Secure AVR BLE IoT Node - Software User Guide

Secure AVR BLE IoT Node - Software User Guide

Introduction

In this document, the firmware modules of an AVR® powered IoT node is introduced. The kit is controlled by an ATtiny1617 host MCU. It is equipped with an ATECC508A CryptoAuthentication™ device, an RN4871 Bluetooth 4.2 Low-Energy module, and a tri-axial accelerometer. In this design, the kit can be easily connected to the app via BLE and the sensor data can be viewed in the app at run-time, which is a real IoT use-case. As an IoT edge node, this firmware provides the essential functions in a typical IoT scenario; Control, Security, Connectivity, and Low Power.

To know how to set up this kit and connect it with the app, refer to Secure AVR BLE IoT Node - Getting Started Guide.

Features

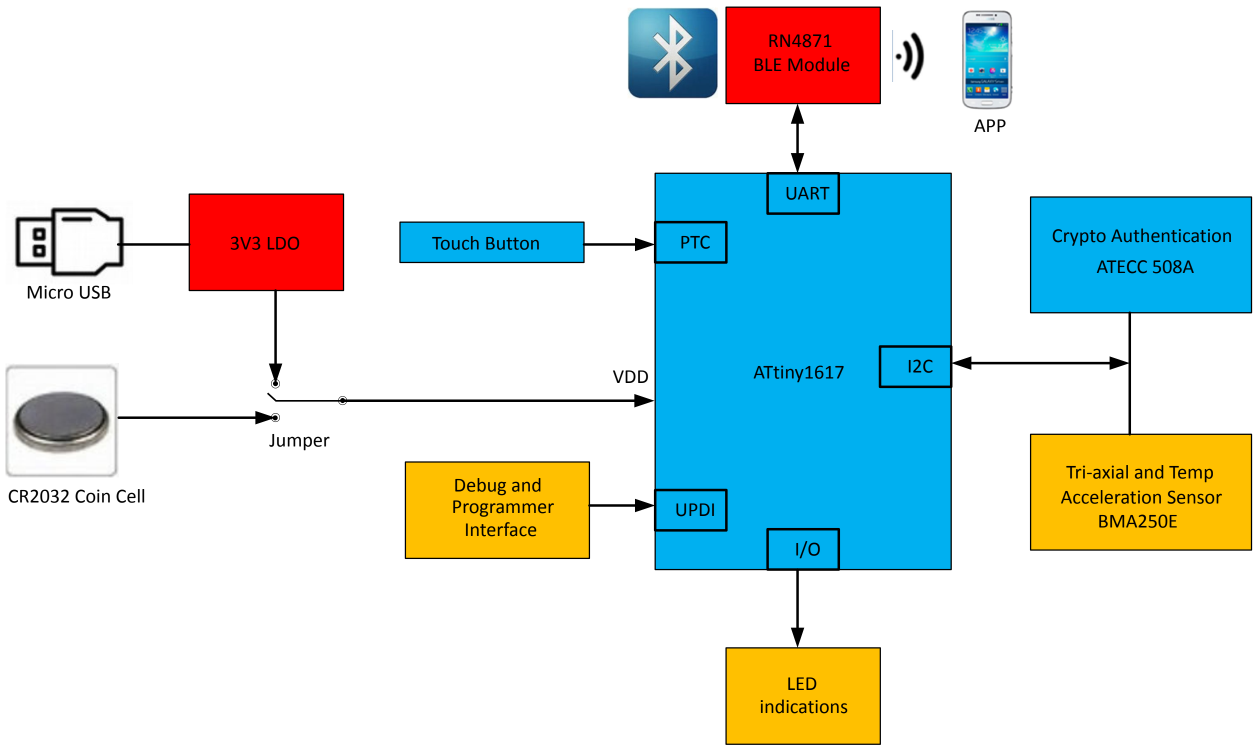
- ATtiny1617 usage in a real IoT application
- Node authentication based on ECDSA
- Bluetooth 4.2 Low-Energy connection
- Low-power operation with coin cell battery
- Run-time sensor reading and data exchange with mobile app
- LED indicators to show operation status

Table of Contents

1. Block Diagram

The secure AVR BLE IoT node is a typical edge node in the IoT scenario. Its block diagram is shown below.

Figure 1-1. Secure AVR BLE IoT Node Block Diagram



The kit is designed to be powered by either a coin cell battery or USB interface. The central part is the ATtiny1617 AVR MCU, which controls all the other devices in this kit. The ATECC508A is responsible for the certificate storage and authentication. The RN4871 BLE module provides connectivity with the phone. A tri-axial accelerometer and three LEDs are used for demo purpose.

For the details of hardware design, refer to the Secure AVR IoT Node - Hardware User Guide.

2. Development Tools

To download and debug the firmware, the following development toolchain can be used:

- Firmware example: Atmel | START.
- Atmel® Studio version 7.0.1188 or above with the latest ATtiny_DFP installed
- IAR Embedded Workbench® AVR 6.80 or above
- Programming and debugging tool: Atmel-ICE or Power Debugger.

For more information about the available toolchain, check the link below.

http://www.microchip.com/wwwproducts/en/ATTINY1617

3. Firmware Architecture

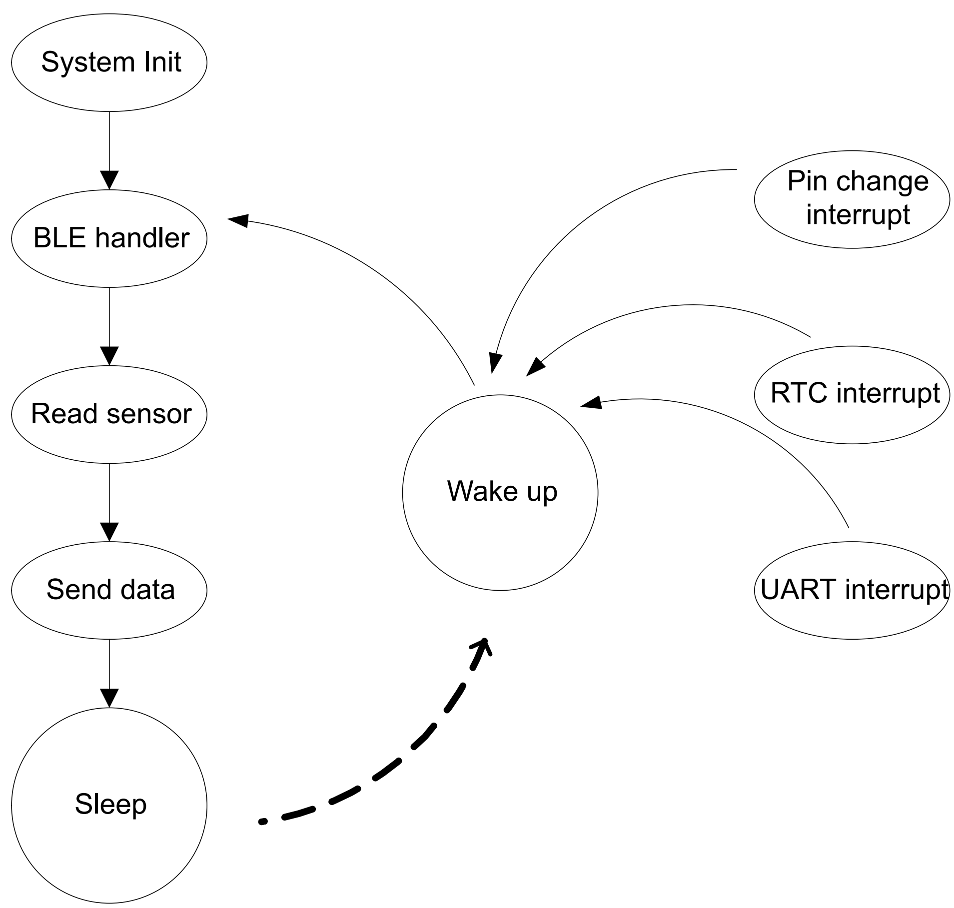
The firmware runs on the ATtiny1617 MCU at 5MHz by dividing the 20MHz internal oscillator by 4. This makes ATtiny1617 to work at wide voltage range (1.8V ~ 5.5V) which benefits the battery usage life. By functions, the firmware can be divided into the following blocks:

- BLE connectivity - UART communication with RN4871 module
- Node authentication - Crypto authentication lib and I^2C communication with ATECC508A
- Tri-axial acceleration sensor - Sensor initialization and reading
- Low-power operation - MCU and on-kit components low-power logic
- LED indicator - System operation status

More details of these function blocks will be introduced in separate sections in Firmware Function Blocks.

The firmware runs in an infinite loop. Below is the simplified main flowchart.

Figure 3-1. Main Flowchart



The 'Standby' sleep mode is used in this implementation. The transfer from 'Sleep' state to 'Wake up' state is triggered by MCU wake-up sources. Here, the pin change interrupt, RTC interrupt, and UART start of frame are used as wake-up sources. Refer to Low Power for more information.

4. Firmware Function Blocks

In this chapter, more details of the firmware functions blocks will be discussed.

4.1 BLE Connectivity

A Microchip RN4871 BLE module is used to enable the connectivity on this kit. The RN4871 is a small form factor, Bluetooth 4.2 Low-Energy module. It can easily be connected to the host MCU via a standard UART interface.

When this kit was developed, the RN4871 BLE module firmware version was 1.18.3. This firmware provides the control interface based on ASCII commands sent over UART. The UART interface settings are defined in the table below:

Table 4-1. RN4871 UART Settings

UART Setting	Value
Baud Rate	38400
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

In this application, private services and characteristics are used. These services and characteristics have to be configured into RN4871 BLE module before using them in the application. Refer to BLE Module Configuration for details. The private services and characteristics are defined in BLE Services and Characteristics.

After the kit is powered up, ATtiny1617 talks to RN4871 via the UART interface. It controls the RN4871 behavior by issuing ASCII commands. RN4871 responds to the command with data or result based on different commands. RN4871 will also generate status messages through the UART. The ATtiny1617 USART RX interrupt is enabled to handle the command response and status response from the UART interface. For complete command reference, check RN4870/71 Bluetooth Low Energy Module User's Guide.

RN4871 keeps advertising an incomplete list of 128-bit UUIDs and complete local name if no connection is established. The supplied app can then scan and discover nearby nodes. When a node is connected with the app, it stops the advertisement and transfers data back and forth with the app based on configured services and characteristics.

If the advertisement runs more than 30s, the RN4871 module will go into low-power advertisement by its own to save power. In addition, ATtiny1617 can put RN4871 into low-power operation if no data transfer via BLE is needed. In this case, the RN4871 runs at low frequency with much lower power consumption. ATtiny1617 will bring RN4871 back into high-frequency operation if a BLE data transfer is needed. Refer to BLE Module Low Power for RN4871 low-power operation.

For more information about RN4871, refer to its product page: http://www.microchip.com/wwwproducts/en/RN4871

4.1.1 BLE Module Configuration

Before using an RN4871 BLE module, it should be configured with dedicated settings, services, and characteristics according to the application needs. Interactive ASCII commands make it easily configured without any complex configuration tools. For this application, the following ASCII commands are used to configure the RN4871 BLE module.

- $$$
- SF,1
- $$$
- SN,AVR_TAG
- SO,1
- PZ
- R,1
- $$$
- PS,F05ABAC0393611E587A60002A5D5C51B
- PC,F05ABAD0393611E587A60002A5D5C51B,12,09
- PS,F05ABAC1393611E587A60002A5D5C51B
- PC,F05ABAD7393611E587A60002A5D5C51B,12,06
- PC,F05ABADA393611E587A60002A5D5C51B,10,01
- PS,F05ABAC2393611E587A60002A5D5C51B
- PC,F05ABAD2393611E587A60002A5D5C51B,1A,12
- LS
- NA,06,1BC5D5A50200A687E5113639C0BA5AF0
- NA,09,4156525F544147
- SB,05
- R,1

Among the above commands, PS is used to set up private services, the PC is used to set up private characteristics, and NA is used to set up advertisement information. For the details of other commands, refer to RN4870/71 Bluetooth Low Energy Module User's Guide.

Note: The default UART baud rate for a factory new RN4871 module is 115200 and the above commands are used for this purpose. If a module with different baud rate has to be configured, change the baud rate accordingly.

4.1.2 BLE Services and Characteristics

BLE uses the concept of profiles to ensure interoperability between different devices. Secure AVR BLE IoT Node uses a custom (private) profile with 128-bit unique UUIDs. It uses a base 128-bit UUID: F05A0000-3936-11E5-87A6-0002A5D5C51B. The BLE profile is a collection of services, and in this design, all services use 128-bit UUIDs. These services are built on top of attributes called characteristics. Each characteristic is identified by its 128-bit UUID. For easy documentation reason, a 16-bit part of UUID is listed in this document. It is embedded in the 128-bit UUID as shown in the example below.

Example: 0xBAC0 maps as F05ABAC0-3936-11E5-87A6-0002A5D5C51B. All UUIDs that are mapped to 128-bit values are marked with *.

The following private services are used in this design.

Table 4-2. Private Services

Services	UUID
Environment Service	BAC0*
Device Motion Service	BAC1*
Authentication Service	BAC2*

The characteristics in each service are listed below in separate tables. Reserved (Res.) byte is always 0x00.

Table 4-3. Environment Service

Characteristics	UUID	Properties	Data
Environment Data	BAD0*	Read/Notify	(9 bytes) Byte0 = Temp LSB Byte1 = Temp MSB Byte2 = Res. Byte3 = Res. Byte4 = Battery LSB Byte5 = Battery MSB Byte6 = Res. Byte7 = Res. Byte8 = Res.

Table 4-4. Device Motion Service

Characteristics	UUID	Properties	Data
Accelerometer	BAD7*	Read/Notify	(6 Bytes) Byte0 = X LSB Byte1 = X MSB Byte2 = Y LSB Byte3 = Y MSB Byte4 = Z LSB Byte5 = Z MSB
Drop Detection	BADA*	Notify	1 Byte

Table 4-5. Authentication Service

Characteristics	UUID	Properties	Data
Auth Data	BAD2*	Notify/R/W	18 Bytes

Authentication data (Auth Data) is used to transfer keys, signatures, and random numbers for node authentication. The data payload in BLE characteristic is not long enough to hold such data bytes in a single transfer, so it is divided in several segments when transferring. Thus Auth Data characteristics are formatted as below:

Type (1byte) + index (1byte) + Auth_data[0~15]

The table below defines Auth Data type and content.

Table 4-6. Auth Data Type and Content

Type	Auth Data Content	Note
0x01	Signer Public Key indexed by 0~3.	Public key length is 64-byte.
0x02	Signer Signature indexed by 0~3.	Signature length is 64-byte.
0x03	Device Public Key indexed by 0~3.	Public key length is 64-byte.
0x04	Device Signature indexed by 0~3.	Signature length is 64 byte.
0x05	*Certificate Information indexed by 0.*	*Info length is 16 byte.*
0x06	***Challenge (from host) indexed by 0~1***	***Challenge length is 32-byte.***
0x07	Challenge Response indexed by 0~3	Challenge response length is 64-byte.

Note: For 32-byte and 64-byte contents, the "Auth data" characteristic will be updated by 2 and 4 transfers with 16 bytes data in each transfer due to the payload limitation.

Certificate information is organized as; Signer Certificate Info (8 bytes) + Device Certificate Info (8 bytes). It is used together with public keys and signatures to reconstruct signer and device certificates at the app side.

4.2 Node Authentication

In this kit, Microchip CryptoAuthentication device ATECC508A is used for node authentication. The node authentication is based on Elliptic Curve Digital Signature Algorithm - ECDSA, which is embedded in ATECC508A. In the real world, it is used to prevents illegal (cloned, faked, or tampered etc.) nodes to be connected to the app and further to the Cloud.

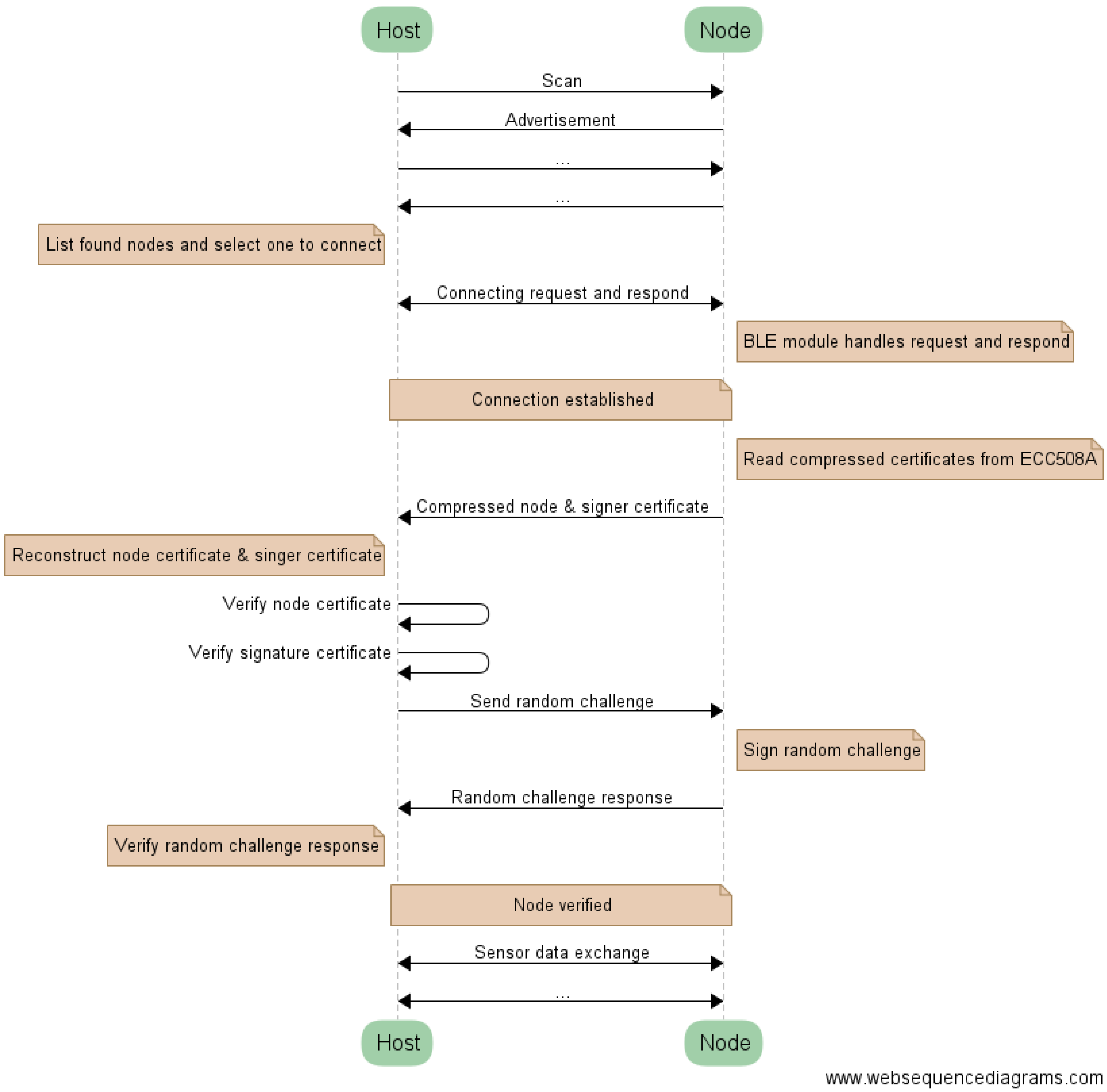
ATECC508A is connected to an I^2C bus as a slave with its address set to 0xC0, while ATtiny1617 works as I^2C master. On the same I^2C bus, a tri-axial acceleration sensor is also connected as I^2C slave. Refer to Acceleration Sensor for more information.

In this implementation, the node authentication happens right after a BLE connection is established between an app and a node each time. After the node authentication is completed successfully, no further authentication check is required until the current connection is terminated.

At the app (host) side, it will first verify the certificates of the node side. At the node side, the dynamic part of the certificates is stored in ATECC508A. It is also known as compressed certificate. After it is sent to the app, the full certificate will be reconstructed there, based on received compressed certificate and static part of the certificate, which is store at the app side. The app can then verify the full certificate. The chained node certificate and signer certificate are verified. If they are all passed, the app sends random numbers to challenge the node. Node signs the challenge and sends the challenge response back to the app. Again, the app verifies the challenge response. If it is passed too, the whole node authentication process completes successfully.

The figure below shows the node authentication process in this application. The Host side is the app, while node side is the AVR Secure BLE IoT Node.

Figure 4-1. Node Authentication Flow



As mentioned above, the ATECC508A device should be provisioned before it can be used in this application. Refer to the application notes below on how to provision it according to specific application needs.

- AN_8971 - ATECC Production Provisioning Guide
- AN_8974 - ATECC Compressed Certificate Definition

- AN_8983 - ATECC508A Node Authentication Example Using Asymmetric PKI

For more information about ATECC508A, refer to its product page: http://www.microchip.com/wwwproducts/en/ATECC508A

The CryptoAuthLib from Microchip is used in this firmware. The lib is provided as middleware to be used with CryptoAuthentication devices, including ATECC508A. Based on application needs, not all the ATECC508A functions are used. Refer to the source code project for the details.

4.3 Acceleration Sensor

As an IoT edge node, a sensor is usually a necessary part. On this kit, a tri-axial acceleration sensor BMA250E is used. It is connected to an I^2C bus as I^2C slave with the address set to 0x18. It shares the I^2C bus with ATECC508A. Refer to Node Authentication for more information. In addition, the BMA250E features two INT pins (INT1 and INT2) enabling motion-based applications without constant I^2C communication with the host MCU. The INT pins are connected to the MCU I/O pins and a PORT pin interrupt can be enabled to handle a corresponding interrupt request.

BMA250E measures acceleration in x-, y-, and z-axis, the three perpendicular axes. Besides that, a temperature sensor is also integrated in BMA250E. When BLE connection is established, both the acceleration data and temperature data are read and sent to the app in 200ms interval. If BLE is disconnected, the sensor data is read in 5s intervals. To save power, the sensor is initialized to work in low-power mode. Refer to Acceleration Sensor Low Power for more details.

To realize tap detection in this firmware, BMA250E low-g interrupt is enabled and it is assigned to the BMA250E INT1 pin with level active-low. Whenever a tap is detected (low-g interrupt), a low level is output on the INT1 pin and the corresponding ATtiny1617 PORT pin is configured to sense the low level, thus a port interrupt will be triggered. In the interrupt handler, the Alarm LED on the board is controlled to blink once to indicate a tap is detected. The tap detection is also operational in low-power mode.

To know more details about BMA250E, refer to its data sheet.

In the firmware, the BMA250E driver package from Bosch Sensortec is used. The original sensor driver link: https://github.com/BoschSensortec/BMA2x2_driver. To achieve smaller code size, some APIs are modified based on the application needs. Refer to the source code project for details.

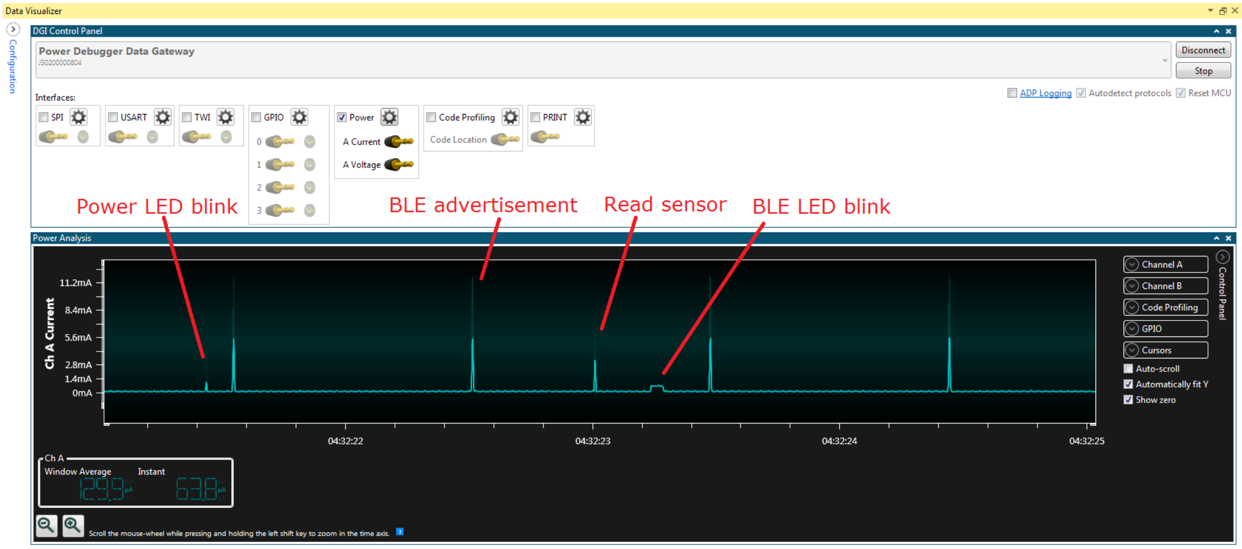
4.4 Low-Power

For IoT edge nodes, the power consumption is a key feature especially when it is battery powered. To minimize the overall system power consumption, the following low-power operations are addressed:

- AVR MCU low-power
- BLE module low-power
- CryptoAuthentication device low-power
- Acceleration sensor low-power

In the figure below one can see the current measurement result on a running kit in idle state. In this state, the BLE connection is not established but discoverable and the acceleration sensor tap detection is functional. The Power Debugger is used to measure the current in Atmel Studio.

Figure 4-2. Current Measurement Result



4.4.1 AVR MCU Low-Power

ATtiny1617 is the main controller in this kit. To save power, the MCU is put into sleep mode as often as possible. ATtiny1617 features three different sleep modes; Idle, Standby, and Power Down. In this firmware, 'Standby' sleep mode is used. Refer to ATtiny1617 Data Sheet for more information about the MCU sleep modes.

In 'Standby' sleep mode, the operation of selected peripherals are configurable. Based on the application needs, RTC is configured to run in Standby by setting the RUNSTBY bit. And RTC interrupt, UART start of frame, and pin change interrupt are configured as wake-up sources.

The RTC overflow interrupt is enabled to periodically wake up the CPU to read the sensor, check the BLE module status, and update the LED status etc. After the main loop tasks are done, the CPU is put into 'Standby' sleep mode while the RTC counter is kept running.

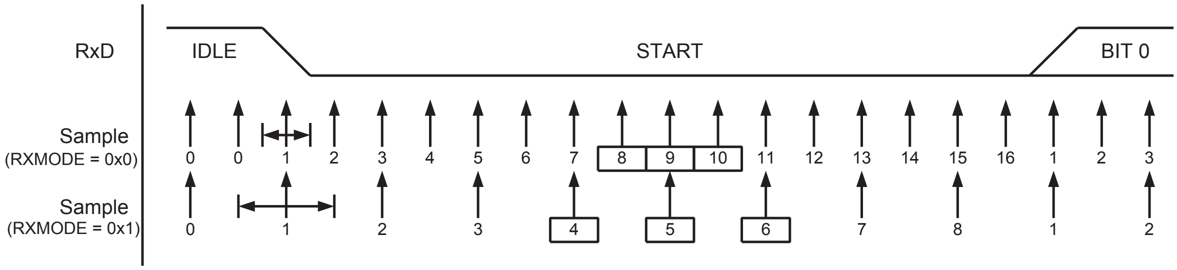
The UART start of frame detection works as wake-up source in the 'Standby' sleep mode. It is used to handle the BLE module status responses via UART. These status responses include, but are not limited to, connection, disconnection, and characteristics access status. By doing so, the MCU can be put into sleep mode while the BLE function remains active. Whenever UART bytes are received from the BLE module, the MCU will wake up and respond to the BLE communication immediately. As documented in the ATtiny1617 Data Sheet, the UART receive complete flag and UART start interrupt flag share the same interrupt line. To enable the USART start frame detection with the data receiving interrupt, the following settings are used in this design.

Table 4-7. USART Start Frame Detection with Data RX Interrupt

SFDEN	RXSIF interrupt	RXCIF interrupt	Comment
1	Disabled	Enabled	System/all clocks waked-up on Receive Complete interrupt

In order to receive data frame right after the UART start of frame wake-up, the UART baud rate should be set slow enough in relation to the oscillator start-up time. The sampling process for the start bit of incoming frame is shown in the figure below.

Figure 4-3. Start Bit Sampling



The sample rate is 16 times the baud rate for normal mode, and eight times the baud rate for double speed mode. The clock recovery logic then uses samples 8, 9, and 10 for normal mode and samples 4, 5, and 6 for double speed mode to decide if a valid start bit is received. If two or three samples have a low level, the start bit is accepted. Here, we calculate the baud rate with 10 samples to have some margin. The equation is as below.

$$\frac{16-10}{16 \cdot \text{Baud Rate}} > \text{wake−up time}$$

In the above equation, the wake-up time from the "Standby" sleep mode consists of six CLK + OSC start-ups. In this design, we use internal 16/20MHz oscillator. The start-up time of this oscillator is analog start-up time plus four oscillator cycles. Therefore we get the wake-up time as six Main Clock cycles (CLK_PER) + analog start-up + four oscillator cycles. That is 6*1/5MHz + 8μs + 4*1/20MHz = 9.4μs.

We can further calculate the baud rate capable of receiving the data frame right after UART start of frame wake-up as below.

$$\frac{6}{16 \cdot \text{Baud Rate}} > 9.4\text{us}$$

$$\text{Baud Rate} < \frac{6}{16 \cdot 9.4\text{us}} = 39894$$

Thus, we choose 38400 as the UART baud rate in this design.

More details can be found in "Asynchronous Clock Recovery" section of ATtiny1617 Data Sheet.

The pin change interrupt is used to wake up MCU from "Standby" sleep mode whenever a predefined motion is detected. The acceleration sensor INT pins are connected to the MCU I/O pins and the corresponding pin can be configured to handle the INT pin change via interrupt request. Refer to Acceleration Sensor for motion detection details.

4.4.2 BLE Module Low-Power

One of the key features of BLE is low-power operation. In this section, the how to utilize BLE module RN4871 low-power feature with the help of a host MCU is explained.

Obviously always transferring data via BLE is not a good strategy to save power. In this design, the data will only be sent to the BLE module via UART when the BLE connection is established. The data transfer frequency is application dependent.

The RN4871 module itself supports two different operation modes; high-frequency operation and low-frequency operation. This feature should be enabled by ASCII command "SO,1" during module configuration. Refer to BLE Module Configuration for details. When the UART_RX_IND pin is high, RN4871 is in low-frequency operation. In this low-frequency operation, the BLE connection can still be maintained, but RN4871 cannot receive data via UART. RN4871 restarts high-frequency operation by pulling its UART_RX_IND pin low. When data transfer via BLE is required, RN4871 is put into high-frequency operation. If BLE data transfer completes, RN4871 is put into low-frequency operation. The UART_RX_IND pin is fully controlled by the ATtiny1716 host MCU based on application needs.

By implementing these low-power system designs, the kit BLE function can be discovered and operational at any time from an application point of view while keeping the low-power operation in the background.

4.4.3 CryptoAuthentication Device Low-Power

Normally there is no special consideration of ATECC508A low-power operation, as it goes into low-power automatically. For more information, contact your local Microchip sales office. However, in this design the I^2C bus is shared by two components. The operation of acceleration sensor affects the operation of ATECC508A.

In this design, after the node authentication is completed, ATECC508A should be put in low-power until current connection is terminated. Note that frequently reading of the acceleration sensor will wake up ATECC508A unnecessarily and unintentionally. To eliminate this extra power consumption, ATECC508A is forced into low-power every time after the acceleration sensor operation is completed.

4.4.4 Acceleration Sensor Low Power

BMA250E has several power modes. As the sensor needs to detect tap (low-g interrupt) all the time, Low Power Mode 1 with balanced performance and power consumption is used. In this mode, it periodically switches between a wake-up phase and a sleep phase. In wake-up phase, all parts of the sensor are kept powered-up and data acquisition is performed continuously. In sleep phase, the whole analog part is powered down and no data acquisition is performed. The time in wake-up phase and sleep phase is configurable. Refer to the BMA250E data sheet for more details.

In this power mode, the interface section is kept alive and the latest acceleration data is always available by reading registers. There is no need to change power modes when reading the sensor data, thus it simplifies the firmware.

4.5 LED Indicator

On this kit, there are three LEDs.

- Power LED - green
- BLE LED - blue
- Alarm LED - red

Among them, Power LED and Alarm LED are controlled by the ATtiny1617 MCU pins. The Power LED is used to indicate MCU sleep/wake-up states. In this design, the Power LED blinks once at the frequency of the MCU wake-up. The Alarm LED is used to indicate tap detection. Every time a tap detection is reported by the acceleration sensor, it blinks once.

BLE LED is directly controlled by BLE module RN4871. It is used to indicate the BLE connection status. After the kit is powered up, the BLE LED blinks once every 3 seconds. If connection with a peer device is established, the BLE LED blinks twice per 1.5 seconds.

5. Get Source Code from Atmel | START

The example code is available through Atmel | START, which is a web-based tool that enables configuration of application code through a graphical user interface. The code can be downloaded for both Atmel Studio 7.0 and IAR IDE via the **Examples**-link below, or the **BROWSE EXAMPLES** button on the Atmel | START front page.

Web page: http://start.atmel.com/

Documentation: http://start.atmel.com/static/help/index.html

Examples: http://start.atmel.com/#examples

In the Examples-browser, search for Secure AVR BLE IoT Node

(press **User Guide** in Atmel | START for detailed requirements for the example project).

Double-click the downloaded .atzip file and the project will be imported to Atmel Studio 7.0.

For information on how to import the project in IAR, press the **Documentation**-link above, select ‘Atmel Start Output in External Tools', and 'IAR Embedded Workbench®'.

6. Revision History

Doc. Rev.	Date	Comments
A	06/2017	Initial document release

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: 978-1-5224-1815-3

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