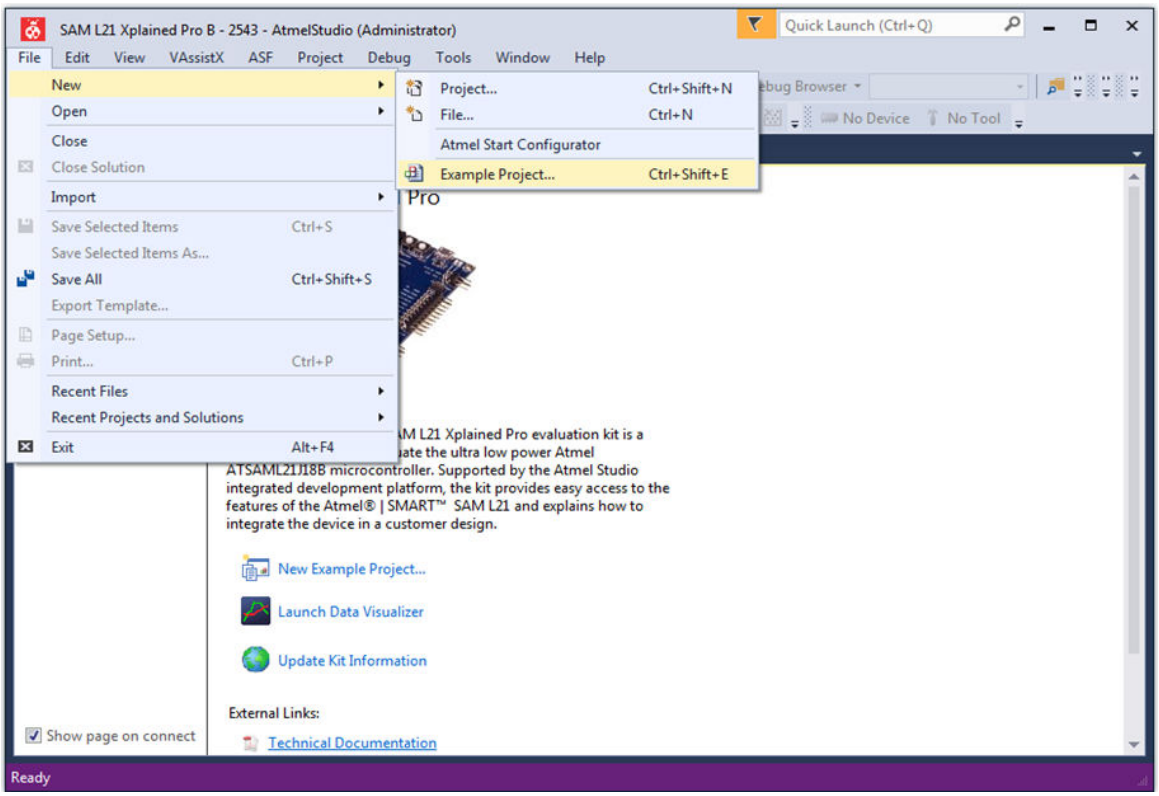
Note: For more information on the previous releases, refer to the *Atmel Studio Release Notes* available on the Microchip web page.

4.2 Build Procedure

Perform the following steps to build an example project. This example build procedure is developed using the SAM L21 Xplained Pro Board, which is also valid for the other supported hardware platforms and IDEs (see Table 2-1).

1. Open Atmel Studio and select `File > New > Example Project`.

Figure 4-1. Creating a New Project



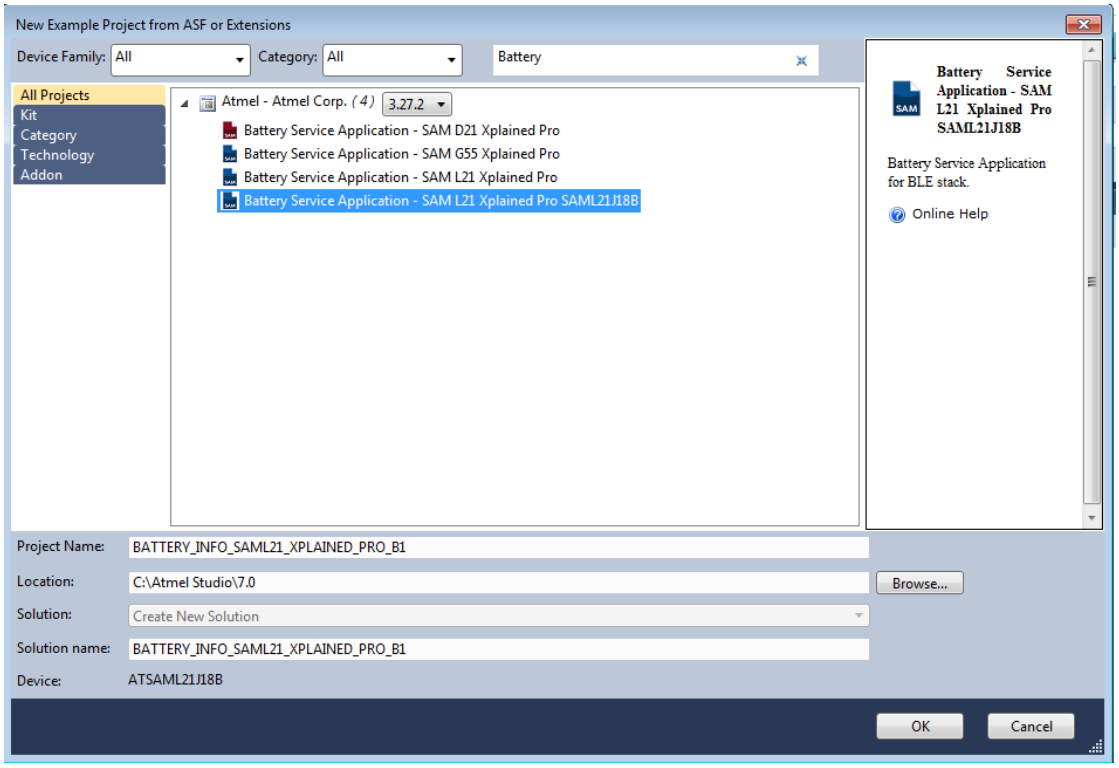
2. In the New Example Project from ASF or the Extensions window:

2.1. Enter the application specific keyword in the search box; for example, Scan Parameter, Battery Service Application, Blood Pressure, etc.

2.2. Select the respective example application of theSAM L21 by expanding the “Atmel - Atmel Corp.” in the **All Projects** tab. This selection automatically populates the Project Name, Location, Solution, Solution Name, and Device.

2.3. Click **OK**.

Figure 4-2. Searching for a Specific Application Example

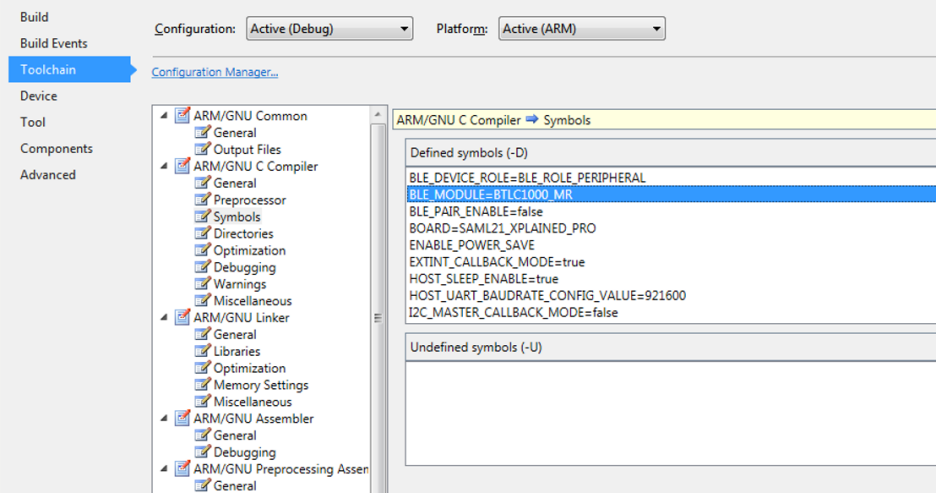


3. Select the “Accept the License Agreement” checkbox and then click **Finish**.
4. Atmel Studio generates the project files for the selected application example that can be used in the SAM L21 Xplained Pro board.
5. Go to `Project > Properties` to choose the hardware configuration switches and number of wires:

   5.1. Set the appropriate build symbols (see following figure):

   - For ATBTLC1000-MR: “BLE_MODULE = BTLC1000_MR”
   - For ATBTLC1000-ZR: “BLE_MODULE = BTLC1000_ZR”

Figure 4-3. Selecting the ATBTLC1000 Board Type



5.2. Choose between the 4-wire or 6-wire modes:

- ATBTLC1000-MR – supports both 4-wire and 6-wire UART modes. The ATBTLC1000-XPRO-ADPTR must be with this board in 4-wire mode.
- ATBTLC1000-ZR – supports only 4-wire mode.

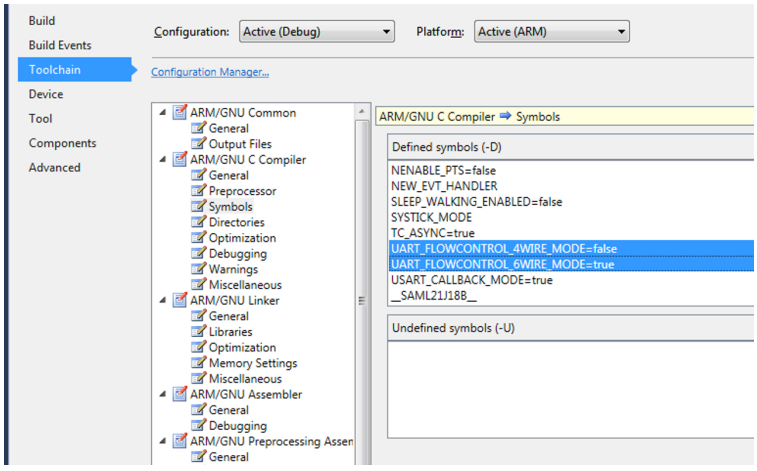
Note: For more information on 4-wire mode, refer to Hardware Flow Control for 4-wire Mode eFuse Write Procedure.

The configurations for 4-wire and 6-wire are as follows:

- 6-wire:
  - UART_FLOWCONTROL_4WIRE_MODE=false
  - UART_FLOWCONTROL_6WIRE_MODE=true
- 4-wire:
  - UART_FLOWCONTROL_4WIRE_MODE=true
  - UART_FLOWCONTROL_6WIRE_MODE=false

Configure UART_FLOWCONTROL_4WIRE_MODE and UART_FLOWCONTROL_6WIRE_MODE symbols in the project properties as shown in the following figure.

Figure 4-4. Configuring the UART Flow Compiler Symbols



6. For the Time Information Profile application, the user must select the compiler symbol based on the following:

– For Android devices: TP_ANDROID
– For iOS devices: NTP_ANDROID

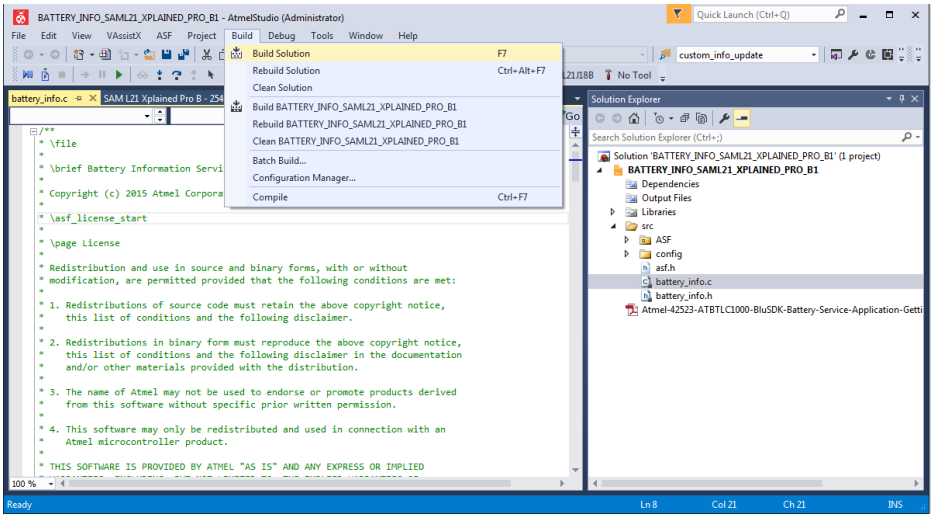
Note: This step is applicable only for the Time Information Profile application.

Figure 4-5. Configuring the UART Flow Compiler Symbols for the Time Information Profile

Note: iOS requires a device supporting the Time Information Profile to include the service solicitation advertisement type in the advertisement data. The above setting provides the configuration to build the Time Information Profile for iOS or Android. The iOS natively supports Time Server and does not require a specific mobile application. To enable the devices that are displayed on the iOS BLE devices page, the service solicitation advertisement data type configuration is necessary.

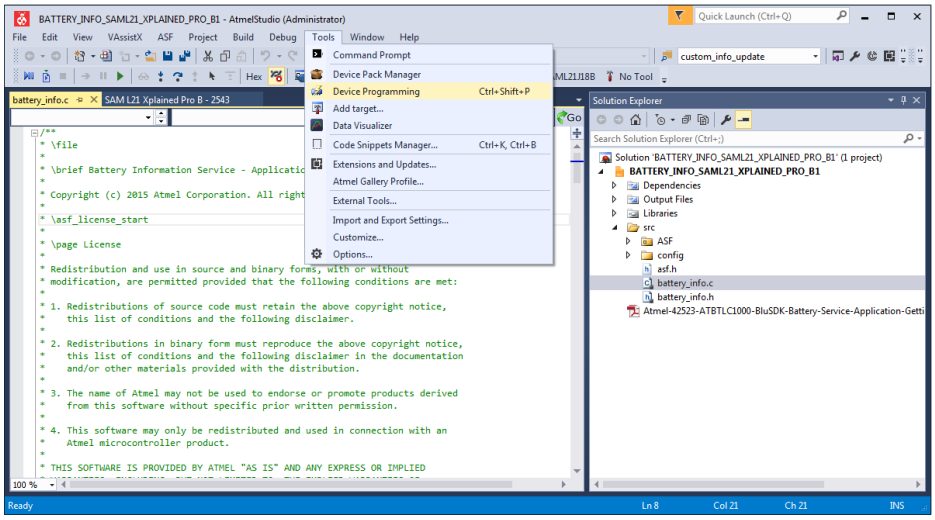
7. To build the solution, go to `Build > Build Solution`.

Figure 4-6. Building Solution for Selected Application Example



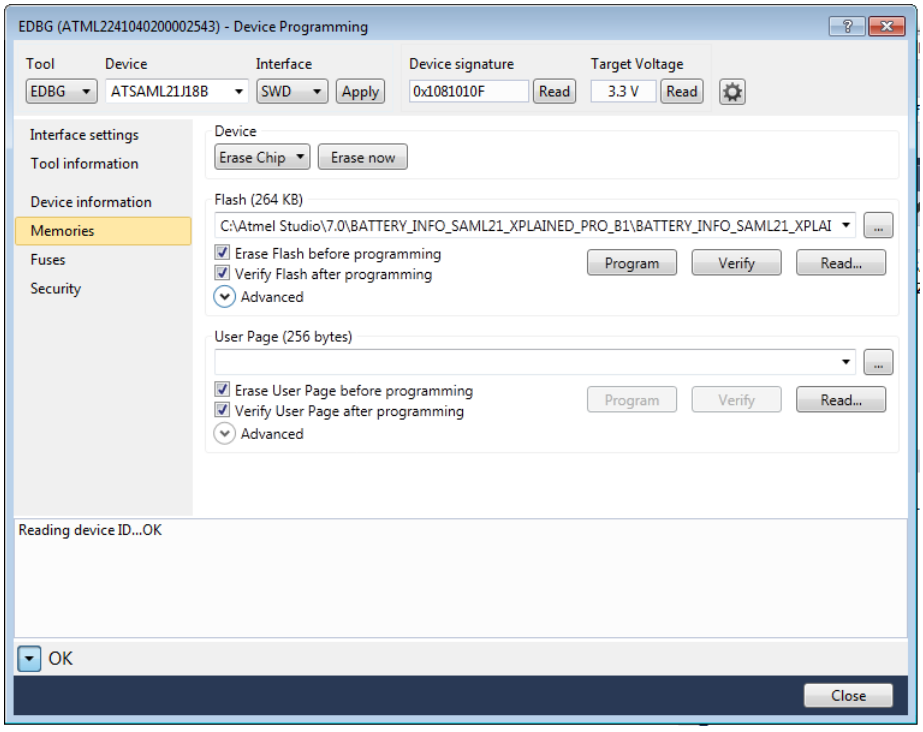
8. The generated solution is downloaded into the SAM L21 XPro board through the USB cable. To program the board, go to `Tools > Device Programming`.

Figure 4-7. Selecting Device Programming



9. In the EDBG (XXXXXXXX) Device Programming window, perform the following steps:
   9.1. Select EDBG in **Tool**.
   9.2. Click **Apply** and then click **Read** to read the *Device Signature*.
   9.3. After reading the *Device*, click **Program** to program the device.

Figure 4-8. Embedded Debugger Device Programming Window



10. After flashing the example application into the SAM L21 Xpro board, it is ready to be used as a BLE device that supports the selected application example.

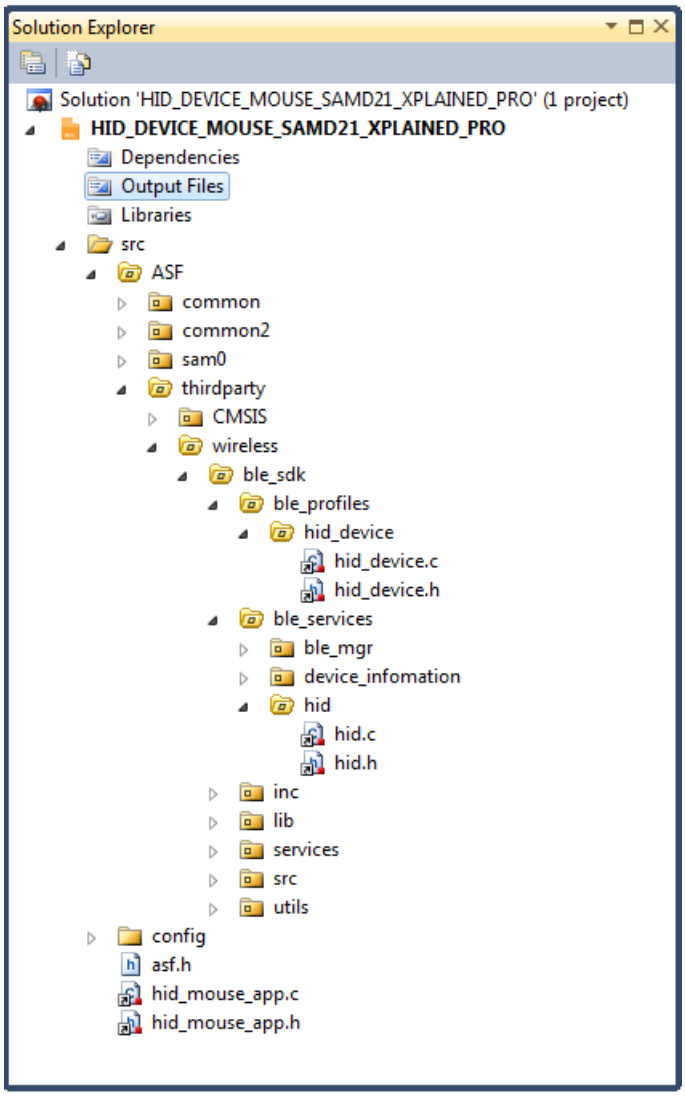
Note:

1. To run the profile application, refer to Running the Demo.
2. In the case of HID and Broadcaster applications, refer to the following configuration sections.

4.2.1 HID Mouse and HID Keyboard Application Configuration

The user needs to modify a few macros in `hid_device.h` (HID profile) for configuring the profile for HID Mouse and HID Keyboard applications as per the desired application use case.

Figure 4-9. HID Mouse Code Hierarchy

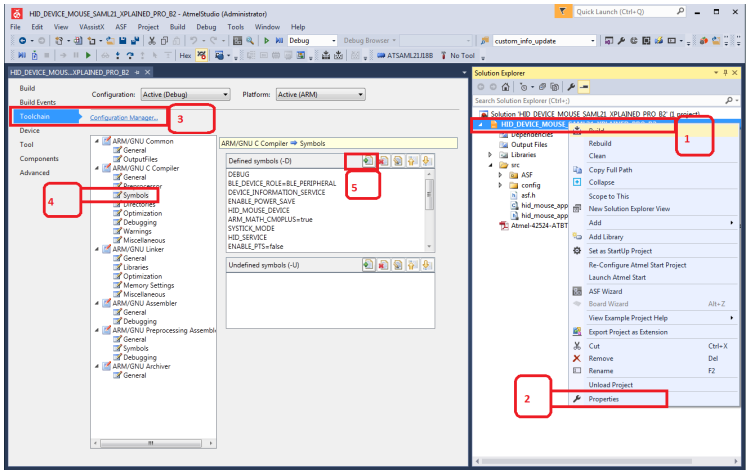


Note: Similar to HID Mouse, the `hid_device.h` file for the HID Keyboard is available in the following directory: `\asf\thirdparty\wireless\ble_sdk\ble_profiles\hid_device\`

The list of macros that must be modified by the user are:

1. By default, the application supports Report mode. If the application requires only Boot mode support, the user can add the macro `BOOT_MODE` in the **Compiler/Symbols** tab, as shown in the following screen.

Figure 4-10. Enabling Boot Mode Support



2. The user must configure the desired number of reports in the application. Currently, a maximum of 10 reports are supported.

```
#define HID_NUM_OF_REPORT               (1)
```

3. The user must configure the desired number of service instances. Currently, a maximum of two services are supported.

```
#define HID_SERV_INST                   (1)
```

Note: After configuring the profiles for HID Mouse and HID Keyboard, follow the steps mentioned in HID Mouse Device and HID Keyboard Device, respectively.

4.2.2 Configuration of the Simple Broadcaster Application

- Simple Broadcaster application advertises the default configuration provided as follows:
  - Non-connectable undirected advertisement event
  - Broadcasts data in advertisement data packets only
  - Broadcasts the following advertisement data types:
    - Complete list of 16-bit service class UUIDs
    - Complete local name
    - Appearance
- The configuration and advertisement data types listed above can be changed by using the macros provided in the `simple_broadcaster_app.h` file that is available in the `\src\config\` directory.

Note: After configuration, follow the steps mentioned in Simple Broadcaster Application.

4.2.3 Configuration of Observer Application

The default scanning parameters of Observer application are:

```
MAX_SCAN_DEVICE         (10)
SCAN_INTERVAL           (96)
SCAN_WINDOW             (96)
SCAN_TIMEOUT            (0x0000)
SCAN_TYPE               (AT_BLE_SCAN_ACTIVE)
```

These parameters can be modified as per the user requirement. These parameter can be updated in the `ble_manager.h` file, which is available in the following directory: `\asf\thirdparty\wireless\ble_sdk\ble_services\ble_mgr\`

Note: After configuration follow the steps in Observer Application.

5. Application Demo

5.1 Demo Setup

The following figure shows how to setup the board and the Microchip SmartConnect App for the purpose of the demo.

Figure 5-1. Demo Setup

Table 5-1. Demo Setup Details for Various Applications

Applications (Keywords)	BLE Node 1	BLE Node 2
Observer Application	Any BLE device can be used as Peripherals	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as an Observer application
Proximity Reporter Application	Supported by the Microchip SmartConnect application for iPhone/Android devices to act as a Proximity Monitor	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as a Proximity Reporter
Proximity Monitor Application	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as a Proximity Monitor	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to acts as a Proximity Reporter
ANCS Profile Application	Supported by the Microchip SmartConnect application for only iPhone devices to act as a Notification Provider	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as a Notification Consumer
Scan Parameters Service Application	Supported by the Microchip SmartConnect application for iPhone/Android devices	Supported by the ATBTLC1000-MR/ZR extension board and microcontrollers to act as Scan Parameter Service
Time Information Profile Application	Supported by the Microchip SmartConnect application for iPhone/Android devices to act as a Time server	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Time client

Applications (Keywords)	BLE Node 1	BLE Node 2
HID Mouse Device Application	Supported by the Microchip SmartConnect application only for Android devices to act as HOGP host	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as HID Mouse Device application
HID Keyboard Device Application	Supported by the Microchip SmartConnect application only for Android devices to act as Notepad text editor app (HOGP Host role)	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as HID Keyboard Device application
Battery Service Application	Supported by the Microchip SmartConnect application for iPhone/Android devices	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Battery Service application
Simple Broadcaster Application	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Simple Broadcaster application	Supported by Scanner application on a mobile phone
Device Information Service	Supported by the Microchip SmartConnect application for iPhone/Android devices	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Device Information Service application
Custom Serial Chat Profile Application	Supported by Custom Serial Chat (CSC) application for iPhone/ Android to send and receive data	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller with CSC application to send and receive data
Heart Rate Profile Application	Supported by the Microchip SmartConnect application for iPhone/Android devices to act as a Heart Rate Data Collector	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Heart Rate Sensor
Blood Pressure Profile Application	Supported by the Microchip SmartConnect application for iPhone/Android devices to act as a Blood Pressure Monitor	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Blood Pressure Sensor
Find Me Profile Application	Supported by the Microchip SmartConnect application for iPhone/Android devices to act as a Find Me Locator	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Find Me Target
Phone Alert Status Profile Application	Supported by the Microchip SmartConnect application only for Android devices to act as a Phone Alert Status server	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Phone Alert Status client
Alert Notification Profile Application	Supported by the Microchip SmartConnect application only for	Supported by the ATBTLC1000-MR/ZR extension board and

Applications (Keywords)	BLE Node 1	BLE Node 2
	Android devices to act as a Notification Provider	microcontroller to act as Notification Consumer
Multi-Role Peripheral Multi-Connect Application	Supported by the ATBTLC1000-MR/ZR extension board, microcontroller, and Microchip SmartConnect application to act as Peripheral or Central. This supports maximum of eight connections	
L2CAP Throughput Application	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as L2CAP Central	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as L2CAP Peripheral
Health Thermometer Profile (HTP) Application	Supported by the Microchip SmartConnect application for iPhone/Android devices to act as a Health Thermometer Collector	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as HTP application
iBeacon Application	Supported by the Microchip SmartConnect application for iPhone/Android devices to act as Beacon Monitor	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Beacon Reporter
AltBeacon Application	Supported by the Microchip SmartConnect application for iPhone/Android devices act as a AltBeacon App (Monitor)	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Reporter
Eddystone Beacon Application	Supported by the Microchip SmartConnect application for iPhone/Android devices	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as Eddystone Beacon Application
Direct Test Mode (DTM) Application	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as a Transmitter (Tx) Test Board. BLE performance analyzer tool connected with target board using COM port	Supported by the ATBTLC1000-MR/ZR extension board and microcontroller to act as a Receiver (Rx) Test Board. BLE performance analyzer tool connected with target board using COM port

5.2 Console Logging

For the purpose of debugging, a logging interface can be implemented in the applications.

The logging interface utilizes the same EDBG port that connects to a supported Xplained Pro (XPro) platform. A serial port monitor application (for example, TeraTerm) is opened and attached to the appropriate COM port enumerated by the device on the PC.

5.3 Running the Demo

Initializing the Device

Perform the following steps to initialize the device:

1. Open any Terminal Application (for example, TeraTerm). Select the COM port enumerated on the PC and set the following parameters:
   - Baudrate 115200
   - Parity None
   - One Stop bit
   - One Start bit
   - No Hardware Handshake
2. Press the Reset button on the supported Xplained Pro (XPro) platforms (see Table 2-1).
   Note: The device is now ready to be used as selected application and starts to scan or advertise on the button press. This button must be pressed only when the “Press button” is displayed on the console log window. The same button is pressed to stop the device scan or advertise.
3. The device is in advertising mode and the device initialization message is displayed on the console window.

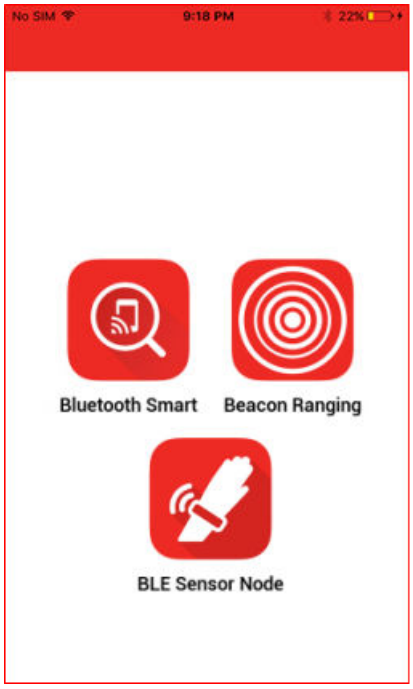
```
Initializing Application
BTLC1000 XPro Module: BTLC1000-ZR
BTLC1000 Host Interface UART Mode:4-Wire, Baudrate:921600
Initializing BTLC1000
BTLC1000 Chip ID: 0x2000B1
BD Address:0xF8F005F60515, Address Type:0
BluSDK Firmware Version:6.1.7035
BLE Started Advertisement
```

Pairing Procedure

Perform the following steps to pair the device with the smartphone application:

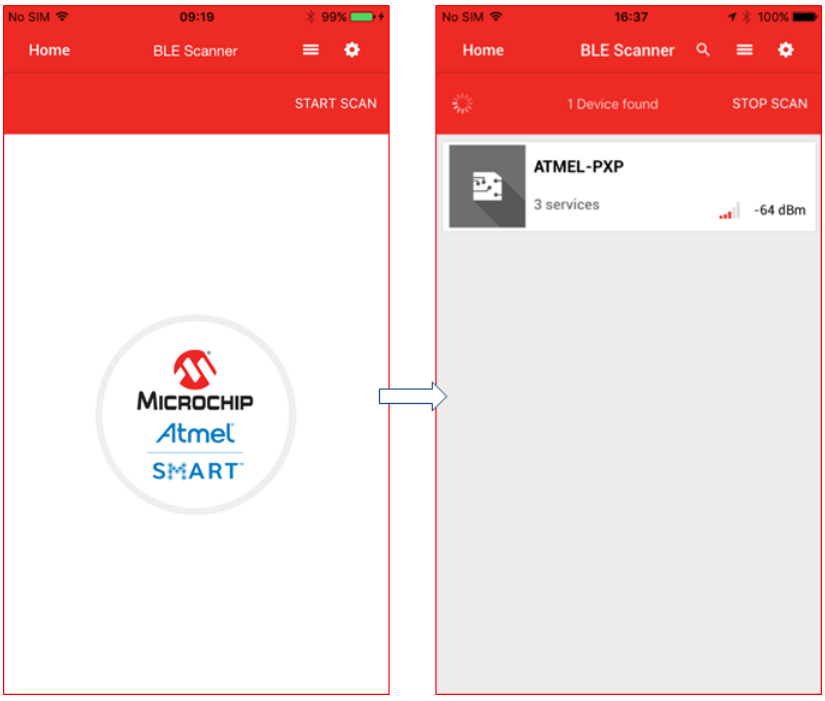
1. Open the Microchip SmartConnect application and click the **Bluetooth Smart** in an application dashboard as illustrated in the following figure.

Figure 5-2. Dashboard of Microchip SmartConnect Application



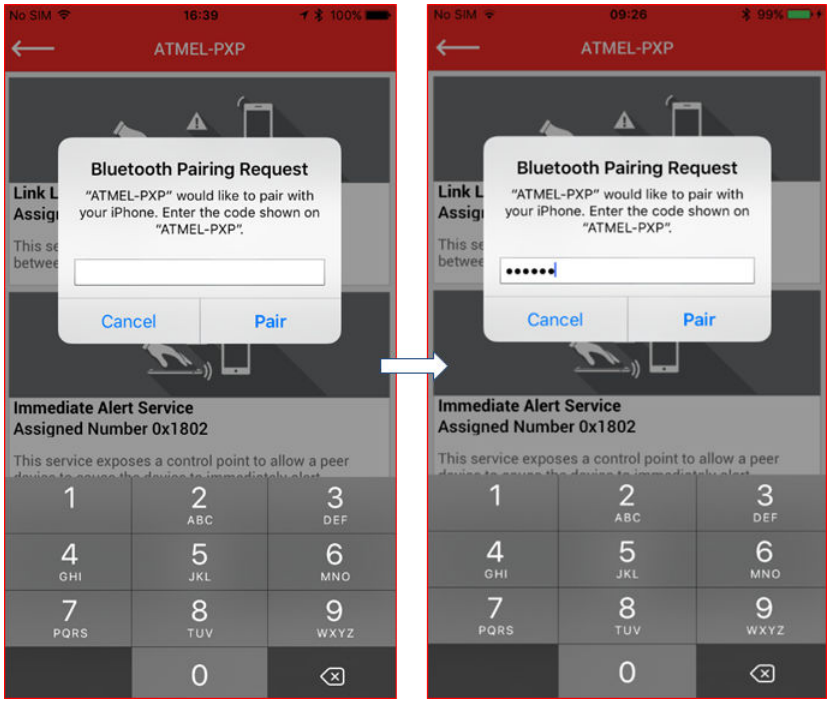
2. To scan for the peripheral devices, click the **START SCAN** option available in scanning page. The device name (for example, ATMEL-PXP) is displayed among the list of scanned devices.

Figure 5-3. Scanning for Devices



3. Select the device name in the scan results, which initiates the pairing procedure. Enter the pass-key “123456” on Bluetooth Pairing Request window and click **Pair**. The mobile app displays "Successful connection" upon successful completion of pairing.

Figure 5-4. Pairing Request



4. On the device side, the console displays the successful completion of the pairing procedure.
5. On the Microchip SmartConnect app, the supported services are displayed for the device.

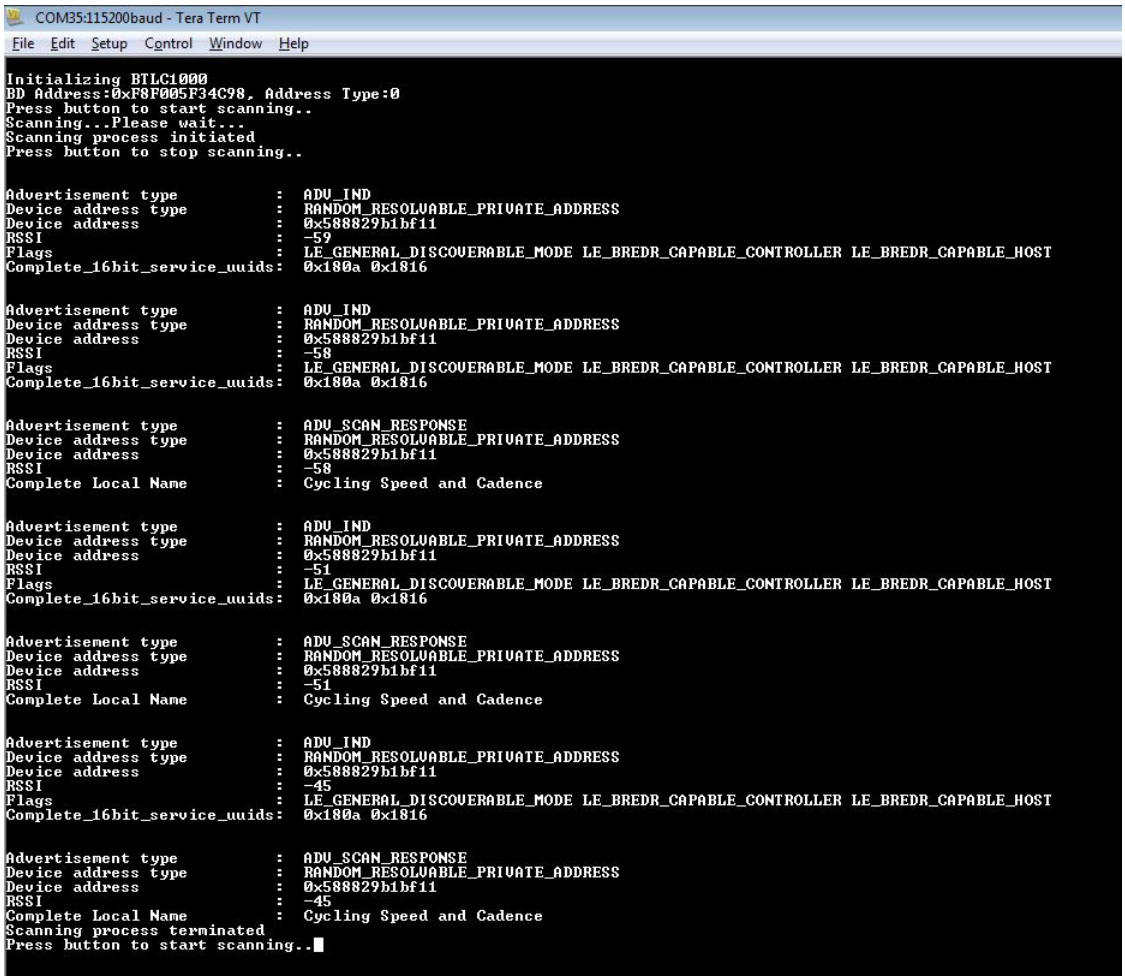
Figure 5-5. Display of Services Supported by the Application

5.3.1 Observer Application

Perform the following steps to run the Observer application demo:

1. Follow the steps (1 and 2) from Initializing the Device.
2. The device is now ready to be used as an Observer and starts to scan for nearby BLE devices.
3. The following figure shows example logs from the Observer application:

Figure 5-6. Observer Console Output

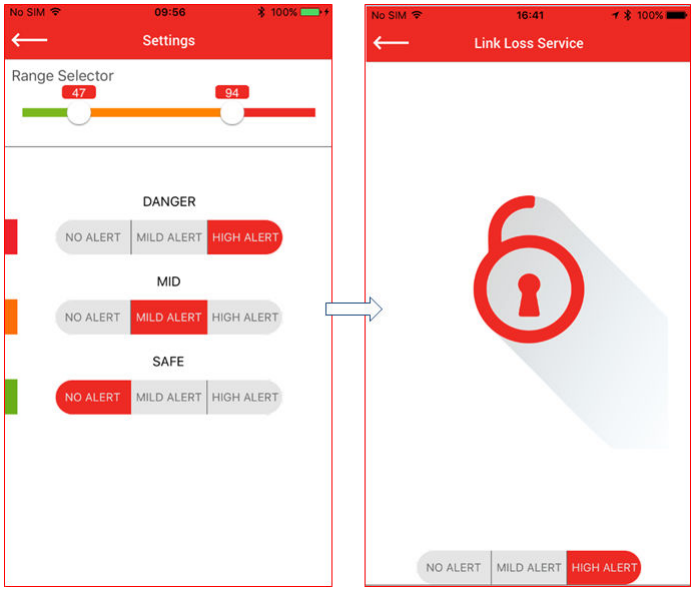


5.3.2 Proximity Reporter Application

Perform the following steps to run the Proximity Reporter application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. Select the desired service (Link Loss or Immediate Alert) for alert level characteristics configuration. Choose a value from the following:
   - HIGH ALERT
   - MILD ALERT
   - NO ALERT

Figure 5-7. Configuring Alert Level Settings



3. After configuration of the desired alert levels, click **Immediate Alert** service and then move the mobile phone away from the Proximity reporter. Based on the distance of separation, Path Loss is plotted on the zone radar (using RSSI values received from the Proximity Reporter). Based on the zone, the Proximity Monitor sends the corresponding alert level. The console log on the Proximity Reporter displays the corresponding alerts and on-board status LED behavior.

Figure 5-8. Proximity Reporter Path Loss Plot Across Safe, Mid, and Danger Zone

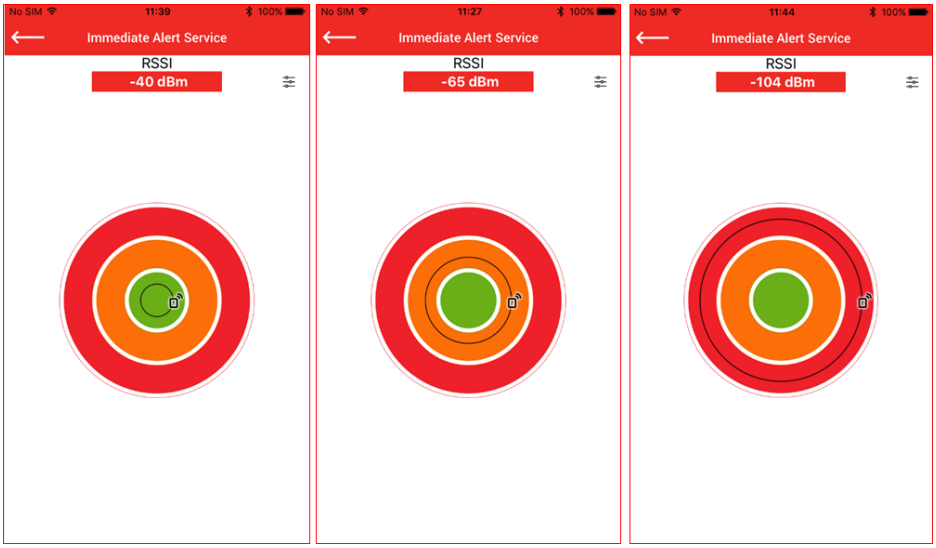
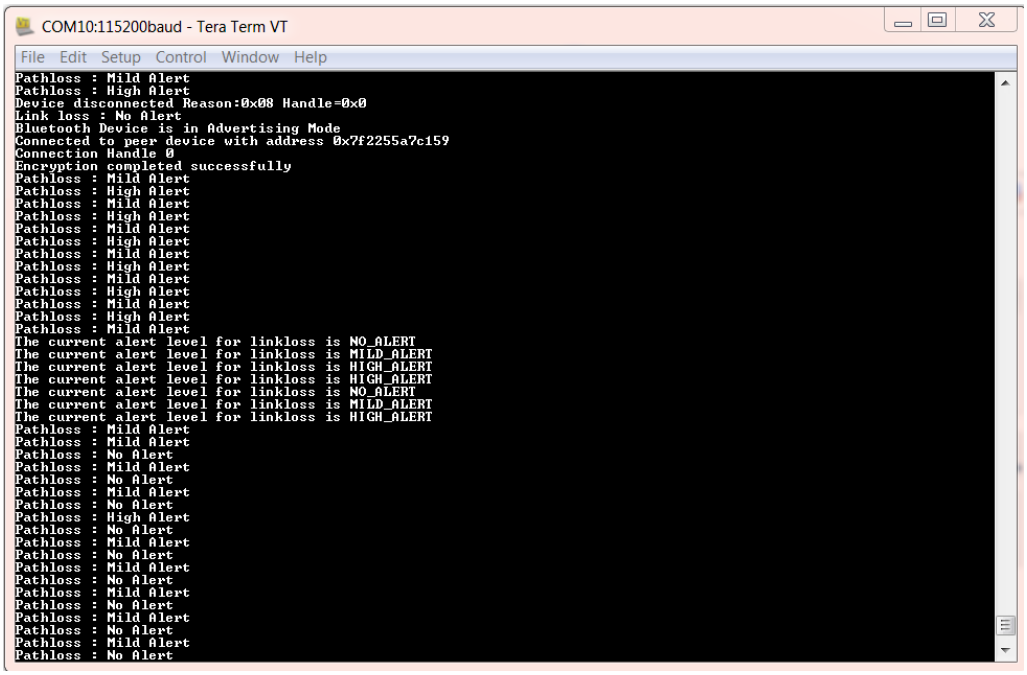


Figure 5-9. Proximity Reporter Path Loss Console Log Alerts Notification



4. After configuration of the desired alert levels, click on the **Link Loss** service and then move the mobile phone away from the reporter. Based on the distance of separation, the Proximity Reporter receives the path loss notifications based on the alert settings. Keep moving until the “Link Loss” pop-up appears. The console log on the Proximity Reporter displays the corresponding alerts and when Link Loss occurs, it reports the disconnection and the on-board status LED behavior. The lock screen emulates a common use-case application where the Link Loss service is used (for example, key fob). When the user is in close proximity, the lock remains open. Subsequently, the user moving out of range can be triggered to close the lock.

Figure 5-10. Link Loss Pop-up on Proximity Monitor

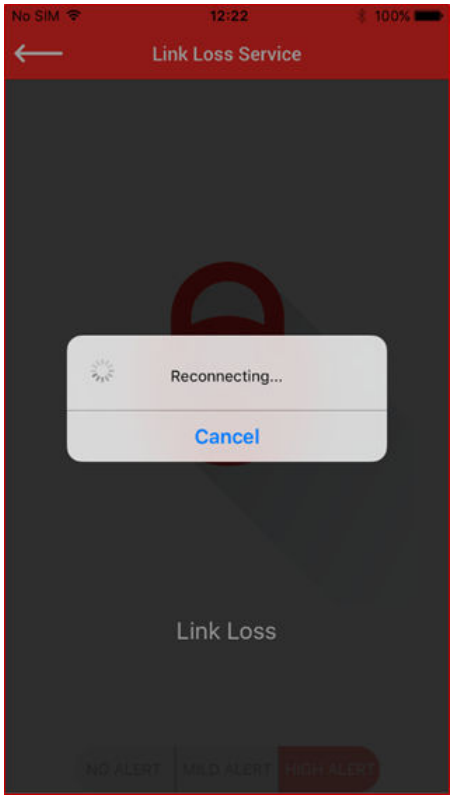
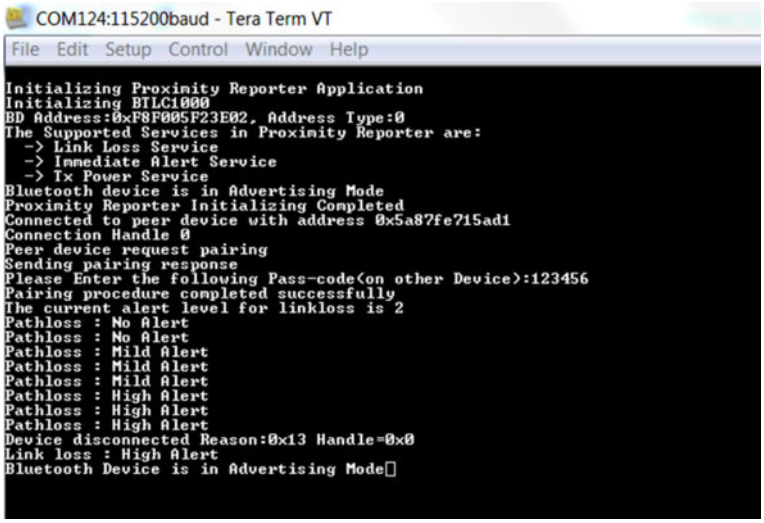


Figure 5-11. Proximity Reporter Console Log for Link Loss



5. After Link Loss, the mobile application attempts to reconnect to the Proximity Reporter. The connection is re-established by moving the mobile phone closer to the reporter.
6. The Tx Power service is used to retrieve the Tx Power of the Proximity Reporter. Click **Tx Power** service icon in the services screen. The Proximity Monitor reads the Tx Power value from the Proximity Reporter and displays the TX POWER LEVEL as shown in the following figure.

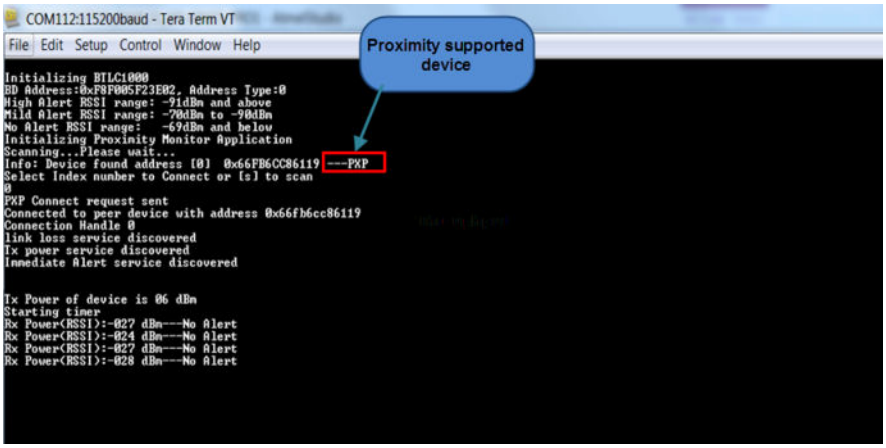
Figure 5-12. Proximity Monitor – Reading Tx Power Service

5.3.3 Proximity Monitor Application

Perform the following steps to run the Proximity Monitor application demo:

1. Connect one ATBTLC1000-MR/ZR device loaded with the Proximity Monitor example application code. Follow the steps (1 and 2) from Initializing the Device.
2. Setup another ATBTLC1000-MR/ZR device with the Proximity Reporter application. Follow all the steps from Initializing the Device. The device starts advertising.
3. The Proximity Monitor device then starts scanning for available devices in the vicinity and displays its Bluetooth Device Address (BD) in the console window. The Proximity Reporter device found during the scan is displayed in the console log window (refer to the following figure). Select the index number of that device to establish connection with it.

   Figure 5-13. Proximity Monitor Connection with a Proximity Reporter

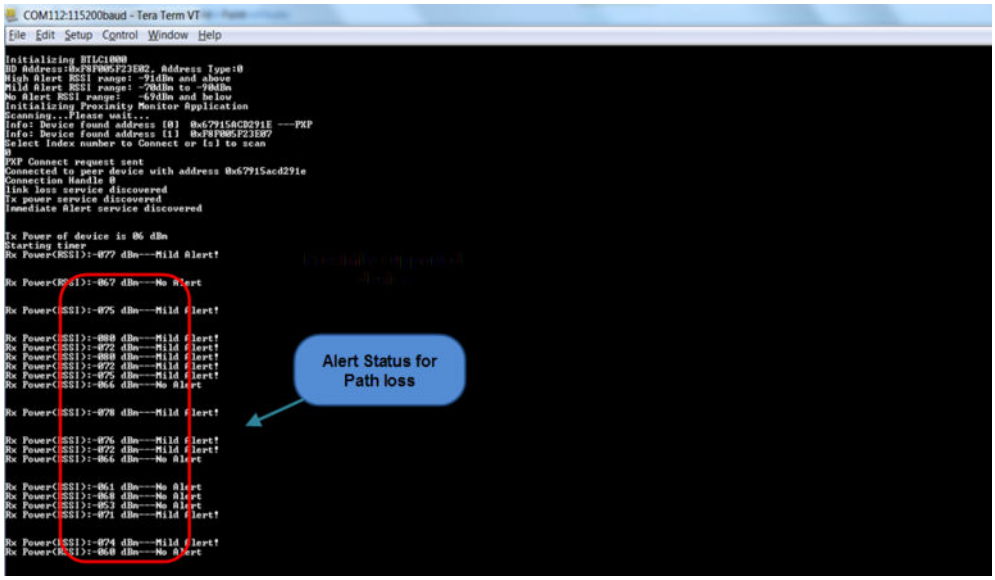


4. When the connection is established, the Proximity Monitor sets the link loss alert value to “HIGH ALERT” at the Proximity Reporter device. The Proximity Monitor also monitors the path loss, if the Proximity Reporter device supports the optional “Immediate Alert” service and “Tx Power” service. The Proximity Reporter example application supports both of these optional services.
   The default alert settings are as follows:

– For HIGH ALERT, set high alert RSSI to -91dBm and above, alert status is indicated by LED, which must be ON.
– For MILD ALERT, set RSSI to -70dBm to -90dBm, alert status is indicated by LED, which must be toggling.
– For NO ALERT, set RSSI to -69dBm and below, alert status is indicated by LED, which must be OFF.

If the reporter device moves out of the proximity of the monitor device, the path loss crosses the threshold values and the corresponding alert value is set; the alert notification is displayed on the console as shown below.

Figure 5-14. Proximity Monitor Setting Alert Levels



5.3.4 ANCS Application

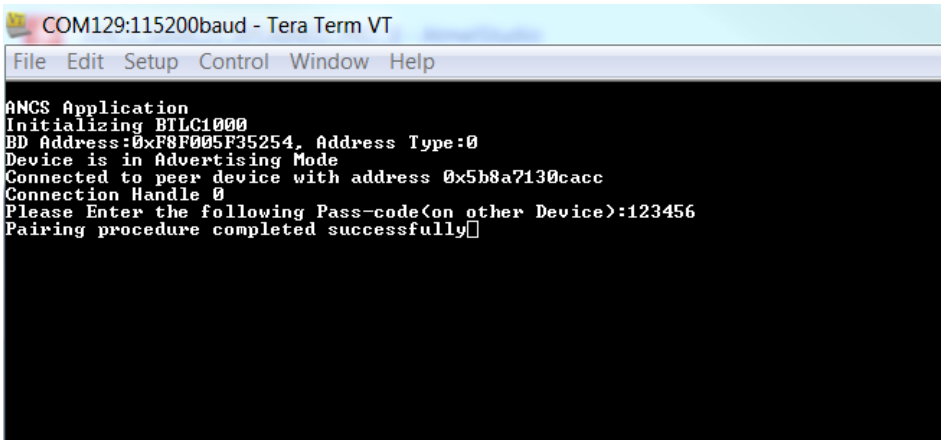
Perform the following steps to run the ANCS application demo:

1. Follow the steps from Initializing the Device.
2. Enable Bluetooth from the Settings page of iPhone. The phone starts to scan for the devices. ATMEL-ANCS appears among the list of devices scanned. Click the **ATMEL-ANCS** to connect to the device.

Figure 5-15. ANCS Device Discovery in iPhone

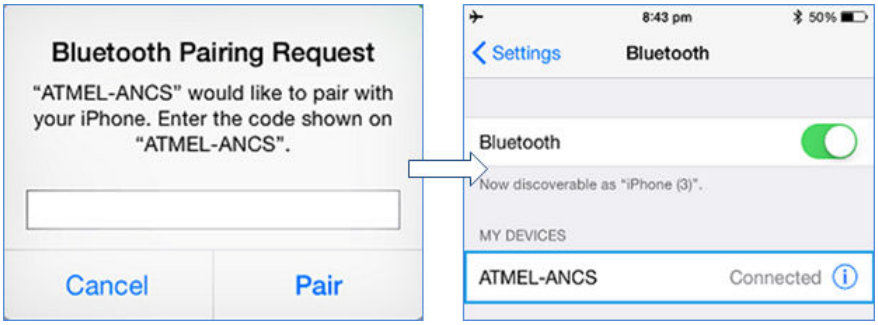
3. When connected, the client side initiates a pairing request with the iPhone. The console log provides guidance for the user to enter the pass-key on the iPhone.

Figure 5-16. Console Display for Pairing in ANCS



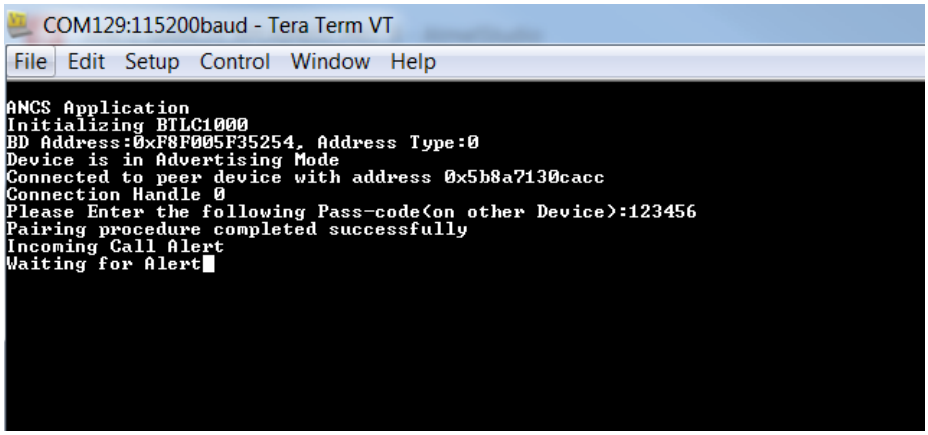
4. Enter the pass-key displayed in the console log on the Bluetooth Pairing Request window of the iPhone and click **Pair**. After the device is connected, “ATMEL-ANCS” appears in the MY DEVICES section on the iPhone.

Figure 5-17. Pairing and Connecting iPhone to ATMEL-ANCS



5. Now, the user can initiate a mobile terminated call to the iPhone. When the iPhone receives a call, the corresponding incoming call alert is indicated on the device side console log window. Once the call is terminated, the device waits for a new alert to occur, as shown in the following screen.

Figure 5-18. Console Display for Notification Received as Incoming Call Alert

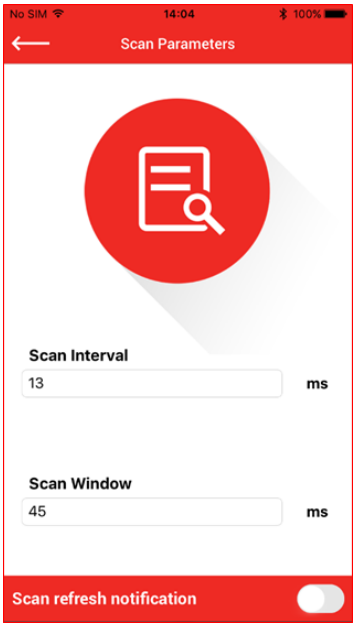


5.3.5 Scan Parameters Application

Perform the following steps to run the Scan Parameters application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. When paired, the application displays the Scan Parameters and the Generic Information service.
3. Click the **Scan Parameters** service. The user receives a notification for the scan refresh characteristic value. The user can disable the notification in the Scan Parameters page, refer the following figure.

Figure 5-19. Scan Refresh Characteristic Notification Options



4. The user can set appropriate value for the Scan Interval and Scan Window characteristics.
5. The newly updated values of Scan Interval and Scan Window must be displayed on the console log of the device side as shown in the following figure.

Figure 5-20. Updated Scan Interval/Window Characteristic Value on Device

```
Initializing Scan Parameter Application
Initializing BTLC1000
BD Address:0xF8F005F23DED, Address Type:0
BLE Started Adv
Connected to peer device with address 0x4e3cd73c6fcc
Connection Handle 0
Peer device request pairing
Sending pairing response
Please Enter the following Pass-code(on other Device):123456
Pairing procedure completed successfully
New scan interval window parameter
Scan Interval 3 ms
Scan Window   3 ms
Scan Refresh Characteristic Value: 0
New scan interval window parameter
Scan Interval 3 ms
Scan Window   3 ms
Scan Refresh Characteristic Value: 0
New scan interval window parameter
Scan Interval 12 ms
Scan Window   3 ms
Scan Refresh Characteristic Value: 0
New scan interval window parameter
Scan Interval 12 ms
Scan Window   3 ms
Device disconnected Reason:0x13 Handle=0x0
BLE Started Adv
```

5.3.6 Time Information Profile Application

Perform the following steps to run the Time Information Profile application demo:

1. Follow steps 1 through 4 from the ANCS Application.
2. Press the SW0 button on the device to read the internally supported characteristic values from the iPhone.

3. The console log on the device side displays the values for all characteristics supported by the iPhone internally.

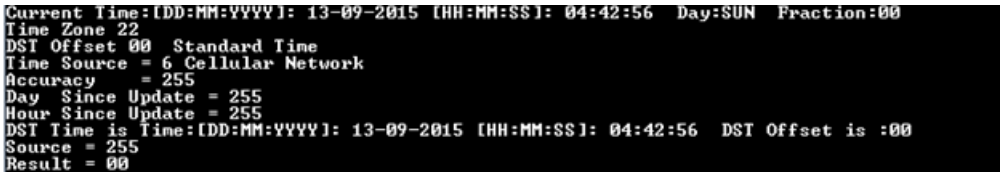
Figure 5-21. Console Display – Date, Time, and Day Information

5.3.6.1 Running the Demo for Android devices

Perform the following steps to run the Time Information Profile application demo for Android devices:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. After the device is connected, the application displays Continuous Time Service, Next DST Change Service, and Reference Time Update Service.
3. The user has to click on the services to read the characteristic values.
4. Press the SW0 button on the supported platform device to read the internally supported characteristic values from the Android device.
5. The console log on the device side displays the values for all the characteristics supported by the device.

Figure 5-22. Console Display - All Supported Characteristic Values

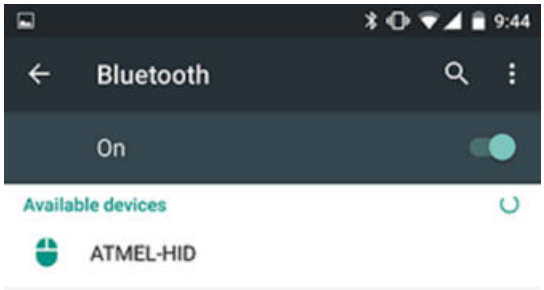


5.3.7 HID Mouse Device Application

Perform the following steps to run the HID Mouse Device application demo:

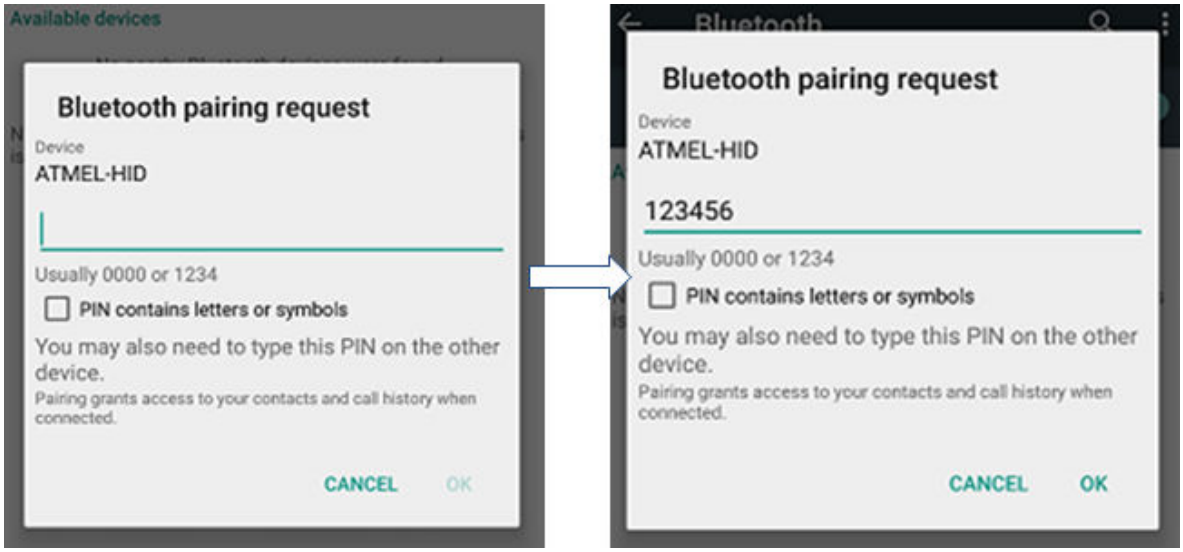
1. Follow the steps from Initializing the Device.
2. In this demonstration, an Android device supporting HOGP is used. The HOGP profile is natively supported in Android version 4.4 (Android KitKat) and higher. The mobile phone must include a Bluetooth chipset supporting Bluetooth 4.0 or higher. On the mobile phone, enable Bluetooth in the Settings page to scan for the devices. “ATMEL-HID” appears among the list of scanned devices. Select **ATMEL-HID** to connect to the supported platform device.

Figure 5-23. HID (Mouse) Device Discovery on Bluetooth Settings Page



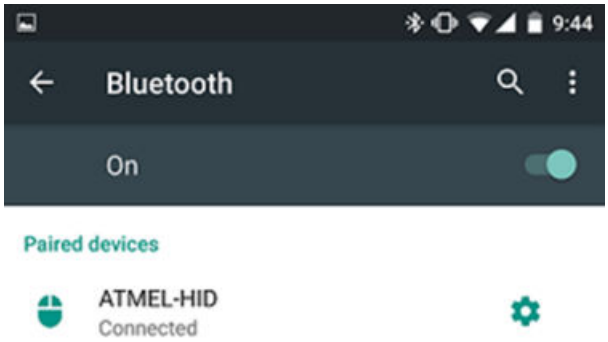
3. Click **ATMEL-HID** to start the pairing procedure. A pop-up requesting the pass-key appears. Enter pass-key “123456” and click **Pair**.

Figure 5-24. Bluetooth Pairing Request



4. After pairing is complete, the connected device is listed under Paired device.

Figure 5-25. Paired Devices



5. The HID device side for the pairing and connection procedure is shown in the console log.

Figure 5-26. HID Mouse Device Console Log

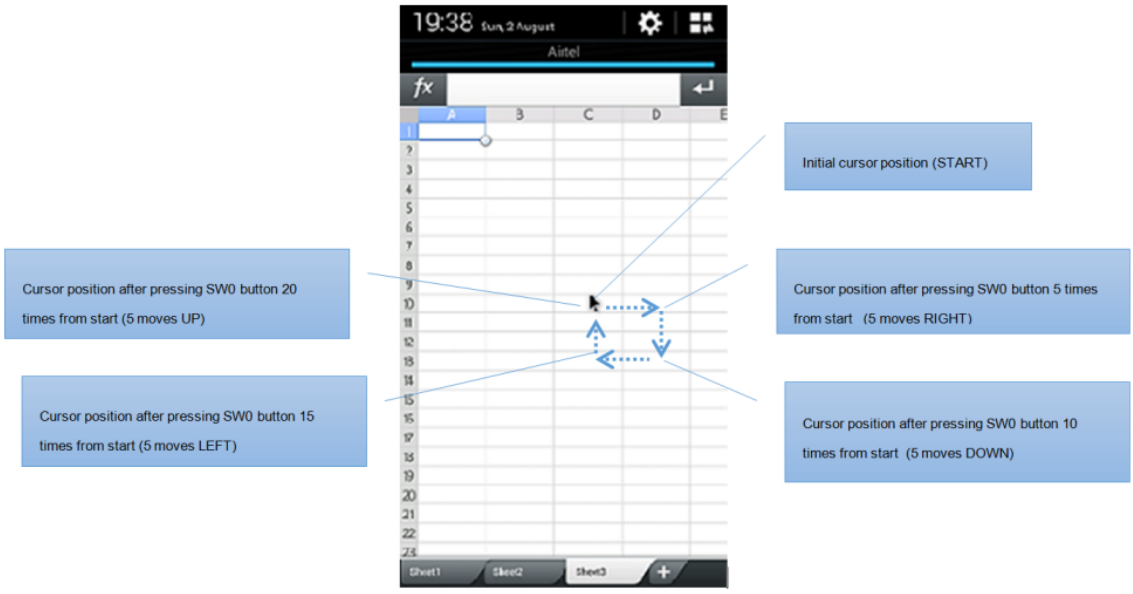
```
Initializing HID Mouse Application
HID Profile Configured
Initializing BTLC1000
BD Address:0xF8F005F35166, Address Type:0
Device Started Advertisement
Connected to peer device with address 0x42ca836224e4
Connection Handle 0
Please Enter the following Pass-code(on other Device):123456
Pairing procedure completed successfully
```

6. After the device is connected to the mobile phone, the user can click on the SW0 button to simulate mouse movement.
7. For every press of the button, the user can see a corresponding cursor movement on the HID host as described below:
   - First 5 button presses – cursor moves right
   - Next 5 button presses – cursor moved down
   - Next 5 button presses – cursor moves left
   - Next 5 button presses – cursor moved up

The same sequence is repeated based on user input. The console log is shown in the following screen.

Figure 5-27. HID Device Console Log for Movement

Figure 5-28. Mouse Movement Simulation

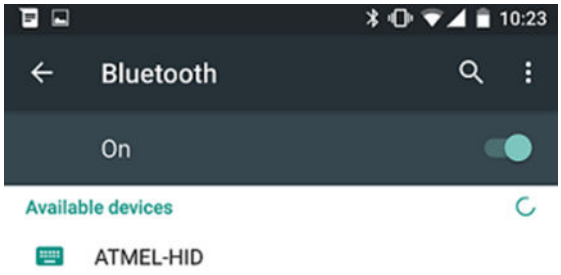


5.3.8 HID Keyboard Device Application

Perform the following steps to run the HID Keyboard device application demo:

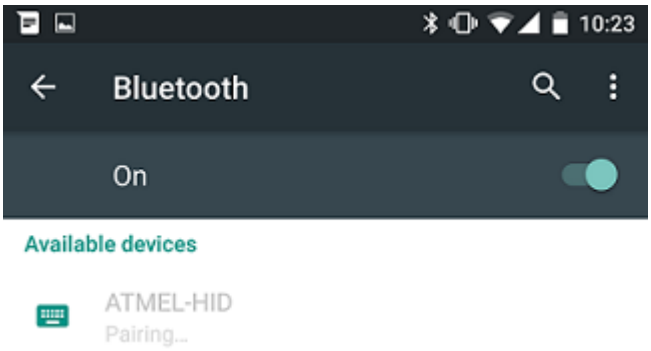
1. Follow the steps from Initializing the Device.
2. In this demonstration, an Android device supporting HOGP is used. The HOGP profile is natively supported in Android version 4.4 (Android KitKat) and higher. The mobile phone must include a Bluetooth chip-set supporting Bluetooth 4.0 or higher. On the mobile phone, enable Bluetooth in the Settings page to scan for the devices. “ATMEL-HID” appears among the list of scanned devices. Select **ATMEL-HID** to connect to the supported platform device.

Figure 5-29. HID (Keyboard) Device Discovery on Bluetooth Settings Page



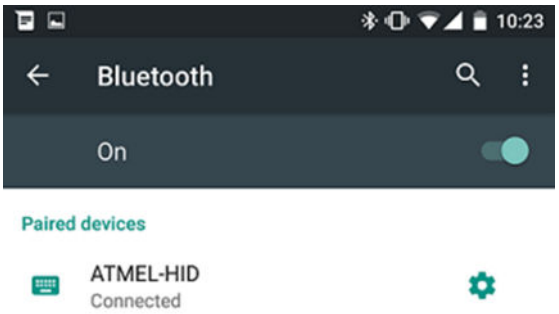
3. Click **ATMEL-HID** to initiate the pairing procedure.

Figure 5-30. Pairing Procedure with HID Device



4. After pairing is complete, the connected device is listed under Paired device.

Figure 5-31. Paired Devices



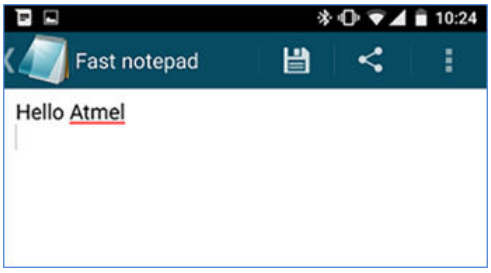
5. The HID device side for the pairing and connection procedure is shown in the console log.

Figure 5-32. HID Keyboard Device Console Log

```
Initializing HID Keyboard Application
HID Profile Configured
Initializing BTLC1000
BD Address:0xF8F005F35166, Address Type:0
Device Started Advertisement
Connected to peer device with address 0x5ac088408fdc
Connection Handle 0
Pairing procedure completed successfully
```

6. After the device is connected, start any notepad application on the mobile phone.
7. Click the **SW0** button on the supported platform device.
8. The user can see a letter for each press in the application “Fast notepad”.
9. The user can see a complete “Hello Atmel” in the application as shown in the following screen.

Figure 5-33. Message Displayed in the Application

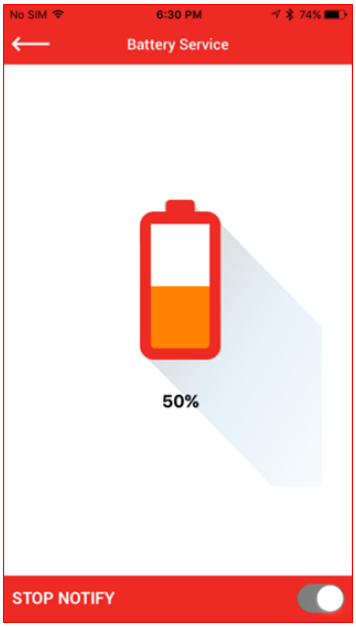


5.3.9 Battery Service Application

Perform the following steps to run the Battery Service Application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. When paired, the application displays the Battery Service and the Generic Information service.
3. Select “Battery Service” to receive notifications for the battery level characteristic. The user can stop receiving the notifications by disabling notifications, as shown in the following figure.

   Figure 5-34. Battery Level Characteristic Notification Options



4. On the device side, the console log displays the periodic battery level updates.

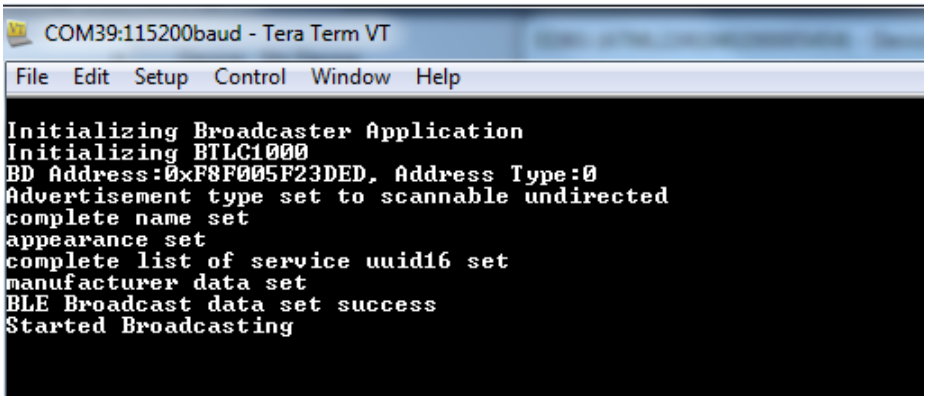
```
Battery Level:0%
Battery Level:1%
Battery Level:2%
Battery Level:3%
Battery Level:4%
Battery Level:5%
```

5.3.10 Simple Broadcaster Application

Perform the following steps to run the Simple Broadcaster application demo:

1. Follow the steps (1 and 2) from Initializing the Device.
2. The device is in advertising mode.
3. The following figure shows example logs from the Simple Broadcaster application.

Figure 5-35. Simple Broadcaster Console Display

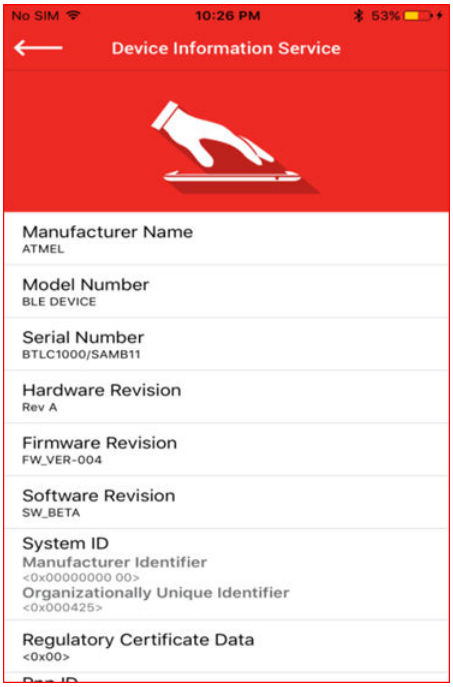


5.3.11 Device Information Service Application

Perform the following steps to run the Device Information Service application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. When paired, the application displays the Device Information Service.
3. When the Device Information Service is selected, the user can view the device information service characteristics as shown in the following screen.

   Figure 5-36. Display of Device Information Service Characteristics



4. The user can refresh the page to get the updated characteristic value of all characteristics.
5. On the device side the console log is displayed as:

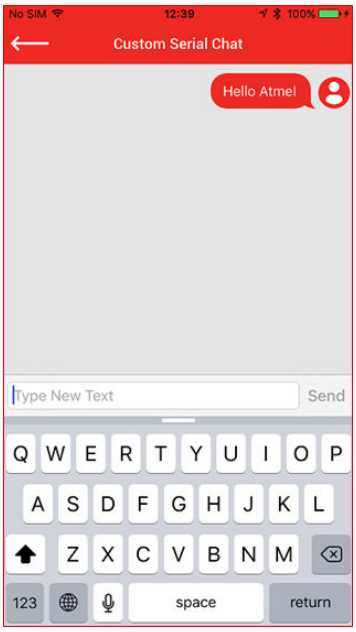
```
Updating Firmware to ver:FW_VER-000
Updating Firmware to ver:FW_VER-001
Updating Firmware to ver:FW_VER-002
Updating Firmware to ver:FW_VER-003
```

5.3.12 Custom Serial Chat Profile Application

Perform the following steps to run the Custom Serial Chat Profile application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. Once pairing is complete, the Custom Serial Chat icon appears on the service list page.
3. Click the **Custom Serial Chat** icon. The chat screen appears where the user can type the text that is to be sent to the remote device and also see the text coming from the remote device.
4. Chat text “Hello Atmel” send to remote device.

Figure 5-37. Sending Data to Device



5. The user can also write the text on the console for the device and press the ENTER key for transmitting the chat text to the mobile application.

Figure 5-38. Console Log for Sending Data to Remote Device

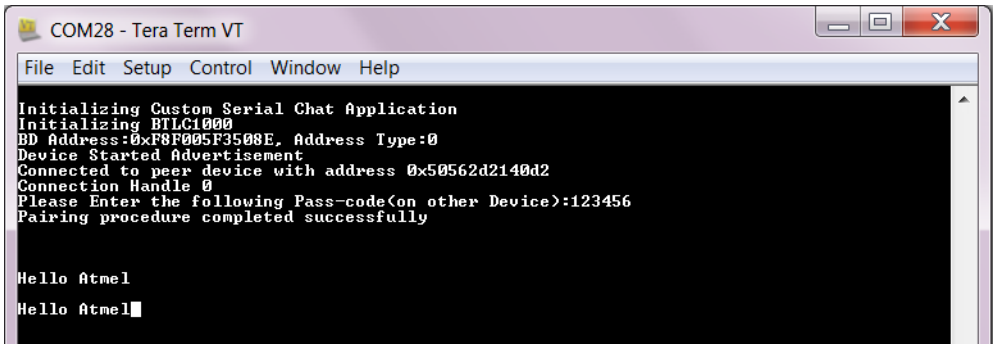
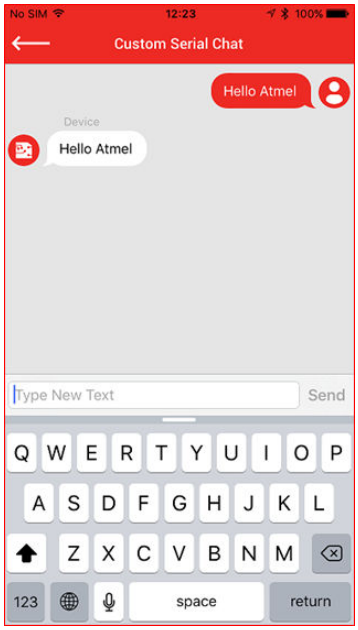


Figure 5-39. Chat Text Received from ATBTLC1000



Note: For more information on the Custom Serial Chat service, refer to the Custom Serial Chat Service Specification.

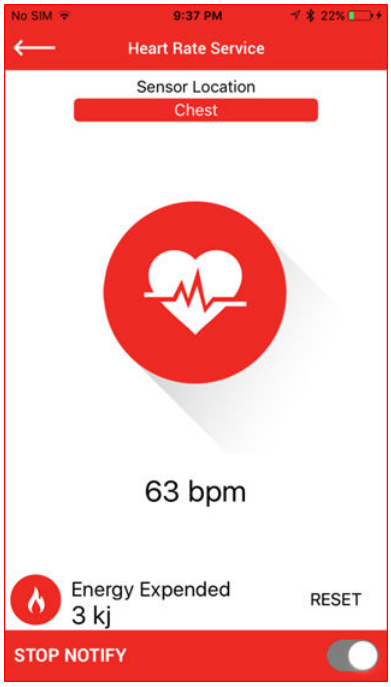
5.3.13 Heart Rate Profile Application

Perform the following steps to run the Heart Rate Profile application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. After the device is connected, the application displays the Heart Rate and the Device Information service is displayed.
3. When notifications are enabled, the HRM values are displayed, as shown in the console and the corresponding mobile app. The LED on the SAM L21 board starts blinking while sending notifications.

```
Notification Enabled
Heart Rate: 50 bpm        RR Values:<100,300>msec User Status:Idle
Heart Rate: 51 bpm        RR Values:<500,700>msec User Status:Idle
Heart Rate: 52 bpm        RR Values:<900,1100>msec User Status:Idle
Heart Rate: 53 bpm        RR Values:<100,300>msec User Status:Idle
Heart Rate: 54 bpm        RR Values:<500,700>msec User Status:Idle
Heart Rate: 55 bpm        RR Values:<900,1100>msec User Status:Idle
Heart Rate: 56 bpm        RR Values:<100,300>msec User Status:Idle
Heart Rate: 57 bpm        RR Values:<500,700>msec User Status:Idle
Heart Rate: 58 bpm        RR Values:<900,1100>msec User Status:Idle
Heart Rate: 59 bpm        RR Values:<100,300>msec User Status:Idle
Energy Expended :3KJ
```

Figure 5-40. Displaying Heart Rate Measurements



4. When the user disable on Stop Notify, the notifications are displayed in the console logs as:

```
Notification Disabled
```

5. During the connection, the SW0 button is used to disconnect the connection. If no connection exists, the SW0 button is used to start advertisement.

5.3.14 Blood Pressure Profile Application

Perform the following steps to run the Blood Pressure Profile application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. When the device is connected, the application displays Blood Pressure, Device Information Service and Generic Information.
3. On entering the Blood Pressure service page, the mobile application enables the notifications and indications for interim cuff pressure and blood pressure characteristics, respectively. The blood pressure sensor device simulated by the device, sends the current blood pressure values after receiving the indications enabling request. The corresponding console logs and mobile application screen are shown in the following screen.

Figure 5-41. Console Log for Blood Pressure Measurements

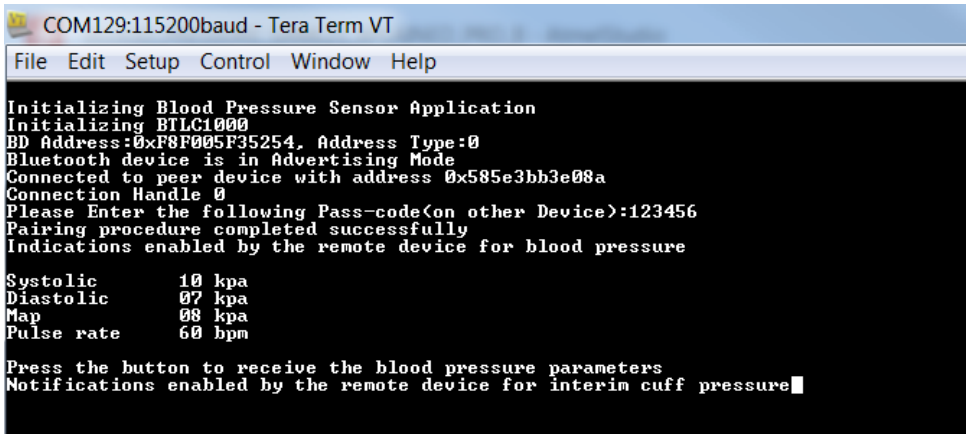
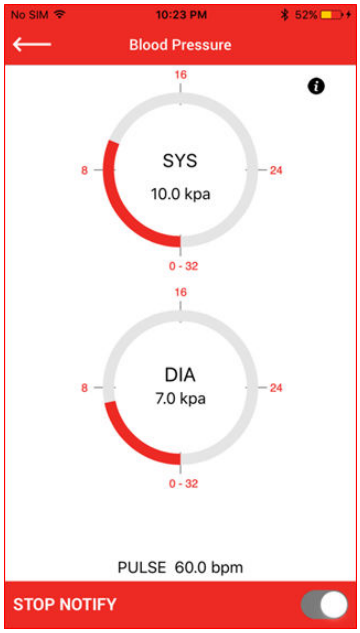


Figure 5-42. Blood Pressure Service Page after Receiving BP Indications



4. The SW0 button can be used on the SAM L21 to receive updated blood pressure measurements. The blood pressure sensor first sends the interim cuff pressure values as notifications and then sends the final blood pressure measurements as indication. The blood pressure measurements sent by the blood pressure sensor are simulated values. The following figures demonstrate the scenario after a SW0 button press.

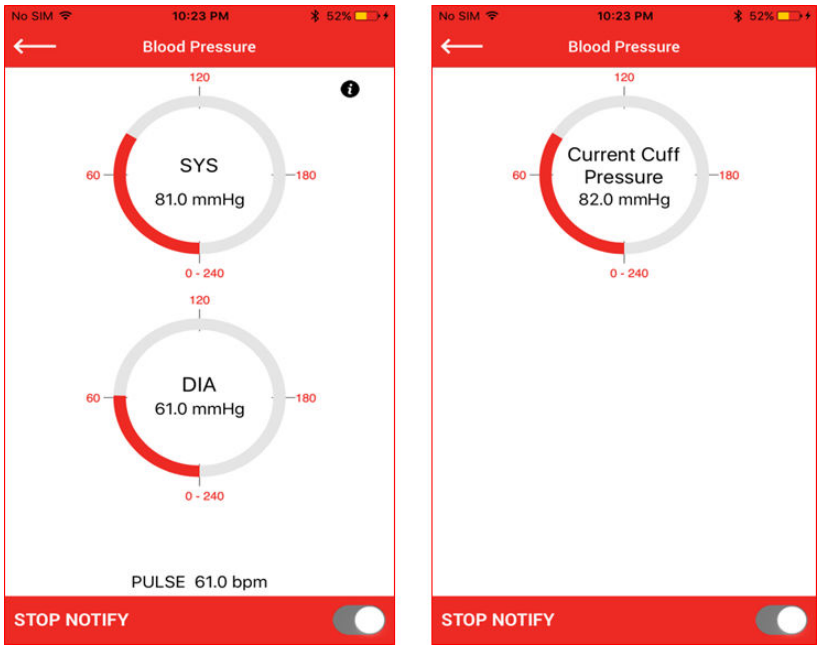
Figure 5-43. Console Log for Blood Pressure Values after Button Press

```
Started sending Interim Cuff Pressure Values
Cuff pressure  80 mmhg
Cuff pressure  81 mmhg
Cuff pressure  82 mmhg
Cuff pressure  83 mmhg
Cuff pressure  84 mmhg
Cuff pressure  85 mmhg
Cuff pressure  86 mmhg
Cuff pressure  87 mmhg
Cuff pressure  88 mmhg


The Blood Pressure Values are:
Systolic       81 mmhg
Diastolic      61 mmhg
Map            71 mmhg
Pulserate      61 bpm

Press the button to receive the blood pressure parameters
```

Figure 5-44. Blood Pressure Service Pages after Receiving Measurement Data on Button Press

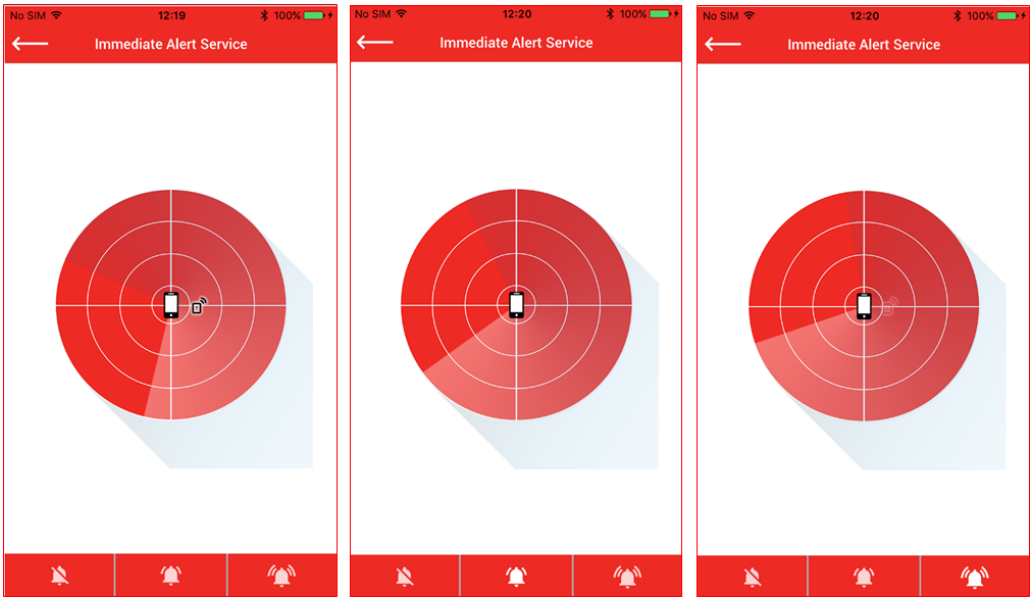


5.3.15 Find Me Profile Application

Perform the following steps to run the Find Me Profile application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. After the device is connected, the application displays a service page that includes Immediate Alert Service and Generic Information.
3. Since the service level connection is established, the user can see the notifications based on the alert level settings as depicted in the following figures.

Figure 5-45. Sending Alerts to Find Me Target ATMEL-FMP



4. On the device side, the console log is displayed as:

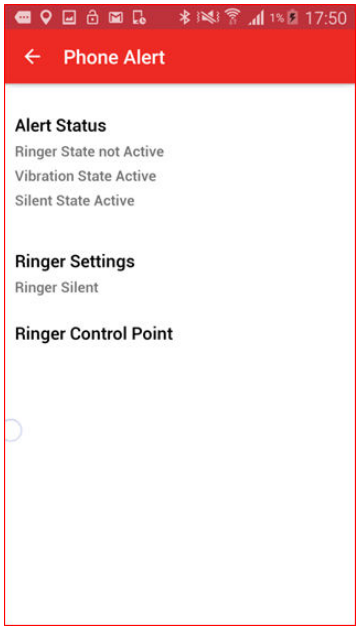
```
Find Me : Mild Alert
Find Me : High Alert
Find Me : No Alert
```

5.3.16 Phone Alert Status Application

Perform the following steps to run the Phone Alert Status application demo:

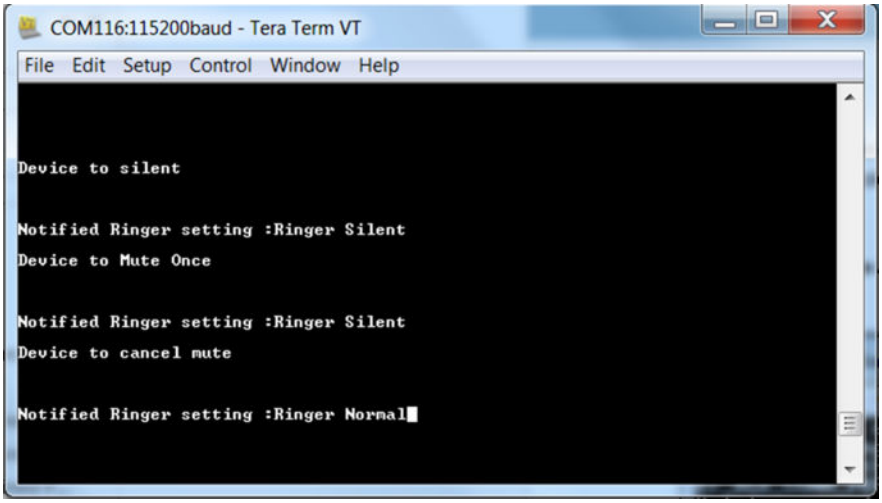
1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. After connection, the application displays the Phone Alert Status Service page.
3. Notifications are automatically enabled and the application reads the values of the “Alert Status”, “Ringer Settings”, and “Ringer Control Point” characteristics, which are updated on the mobile application, as illustrated in the following figure.

Figure 5-46. Displaying the Characteristics of the Phone Alert Service



4. Press the SW0 button. The device is set to different modes by using the notifications and the corresponding console logs are displayed.

Figure 5-47. Phone Alert Status Console Log

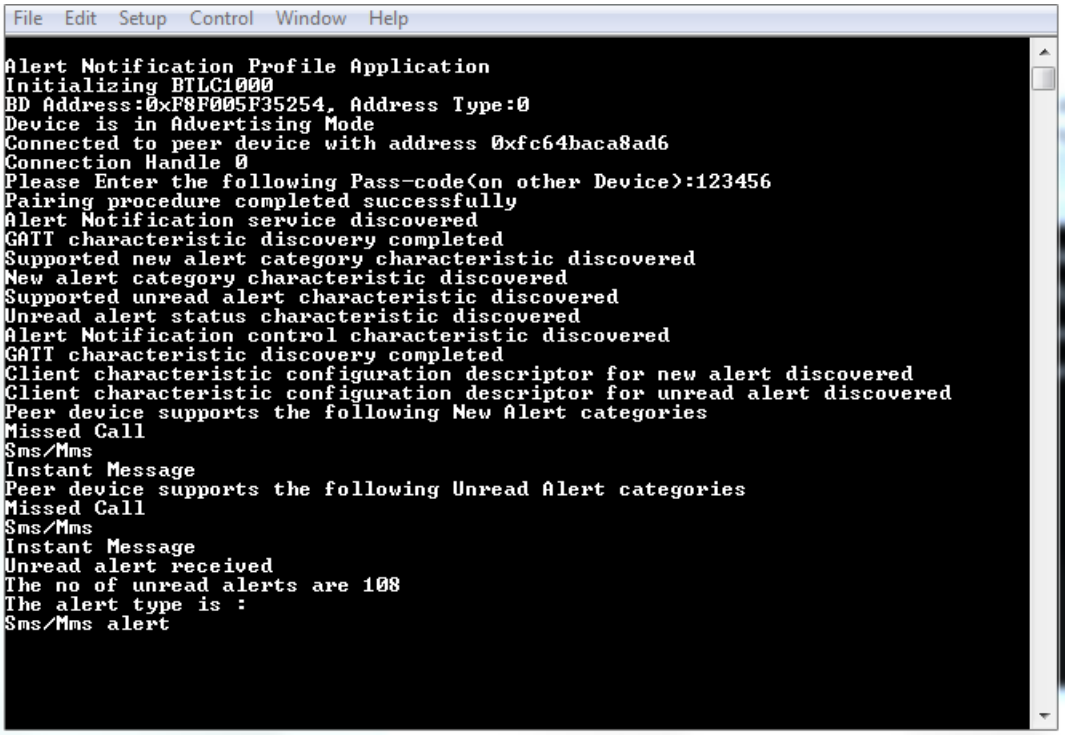


5.3.17 Alert Notification Profile Application

Perform the following steps to run the Alert Notification Profile application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.
2. When connected, the application displays the Alert Notification service page. The console log displays the new and unread alert categories.

Figure 5-48. Alert Notification Categories

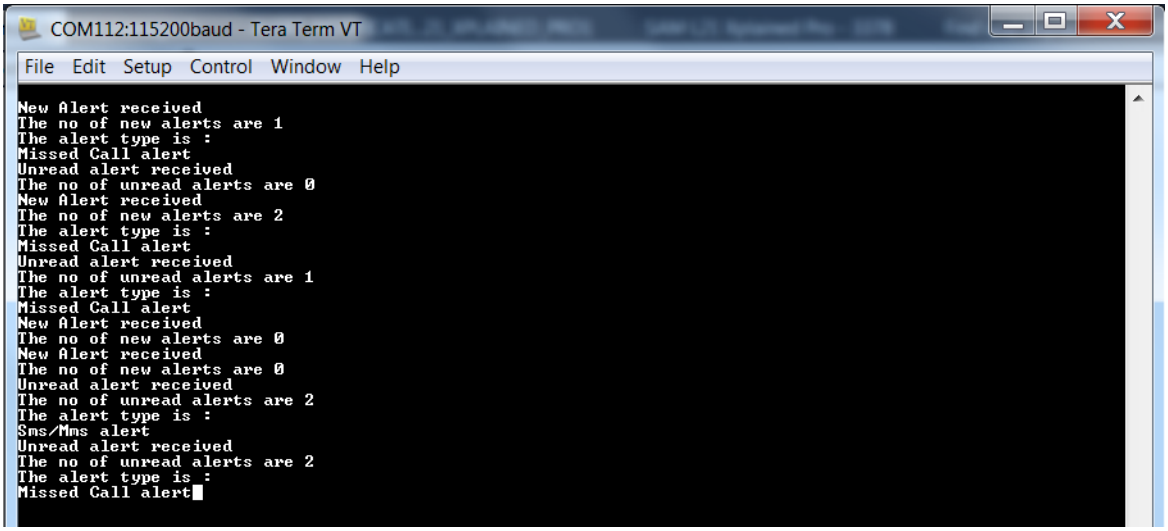


3. Enable the notifications by using the SW0 button. The mobile application reflects the status as shown.

Figure 5-49. Alert Notification Screen on Microchip SmartConnect Application

4. The user can trigger a missed call to the Android device or send an SMS. The corresponding notification then gets displayed on the device side in the console logs.

Figure 5-50. Console Display for Missed Call Alert and SMS Alert Notifications

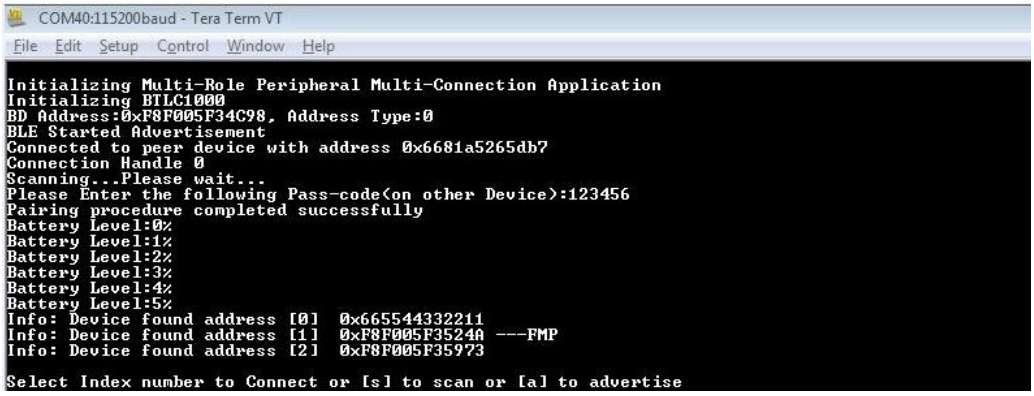


5.3.18 Multi-Role Peripheral Multi-Connect Application

Perform the following steps to run the Multi-Role Peripheral Multi-Connect application demo:

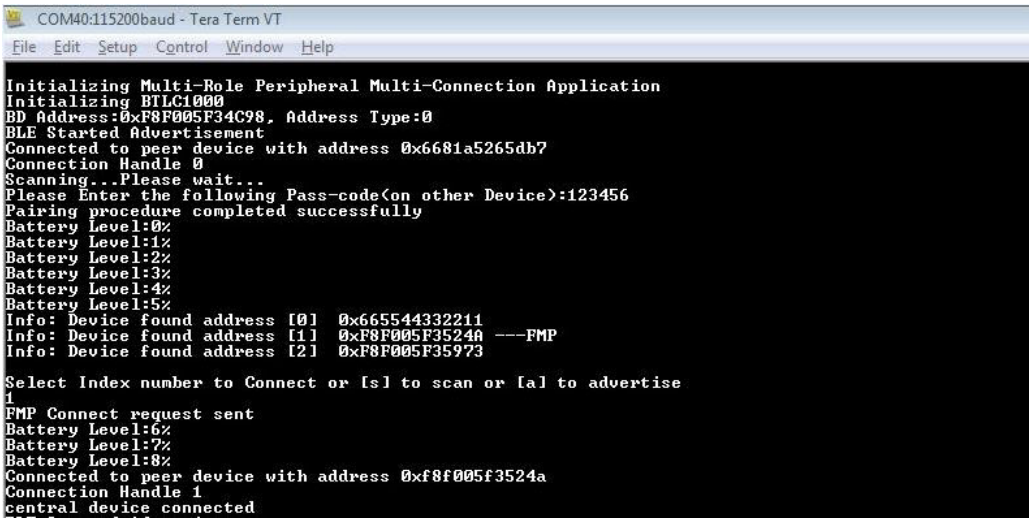
1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo. The device initially acts as a GAP Peripheral and starts advertisement with Battery Service UUID in the advertisement data. Perform all the steps from Battery Service Application.
2. The device starts scanning and displays the devices found, as shown in the following figure.

   Figure 5-51. Multi-Role Peripheral Multi-Connect Application – Scanning Devices



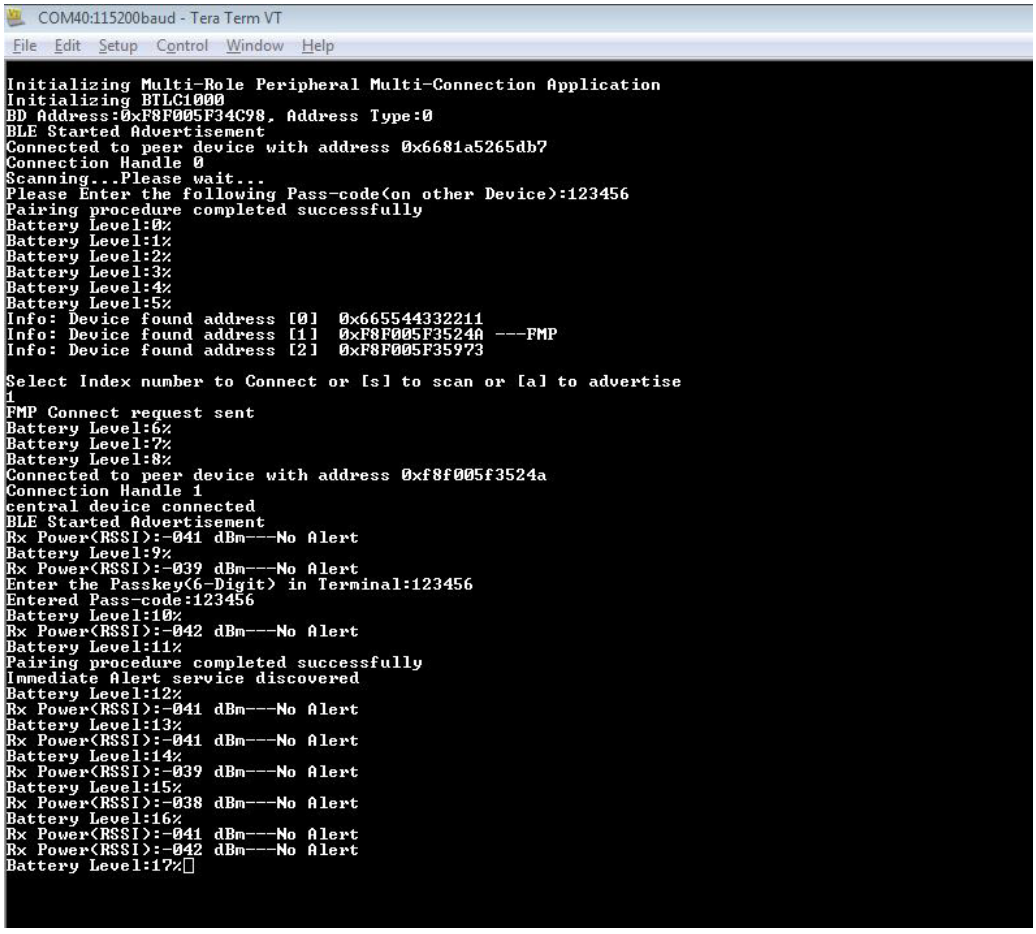
3. Setup another ATBTLC1000-MR/ZR device with the Find Me application example loaded. Follow the steps (1 through 3) from Running the Demo. The device starts advertising.
4. The GAP Central (Find Me Locator) scans and then displays the list of all BLE devices that are advertising. Find Me Target devices (GATT server role) are indicated with tag “---FMP”. Select the appropriate index number for the Find Me Target. GAP Central (Find Me Locator) connects to the selected peer device.

Figure 5-52. Connecting GAP Central (Find Me Locator) with GAP Peripheral (Find Me Target)



5. The ATBTLC1000 as a GAP Central pairs with the connected peripheral. The ATBTLC1000-MR/ZR then acts as a GAP Peripheral by advertising with Battery Service UUID in the advertisement data. Now the ATBTLC1000-MR/ZR sends alert levels as a GAP Central, sends battery level notifications to the device connected as a GAP Peripheral and also starts advertising with Connectable advertisement packets.

Figure 5-53. Device acting as Multi-Role to accept connections from GAP Central devices (Mobile)



6. The ATBTLC1000-MR/ZR acting as a GAP Peripheral (BAS) can connect to seven GAP central devices (mobile devices through the Microchip SmartConnect application). Now, the ATBTLC1000-MR/ZR continues to behave as Find Me Locator (GAP Central) and Battery Service Application (GAP Peripheral) simultaneously with eight active connections. Continuous data transfer happens on all the links by the ATBTLC1000-MR/ZR and even if one link gets disconnected, the data transfer happens on the other links.

5.3.19 L2CAP Throughput Application

This demonstration requires two ATBTLC1000-MR/ZR devices. Program one ATBTLC1000-MR/ZR device with the L2CAP Peripheral and another one with the L2CAP Central application example. Perform the following steps to run the Throughput application demo:

1. Follow the steps (1 and 2) from Initializing the Device for both devices.
2. The device initializes and start-up.
3. The Central device starts scanning and subsequently connects with the desired peripheral device. The following log shows that both devices connected to confirm the connection status.

Figure 5-54. L2CAP Central Connection with L2CAP Peripheral

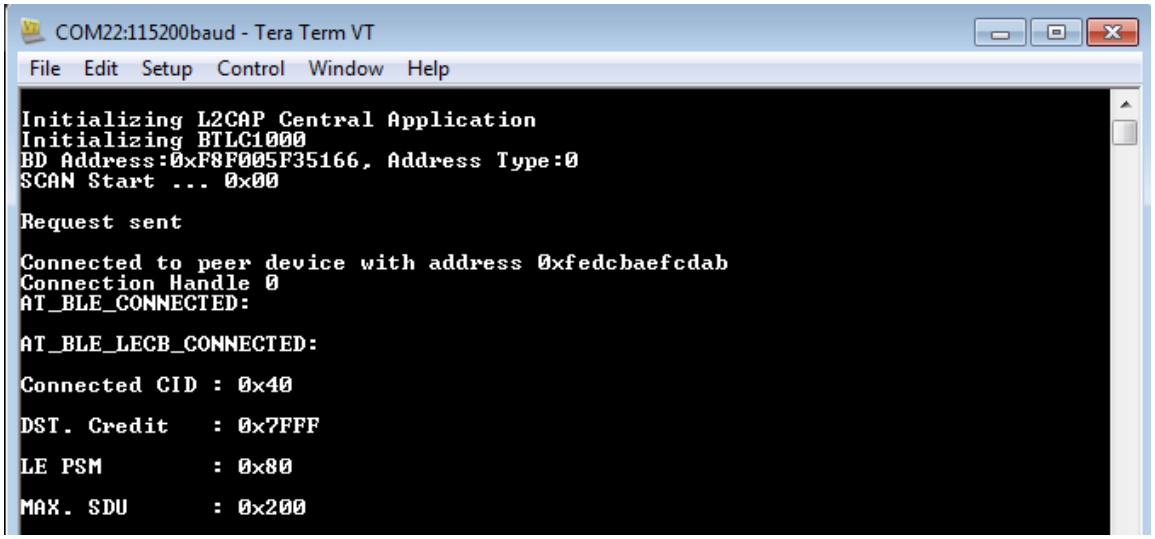
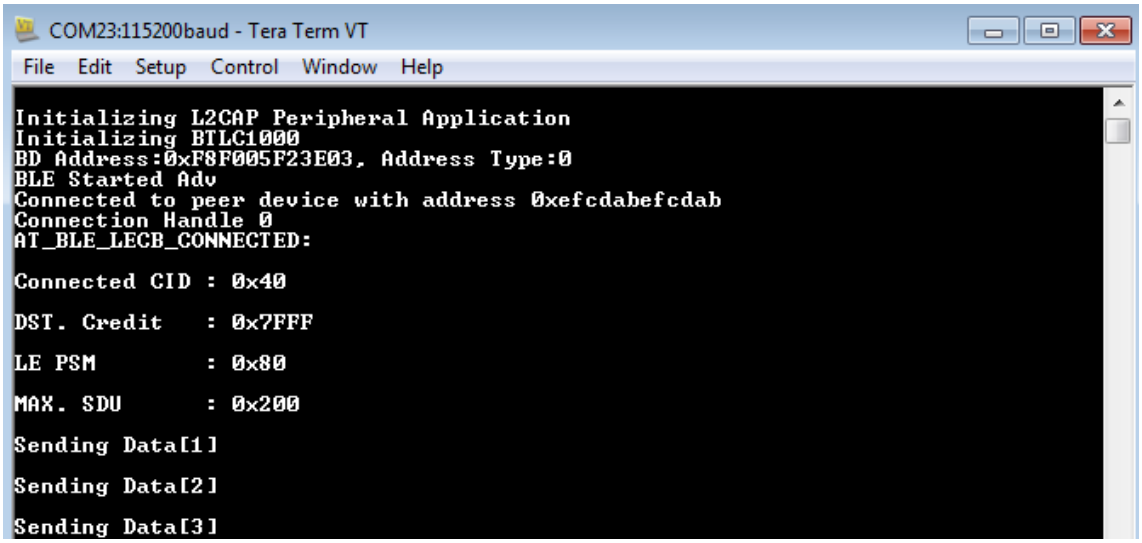


Figure 5-55. L2CAP Peripheral Connection with a L2CAP Central



4. Once the connection is established, the peripheral device keeps sending the specified data and the central device receives the same data in a given time. Eventually, calculated Throughput is displayed on the console for both central and peripheral.

Figure 5-56. L2CAP Peripheral Final Throughput Value

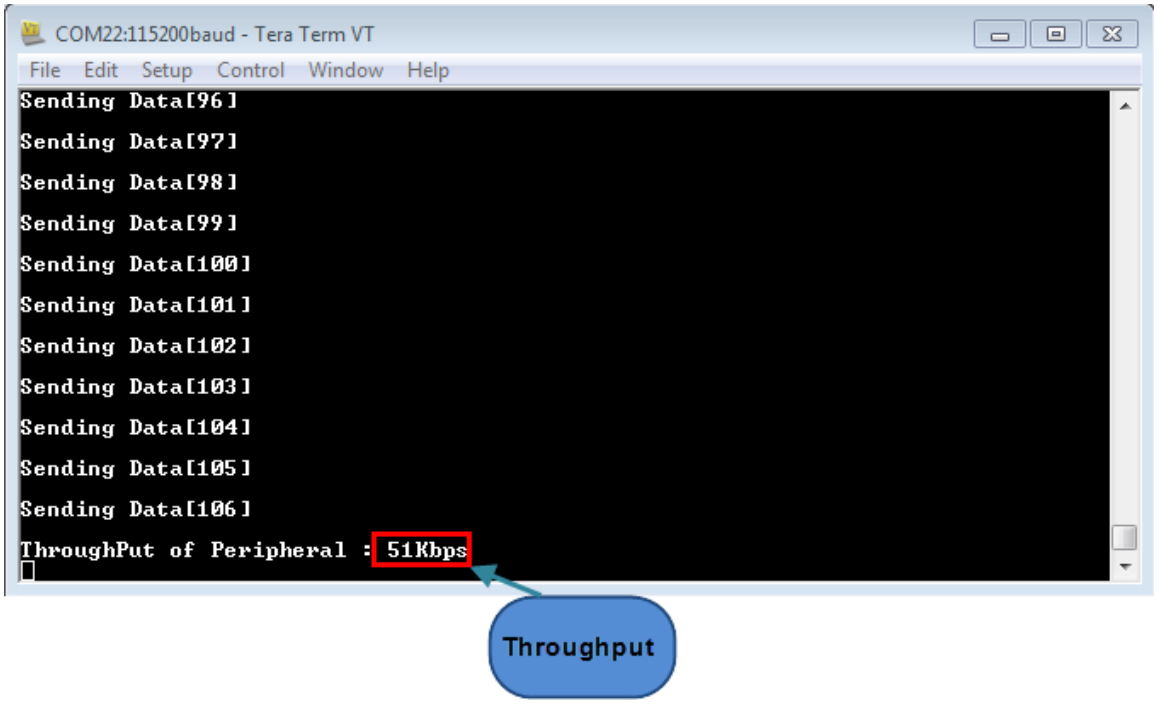
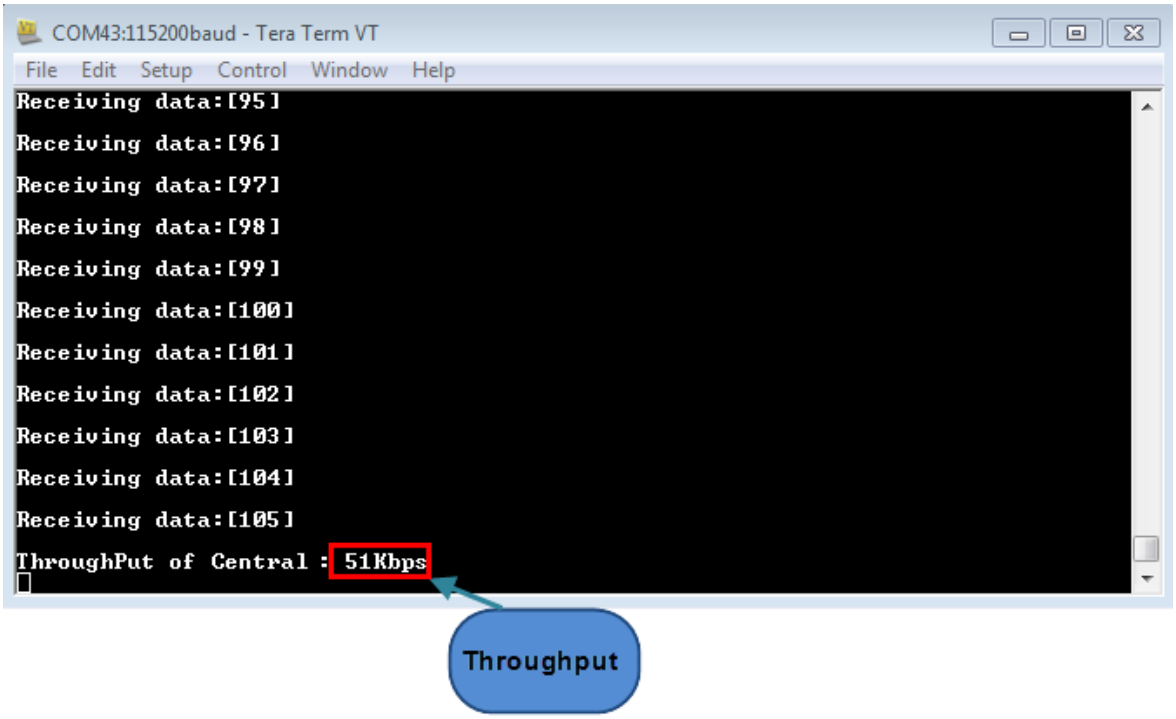


Figure 5-57. L2CAP Central Final Throughput Value



5.3.20 Health Thermometer Profile Application

Perform the following steps to run the Health Thermometer Profile application demo:

1. Establish the connection between the device and mobile phone using the procedure listed in Running the Demo.

2. When paired, the application displays the Health Thermometer Service and the Generic Information service.
3. The temperature value, RSSI and the device name are displayed on the mobile application. The console log is shown in the following screen.

Figure 5-58. Health Thermometer Connected Services

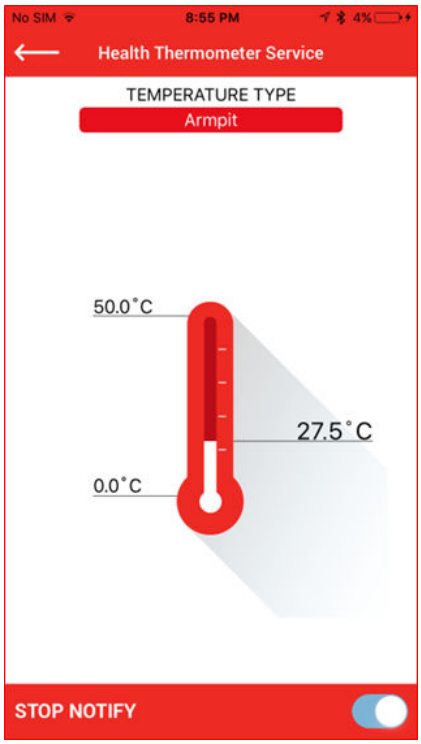
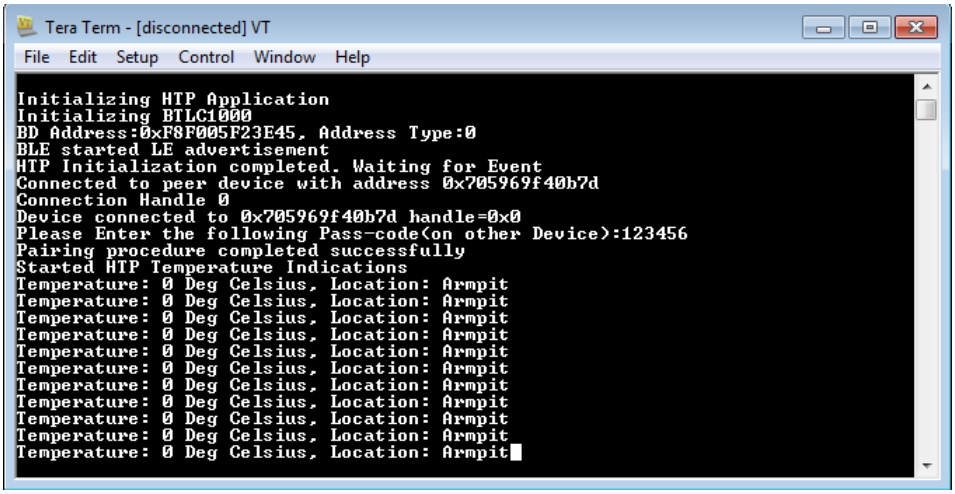


Figure 5-59. Console Log after Connection, Pairing, and with Notifications



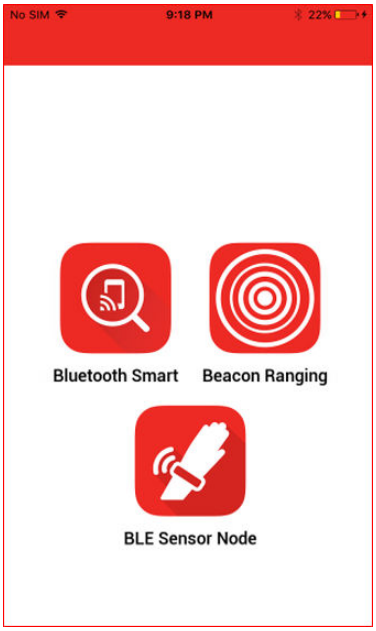
4. To change the body measurement location, press the user button (SW0) on the SAM L21. The new value is updated in the application.
5. On the Microchip SmartConnect application, going back to the scanning screen disconnects the device with mobile application.

5.3.21 iBeacon Application

Perform the following steps to run the iBeacon application demo:

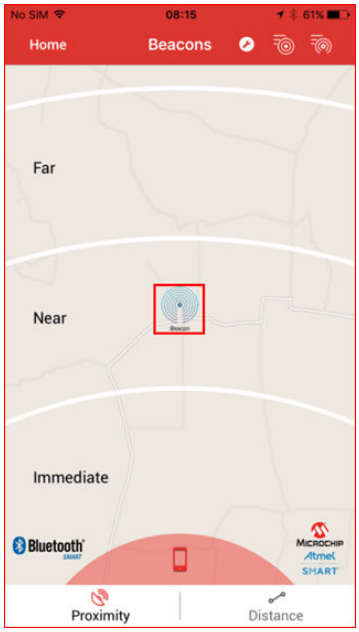
1. Follow the steps (1 and 2) from Initializing the Device.
2. Start the Beacon application on the iPhone.

Figure 5-60. Beacon Radar Profile App Launch Screen



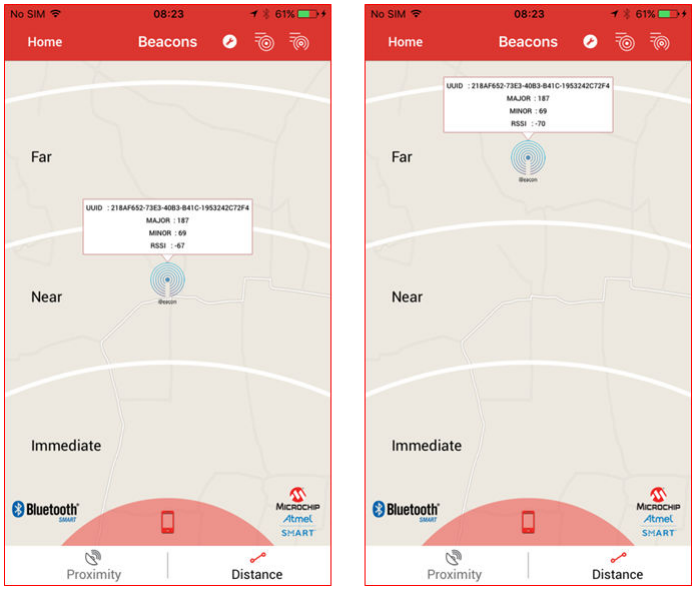
3. Click **Beacon Ranging**. The beacon application is launched to show the positioning of the beacon device with respect to the mobile phone and supports the following modes:
   - **Proximity** – used to display beacon specific information when the mobile device comes in close proximity to a given beacon. This mode also shows the corresponding product related information that is configured for this particular beacon device.
   - **Distance** – used to indicate the distance between the beacon device and the mobile.

Figure 5-61. Beacon Radar Application Initial Screen



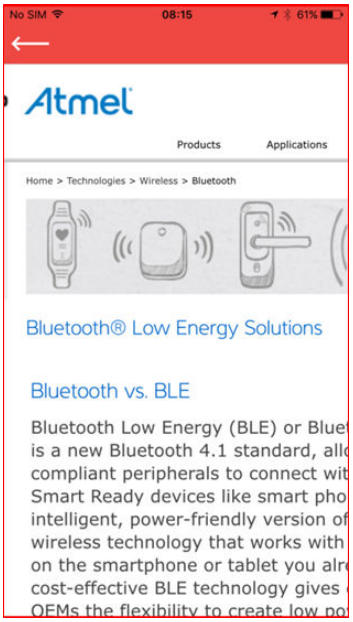
4. Click **iBeacon** to check the Major, Minor and RSSI value. The RSSI value is automatically updated based on the movement of the scanner device, as shown in the following figure.

Figure 5-62. Beacon Radar Application in Distance Mode



5. Inside the proximity mode, if the scanner device is very near to the beacon. The user can see the product information when the user is in close proximity to a given beacon device. When the user moves away from the beacon device information content is not shown any more. It is an indication that the user is moved away from the beacon device. The user can optionally close the message by clicking on close.

Figure 5-63. Beacon Radar Application in Proximity Mode



5.3.22 AltBeacon Application

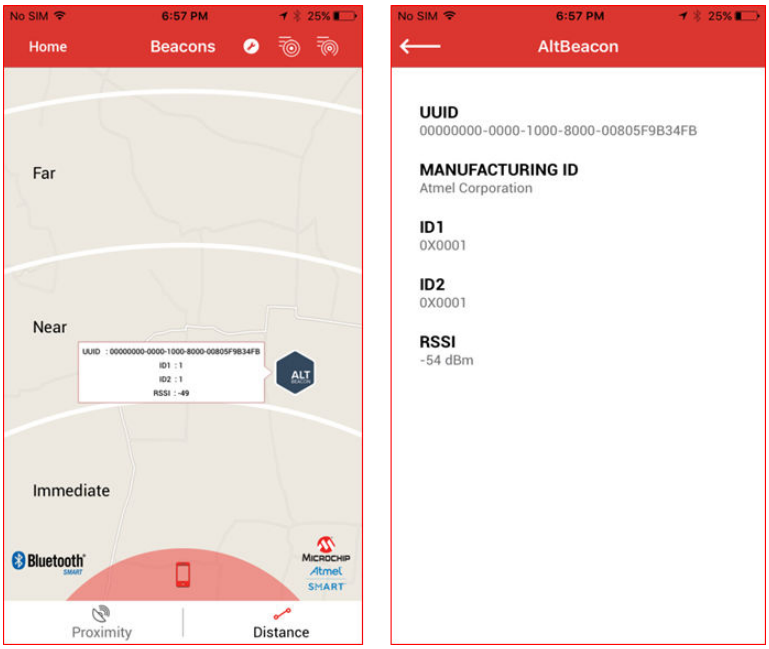
Perform the following steps to run the AltBeacon application demo:

1. Follow the steps (1 and 2) from Running the Demo.
2. The beacon application initialization is displayed in the console.

```
Initializing AltBeacon Application
BLE AltBeacon Advertisement started
```

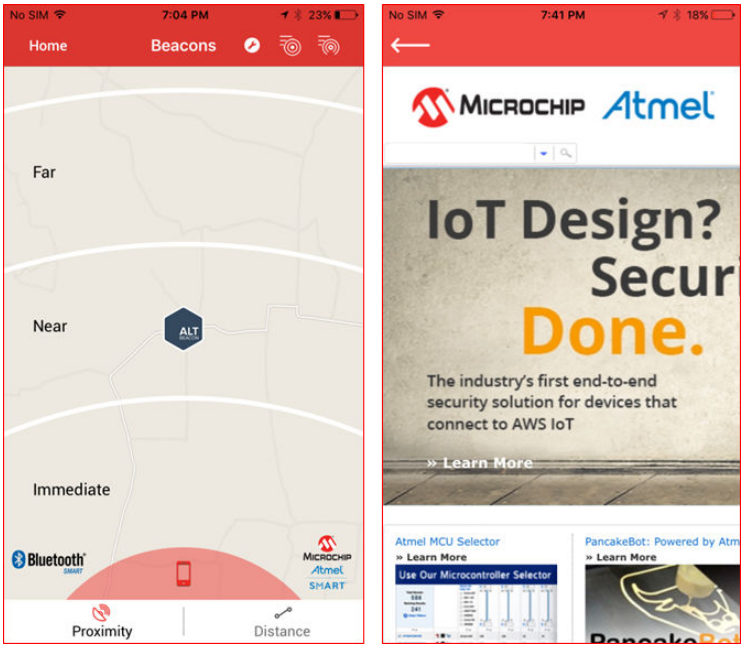
3. Start the Beacon application on the mobile phone (see Figure 5-50). In this demonstration, an iPhone is used to run the application.
4. Tap on the AltBeacon icon for Major, Minor and UUID Value. The RSSI values are automatically updated based on the movement of the scanner device. For more details about the AltBeacon device, the user can tap on the pop-up message (which shows UUID, ID1 and ID2 values), as shown in the following figures.

Figure 5-64. AltBeacon Radar Application in Distance Mode



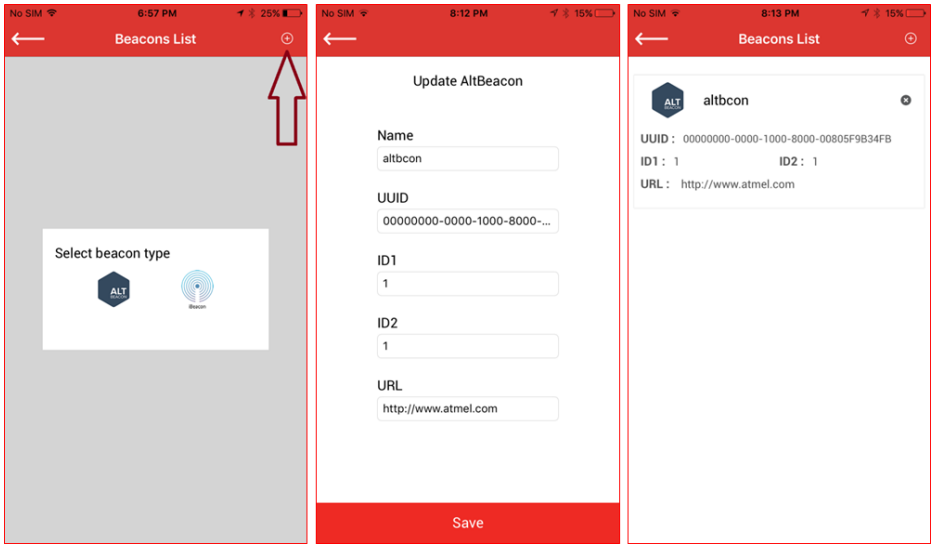
5. In proximity mode, the application opens the configured URL whenever the user comes in close proximity to the configured beacon device. When the user moves away from the beacon device, the configured beacon is not shown. It is just an indication that the user moved away from beacon device.

Figure 5-65. AltBeacon Radar Application in Proximity Mode



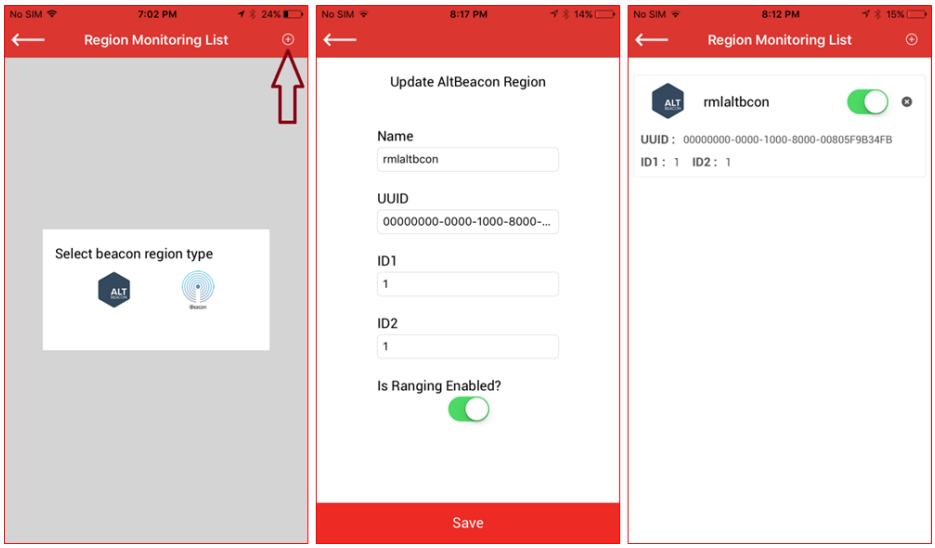
6. A new AltBeacon can be added to the Beacon list using the add button, as illustrated in the following figure.

Figure 5-66. Adding new beacon



7. A new AltBeacon can be added into the Region monitoring list using the add button, as illustrated in the following figure.

Figure 5-67. Adding new beacon in Region Monitoring List



Note: The Region Monitoring List is supported on iOS, and not on Android devices.

5.3.23 Eddystone Beacon Application

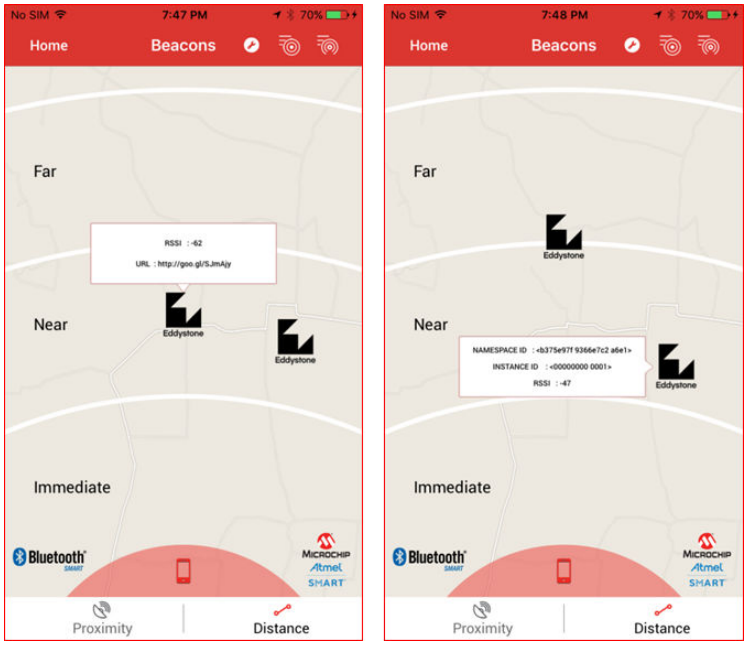
Perform the following steps to run the Eddystone Beacon application demo:

1. Follow the steps (1 and 2) from Running the Demo.
2. The beacon application initialization is displayed on the console.

```
Initializing BTLC1000
BD Address:0xF8F005F34CC1, Address Type:0
Eddystone beacon started
Adv count: 1
Tx URL
Adv count: 22
Tx TLM
```

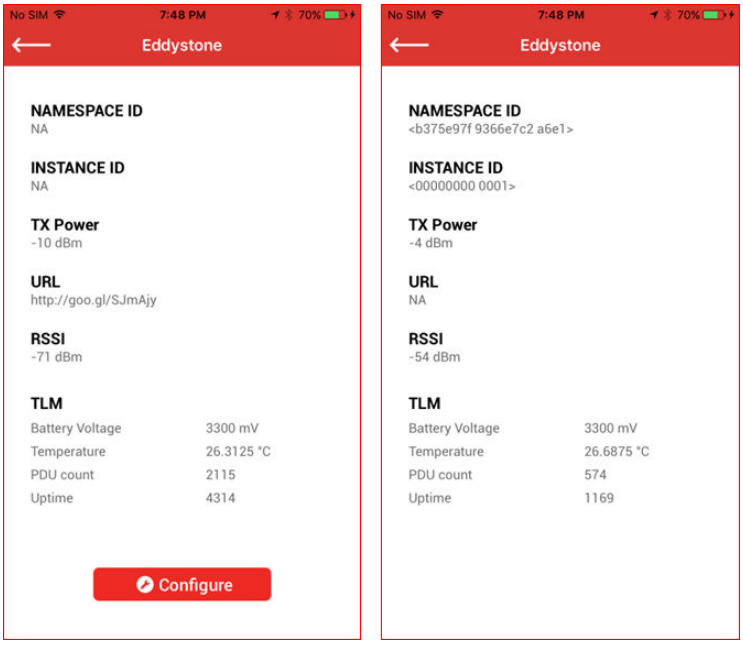
3. Start the Beacon application on the mobile phone (see Figure 5-50). In this demonstration, an iPhone is used to run the application.
4. Open the **Beacons** navigation tab to view the ranging screen. The Eddystone beacon device is shown on the ranging screen with the Eddystone icon. The position of the beacon is based on the strength of the signal received from RSSI. Click the **Beacon** icon to see a pop-up window showing the identity of the frame; in the case of the `EDDYSTONE_URL_APP`, the shortened URL value is shown and in the case of the `EDDYSTONE_UID_APP`, Namespace ID and Instance ID is shown.

Figure 5-68. Eddystone Beacons (both UID and URL beacons) ranged by Microchip SmartConnect Application



5. Click the beacon pop-up window to view detailed information. The detailed view shows UID/URL and telemetric information like battery voltage, beacon temperature, time since power-on, etc.. This telemetric information is obtained from the Eddystone-TLM frames which are interleaved with Eddystone identifying frames (UID/URL).

Figure 5-69. Detailed view of the Eddystone URL and UID beacon



6. In the `EDDYSTONE_URL_APP`, the detailed beacon information screen shows a Configure button. Click the **Configure** button. It requests that the user puts the beacon into Configuration mode. The SW0 hardware button present on the SAM L21 Xplained Pro board has to be long pressed (around 3 seconds) to enter into Configuration mode.
7. Connect to the beacon in Configuration mode as shown in Figure 5-70. Once connected, the configurable beacon parameters are listed out as shown in Figure 5-71.

Figure 5-70. Connecting to Beacon in Configuration Mode

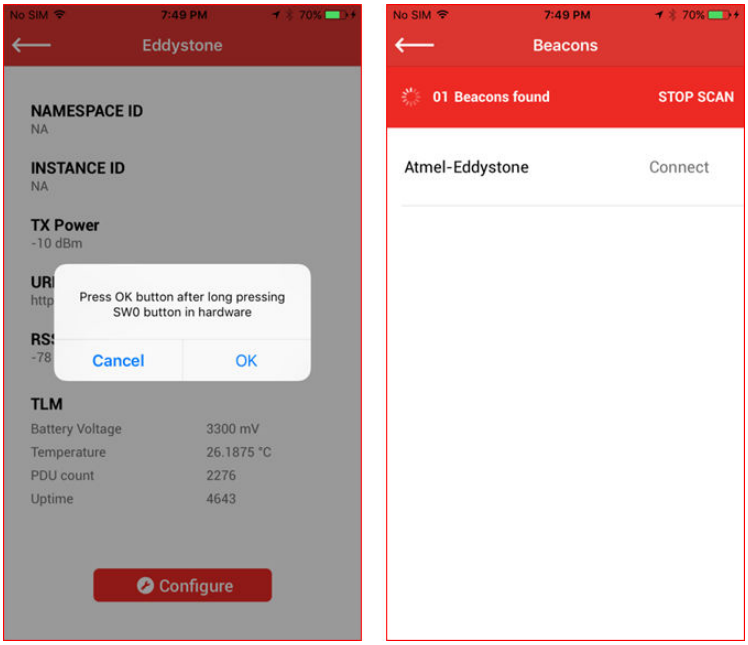
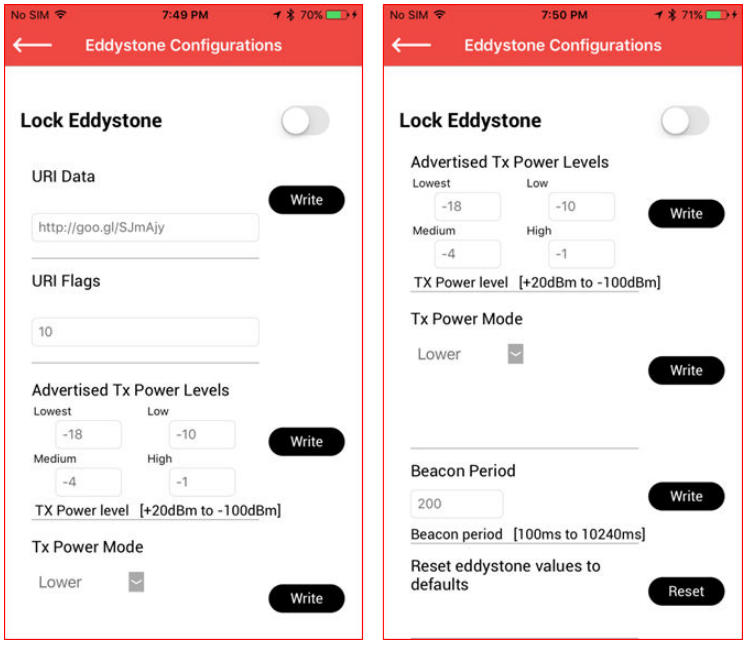
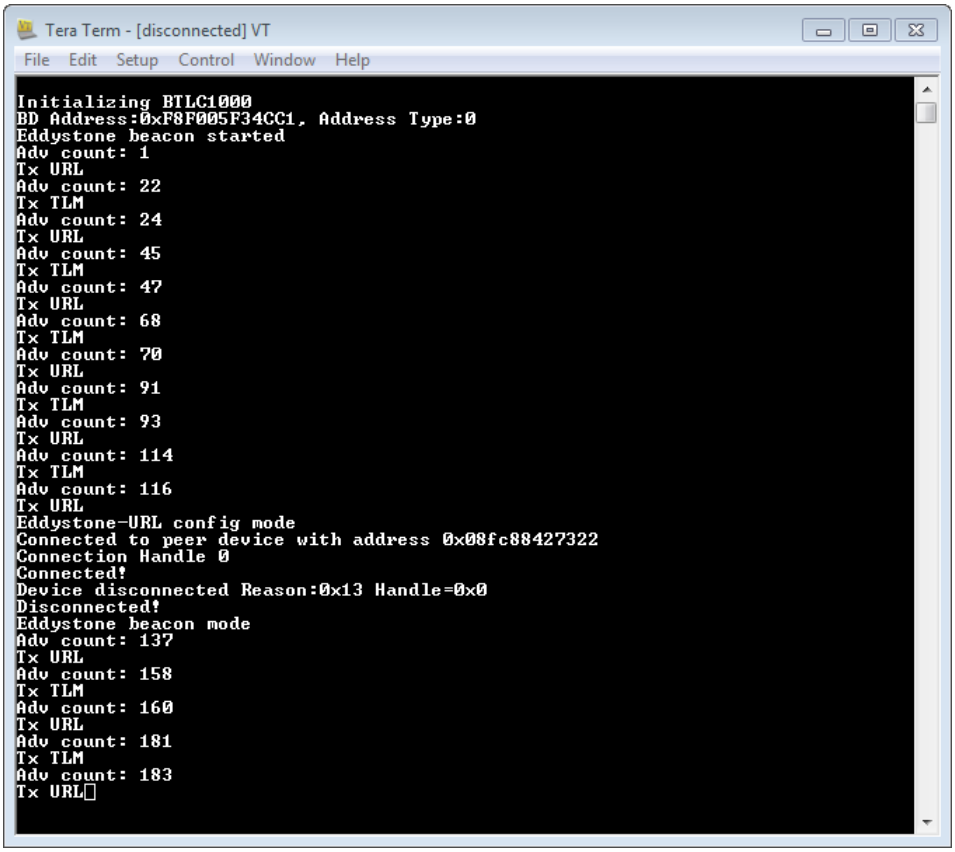


Figure 5-71. Beacon Configuration Screen



8. Update the URL, Tx Power mode, beacon period, etc. and then save. Now, disconnect from the beacon and enter the ranging screen. Once disconnected, the beacon device (ATBTLC1000-MR/ZR) enters into Beacon mode and start sending Eddystone URL frames with the updated values. The ranging console log screen shows the beacon with new URL value.

Figure 5-72. Eddystone Beacon Console Log



9. The beacon configuration page also provides a reset button that can set all the parameters to its default factory settings.

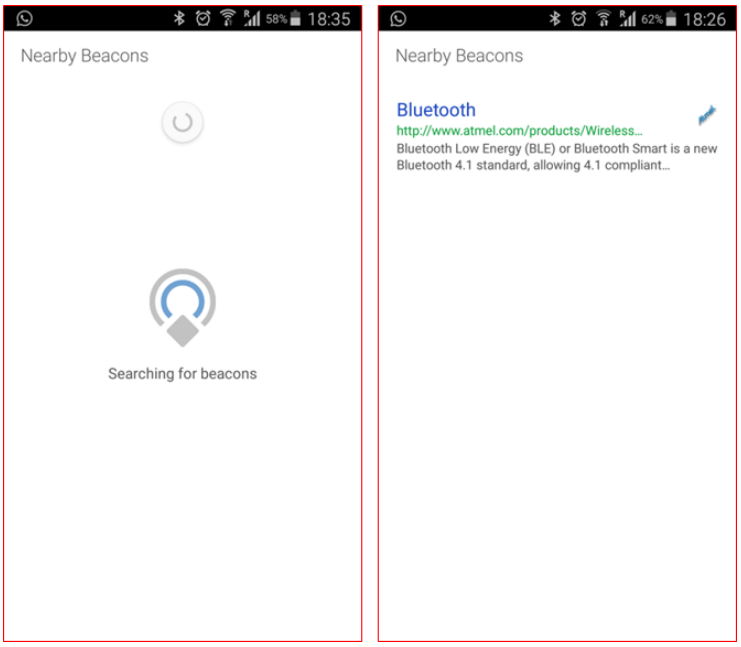
5.3.23.1 Demo with Physical Web Application

Eddystone is the backbone of the Physical Web initiative from Google. For more information on the Physical Web, refer to https://google.github.io/physical-web/.

The following demo shows how the Eddystone application running on an ATBTLC1000-MR/ZR device works seamlessly with the Physical Web Android application.

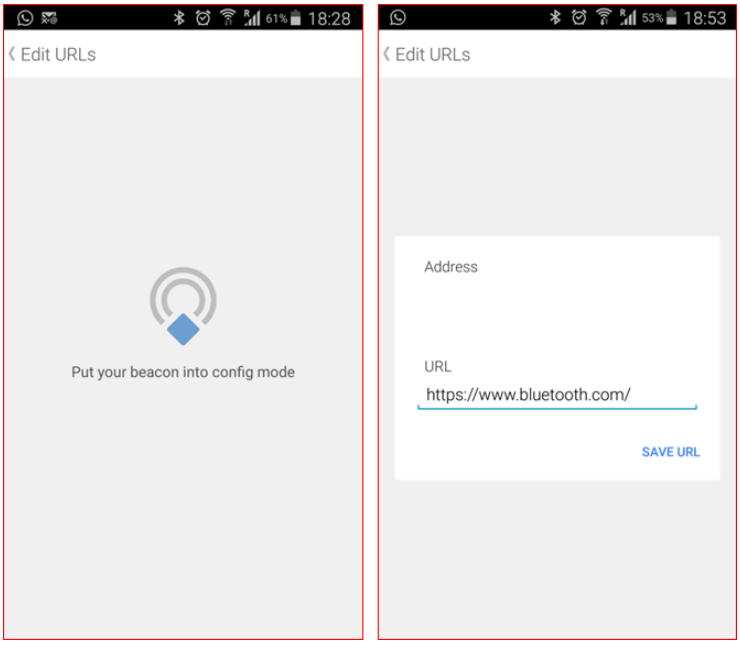
1. Install the Physical Web Android application in a BLE compatible android device.
2. Build and run the `EDDYSTONE_URL_APP` in the hardware setup.
3. Open the Physical Web app to detect the URL emitted by the beacon node, as shown in the following figure.

Figure 5-73. Physical Web App Detecting Eddystone-URL



4. Click the **Menu** button to open the “Edit URL” option; this requests that the user to put the beacon in Configuration mode. Pressing the SW0 button on the Xplained Pro board for 3 seconds (long press) puts the beacon device in Configuration mode.
5. The URL configuration window will pop-up once the Android device establishes connection with the beacon’s configuration service, as shown in the following figure. Change the URL value to a different one; make sure to use a shortened URL as the size of encoded URL is limited to 17 bytes. Google’s URL shortener can be used for this purpose https://goo.gl/.

Figure 5-74. URL Configuration on Physical Web App



5.3.24 Direct Test Mode Application

This demonstration requires two ATBTLC1000-MR/ZR devices loaded with the Direct Test Mode example application code. Perform the following steps to run the DTM with the Performance Analyzer tool.

1. Start the performance analyzer in Atmel Studio.

Figure 5-75. Selecting Studio Performance Analyzer Tool

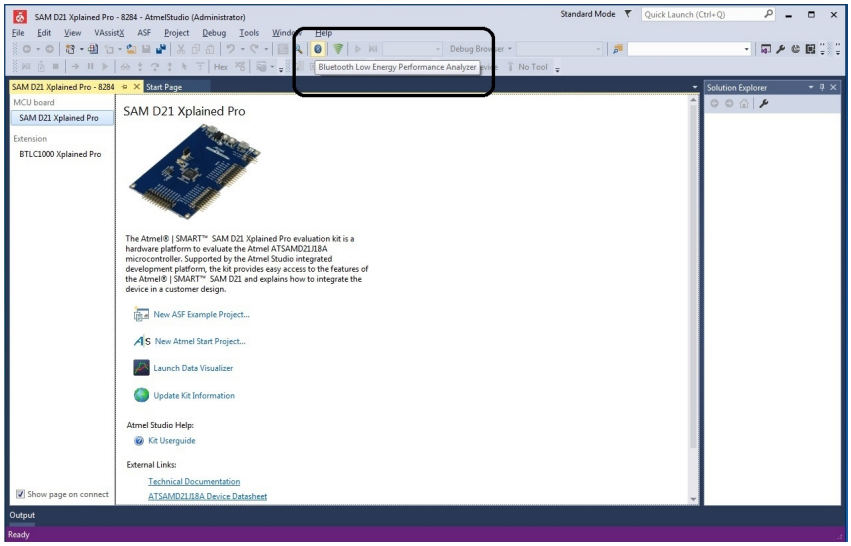
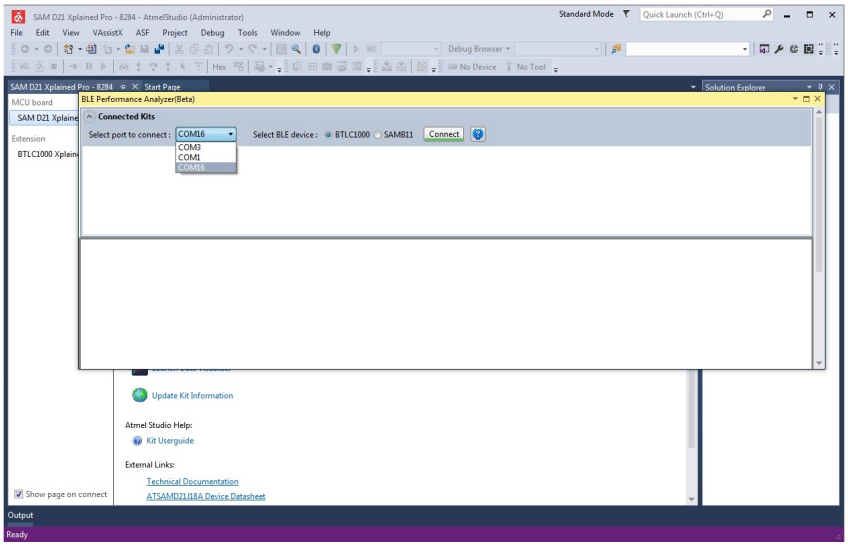


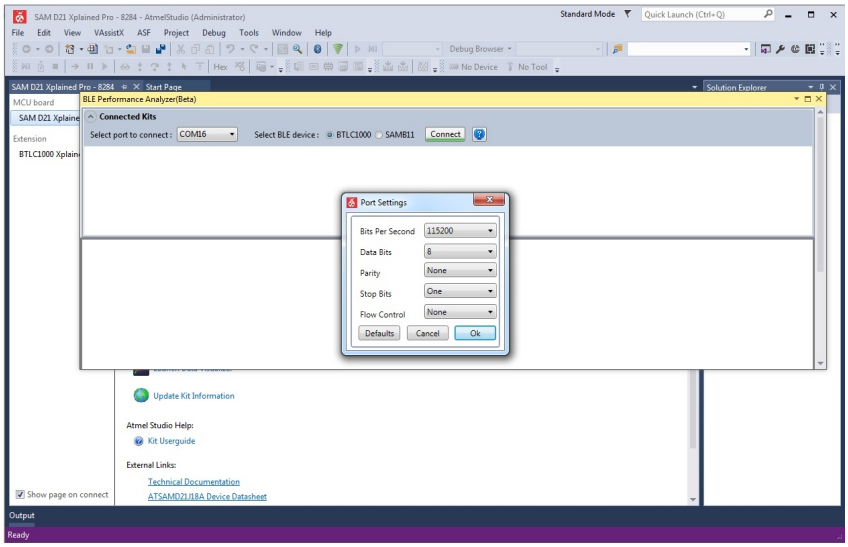
Figure 5-76. BLE Performance Analyzer Tool Window



2. Next, initialize the UART. Enter the COM port number and press “Init UART”. A successful initialization is indicated by receiving a chip response, as shown in the following figure.

Tip: Check the COM port number from the Device Manager.

Figure 5-77. Initializing UART



3. Start the Direct Test Mode, configure one board as Tx and the other one as Rx. Make sure to select the same RF Channel for both Rx and Tx during the test and start the Rx test before the Tx test in order not to miss any packets. The user must see non-zero packets received at the Rx side notifying successful transmission and reception.
   Note: Any side can be replaced by standard compliant test equipment.
4. Select **Tx Power (dBm)** other than -55 dBm, if the devices are relatively far from each other.

Figure 5-78. Tx Power Configuration

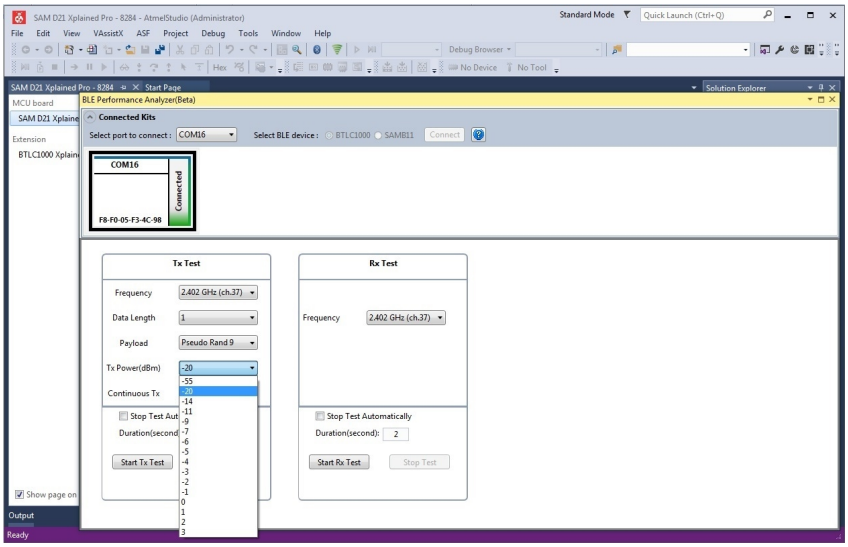
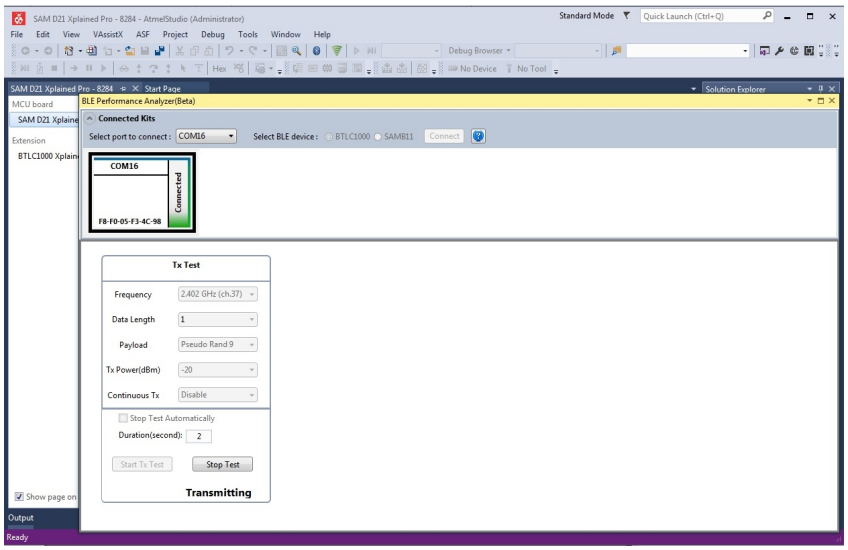
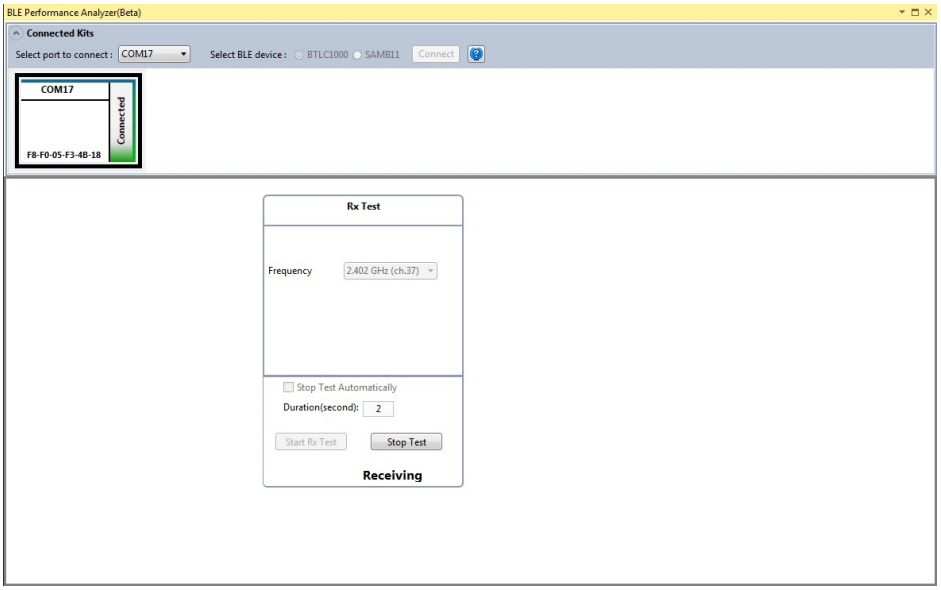


Figure 5-79. Starting Tx Test



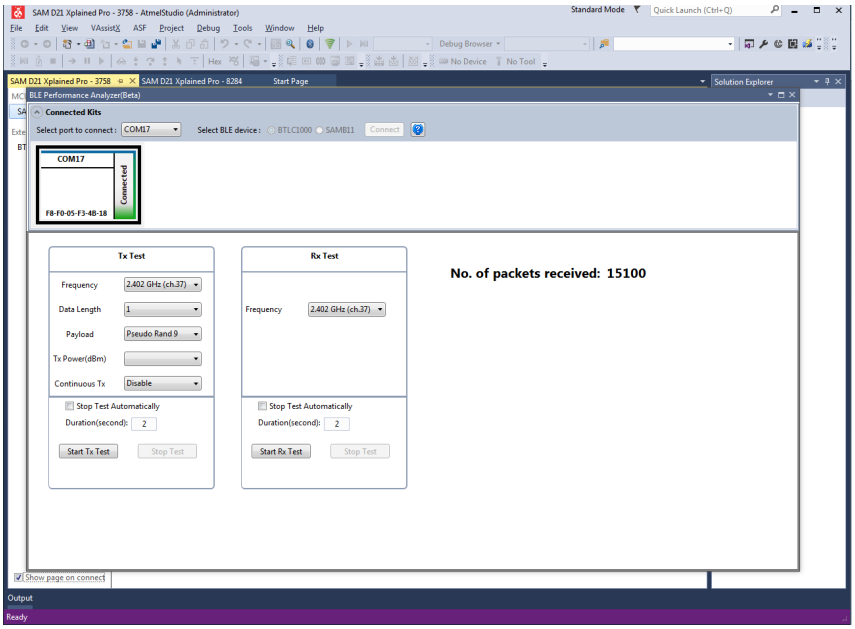
5. On the Rx side, select the appropriate COM port with the same default settings. Open the window having both Tx and Rx options. Click **Start Rx Test** and ensure that the packets are transferred for a certain time period from the Tx device.

Figure 5-80. Starting Rx Test



6. Click **Stop Test** to display the number of successful received packets.

Figure 5-81. Number of Packets Received



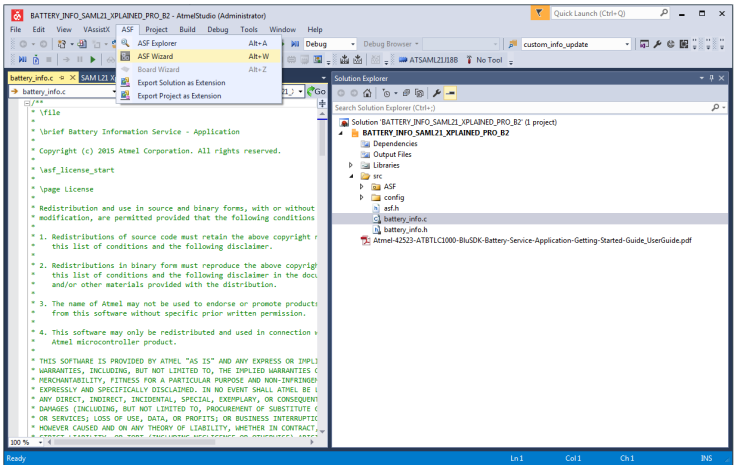
Important: The PER is calculated assuming that the transmitter side sends 1500 packets for testing using R&S CBT equipment. For peer testing, ignore the PER reading.

6. Adding a BLE Standard Service

The user can add another service such as the "Device Information Service" or "Battery Service" to the application by using the ASF wizard as mentioned in the following screen:

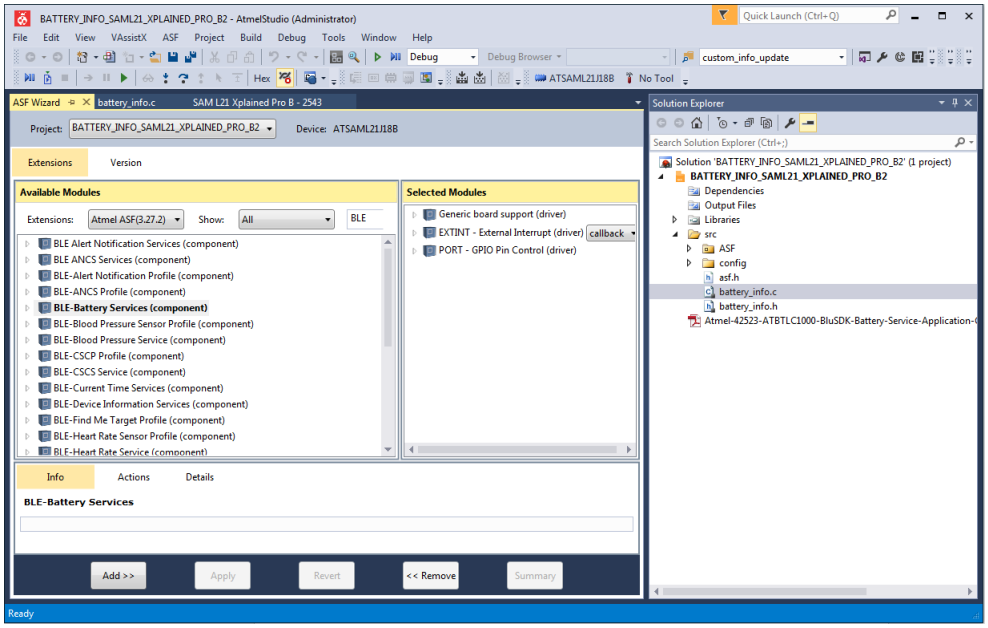
1. Go to the Atmel Studio `ASF > ASF Wizard` as shown in the following figure.

Figure 6-1. Invoking ASF Wizard



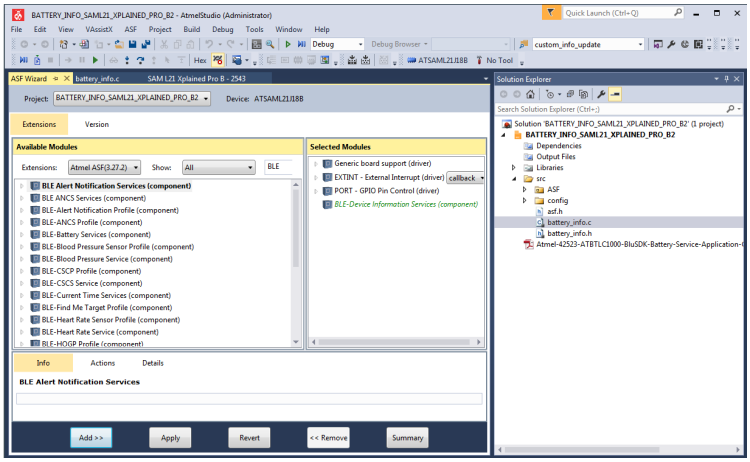
2. In the *ASF Wizard* window, enter “BLE” in the search box, as shown in the following figure.

Figure 6-2. ASF BLE Services and Components Window



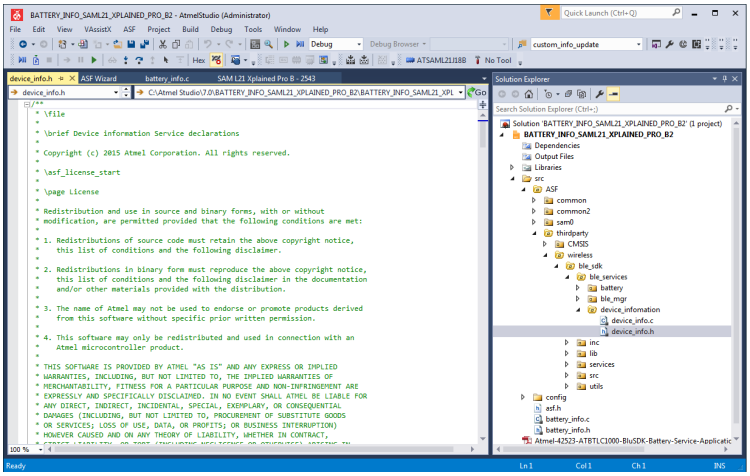
3. Select the required BLE Services/Profiles Component, as shown in the following figure.
   3.1. Select **Device Information Services**.
   3.2. Click **Add > Apply > OK**.

Figure 6-3. Adding BLE-Device Information Service and its Component



4. Newly added BLE service component must be available in the following directory: `src\thirdparty\wireless\ble_sdk\ble_services\device_information`, as shown in the following figure.

Figure 6-4. Hierarchy of Newly Added Service Component



5. Use the APIs, as mentioned in the Device Information Service (`device_info.h`), for incorporating this functionality, if required in the application.

```
/**@brief Update the DIS characteristic value after defining the services using
dis_primary_service_define
*
* @param[in] dis_serv dis service instance
* @param[in] info_type dis characteristic type to be updated
* @param[in] info_data data need to be updated
* @return @ref AT_BLE_SUCCESS operation completed successfully.
* @return @ref AT_BLE_FAILURE Generic error.
*/
at_ble_status_t dis_info_update(dis_gatt_service_handler_t *dis_serv , dis_info_type
info_type,
dis_info_data* info_data, at_ble_handle_t conn_handle);
/**@brief DIS service and characteristic initialization (Called only once by user).
*
* @param[in] device_info_serv dis service instance
*
* @return none
*/
void dis_init_service(dis_gatt_service_handler_t *device_info_serv );
/**@brief Register a dis service instance inside stack.
```

```
*
* @param[in] dis_primary_service dis service instance
*
* @return @ref AT_BLE_SUCCESS operation completed successfully
* @return @ref AT_BLE_FAILURE Generic error.
*/
at_ble_status_t dis_primary_service_define(dis_gatt_service_handler_t *dis_primary_service);
```

7. Custom Serial Chat Service Specification

7.1 Service Declaration

The Custom Serial Chat profile consists of a custom serial chat service. Both the mobile app and the host (HOST MCU + ATBTLC1000-MR/ZR) need to expose this service. The custom serial chat service is instantiated as a primary service.

The UUID value assigned to custom serial chat service is:
`fd5abba0-3935-11e5-85a6-0002a5d5c51b`.

7.2 Service Characteristic

The following characteristics are exposed in the Custom Serial Chat service. Only one instance of each characteristic is permitted within this service.

Table 7-1. Custom Serial Chat Service Characteristics

Characteristic Name	Requirement	Mandatory Properties	Security Permission
Endpoint	M	Notify	Depend on BLE_PAIR_ENABLE macro
Client characteristic configuration descriptor	M	Read, Write	None

Note:

1. The security permission depends on `BLE_PAIR_ENABLE` macro, defined inside the `ble_manager.h`.
2. If `BLE_PAIR_ENABLE` is set true, then the security permission of Endpoint characteristic is readable with authentication and writable with authentication.
3. If `BLE_PAIR_ENABLE` is set false, then the security permission of Endpoint characteristic is none.

7.3 Endpoint

The Endpoint characteristic is used to transmit the chat data provided by the user on the terminal (device side) and on the mobile chat screen (mobile side).

The UUID value assigned to Endpoint characteristic is `fd5abba1-3935-11e5-85a6-0002a5d5c51b`.

7.3.1 Characteristic Behavior

When the client characteristic configuration descriptor is configured for the notification by a remote device, the user can send chat text message to the remote device.

Note: The chat text is sent as a notification from the sender (mobile app or ATBTLC1000-MR/ZR based device). Hence, the client characteristic configuration descriptor is always configured for notifications (in the Custom Serial Chat service instance on the mobile application and host).

7.4 Characteristic Descriptors

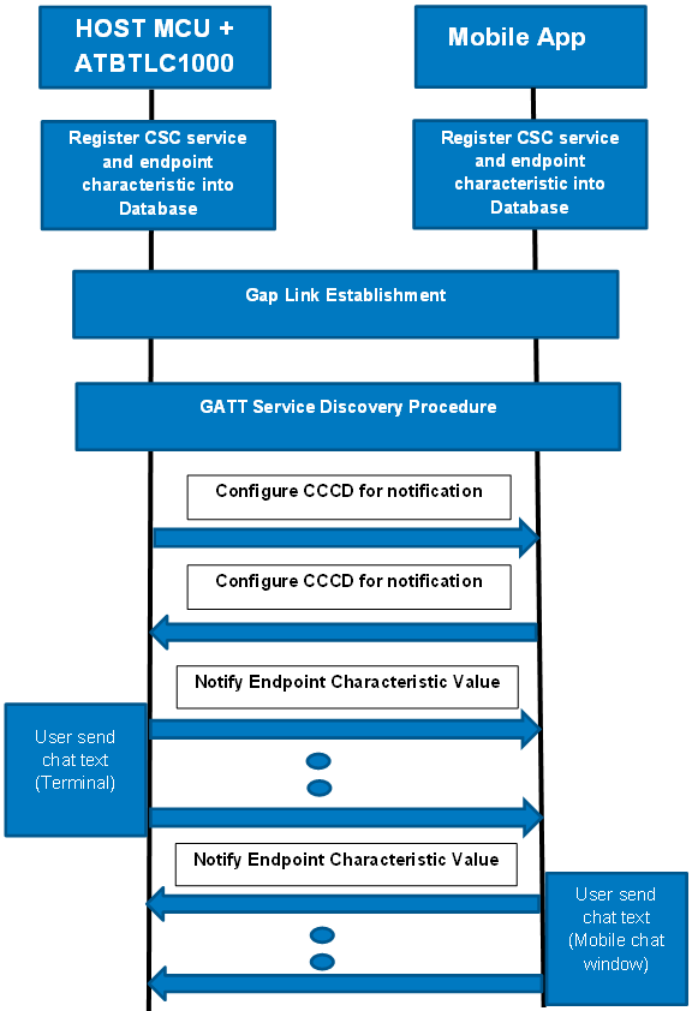
7.4.1 Client Characteristic Configuration Descriptor

The client characteristic configuration descriptor is included in the Endpoint characteristic.

7.5 Sequence Flow Diagram

The following figure illustrates the sequence flow diagram of Custom Serial Chat profile.

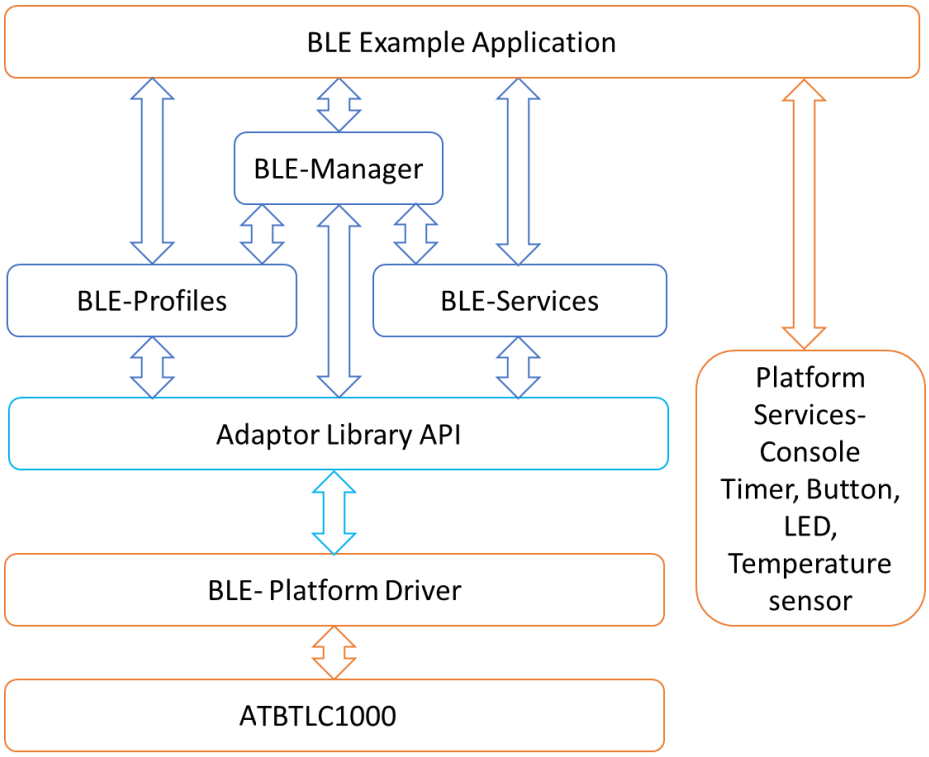
Figure 7-1. Sequence Flow Diagram



8. BluSDK Software Architecture

The following diagram illustrates the various layers in the BluSDK Architecture for implementing various applications. The External host can be supported hardware platforms and IDEs (see Supported Hardware Platforms and IDEs).

Figure 8-1. BluSDK Software Architecture



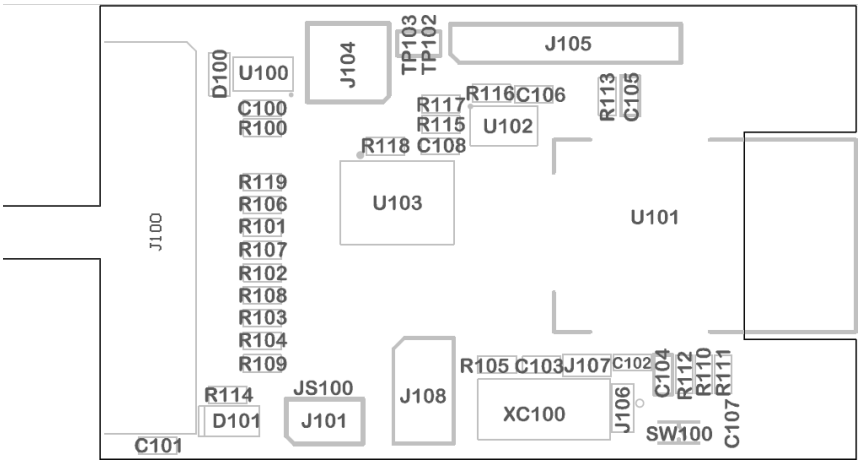
9. Hardware Flow Control for 4-Wire Mode eFuse Write Procedure

This procedure is applicable only for the MR/CSP/QFN packages. The ATBTLC1000-ZR module comes with a 4-wire mode eFuse by default from the factory.

Caution: While writing data to the eFuse, the data written can never be changed (that is, if a value of '1' is written to a specific eFuse, those contents can never be reverted back to its original value). To configure the eFuse controller for accessing the eFuse contents, the user must enter the valid arguments for eFuse configuration.

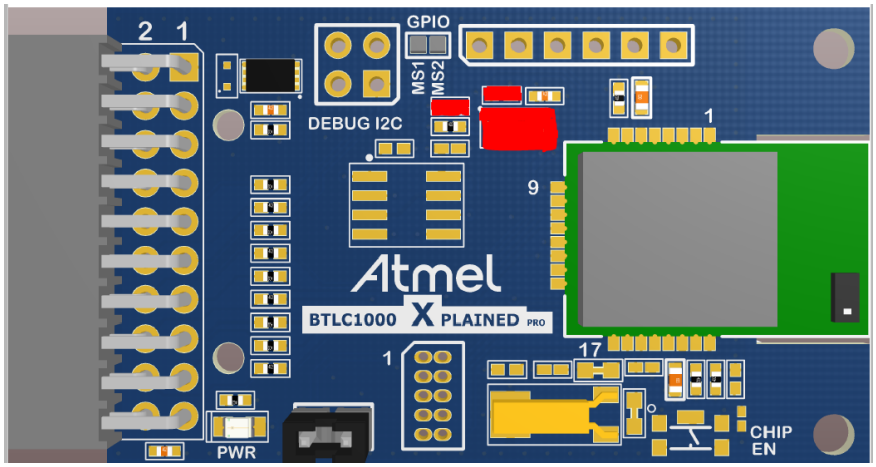
1. Connect the SAM-ICE to the J108 header and the ATBTLC1000 XPRO to any MCU board to power it up. For the J108 location, refer to the following assembly drawing. Ensure that the “ATBTLC1000 chip enable” and “Wakeup pin” are driven high throughout the eFuse process.

Figure 9-1. ATBTLC1000 XPRO Board Assembly Diagram



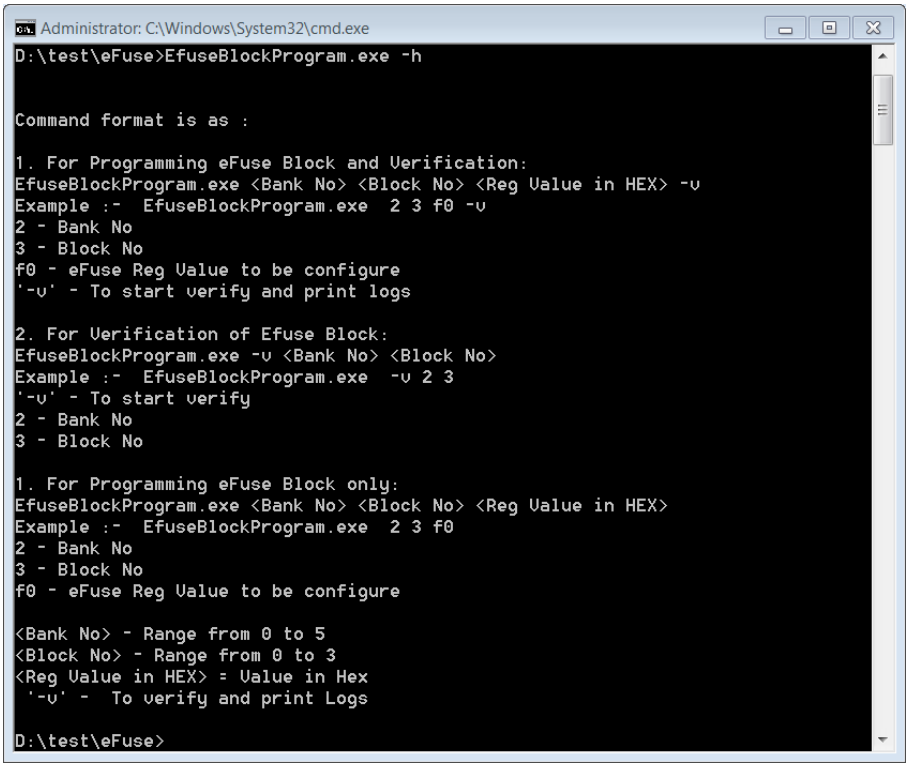
2. Remove pins U102, R116, and R117 on the ATBTLC1000 XPRO to disconnect the temperature sensor chip from the GPIOs used for flow control, as shown in following figure (highlighted in red).

Figure 9-2. ATBTLC1000 3D View XPRO Board



3. The supplied command line tool (`EfuseBlockProgram.exe`) is used to write the ATBTLC1000 eFuse to configure the flow control signals.
4. To invoke the help information from the EfuseBlockProgram, enter `EfuseBlockProgram.exe -h` in the command line.

Figure 9-3. Invoking Help Information

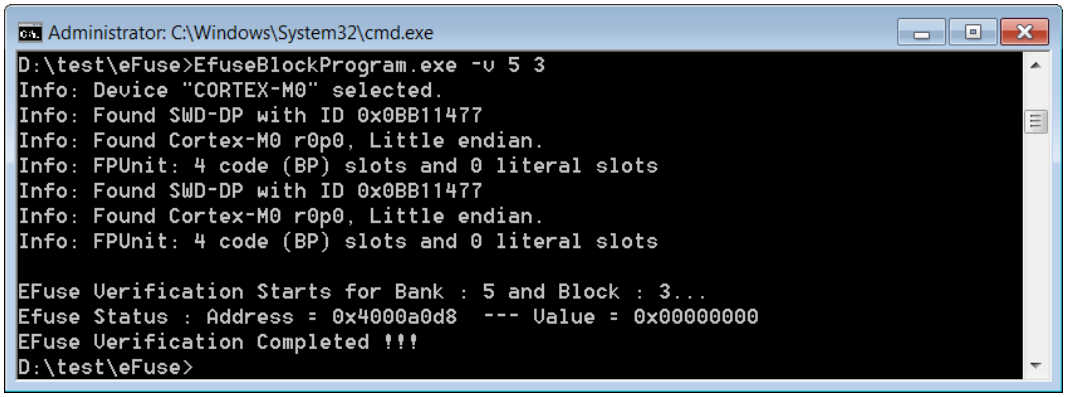


5. Reading eFuse value:
   5.1. Command Syntax: `EfuseBlockProgram.exe –v <Bank No> <Block No>`.
   - `Bank No`: Range from 0 to 5
   - `Block No`: Range from 0 to 3
   - `–v`: For verification of a programmed block

   Note: For the ATBTLC1000 UART hardware flow control 4-wire mode, Bank-5 and Block-3 are configured.

   5.2. To enable the ATBTLC1000 UART hardware flow control eFuse configuration, use command `EfuseBlockProgram.exe –v 5 3`. The example output shown in following figure is for reading the eFuse.

   Figure 9-4. Reading eFuse Values



6. Writing and verifying the ATBTLC1000 eFuse value:

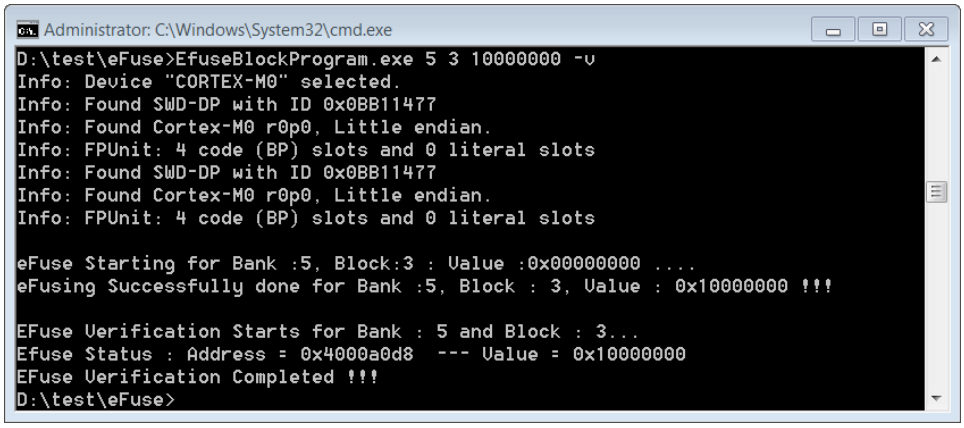
6.1. When writing the data to eFuse, the data written can never be changed (that is, if a value '1' is written to a specific eFuse, those contents can never be reverted back to its original value. To configure the eFuse controller for accessing the eFuse contents, the user must enter the valid arguments for eFuse configuration.

6.2. Command Syntax: `EfuseBlockProgram.exe <Bank No> <Block No> <Reg Value in HEX> -v`.

- `Bank No`: Range from 0 to 5
- `Block No`: Range from 0 to 3
- `Reg Value`: eFuse to be configured in Hex
- `-v`: For verification of a programmed block

6.3. To enable the ATBTLC1000 UART hardware flow control 4-wire mode, Bank-5 and Block-3 must be written as “10000000” (inputs are in `hex`). The example write configuration to enable the hardware flow control is shown in following figure.

Figure 9-5. Writing eFuse Values



10. Document Revision History

Rev A - 07/2017

Section	Changes
Document	Initial Release

11. Object of Declaration

EU Declaration of Conformity for ATBTLC1000-MR/ZR Evaluation Boards

This declaration of conformity is issued by the manufacturer.

The development/evaluation tool is designed to be used for research and development in a laboratory environment. This development/evaluation tool is not intended to be a finished appliance, nor is it intended for incorporation into finished appliances that are made commercially available as single functional units to end users. This development/evaluation tool complies with EU EMC Directive 2004/108/EC and as supported by the European Commission's Guide for the EMC Directive 2004/108/EC (8th February 2010).

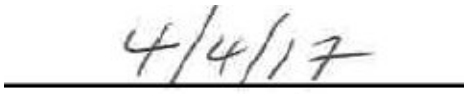
This development/evaluation tool complies with EU RoHS2 Directive 2011/65/EU.

This development/evaluation tool, when incorporating wireless and radio-telecom functionality, is in compliance with the essential requirement and other relevant provisions of the R&TTE Directive 1999/5/EC and the FCC rules as stated in the declaration of conformity provided in the module datasheet and the module product page available at www.microchip.com.

For information regarding the exclusive, limited warranties applicable to Microchip products, please see Microchip's standard terms and conditions of sale, which are printed on our sales documentation and available at www.microchip.com.

Signed for and on behalf of Microchip Technology Inc. at Chandler, Arizona, USA.

Rodger Richey
Director of Development Tools



Date

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks

ISBN: 978-1-5224-1882-5

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Asia Pacific Office** Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon **Hong Kong** Tel: 852-2943-5100 Fax: 852-2401-3431 **Australia - Sydney** Tel: 61-2-9868-6733 Fax: 61-2-9868-6755 **China - Beijing** Tel: 86-10-8569-7000 Fax: 86-10-8528-2104 **China - Chengdu** Tel: 86-28-8665-5511 Fax: 86-28-8665-7889 **China - Chongqing** Tel: 86-23-8980-9588 Fax: 86-23-8980-9500 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 Fax: 86-571-8792-8116 **China - Hong Kong SAR** Tel: 852-2943-5100 Fax: 852-2401-3431 **China - Nanjing** Tel: 86-25-8473-2460 Fax: 86-25-8473-2470 **China - Qingdao** Tel: 86-532-8502-7355 Fax: 86-532-8502-7205 **China - Shanghai** Tel: 86-21-3326-8000 Fax: 86-21-3326-8021 **China - Shenyang** Tel: 86-24-2334-2829 Fax: 86-24-2334-2393 **China - Shenzhen** Tel: 86-755-8864-2200 Fax: 86-755-8203-1760 **China - Wuhan** Tel: 86-27-5980-5300 Fax: 86-27-5980-5118 **China - Xian** Tel: 86-29-8833-7252 Fax: 86-29-8833-7256	**China - Xiamen** Tel: 86-592-2388138 Fax: 86-592-2388130 **China - Zhuhai** Tel: 86-756-3210040 Fax: 86-756-3210049 **India - Bangalore** Tel: 91-80-3090-4444 Fax: 91-80-3090-4123 **India - New Delhi** Tel: 91-11-4160-8631 Fax: 91-11-4160-8632 **India - Pune** Tel: 91-20-3019-1500 **Japan - Osaka** Tel: 81-6-6152-7160 Fax: 81-6-6152-9310 **Japan - Tokyo** Tel: 81-3-6880- 3770 Fax: 81-3-6880-3771 **Korea - Daegu** Tel: 82-53-744-4301 Fax: 82-53-744-4302 **Korea - Seoul** Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934 **Malaysia - Kuala Lumpur** Tel: 60-3-6201-9857 Fax: 60-3-6201-9859 **Malaysia - Penang** Tel: 60-4-227-8870 Fax: 60-4-227-4068 **Philippines - Manila** Tel: 63-2-634-9065 Fax: 63-2-634-9069 **Singapore** Tel: 65-6334-8870 Fax: 65-6334-8850 **Taiwan - Hsin Chu** Tel: 886-3-5778-366 Fax: 886-3-5770-955 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 Fax: 886-2-2508-0102 **Thailand - Bangkok** Tel: 66-2-694-1351 Fax: 66-2-694-1350	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **France - Saint Cloud** Tel: 33-1-30-60-70-00 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-67-3636 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-7289-7561 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820