

# ***INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE***

*Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018,  
publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.*

---

*Departamento de Electrónica, Sistemas e Informática*

*MAESTRÍA EN DISEÑO ELECTRÓNICO*



## ***BLE POWER MEASUREMENTS ON COMMERCIAL BOARDS***

*Tesis que para obtener el grado de  
MAESTRO EN DISEÑO ELECTRÓNICO*

*Presentan: IOSETH IBIS PEGUEROS LEPE*

*Director: DR OMAR HUMBERTO LONGORIA GANDARA*

*San Pedro Tlaquepaque, Jalisco. Junio de 2019*



## **ABSTRACT**

*The general goal of this thesis work is to provide a methodology to perform measurements on various commercial Bluetooth low energy (BLE) platforms. The electronic commercial boards presented in this work are manufactured by Intel-Arduino, NXP, Cypress, Microsoft-RaspberryPi3 and Texas Instruments.*

*The average current and the total power consumed are obtained by capturing and post-processing oscilloscope waveforms of BLE connection and advertising events. The measurement results will be used to estimate the battery life using different advertising and connection interval parameters and power consumed during a particular event.*

*The users of this thesis will have a solid and well-documented reference to help them choose one of the presented platforms or replicate the physical experiments on different platforms.*



# CONTENT

<i>Instituto Tecnológico y de Estudios Superiores de Occidente</i> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>iii</b>
<i>Content</i> .....	<b>v</b>
<b>Introduction</b> .....	<b>7</b>
<b>1. Bluetooth low energy theory</b> .....	<b>8</b>
1.1. <i>BLE ARCHITECTURE</i> .....	8
1.1.1 <i>Host</i> .....	9
1.1.2 <i>Controller</i> .....	11
1.1.3 <i>Protocol events</i> .....	13
<b>2. Measurement equipment and BLE development boards</b> .....	<b>18</b>
2.1. <i>MEASUREMENT EQUIPMENT</i> .....	18
2.2. <i>TI CC2540 MINI DEVELOPMENT KIT</i> .....	19
2.2.1 <i>TI CC2540 board description</i> .....	19
2.2.2 <i>TI CC2540 SoC description</i> .....	20
2.2.3 <i>Tools and Software for the CC2540 MINI dev kit</i> .....	20
2.3. <i>GENUINO/ARDUINO 101</i> .....	21
2.3.1 <i>Genuino 101 board Description</i> .....	21
2.3.2 <i>Genuino 101 SoC description</i> .....	22
2.3.3 <i>Tools and Software for Arduino 101</i> .....	23
2.4. <i>NXP FRDM-KW41Z: FREEDOM DEVELOPMENT KIT</i> .....	24
2.4.1 <i>FRDM-KW41Z Freedom Development board description</i> .....	24
2.4.2 <i>KW41Z SoC description</i> .....	25
2.4.3 <i>Tools and Software for FRDM-KW41Z Freedom Development board</i> .....	25
2.5. <i>CYPRESS CY8CKIT-042-BLE-A PIONEER KIT</i> .....	26
2.5.1 <i>CY8CKIT-042-BLE-A board description</i> .....	26
2.5.2 <i>CY8CKIT-042-BLE-A SoC description</i> .....	27
2.5.3 <i>Tools and Software for CY8CKIT-042-BLE-A</i> .....	27
2.6. <i>MICROSOFT IOT PACK FOR RASPBERRY PI 3</i> .....	28
2.6.1 <i>Raspberry Pi 3 board description</i> .....	28
<b>3. Power Measurements</b> .....	<b>29</b>
3.1. <i>TI CC2540 APPLICATION NOTE AS A REFERENCE</i> .....	29
3.1.1 <i>Test setup</i> .....	29
3.1.2 <i>Advertising and connection waveforms</i> .....	30
3.1.3 <i>Power consumption computation</i> .....	31
3.2. <i>POWER CONSUMPTION AND CURRENT EQUATIONS USED IN THIS WORK</i> .....	33
3.3. <i>TI CC2540 MINI DEVELOPMENT KIT</i> .....	35
3.3.1 <i>Schematics and measurement point</i> .....	35
3.3.2 <i>Modifications for measurements</i> .....	35
3.3.3 <i>Measurement results</i> .....	38

3.3.4	<i>Advertising</i>	40
3.3.5	<i>Connection</i>	41
3.4.	<b>GENUINO/ARDUINO 101</b>	42
3.4.1	<i>Schematics and measurement point</i>	42
3.4.2	<i>Modifications for measurements</i>	42
3.4.3	<i>Measurement results</i>	44
3.5.	<b>FRDM-KW41Z: FREEDOM DEVELOPMENT KIT</b>	50
3.5.1	<i>Schematics and measurement point</i>	50
3.5.2	<i>Modifications for measurements</i>	51
3.5.3	<i>Measurement results</i>	53
3.5.4	<i>Advertising</i>	53
3.5.5	<i>Connection</i>	55
3.6.	<b>CY8CKIT-042-BLE-A PIONEER KIT</b>	57
3.6.1	<i>Schematics and measurement point</i>	57
3.6.2	<i>Modifications for measurements</i>	58
3.6.3	<i>Measurement results</i>	60
3.6.4	<i>Advertising</i>	60
3.6.5	<i>Connection</i>	62
3.7.	<b>POWER CONSUMPTION SUMMARY</b>	64
3.7.1	Battery life estimation	64
3.7.2	Power consumption	66

# INTRODUCTION

Bluetooth core specification 4.0 introduced a low energy extension that includes: low energy physical layer, low energy link layer, AES encryption, attribute protocol (ATT), generic attribute (GATT) profile, Security Manager (SM) and other significant enhancements. Bluetooth low energy (BLE), which is the name of this new version of the Bluetooth protocol, was developed by the Bluetooth Special Interest Group (SIG) for short-range communication and is designed as a low-power solution for control and monitoring applications. This technology will be one of the key enablers of the internet of things (IoT) along with other communication protocols such as ZigBee, RFID, NFC and 6LoWPAN mentioned in a wide variety of IoT related publications. IoT is expected to enable physical things to collect, process and exchange data over the internet. For this reason, it's very important to understand the power consumption of the tasks that the sensors will be performing.

Several research articles have addressed the power consumption of the BLE protocol, by using both experimental and analytical methods. As an example, in [Gomez-12], the authors used the Texas Instruments (TI) CC2540 platform to perform power measurements and compared them with the results obtained in [Kamath-10] to calculate the battery life of a BLE slave. [Schrader-16] used two boards from Nordic semiconductor, nrf51, and nrf52 development boards, to measure power consumption on Advertising events to estimate the lifetime of beacons. In [Afonso-16], the authors used the CC2540 power measurements obtained in [Kamath-10] to contribute with an experimental evaluation of the reliability of BLE with small connection intervals and multiple sensor nodes.

As can be seen, experimental measurements are useful for those interested in backing up their theoretical calculations. For example, [Tei-15] used experimental measurements "available in the market" to estimate the lifetime of a BLE temperature sensor for a manufacturing application.

In this thesis work, physical power measurements will be performed in BLE commercial boards using Texas Instruments CC2540 as a base reference.

In order to better understand these measurements, the BLE stack and its communication events (advertising and connection) are reviewed in Chapter I.

Chapter II will give an overview of the development board features such as the type of processor they use, peripherals, software stack, etc., and the equipment used for the measurements.

Chapter III will contain a description of how the power was calculated and the measurement results.

Finally, Chapter IV will give conclusions of this work.

# 1. Bluetooth low energy theory

In this Chapter, the BLE protocol stack is reviewed to give an overall idea of this communication protocol and basic concepts. For this entire Chapter, the BLE core 4.1 [BLESIG-13] specification was used as a reference and the material here presented is fully based on this specification.

## 1.1. BLE Architecture

Bluetooth low energy protocol stack is composed of two main pieces: the LE controller and the Host. The LE controller is the lower layer in the stack (see Fig. 1-1), it is in charge of encoding and decoding BLE packets and transform the data into a baseband signal. The Host is the upper layer of the BLE protocol stack (see Fig. 1-1) and it is connected to the LE controller via the Host controller interface (HCI).

The HCI communicates the host and the controller and can be implemented with a USB-A, UART protocol, Secure Digital (SD) or three-wire UART.

The Application layer contains the use-case Software developed by the user to interface the BLE protocol stack.

Each part has its own internal components as shown in Fig. 1-1 and explained below. Host description is found in Section 1.1.1, LE controller in Section 1.1.2. After reviewing the stack, the communication events of the BLE protocol will be explained in Section 1.1.3.

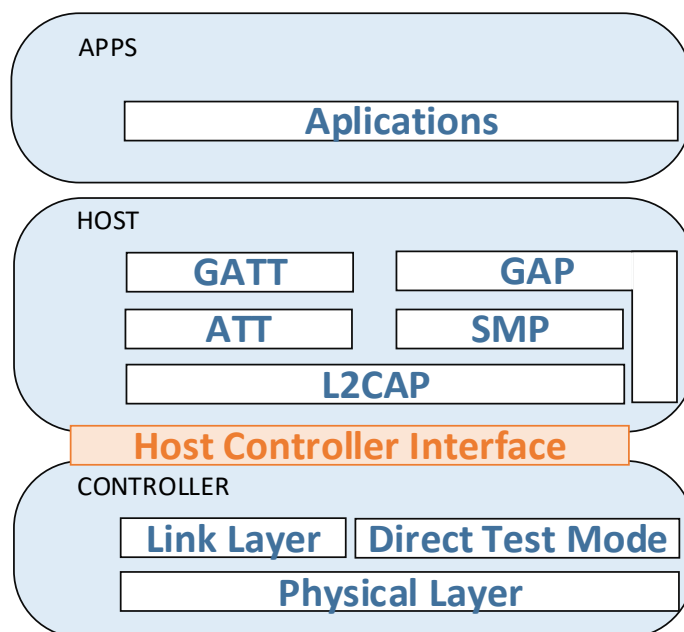


Fig. 1-1 BLE stack.



### 1.1.1 Host

As Fig. 1-1 shows, the Host is located above the LE controller and interfaced through the HCI. It defines different roles for devices and for different purposes. The services and characteristics of the device are managed by the GATT and the ATT. The roles of the devices based on the behavior of the transmission or reception of data are defined in the GATT. Security for authentication, pairing and encryption is managed by SMP and muxed/fragmented in L2CAP.

#### *Generic access profile (GAP)*

The purpose of this profile is to describe how devices are to behave in standby and connecting states in order to guarantee that links and channels always can be established between Bluetooth devices. Special focus is given on discovery, link establishment, and security procedures.

This profile defines three types of devices: "basic rate" or BR/EDR, "low energy" or LE only and "basic rate and low energy combined" or BR/EDR/LE.

GAP defines four roles for devices operating over an LE physical transport:

**Broadcaster.** A device operating in this mode sends advertising events which are described in Section 1.1.3

**Observer.** A device operating in this role receives advertising events which are described in Section 1.1.3

**Peripheral.** Any device that accepts and establishes a connection is referred to as a peripheral. A device operating as a peripheral will be the slave.

**Central.** Any device that initiates the establishment of a connection. A device operating as central will be the master.

#### *Generic attribute profile (GATT)*

The GATT profile defines services with multiple characteristics. A characteristic contains some value and additional information about the value. Services and their characteristics are contained in a Profile. At the same time, the profiles (and all of its elements) are contained by attributes which are used by the attribute protocol described below. Fig. 1-2 shows the hierarchy of a profile.

GATT implements two roles: Client and Server, which are different to the master and slave roles defined in the link layer, a device can act in both roles at the same time or switch between them. These roles are used only for controlling the discovery, read and write operation of attributes.

As an example let's think of a smartphone and a temperature sensor. The sensor would be the server of the temperature, it will provide information about the characteristics exposed as part of the temperature service. The smartphone, the temperature client, will initiate procedures to configure the sensor or to read the sensor values. The client can write some characteristics for configuration and request reading values of a characteristic.

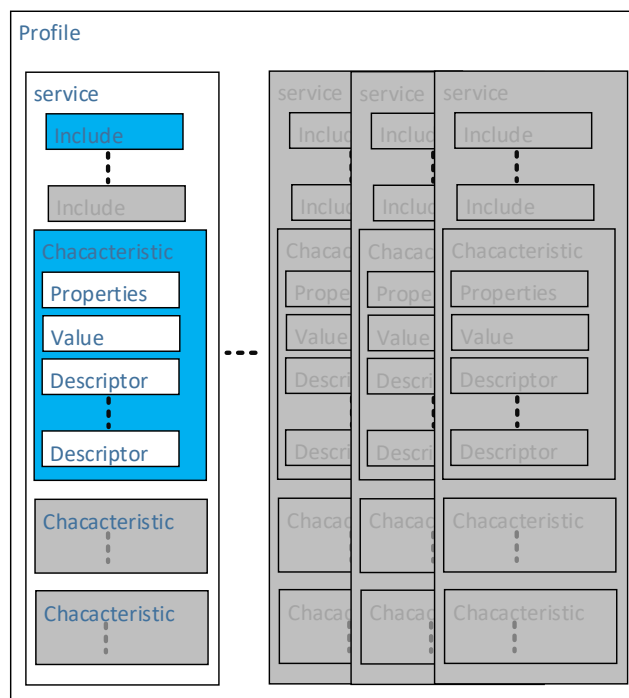


Fig. 1-2 GATT profile hierarchy.

### ***Attribute protocol (ATT)***

A client-server protocol for discovering, reading and writing attributes between devices. The attributes are defined in the GATT (as explained above) and are identified by a UUID (universally unique identifier). This protocol allows the server to expose a set of attributes to a client. A UUID is used to identify every attribute type and is considered unique. A UUID is a 128-bit value and the base value for Bluetooth is:

00000000-0000-1000-8000-00805F9B34FB

As specified in the Bluetooth spec [BLESIG-13]. 16-bit and 32-bit attribute UUIDs can use this common base.

$$128\text{-bit UUID} = 16\text{-bit Attribute UUID} * 2^{96} + \text{Bluetooth\_base\_UUID}$$

$$128\text{-bit UUID} = 16\text{-bit Attribute UUID} * 2^{96} + \text{Bluetooth\_base\_UUID}$$

### ***Security Manager (SM)***

The Security Manager defines the protocol and behavior to manage pairing, authentication, and encryption between LE devices.

The methods of pairing, key distribution, the protocol for those methods and a security toolbox used by those methods are defined in this layer. The key distribution is used to perform identity and encryption functionalities. Each device in a radio communication generates and controls the

keys it distributes. Pairing is performed to establish keys which can then be used to encrypt a link of communication.

### *Logical link control and adaptation protocol L2CAP*

This layer provides connection-oriented and connectionless data services to upper layer protocols with protocol multiplexing capability and segmentation and reassembly operation.

This layer is in charge of handling packets up to 64 kilobytes, these packets (L2CAP service data units, SDU) come from upper layer applications. LE controller (described in section 1.1.2) can't handle big size packets, and L2CAP is in charge of the segmentation of the upper layer packets into smaller packets with a payload size that the controller can handle, and the inverse process of assembling smaller size packets from the controller into packets formatted for the upper layer. L2CAP handles the flow control and retransmission, which is basically assigned data to a physical channel (described in section 1.1.2) and retransmit in case of an error.

### *1.1.2 Controller*

#### *Link layer*

This layer interfaces with the physical layer (explained below) and it is described as a state machine. The state machine implements the next states: Stand-by, Advertising, Scanning, Initiating, and Connection. The state transition diagram is shown in Fig. 1-3.

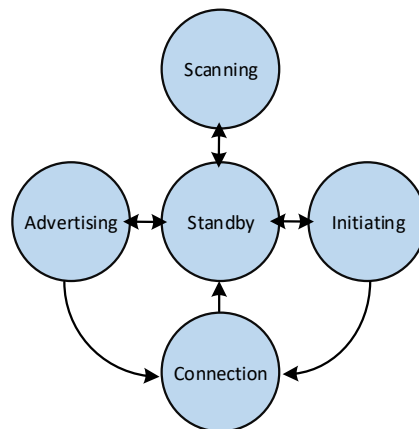


Fig. 1-3 Link layer state machine states.

In the standby state, the devices will not transmit or receive any packets.

Depending on what state a device is in, it will get a different role in the protocol. Devices in Advertising state will send advertising packets and will take the role of an Advertiser. Devices in Scanning state will listen for advertising packets from Advertisers.

Devices in the initiating state are called initiators and will listen for advertising packets from specific devices and respond them to initiate a connection. The connection state can be entered from the initiating or advertising states. In this state, it is considered a connection as established.

In this layer two roles are defined for a device in connection state:

**Master:** when the device goes from initiating to the connection state.

**Slave:** when the device goes from advertising to the connection state.

BLE defines 40 channels for packet transmission (see the section: *Physical layer*) 37 (0 to 36) of them are used for transmitting data packets and 3 (37 to 39) are used for transmitting advertising packets. Link layer selects and uses one physical channel at a given time.

The link layer is in charge of handling advertising and data packets. It calculates and appends the CRC (cyclic redundancy check) for error checking and optionally the MIC (message integrity check) for encryption/decryption purposes.

The link layer assembles the advertising and data packets, which consist of four components: preamble, access address, protocol data unit PDU (includes an optional MIC), and CRC.

The fields of the packets won't be covered in this work. But it's important to mention that PDU contains the actual data payload, plus a header that differentiates the type of packets for advertising and connection events covered in 1.1.3. Advertising and Data PDUs can be at least 2 bytes and maximum 39 bytes in length.

### ***Physical layer***

The physical layer implements the analog circuitry used for modulation and demodulation. A Gaussian Frequency Shift Keying (GFSK) modulation scheme is implemented with a bandwidth-bit period product  $BT=0.5$ . The modulation index is between 0.45 and 0.55.

Low energy system operates in the 2.4 GHz band, also known as ISM (industrial, scientific, medic). And uses 40 RF channels with center frequencies of  $2402 + k*2$  MHz, where  $k=0\dots, 39$ . The bit rate at which the data is transmitted is 1Mbps +/- 1ppm.

The physical layer allows for higher level protocols and applications to transmit and receive upper layer data packets. It also allows for per-channel flow control and retransmission.

### 1.1.3 Protocol events

Section 1.1.2 (*Link layer*), it was explained how a BLE device can be in a connection state. In this state, data packets are exchanged between the devices during a time window called connection event. But before the exchange of data packets during a connection event is performed, two devices have to establish the connection. There has to be an initiation request (a type of packet) during a time window called advertising event.

The time window for an advertising and connection event is the duration from the start of the event until the event is closed and it is slightly different for the two types of event. Both events (advertising and connection) are explained in more detail below.

Advertising and connection events are the most relevant aspects from BLE for this thesis work since these are fundamental for understanding the power measurements described in Chapter 3.

#### Advertising events

There are 4 types of advertising events:

- Connectable Undirected Event
- Connectable Directed Event
- Non-connectable Undirected Event
- Scannable Undirected Event

All of these types of advertising events use a different PDU (described in the link layer section above) type of message and cause different behavior on a device that listens waiting for them, i.e. a scanner. But they essentially have similar characteristics.

In an Advertising event, the advertiser will send advertising PDUs on the 3 advertising channels. From the start of a transmission on one channel to the next there will be a time separation less or equal to 10ms. The host will select the lowest channel index to start the advertising event and will end the advertising event once it has used the last advertising channel index. Fig. 1-4 shows a diagram of an advertising event without any connection or scan requests sent by a scanner, which means there is no interaction with a scanner. In Fig. 1-4 the event is started when the first advertising packet is sent and closed after the third packet was sent on the last advertising channel.

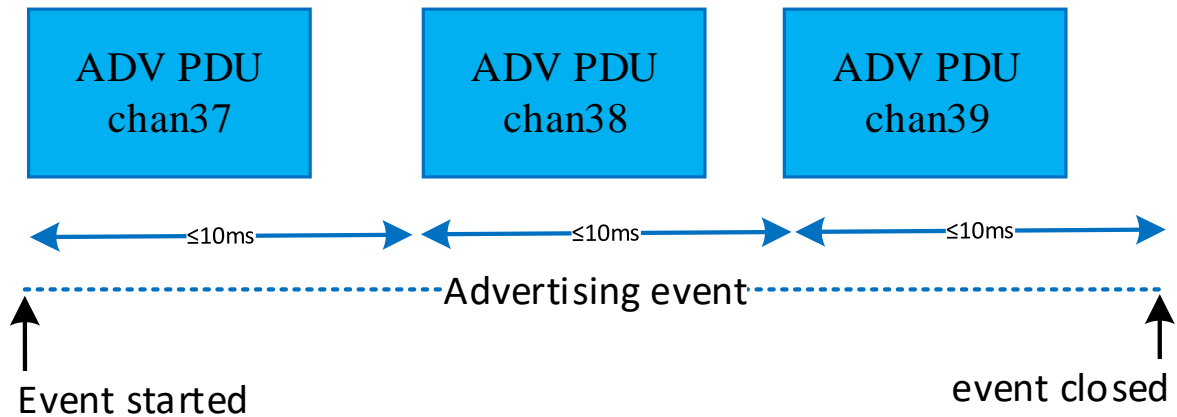


Fig. 1-4 Advertising event in Bluetooth low energy

If there is an interaction with a scanner, scan requests and responses will come after any of the advertised packets. Fig. 1-5 shows an advertising event in which after a time separation called interframe space ( $T_{IFS}$ ) when the second advertising PDU is transmitted, a scan request message is received and after one interframe space, the slave sends a scan response. This scan requests and response can happen right after any of the advertising channels is used and the Advertising event completes.

An advertising event will end if the advertiser receives a connection request after sending an advertising PDU. This means that the advertiser will not send more advertising PDUs on the rest of the advertising channels.

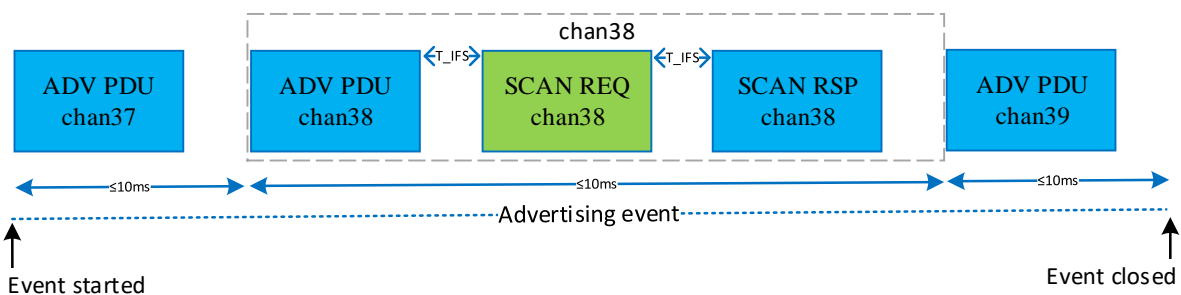


Fig. 1-5 Scan request and response in the middle of an advertising event.

The advertiser response to a scan request depends on what type of advertising packets was sent, refer to the Bluetooth core specification for more information.

Advertising events are separated by an advertising interval (*advInterval*) and an advertising delay (*advDelay*) as shown in Fig. 1-6. The *advInterval* is in the range of 20 ms and 10.24 s with a step of 0.625 s. The *advInterval* should not be smaller than 100ms.

The *advDelay* is a pseudo-random value with a range of 0 ms to 10 ms.

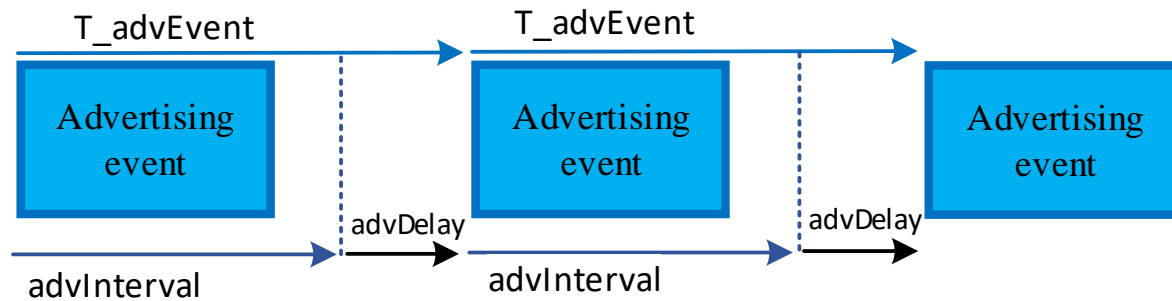


Fig. 1-6 Timing between advertising events

### Connection events

In the same way that an advertising event gets a scan request as shown in Fig. 1-5 it can also get a connection request and transition from advertising to connection events. Connection events use data channels for packet transmission.

After the advertiser (the device who starts the advertising events) receives a connection request from the initiator, a connection is considered to be created. The connection is created, but the connection is considered established once a data PDU is received from the peer. Once in the connection state, the devices will act in master and slave roles respectively. In other words, a connection is created when the advertiser receives a connection request and established upon the first data packet exchanged.

The connection request packet includes parameters to determine when the master can send its first packet, which is the Transmit window size (*transmitWindowSize*) and the transmit window offset (*transmitWindowOffset*).

After  $1.25\text{ms} + \text{transmitWindowOffset}$  a transmit window will start and the duration of it will be specified by the *transmitWindowSize* parameter, in which the first data packet will be sent by the master. The *transmitWindowOffset* has a range from 0 ms to connection interval (*connInterval*) with a step of 1.25 ms. the *transmitWindowSize* range goes from 1.25ms to the lesser of 10ms and (*connInterval* - 1.25ms).

Fig. 1-7 shows the creation of a connection with the transmission of a connection request and the connection setup with the transmission of the first data packet after the *transmitWindowOffset*. The second data packet is sent after one *connInterval*, and the same for all the next connection events.

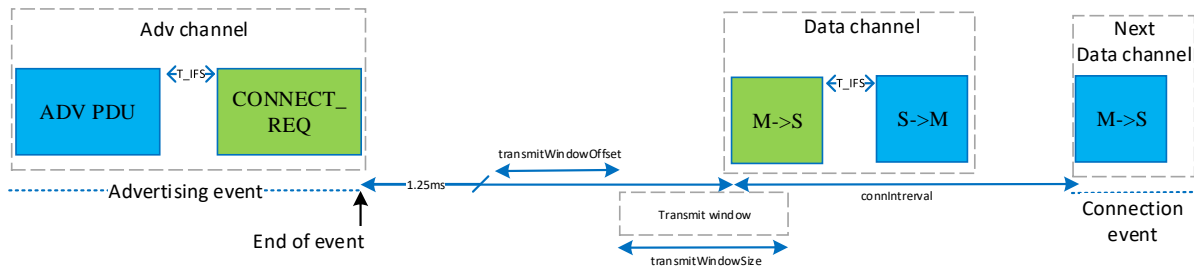


Fig. 1-7 Connection creation in Bluetooth low energy

The start of connection events is called anchor point, and at this point, the master starts transmitting a data packet to the slave.

The timing of the connection events is determined by connection event interval (*connInterval*) and slave latency (*connSlaveLatency*). All connection events will have a separation equal to *connInterval* + *connSlaveLatency*. *ConnInterval* is 7.5ms to 4.0 s with a 1.25 ms step. *ConnSlaveLatency* is a parameter that defines a number of consecutive connection events that the slave is not required to listen for the master and should be less than 500. In other words, slave latency or *connSlaveLatency* is the number of connection events that the slave is allowed to ignore, i.e., not send a data packet, without the master closing or ending the connection.

The connection interval plus slave latency should not exceed the supervision timeout (*connSupervisionTimeout*) otherwise the connection is considered lost. Connections can break unexpectedly due to numerous reasons that's why it is important that the devices monitor the status of the connection and avoid reaching the *connSupervisionTimeout*.

Connection events will use data channels. The host of the device identifies used and unused channels: used for connection and not used for connection respectively. After identifying the channels, it will report back to the link layer what these channels are.

Used channels are mapped to channel indexes. The channel used for a connection event is the *unmappedChannel* which is calculated with the next formula

$$UnmappedChannel = (lastUnmappedChannel + hopIncrement) \text{ mod } 37$$

The *lastUnmappedChannel* is the *unmappedChannel* from the previous connection event, which in case of being the first connection event should be equal to 0. The *hopIncrement* is a number from 5 to 16 sent in the initiating message sent by the master. Fig. 1-8 shows two consecutive connection events.



# 1. BLUETOOTH LOW ENERGY THEORY

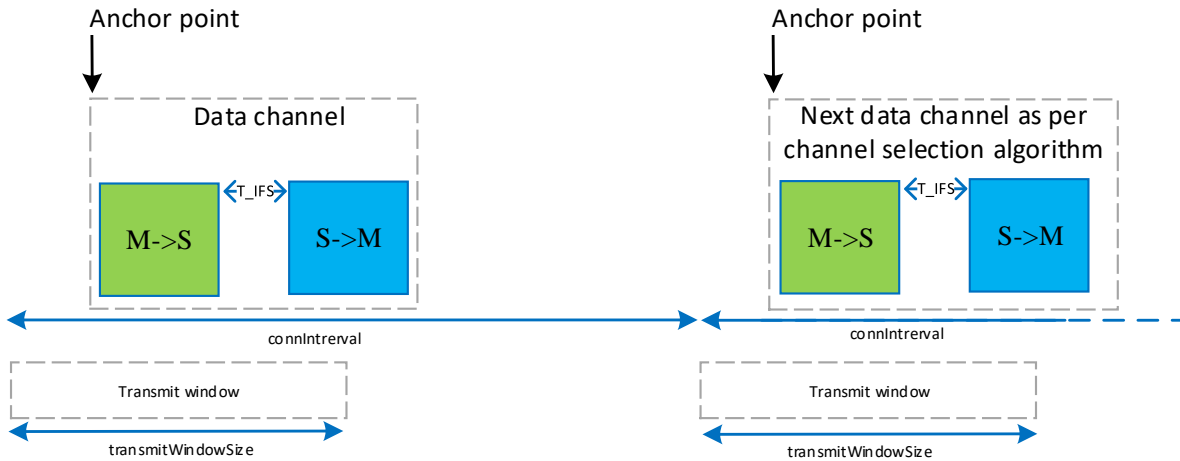


Fig. 1-8 Timing between connection events

Fig. 1-9 shows an example of a connection of two BLE devices from the advertising packets transmission, connection creation, connection establishment, and until the transmission of constant data in the connection events. In this example, it can be seen the states of the slave's link layer going from advertising to initiating and finally staying in connection (standby was omitted since it's assumed the device went straight to advertising after waking up).

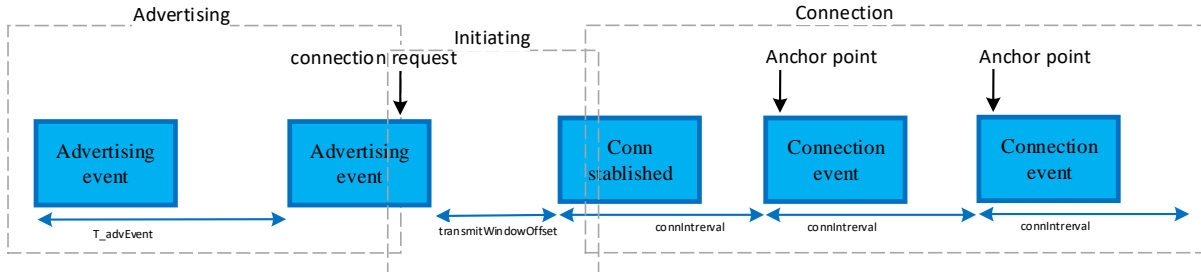


Fig. 1-9 from advertising to connection event sequence

## 2. Measurement equipment and BLE development boards

In this Chapter, the 4 different BLE development boards are reviewed. The review includes the parts of the development kits, they consist of several items such as dongles, cables, programmers, etc., board features and components as well as the main SoC characteristics. This chapter starts with a description of the measurement equipment used for capturing data to process/compute the power consumption on the presented boards.

### 2.1. Measurement equipment

The measurements in this thesis work were performed by a Tektronix DPO7252 oscilloscope. The oscilloscope features of interest for this work are 4 input channels, bandwidth of 18 GHz and a maximum sample rate of 50 GS/s (for 1 or 2 channels) 25M points as maximum record length with standard configuration (with 1 or 2 channels) [tektronix-09]. The oscilloscope has a different maximum in case of using more than 2 channels (i.e. 3 or 4 channels) but only 1 channel was used in this work. Fig. 2-1 shows the measurement equipment.



Fig. 2-1 Tektronix DPO7252 Oscilloscope used for measurements. Image taken from [Tektronix-09].

Standard passive probes were used with main features being a bandwidth up to 1GHz, input impedance of 10 M ohm and input capacitance as low as 3.9pF [tektronix-18], Fig. 2-2 shows the same single test probe used in this thesis work.

## 2. MEASUREMENT EQUIPMENT AND BLE DEVELOPMENT BOARDS



Fig. 2-2 Tektronix passive probes used for measurements. Image taken from [Tektronix-18]

### 2.2. TI CC2540 Mini development kit

#### 2.2.1 TI CC2540 board description

The CC2540 MINI dev kit is composed of a USB dongle with BLE capability, a key Fob and a debug dongle shown in Fig. 2-3. The dongle is programmed to be used with the provided tools for testing applications loaded on the key Fob. Key Fob includes a CC2540 SoC and the USB dongle a CC2541, which are identical except for the implemented interface I2C and USB respectively. (From this point and on the Key Fob will be referred to as CC2540 board and the USB dongle the same way.



Fig. 2-3 CC2540 MINI development kit main elements. Image taken from [TI-13b]

## 2.2.2 TI CC2540 SoC description

TI CC2540 and CC2541 were among the first BLE SoCs in the market. Some of the main HW features (shown in Fig. 2-4) this SoC includes are a High-Performance and low-Power 8051 compatible microcontroller core, 8KB of SRAM, 12-BIT ADC with eight channels, a couple of General-Purpose timers, two USARTs, AES security coprocessor for BLE data encryption, among others.

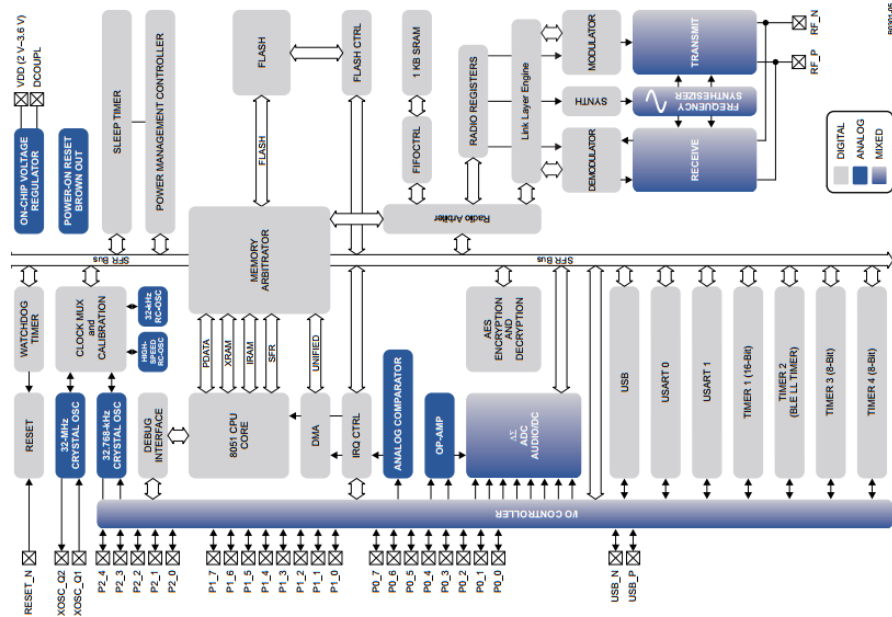


Fig. 2-4 CC2540 SoC subsystems. Image taken from [TI-15].

## 2.2.3 Tools and Software for the CC2540 MINI dev kit

TI provides schematics, bill of materials, CAD and Gerber files [TI-18]. A BLE stack is also available for download along with a few application examples code and compiled files ready for flashing the devices.

Some of the examples provided are Blood Pressure, Heartrate monitor, thermometer, etc. And they can be further developed and compiled/built with IAR embedded workbench for 8051. Although there is a trial version of this development tool, it is not possible to compile any of the BLE examples due to constraints in memory usage. All of the sample code exceeds this limit and forces the project developers to buy a License.

TI provides tools for programming the flash memory, upgrading firmware and bootloader on evaluation boards SoCs including CC2540/41, a Packet Sniffer to capture and display over-the-air

## 2. MEASUREMENT EQUIPMENT AND BLE DEVELOPMENT BOARDS

packets and a windows application used to evaluate and configure RF devices and help designers to evaluate the radio at an early stage in the design process.

### 2.3. *Genuino/Arduino 101*

#### 2.3.1 *Genuino 101 board Description*

Genuino or Arduino 101 is a development board with an Intel Curie SoC and the form factor and peripheral list of the popular UNO. Not only is this board compatible with all of the UNO shields and accessories but the Curie adds Bluetooth low energy connectivity. The board is powered either via a USB cable or an external AC-DC converter. See picture Fig. 2-5.

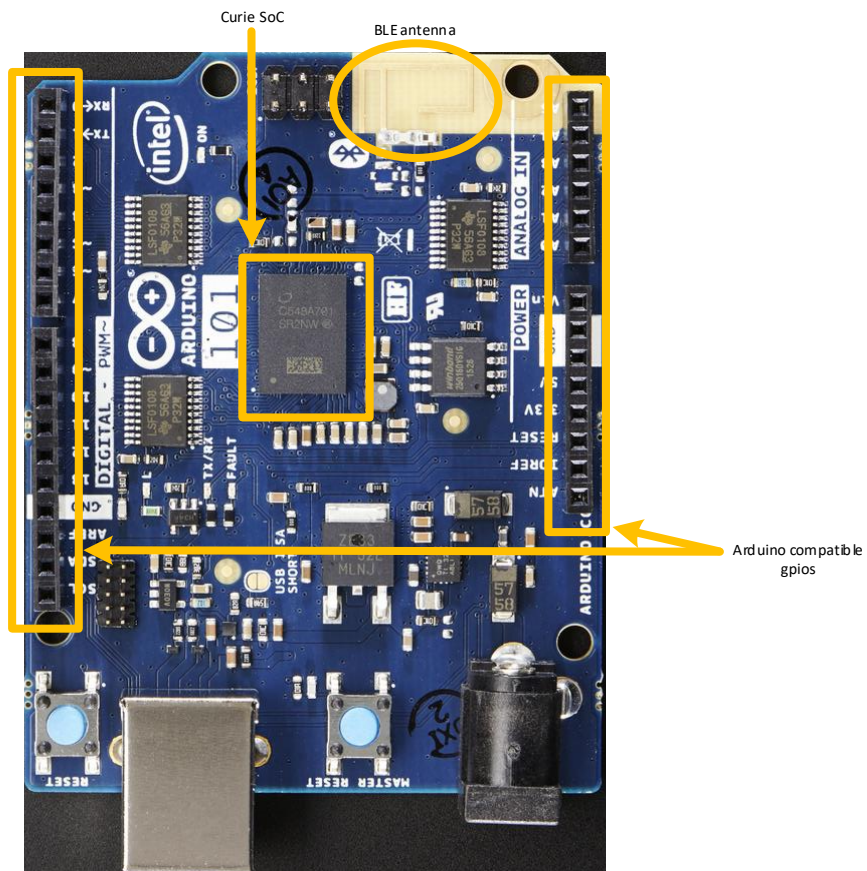


Fig. 2-5 Genuino 101 development board. Image downloaded from [Arduino-18].

### 2.3.2 Genuino 101 SoC description

The Intel Curie SoC (from this point on referred to as Curie) is described as a complete, low-power solution that comes with compute, motion sensor, BLE, battery-charging, and pattern matching capabilities for sensor data Analysis.

Curie subsystems are a Quark SE C1000 processor core, Nordic NRF5182 BLE module, Bosh BM160 6-AXIS accelerometer/gyroscope, TI Q2510H battery charger, and a couple of power supervisors and LDOs. In Fig. 2-6 we have a high-level block diagram of the Curie main blocks.

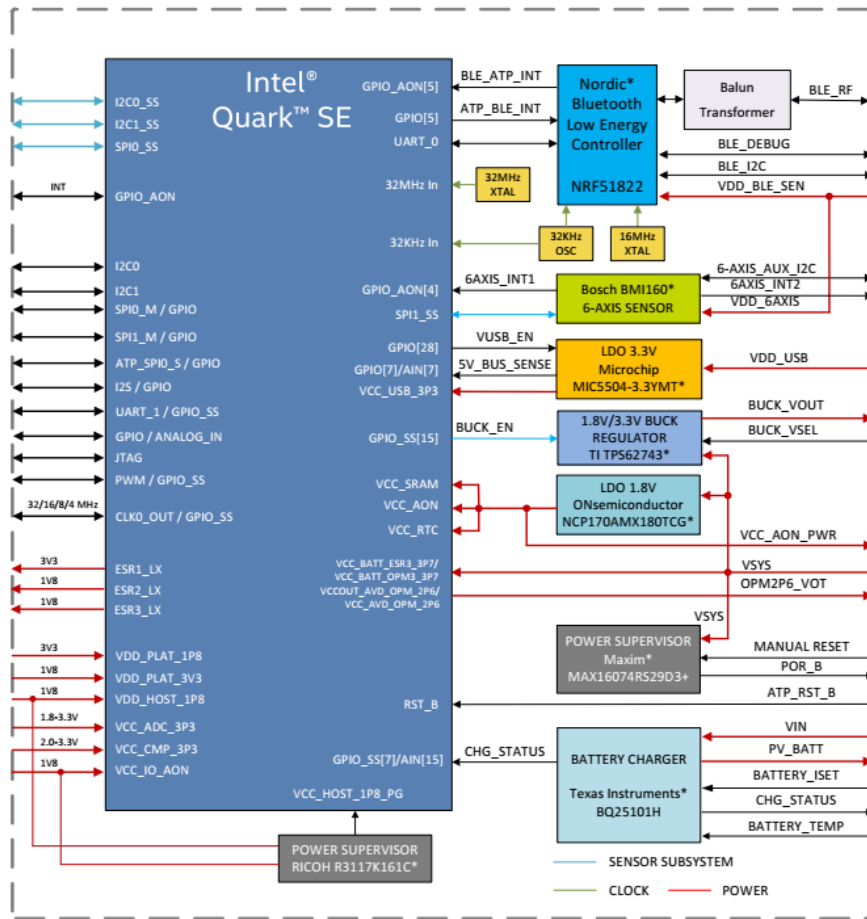


Fig. 2-6 Intel Curie block diagram. Image taken from [Intel-16].

## 2. MEASUREMENT EQUIPMENT AND BLE DEVELOPMENT BOARDS

Being Quark the main subsystem which contains two cores, an x86 ISA-compatible CPU and a 32-bit ARC architecture core, both running at 32MHz. Built-in pattern recognition and pattern matching accelerators are also included. Interfaces for the two processors include i2c, SPI, UART, I2S, and GPIOs. The memory subsystem of 384 KB Flash + 8 KB ROM and 80KB SRAM. A set of timers, watchdog, RTC and 19 channel 12-bit ADC.

### *2.3.3 Tools and Software for Arduino 101*

Arduino is an open-source platform, meaning the hardware and the software are open and free users to develop their applications. CAD and schematic files are available online as well as the Arduino IDE which comes with examples to start using the board out of the box [Arduino-18]. Basic peripheral control, analog input, serial communications to Robot/Servo control are some examples.

Additionally, the Zephyr project provides support for Intel-based (x86) boards including Arduino 101 board. Zephyr is a small-footprint kernel designed for use on resource-constrained systems [Zephyr-17]. It provides security, it is an open source free to use for commercial and non-commercial solutions.

Zephyr comes with a Bluetooth low energy stack compatible with Nordic Semiconductor nRF51, subsystem integrated inside the Curie.

Zephyr project relies on Cmake as a tool for building projects. The Zephyr SDK is provided for convenience and ease of use [Zephyr-17]. It allows the user to choose the default compilation toolchain or make use of a third party option like GCC ARM Embedded compiler or Intel System Studio for Microcontrollers (ISSM).

## 2.4. NXP FRDM-KW41Z: Freedom development kit

### 2.4.1 FRDM-KW41Z Freedom Development board description

The Freedom Development Kit for Kinetis® developed by NXP semiconductor is compatible with Arduino UNO shields and accessories. It has Arduino compatible headers, onboard 3D accelerometer + 3D magnetometer, onboard serial flash, SMA RF connector, 2 push buttons, 2 TSI buttons, an RGB LED and a couple debug interfaces.

The SMA connector is optional since the board has an F-Antenna printed on the PCB.

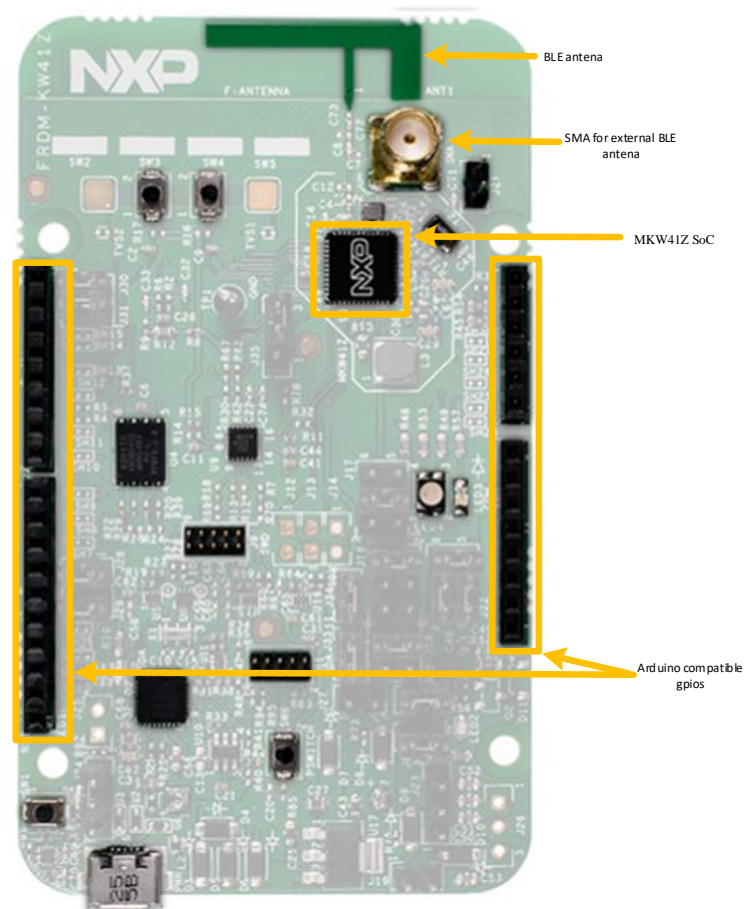


Fig. 2-7 NXP freedom development board. Image downloaded from [NXP-18b].



## 2. MEASUREMENT EQUIPMENT AND BLE DEVELOPMENT BOARDS

### 2.4.2 KW41Z SoC description

The MKW41Z contains an Arm core processor, wireless connectivity (BLE and IEEE 802.15.4) and analog modules on a single chip. The main core is an ARM Cortex-M0+ running at 48 MHz. The multi-protocol radio subsystem support Bluetooth low Energy 4.2 and 802.15.4 based network like Thread and ZigBee. Memories of up to 512 KB flash and up to 128 KB SRAM. 16-bit ADC, 12-bit DAC, and a 6-bit analog comparator are the analog submodules. It also supports wired interfaces such as I2C, SPI, and LPUART and has a few types of timers like a low-power timer, real-time clock, periodic interrupt timers, and Flex timers. Fig. 2-8 shows all of these blocks.

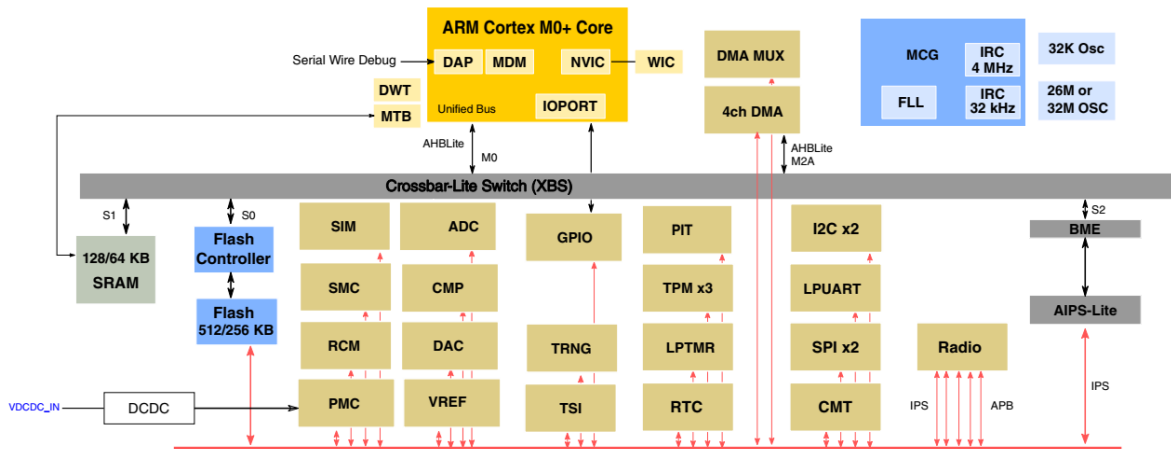


Fig. 2-8 MKW41Z SoC subsystems block diagram. Image taken from [NXP-18].

### 2.4.3 Tools and Software for FRDM-KW41Z Freedom Development board

NXP provides all the printed circuit board information and schematics FRDM-KW41Z, which means this is an open hardware design. The software they provide is the MCUXpresso SDK which includes all the software stacks required for developing BLE, IEEE 802.15.4, Thread solutions and a couple RTOS. There is the option to use the MCUXpresso IDE or installing a different toolchain such as IAR (IAR SYSTEMS), ARM Keil, Green Hills, among other commercial IDEs for ARM Cortex-based microcontrollers. All of the tools and documentation are found in [NXP-18b].

## 2.5. Cypress CY8CKIT-042-BLE-A Pioneer kit

### 2.5.1 CY8CKIT-042-BLE-A board description

Cypress CY8CKIT-042-BLE-A Pioneer kit (from this point on named as Pioneer board) comes with an Arduino compatible board, the PSoC 4 BLE module and the CY5677 CySmart BLE 4.2 USB Dongle.

The Pioneer board features an ARM Cortex-M3 processor, CapSense slider, one push button, an RGB LED, a CapSense proximity header, a coin cell reset button for BLE device, headers for the PSoC 4 BLE module and the actual BLE SoC. Pioneer board key elements are shown in Fig. 2-9.

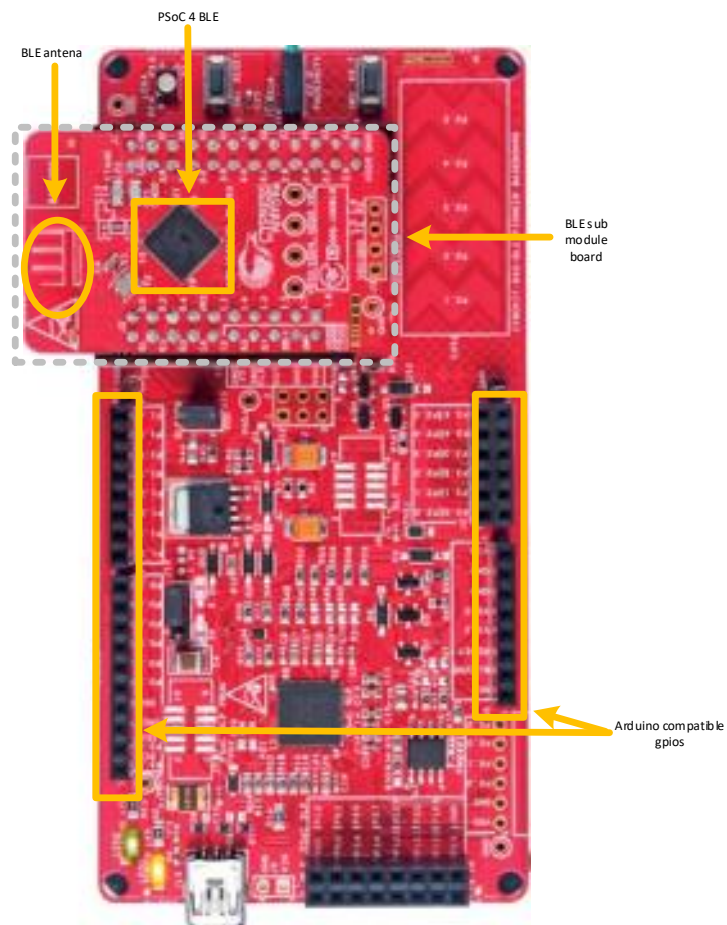


Fig. 2-9 Cypress Pioneer development board. Image downloaded from [Cypress-16c].

## 2. MEASUREMENT EQUIPMENT AND BLE DEVELOPMENT BOARDS

### 2.5.2 CY8CKIT-042-BLE-A SoC description

The processor on the Pioneer board has KitProg firmware and is used as the onboard programmer/debugger and USB-Serial interface to the Processor on the BLE module which is the one carrying the main core processor with an interface to the Arduino compatible headers and the peripherals on the Pioneer board.

The PSoC 4 BLE (see Fig. 2-10) itself has an ARM Cortex-M0 running at 48MHz, BLE radio, CapSense, 12-bit ADC, 2 current DACs, 2 comparators, 4 Timer/Counter PWMs, digital blocks and 2 Serial communications Blocks.

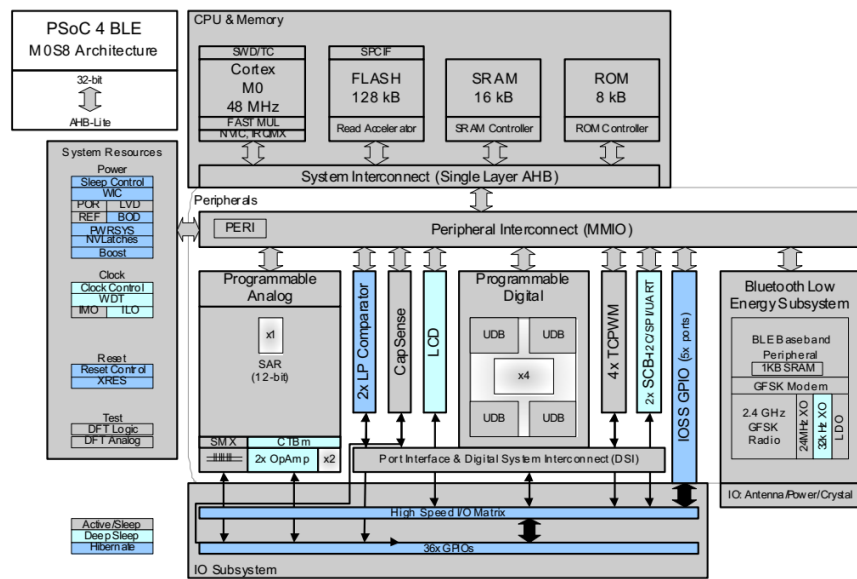


Fig. 2-10 PSoC4 SoC subsystems. Image taken from [Cypress-16d].

### 2.5.3 Tools and Software for CY8CKIT-042-BLE-A

The USB dongle is meant to be used with the CySmart IDE to test the samples loaded on the Pioneer board.

The PSoC Creator 4.1 is the tool for development of applications and comes with a couple preloaded examples to use the Pioneer board as a peripheral or central mode.

Cypress along with the development tools also provides all of the board design files (schematics, layout files, Gerber files, BOM) and guides for creating/customizing your own BLE examples [Cypress-16d].

## 2.6. Microsoft IOT pack for Raspberry Pi 3

### 2.6.1 Raspberry Pi 3 board description

The Microsoft IOT pack for Raspberry Pi 3 is a kit that comes with some passive elements like photocells, jumpers, capacitor, LEDs, potentiometer, cables like Ethernet and micro USB for charging and a MicroSD memory card. The board Raspberry Pi 3 (Fig. 2-11) has peripherals like USB ports, Ethernet ports, HDMI port a camera module port, Audio jack, a port for SD cards and a set of GPIO pins. The main chip on this board is an ARM-based BCM2937B0 chip manufactured by Broadcom.

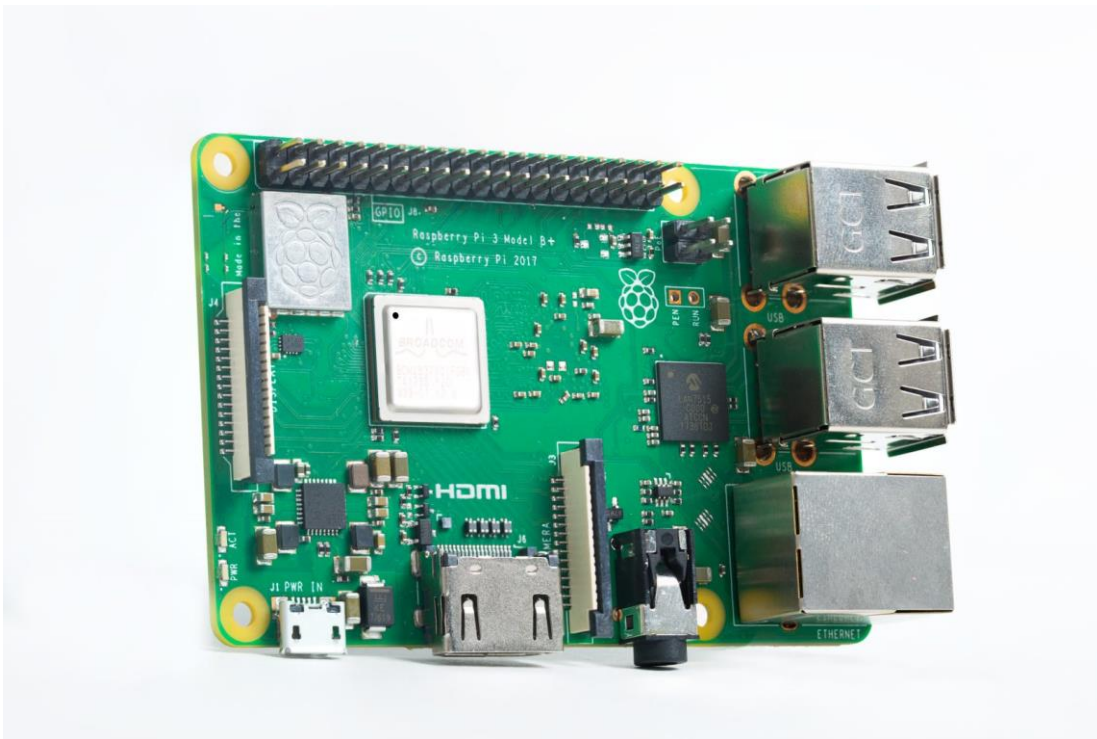


Fig. 2-11 Raspberry Pi 3 development board. Image downloaded from [Adafruit-18].

The Broadcom chip on this board is a 64-bit ARM-based core running at up to 1.4 GHz. The chip has also a video core running at 400 MHz.

This board as just described above and based on the applications it's used for, like running programs on a loaded operating system, is considered more of a computer device. Due to its high performance, hence high power consumption, and the fact that the schematics of the board aren't available, no power measurements were attempted on this board.

### 3. *Power Measurements*

This chapter starts by showing the results of replicating the measurements presented by TI [Kamath-10] and the formulas used to calculate/compute the current and power in this thesis work. This chapter also describes what board reworks were necessary for each board in order to have a measurement point just like the reference CC2540 board, the captured and post-processed waveforms and the results obtained for each board.

#### 3.1. TI CC2540 Application note as a reference

##### 3.1.1 Test setup

In this work, just like the many others mentioned in the introduction section of this work, TI's CC2540 application note AN092 [Kamath-10] was used as a reference to validate the setup used here to perform the power measurements and results comparison.

The Fig. 3-1 shows how a diagram with the test setup is used in the Application note. The idea is to get the voltage drop across a resistor with a known resistance and use ohm's law to calculate the current flowing into the BLE device. 10 ohm is a good value for this tests, it's small enough and shouldn't affect the board circuitry and large enough to provide decent precision according to [Kamath-10] this is validated by testing the board works after a rework.

Section 3.3.1 shows a more detailed description of the setup and 3.1.3 about the Equations for calculating the Current and Power consumption.

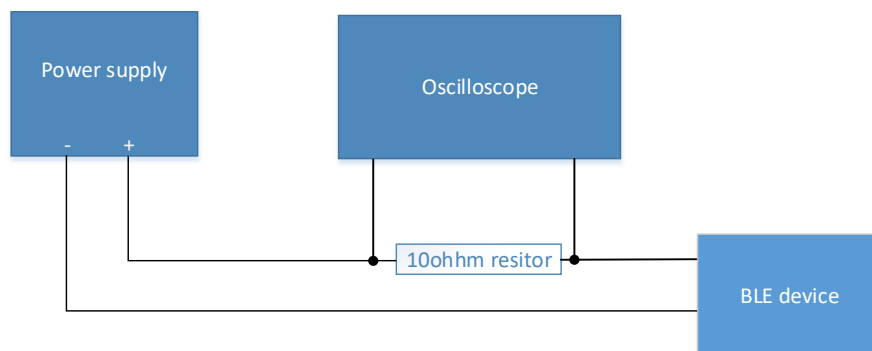


Fig. 3-1 High level diagram of current measurement Setup

### 3.1.2 Advertising and connection waveforms

Two types of waves were captured in this application note, those from Advertising and Connection events. By using the oscilloscope's trigger capabilities it is possible to catch either of these two waveforms. Picture Fig. 3-2 and Fig. 3-3 are the waveform examples provided by Texas Instruments [Kamath-10].

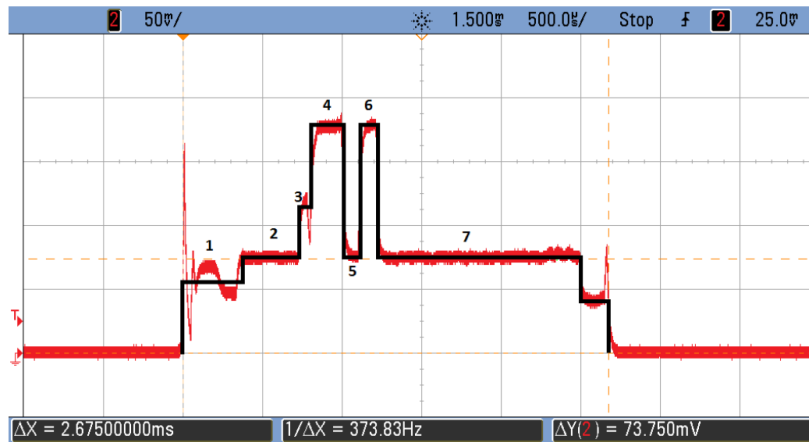


Fig. 3-2 Connection event TI CC2540. Image taken from [Kamath-10].

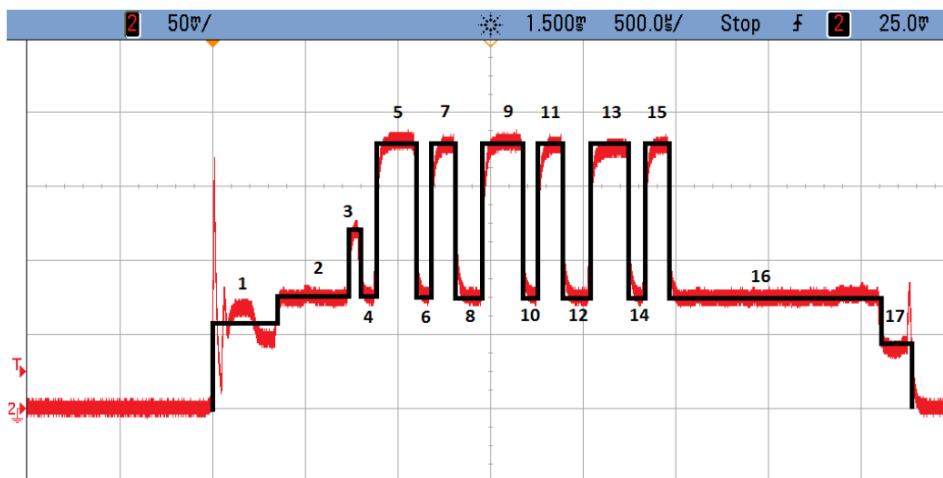


Fig. 3-3 Advertising event TI CC2540. Image taken from [Kamath-10].

#### 3.1.3 Power consumption computation

A spreadsheet is available in Texas Instruments' web page [TI-18] which has a couple of blank charts for capturing the user's own measurements and estimate battery life for their own use-case. For the Connection event, the chart is filled out with the 8 states/sections shown in Fig. 3-4, the connection used in this example (1000 ms) yields to 401 days of life for a 230 mAh battery. The current calculation results for the Advertising event is not provided.

The battery life calculation in TI's application note [Kamath-10], seems to follow the same equations ( 3-1 ) and ( 3-3 ) found in [Afonso-16]. ( 3-3 ) calculates the overall average current consumption to then get the total time a battery could last with ( 3-4 ). ( 3-2 ) is also used for this thesis work and is explained in 3.2

$$I_{on} = \frac{I_{on}T_{on} + I_{sleep}(T_{eci}-T_{on})}{T_{eci}} \quad (3-1)$$

$$I_{on} = \sum_{i=1}^{Ns} \frac{T_{si}I_{si}}{T_{on}} \quad (3-2)$$

$$T_{bat} = \frac{C_{bat}}{I_{avg}} \quad (3-3)$$

Texas Instruments estimates the average current of each state in the connection event waveform, this is the data that will be used for reference in this thesis work. Fig. 3-4 shows the average current results which are considered in [Kamath-10] as fairly accurate estimates: “using the divisions shown and guessing an average current value should provide fairly accurate estimates”.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
2														
3		<b>Status:</b>	<b>OK</b>										<b>Note: This measurement data to the example data in <i>Applic Bluetooth Low Energy Power number SWRA347</i></b>	
5		Battery capacity (mAh):	230											
6		Connection Interval (ms):	1000											
7		Sleep Current with timer running (mA)	0.001											
9			Case 1			Case 2			Case 3					
10			Time (us)	Current (mA)	Percent of events	Time (us)	Current (mA)	Percent of events	Time (us)	Current (mA)	Percent of events			
11					50			50			0			
12		State 1 (wake-up)	400	6		2400	400	6		2400	0	0	0	
13		State 2 (pre-processing)	315	7,4		2331	340	7,4		2516	0	0	0	
14		State 3 (pre-Rx)	80	11		880	80	11		880	0	0	0	
15		State 4 (Rx)	275	17,5		4812,5	190	17,5		3325	0	0	0	
16		State 5 (Rx-to-Tx)	105	7,4		777	105	7,4		777	0	0	0	
17		State 6 (Tx)	115	17,5		2012,5	115	17,5		2012,5	0	0	0	
18		State 7 (post-processing)	1325	7,4		9805	1280	7,4		9472	0	0	0	
19		State 8 (pre-sleep)	160	4,1		656	165	4,1		676,5	0	0	0	
20		State 9	0	0		0	0	0		0	0	0	0	
21		State 10	0	0		0	0	0		0	0	0	0	
22		State 11	0	0		0	0	0		0	0	0	0	
23		State 12	0	0		0	0	0		0	0	0	0	
24		State 13	0	0		0	0	0		0	0	0	0	
25		State 14	0	0		0	0	0		0	0	0	0	
26		State 15	0	0		0	0	0		0	0	0	0	
27		State 16	0	0		0	0	0		0	0	0	0	
28		State 17	0	0		0	0	0		0	0	0	0	
29		State 18	0	0		0	0	0		0	0	0	0	
30		State 19	0	0		0	0	0		0	0	0	0	
31		State 20	0	0		0	0	0		0	0	0	0	
32						23674				22059				
33		Total time of connection event	2775				2675				0			
34		Average Current draw during connection event (mA):		8,5311712				8,246355			0			
35		Average current draw accounting for sleep (mA):				0,024671				0,023056			0,001	
37		<b>Average current draw during connection (mA):</b>	<b>0,023863775</b>											
38		<b>Expected battery life (hours):</b>	<b>9638,039246</b>											
39		<b>Expected battery life (days):</b>	<b>401,5849686</b>											

Fig. 3-4 battery life estimation spreadsheet TI CC2540. Image taken from [Kamath-10].



3.2. Power consumption and Current Equations used in this work

Let  $E$  (see ( 3-4 ) ) be a vector with all the samples of a BLE event, the vector will be split into sections as shown in Fig. 3-5. We will have a different number of section depending on the type of event (either connection or advertising).

$$E = a_0, a_1 \dots a_n \tag{3-4}$$

Each section  $a_n$  has  $N$  number of samples. Separated by a sampling period  $T_s$  so the total time an event takes is calculated with ( 3-5 ) summing the length of each section, calculated with ( 3-6 ).

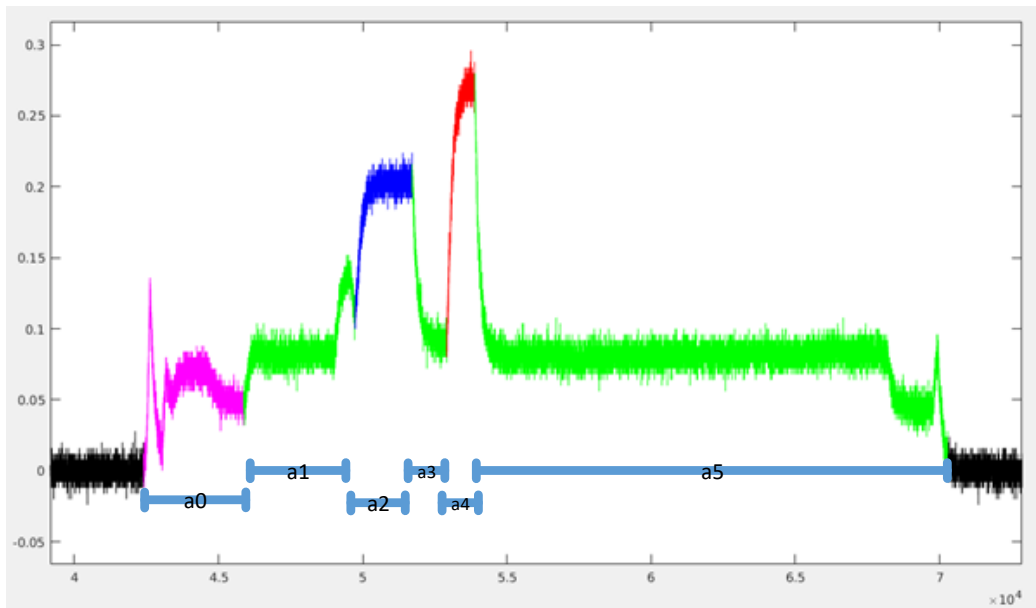


Fig. 3-5 example of sections found in a connection event waveform

$$T_E = T_{a0} + T_{a1} \dots + T_{an} \tag{3-5}$$

Where

$$T_{an} = N * T_s \tag{3-6}$$

The current and power of each section are calculated with equations ( 3-7 ) and ( 3-8 ) respectively. The current calculated with ( 3-7 ) for each section of  $E$  will be used for comparing to the results presented in TI's application note [*Kamath-10*] (see the data comparison in Table 3-1 section 3.3.3), and in this way validate the experimental data obtained in this thesis work.

$$I_{a_n} = \sum_{i=0}^{N-1} \frac{a_n(i)}{r} \quad (3-7)$$

Power ( 3-8 ) will be used to get the total power consumption of advertising and connection events measurements from each one of the proposed platforms. The equation ( 3-9 ) contains information about the consumption of each section for easier comparison between the development boards

$$P_{a_n} = \sum_{i=0}^{N-1} \frac{a_n(i)^2}{r} \quad (3-8)$$

$$P_E = P_{a_1} + P_{a_2} \dots + P_{a_n} \quad (3-9)$$

#### 3.3. TI CC2540 Mini development kit

##### 3.3.1 Schematics and measurement point

In order to perform the measurements on the CC2540 from the secondary power input, it is necessary to remove a capacitor and install a resistor. This change has the purpose of inhibiting the power input from the coin cell battery and use the voltage pin from the debugger as the actual power input. From the schematics (see Fig. 3-6), removing C7 capacitor and installing a resistor in P1 changes the input power on the DCDC regulator (VDD input) to the input coming from the debugger port. This allows the current measuring on the 10-ohm resistor.

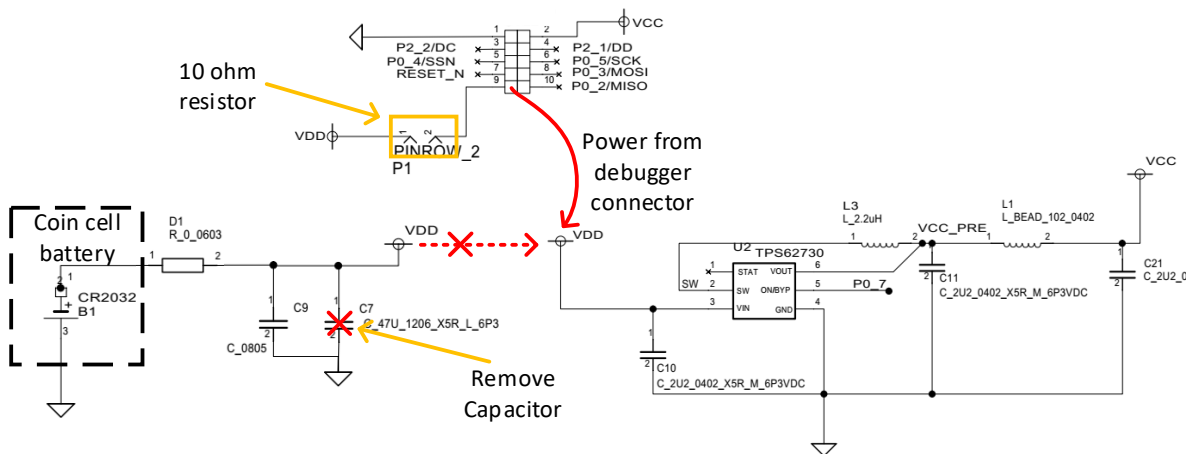


Fig. 3-6 Power supply connection for CC2540. Schematics modified from [TI-13].

##### 3.3.2 Modifications for measurements

A low resistance resistor should be placed on the P1 empty space in order to hook the oscilloscope probes and measure the voltage drop and be able to calculate the current. The power supply voltage is connected to the resistor and GND pin. Now the setup is ready to connect to the master and get the power consumption on an Advertising or Connection event. The rework of the board in Fig. 3-7 shows how a 10-ohm SMT resistor was glued on the board and soldered to the Vcc pin on one side and to a cable on the other in order to provide a test point for the oscilloscope, as well as the removed capacitor to block the power from the coin cell battery.

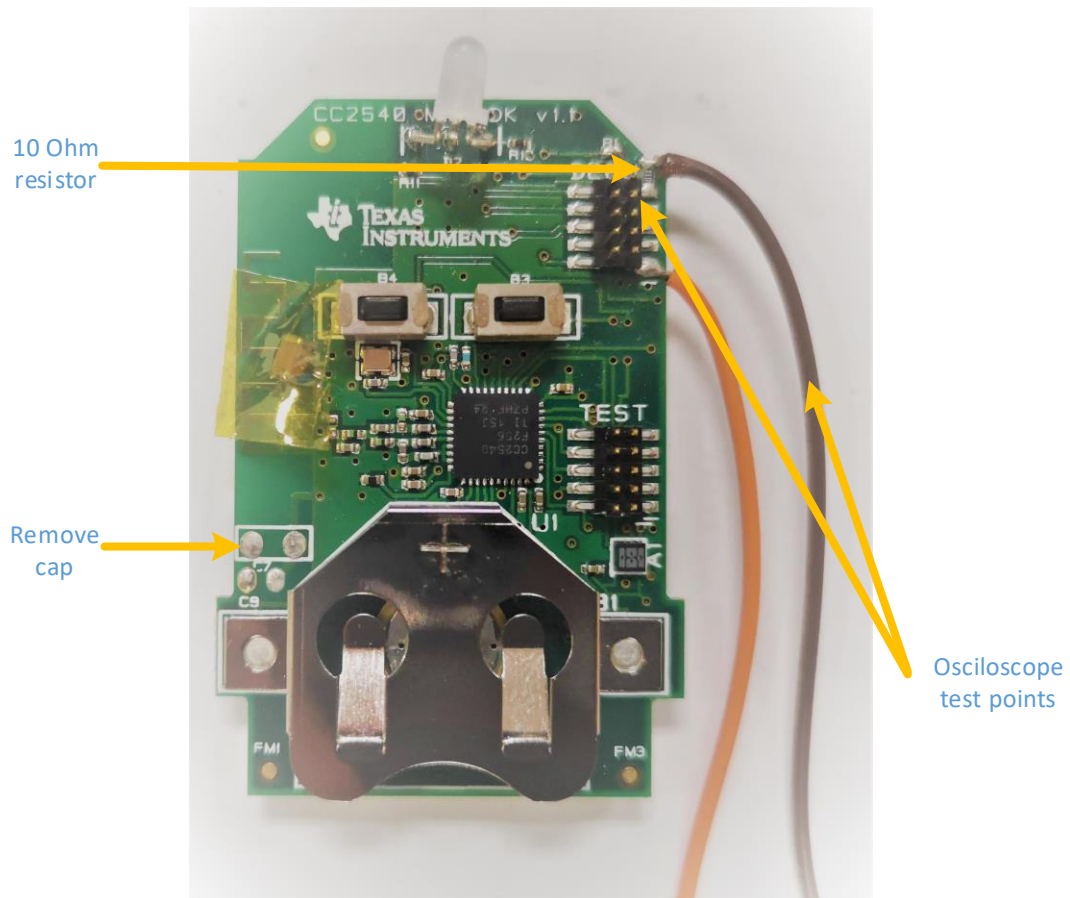


Fig. 3-7 Modification on CC2540 according to TI's application notes

### 3. Power Measurements

Fig. 3-7 shows the measurement setup when capturing a connection event on the CC2540 which is here powered by batteries for easier oscilloscope probe attachment.

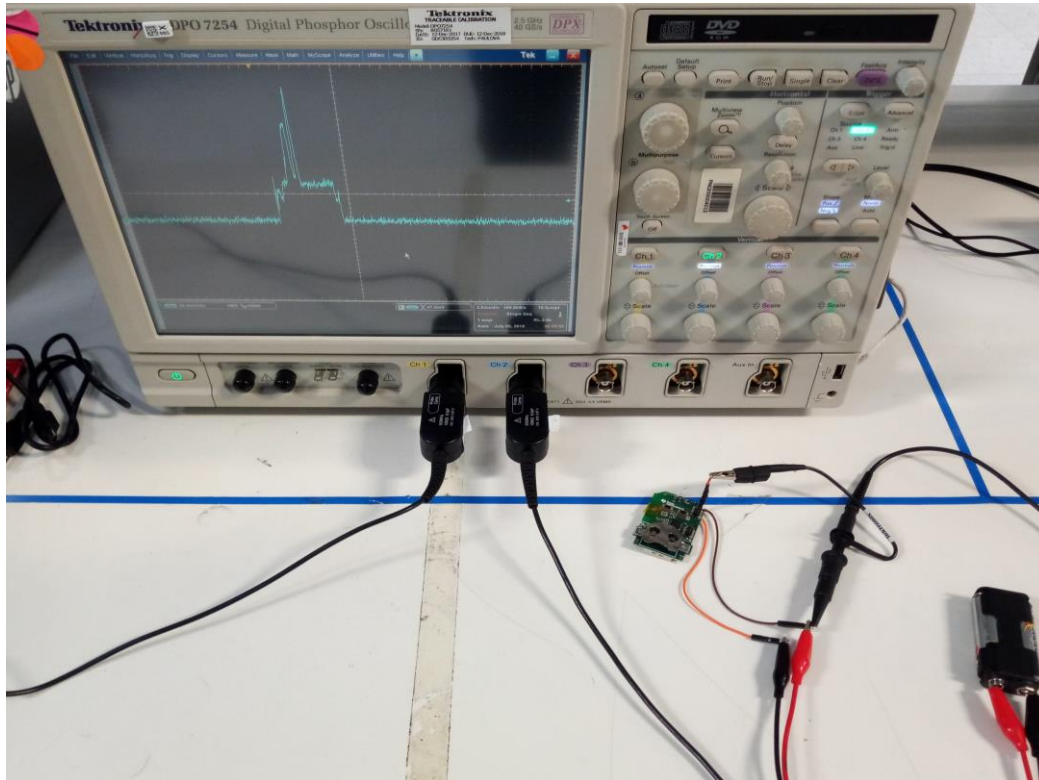


Fig. 3-8 Measurement Setup on CC2540 board

### 3.3.3 Measurement results

Table 3-2 shows the battery life estimation spreadsheet for a connection event with a connection interval of 1s and a battery with 230 mAh capacity. The total time of the connection event varies by 15us and the average current by 0.7 mA. Although the battery life in days yields 31 days less. This comparison gives confidence in the method used for the measurements in this thesis.

TI application note results

Status:	OK		
Battery capacity (mAh):	230		
Connection Interval (ms):	1000		
Sleep Current with timer running (mA)	0.001		
		Case 1	
		Time (us)	Current (mA)
			Percent of events
			100
State 1 (wake-up)	400	6	2400
State 2 (pre-processing)	315	7.4	2331
State 3 (pre-Rx)	80	11	880
State 4 (Rx)	275	17.5	4812.5
State 5 (Rx-to-Tx)	105	7.4	777
State 6 (Tx)	115	17.5	2012.5
State 7 (post-processing)	1325	7.4	9805
State 8 (pre-sleep)	160	4.1	656
State 9	0	0	0
State 10	0	0	0
State 11	0	0	0
State 12	0	0	0
State 13	0	0	0
State 14	0	0	0
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
			23674
Total time of connection event	2775		
Average Current draw during connection event (mA):		8.5311712	
Average current draw accounting for sleep (mA):			0.024671
<b>Average current draw during connection (mA):</b>	<b>0.024671225</b>		
<b>Expected battery life (hours):</b>	<b>9322.601533</b>		
<b>Expected battery life (days):</b>	<b>388.4417305</b>		

Thesis measurement results

Status:	OK		
Battery capacity (mAh):	230		
Connection Interval (ms):	1000		
Sleep Current with timer running (mA)	0.001		
		Case 1	
		Time (us)	Current (mA)
			Percent of events
			100
State 1 (wake-up)	347.1	5.6902	1975.068
State 2 (pre-processing)	386.8	8.8083	3407.05
State 3 (Rx)	198.1	19.3726	3837.712
State 4 (Rx-to-Tx)	120.4	10.7997	1300.284
State 5 (Tx)	97.2	23.8115	2314.478
State 6 (post-processing+pre-sleep)	1641	7.8961	12957.5
State 7	0	0	0
State 8	0	0	0
State 9	0	0	0
State 10	0	0	0
State 11	0	0	0
State 12	0	0	0
State 13	0	0	0
State 14	0	0	0
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
			25792.09
Total time of connection event	2790.6		
Average Current draw during connection event (mA):		9.24249	
Average current draw accounting for sleep (mA):			0.026789
<b>Average current draw during connection (mA):</b>	<b>0.026789302</b>		
<b>Expected battery life (hours):</b>	<b>8585.516679</b>		
<b>Expected battery life (days):</b>	<b>357.7298616</b>		

Table 3-2 Battery life comparison of thesis results versus TI's published data for connection events on the CC2540

### 3. Power Measurements

We used the same spreadsheet to calculate the battery life of a device which remains in advertising state having in mind that the separation between advertising events is called advertising delay (*advDelay*) as explained in 1.1.3. Table 3-3 shows Battery life estimation is 194 days with a 1s *advDelay*.

<b>Status:</b>	<b>OK</b>		
Battery capacity (mAh):	230		
Connection Interval (ms):	1000		
Sleep Current with timer running (mA)	0		
	Case 1		
	Time (us)	Current (mA)	Percent of events
			100
State 1 (wake-up)	348.4	5.7371	1998.806
State 2 (pre-processing)	852	8.5412	7277.102
State 3 (Tx)	202	26.1548	5283.27
State 4 (Tx-to-Rx)	80	13.1676	1053.408
State 5 (Rx)	135.2	18.5329	2505.648
State 6 (Rx-to-Tx)	142.2	10.2805	1461.887
State 7 (Tx)	202.2	25.8336	5223.554
State 8 (Tx-to-Rx)	80.4	13.7504	1105.532
State 9 (Rx)	134.8	18.4163	2482.517
State 10 (Rx-to-Tx)	145.6	10.474	1525.014
State 11 (Tx)	203.2	25.6091	5203.769
State 12 (Tx-to-Rx)	80.2	13.1773	1056.819
State 13 (Rx)	134.8	18.4533	2487.505
State 14 (post-processing)	1362	7.77	10582.74
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
Total time of connection event	4103		49247.57
Average Current draw during connection event (mA):		12.00282	
Average current draw accounting for sleep (mA):			0.049248
<b>Average current draw during connection (mA):</b>	<b>0.049247572</b>		
<b>Expected battery life (hours):</b>	<b>4670.281008</b>		
<b>Expected battery life (days):</b>	<b>194.595042</b>		

Table 3-3 Battery life estimation of an advertising event on the CC2540

### 3.3.4 Advertising

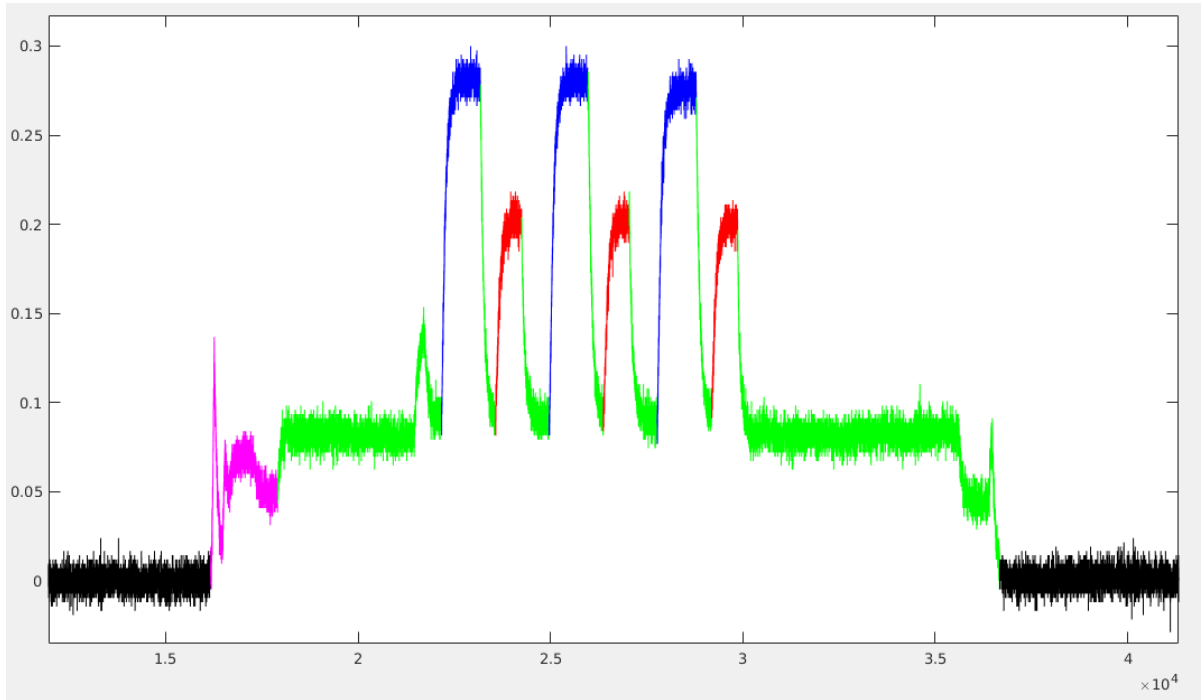


Fig. 3-9 Advertising event on CC2540



3.3.5 Connection

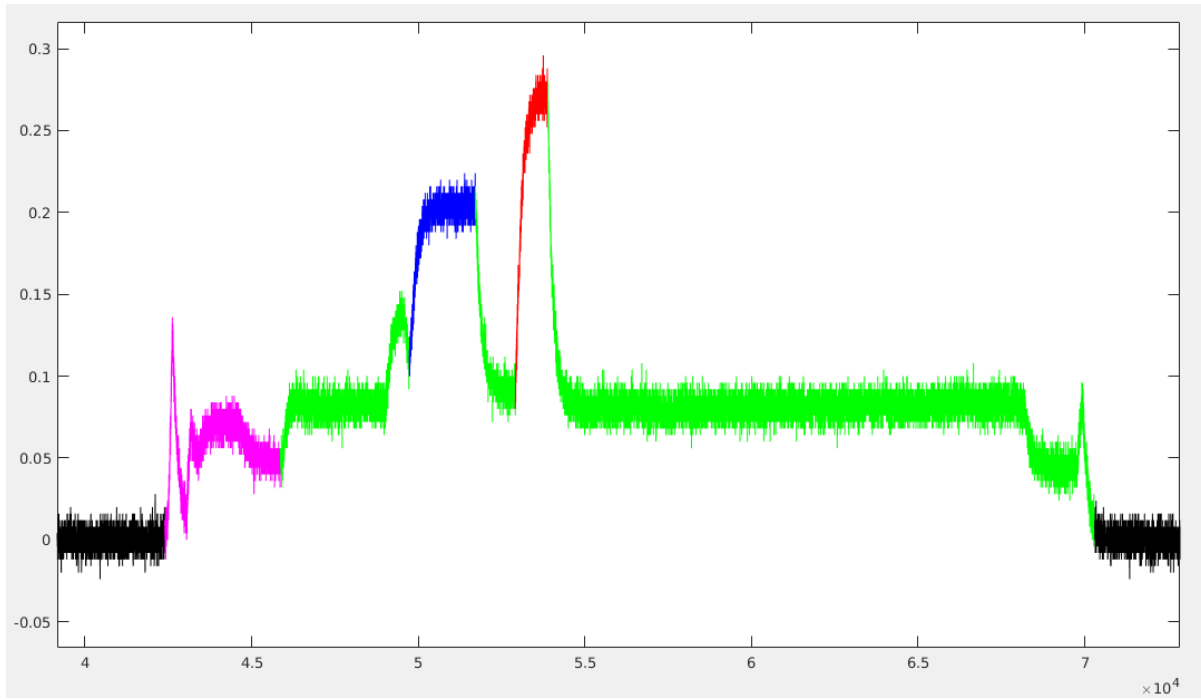


Fig. 3-10 Connection event on CC2540

### 3.4. Genuino/Arduino 101

#### 3.4.1 Schematics and measurement point

By reviewing the schematics of the Genuino 101 [Kashi-16], it was found that the USB voltage got to this resistor R58 (see picture below) and from here connected to VDD\_BLE\_SEN which powers the BLE submodule and the gyroscope.

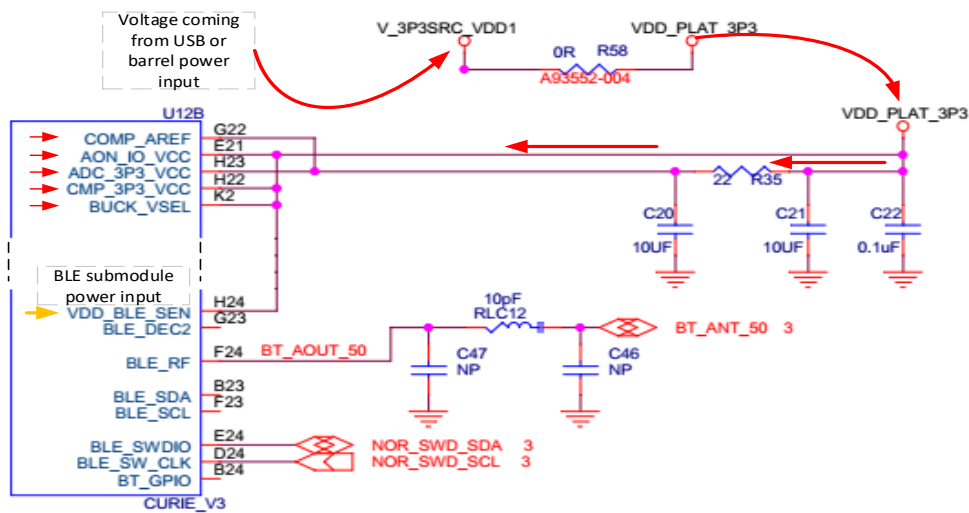


Fig. 3-11 Power supply connection for BLE submodule in Genuino 101. Schematic taken from [Kashi-16].

#### 3.4.2 Modifications for measurements

The resistor mentioned in the previous section was identified using one of the CAD documents and replaced by a 10-ohm resistor. The factory value of the resistor was 0 ohm.

### 3. Power Measurements

The oscilloscope probes were hooked up on each end in order to measure the voltage drop across the resistor shown in Fig. 3-12.

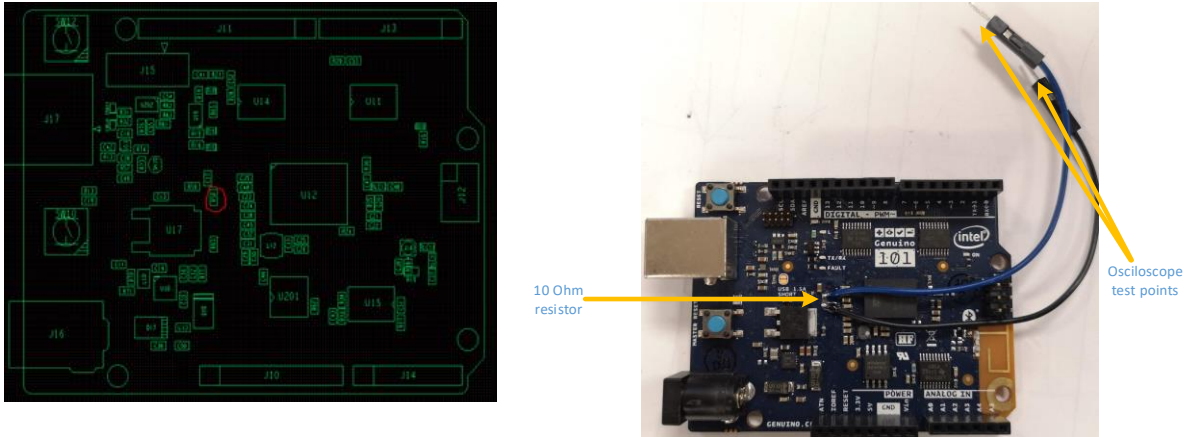


Fig. 3-12 Resistor replaced for power measurements on Genunino 101

The measurement setup in Fig. 3-13 shows an advertising event captured on the oscilloscope and the probes connected on the board.

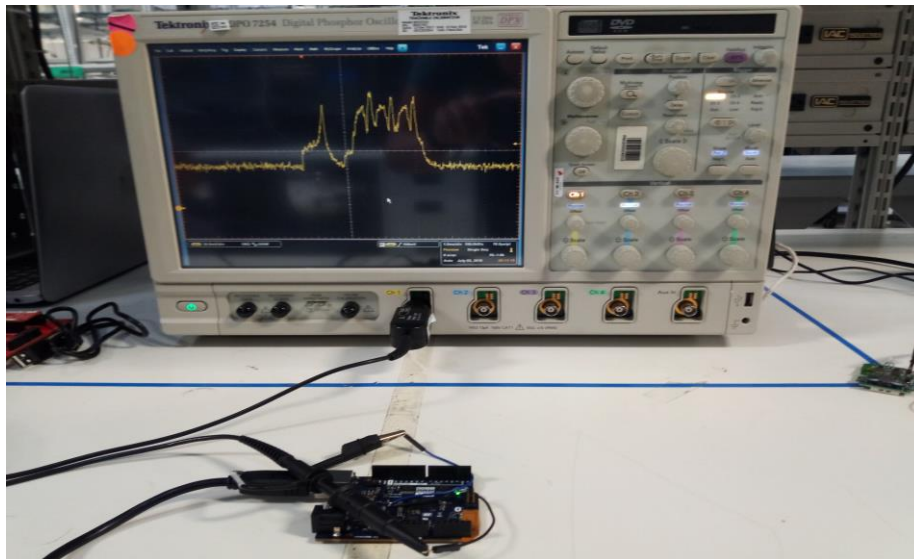


Fig. 3-13 Measurements setup on Genunino 101

### 3.4.3 Measurement results

For Arduino 101 an Advertising and Connection events waveforms were captured and analyzed. Additionally, two different BLE SW stacks were tested: Zephyr and the Arduino's BLE default SW stack, 4.2 and 4.0 respectively.

#### ***Connection Event with Arduino's BLE stack.***

Connection Event waveform and results are shown in the picture and table below

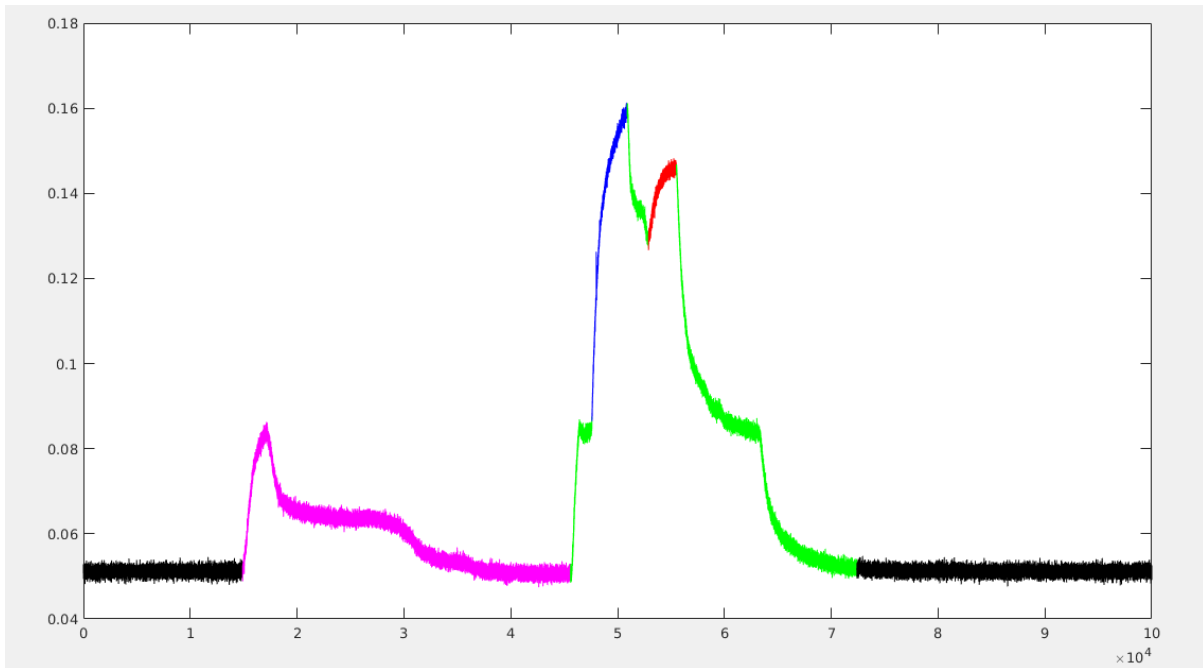


Fig. 3-14 Connection event with Arduino's BLE stack

### 3. Power Measurements

#### *Connection Event with Zephyr BLE stack*

Advertising Event waveform and results are shown in the picture and table below

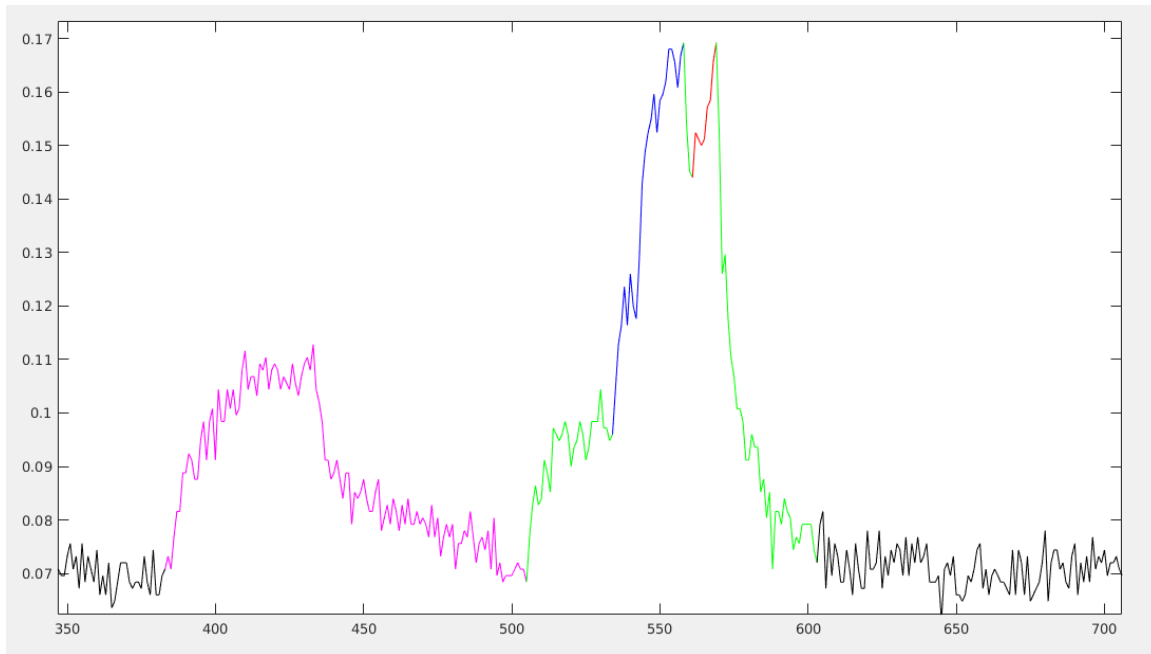


Fig. 3-15 Connection event with Zephyr's BLE stack

### Arduino BLE stack

Status:	OK		
Battery capacity (mAh):	230		
Connection Interval (ms):	100		
Sleep Current with timer running (mA)	0.001		
	Case 1		
	Time (us)	Current (mA)	Percent of events
			100
State 1 (wake-up)	1535	5.9303	9103.011
State 2 (pre-processing)	100	7.7163	771.63
State 3 (Tx)	165	14.09	2324.85
State 4 (Tx-to-Rx)	100	13.8395	1383.95
State 5 (Rx)	130	14.1474	1839.162
State 6 (post-processing)	845	7.4124	6263.478
State 7	0	0	0
State 8	0	0	0
State 9	0	0	0
State 10	0	0	0
State 11	0	0	0
State 12	0	0	0
State 13	0	0	0
State 14	0	0	0
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
			21686.08
Total time of connection event	2875		
Average Current draw during connection event (mA):		7.5429845	
Average current draw accounting for sleep (mA):			0.217832
<b>Average current draw during connection (mA):</b>	<b>0.217832055</b>		
<b>Expected battery life (hours):</b>	<b>1055.859295</b>		
<b>Expected battery life (days):</b>	<b>43.99413729</b>		

### Zephyr BLE stack

Status:	OK		
Battery capacity (mAh):	230		
Connection Interval (ms):	100		
Sleep Current with timer running (mA)	0.001		
	Case 1		
	Time (us)	Current (mA)	Percent of events
			100
State 1 (wake-up)	1,230	8.8068	10832.36
State 2 (pre-processing)	300	9.224	2767.2
State 3 (Rx)	250	14.2032	3550.8
State 4 (Rx-to-Tx)	40	15.3	612
State 5 (Tx)	90	15.5467	1399.203
State 6 (post-processing)	350	9.3291	3265.185
State 7	0	0	0
State 8	0	0	0
State 9	0	0	0
State 10	0	0	0
State 11	0	0	0
State 12	0	0	0
State 13	0	0	0
State 14	0	0	0
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
			22426.75
Total time of connection event	2260		
Average Current draw during connection event (mA):		9.9233416	
Average current draw accounting for sleep (mA):			0.225245
<b>Average current draw during connection (mA):</b>	<b>0.22524492</b>		
<b>Expected battery life (hours):</b>	<b>1021.110709</b>		
<b>Expected battery life (days):</b>	<b>42.54627955</b>		

Table 3-4 Battery life comparison of a connection event Zephyr vs Arduino BLE stack

*Advertising event with Arduino's BLE stack*

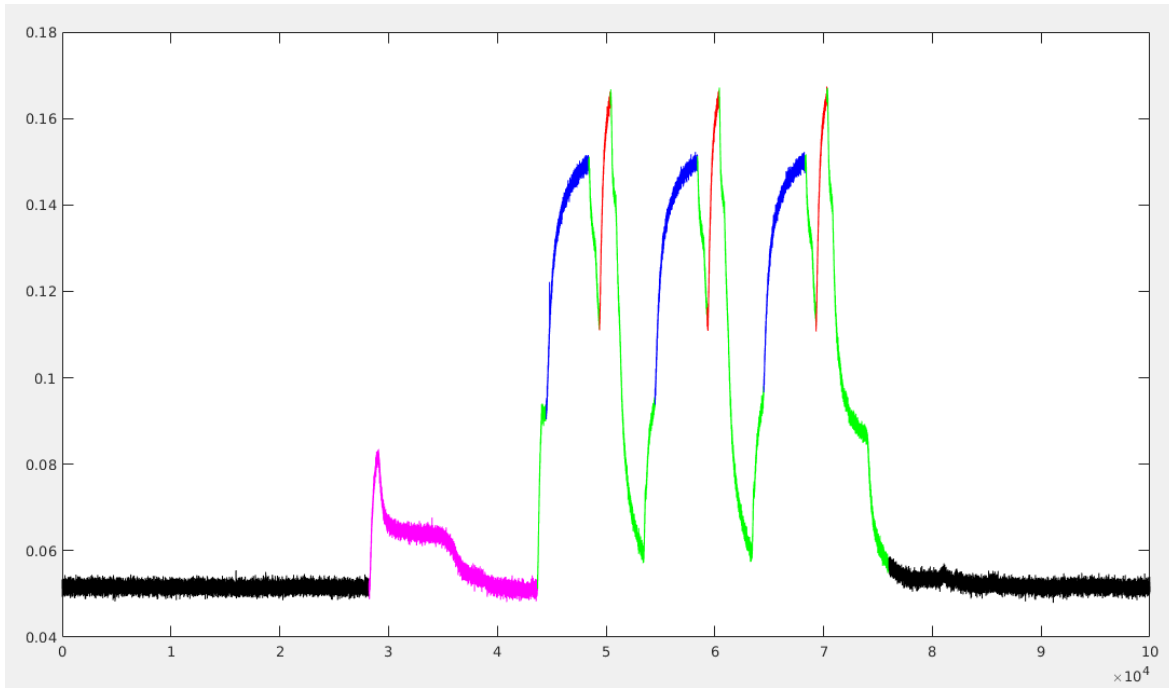


Fig. 3-16 Advertising event with Arduino's BLE stack

*Advertising event with Zephyr's BLE stack*

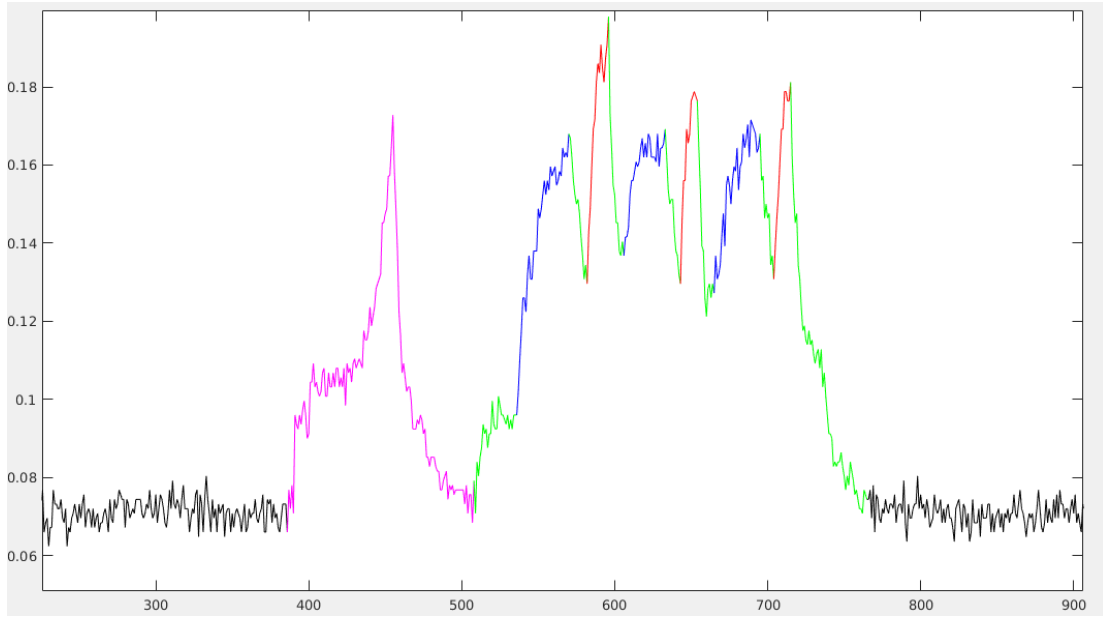


Fig. 3-17 Advertising event with Zephyr's BLE stack



### 3. Power Measurements

#### Arduino BLE stack

Status:	OK		
Battery capacity (mAh):	230		
Connection Interval (ms):	100		
Sleep Current with timer running (mA)	0.001		
	Case 1		
	Time (us)	Current (mA)	Percent of events
			100
State 1 (wake-up)	1550	5.9157	9169.335
State 2 (pre-processing)	80	8.3663	669.304
State 3 (Tx)	390	13.7288	5354.232
State 4 (Tx-to-Rx)	100	13.0859	1308.59
State 5 (Rx)	100	14.8857	1488.57
State 6 (Rx-to-Tx)	410	8.949	3669.09
State 7 (Tx)	390	13.8874	5416.086
State 8 (Tx-to-Rx)	90	13.1772	1185.948
State 9 (Rx)	110	14.7581	1623.391
State 10 (Rx-to-Tx)	410	8.8895	3644.695
State 11 (Tx)	380	13.9525	5301.95
State 12 (Tx-to-Rx)	100	13.1998	1319.98
State 13 (Rx)	100	14.8747	1487.47
State 14 (post-processing)	570	9.0151	5138.607
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
Total time of connection event	4780		46777.25
Average Current draw during connection event (mA):		9.7860351	
Average current draw accounting for sleep (mA):			0.468725
<b>Average current draw during connection (mA):</b>	<b>0.46872468</b>		
<b>Expected battery life (hours):</b>	<b>490.6931719</b>		
<b>Expected battery life (days):</b>	<b>20.44554883</b>		

#### Zephyr BLE stack

Status:	OK		
Battery capacity (mAh):	230		
Connection Interval (ms):	100		
Sleep Current with timer running (mA)	0.001		
	Case 1		
	Time (us)	Current (mA)	Percent of events
			100
State 1 (wake-up)	1,230	10.1512	12485.98
State 2 (pre-processing)	290	9.1572	2655.588
State 3 (Tx)	350	14.4	5040
State 4 (Tx-to-Rx)	130	14.8246	1927.198
State 5 (Rx)	150	17.368	2605.2
State 6 (Rx-to-Tx)	110	15.3164	1684.804
State 7 (Tx)	280	15.9343	4461.604
State 8 (Tx-to-Rx)	110	14.6945	1616.395
State 9 (Rx)	120	16.47	1976.4
State 10 (Rx-to-Tx)	120	13.83	1659.6
State 11 (Tx)	310	15.5652	4825.212
State 12 (Tx-to-Rx)	100	14.736	1473.6
State 13 (Rx)	120	16.33	1959.6
State 14 (post-processing)	510	10.0988	5150.388
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
Total time of connection event	3930		49521.57
Average Current draw during connection event (mA):		12.600907	
Average current draw accounting for sleep (mA):			0.496176
<b>Average current draw during connection (mA):</b>	<b>0.49617635</b>		
<b>Expected battery life (hours):</b>	<b>463.5448667</b>		
<b>Expected battery life (days):</b>	<b>19.31436944</b>		

Table 3-5 Battery life comparison of an advertising event Zephyr vs Arduino BLE stack

### 3.5. FRDM-KW41Z: FREEDOM DEVELOPMENT KIT

#### 3.5.1 Schematics and measurement point

The Freedom board can be powered from a USB port connection or a coin cell battery. If the coin cell battery powers up the system, a jumper J27 shown in Fig. 3-18 can be used to connect a 10-ohm resistor and perform the measurements.

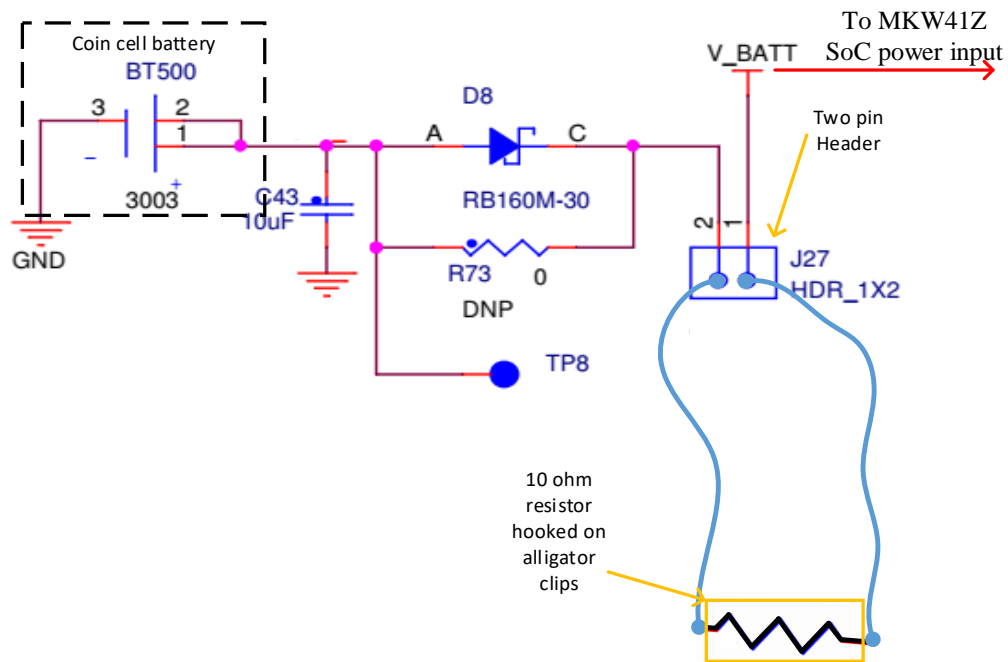


Fig. 3-18 Power supply connection for MKW41Z. Schematic modified from [NXP-18].

### 3. Power Measurements

#### 3.5.2 Modifications for measurements

The Freedom board 10-ohm resistor was connected as in Fig. 3-19. Jumper j27 (see Fig. 3-19) is removed and the resistor attached with alligator clips.

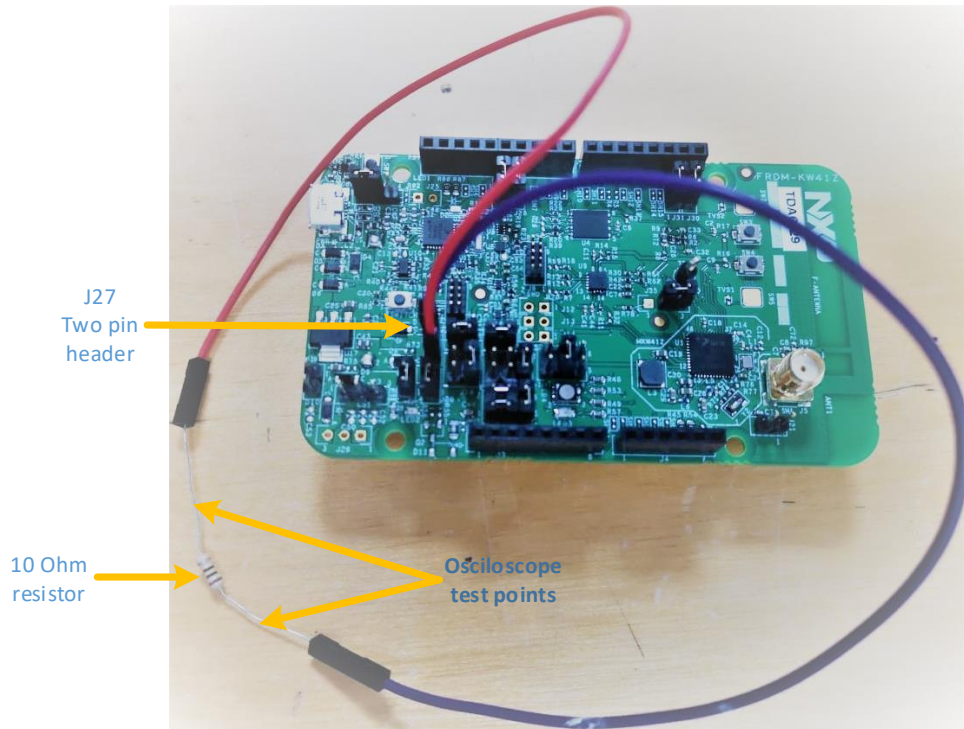


Fig. 3-19 Resistor placed for power measurements on Freedom development board

The measurement setup in Fig. 3-20 shows an advertising event captured on the oscilloscope and the probes connected to the resistor that replace the jumper on J27.

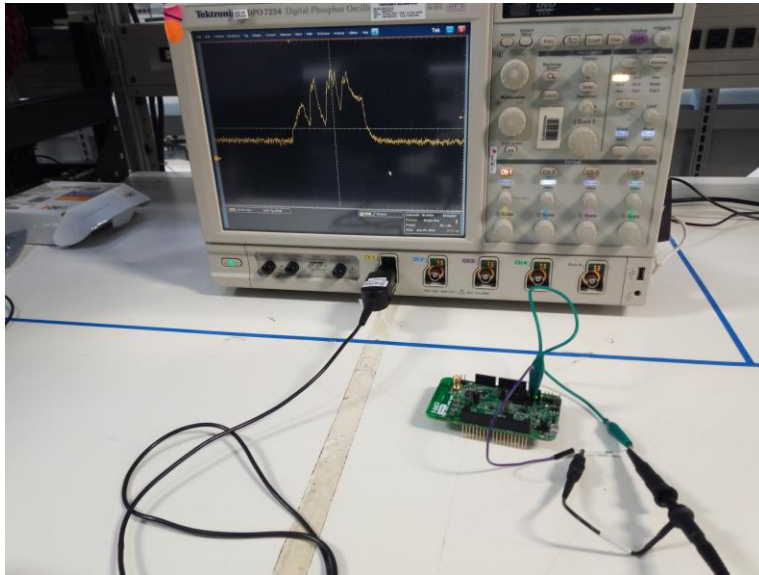


Fig. 3-20 Measurements setup on Freedom development board

3.5.3 Measurement results

3.5.4 Advertising

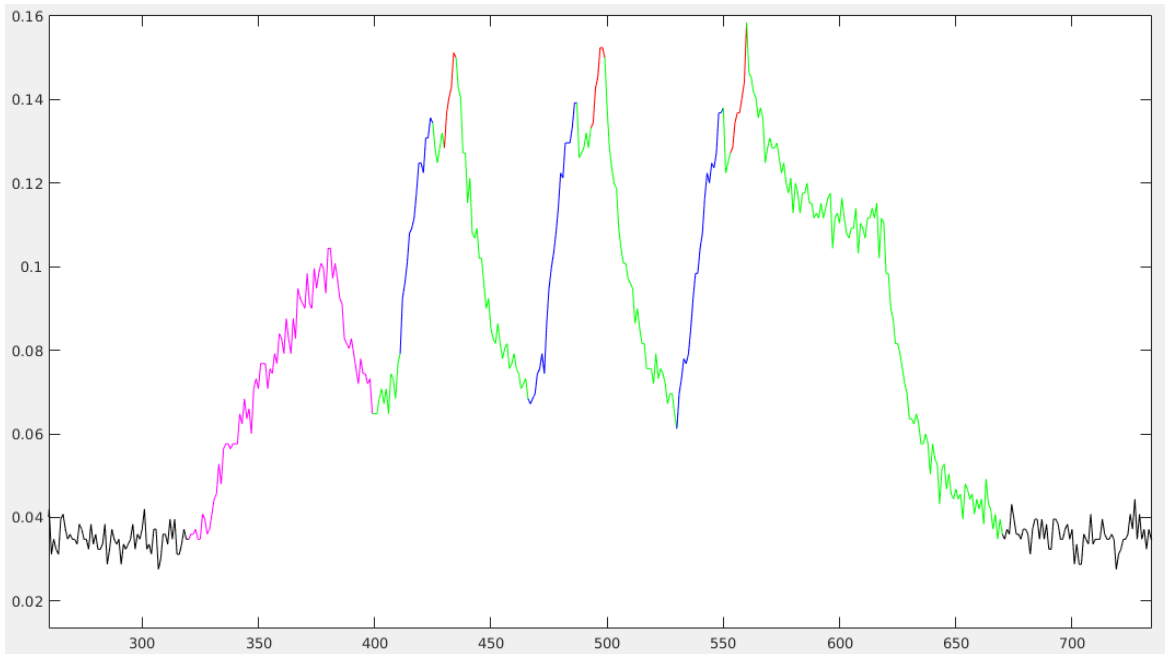


Fig. 3-21 Advertising event on Freedom board.

<b>Status:</b>	<b>OK</b>		
Battery capacity (mAh):	230		
Connection Interval (ms):	100		
Sleep Current with timer running (mA)	0.001		
	Case 1		
	Time (us)	Current (mA)	Percent of events
			100
State 1 (wake-up)	1600	7.233	11572.8
State 2 (pre-processing)	260	6.9877	1816.802
State 3 (Tx)	300	11	3436.8
State 4 (Tx-to-Rx)	120	13	1552.8
State 5 (Rx)	120	14	1699.2
State 6 (Rx-to-Tx)	640	9.5438	6108.032
State 7 (Tx)	440	10	4454.384
State 8 (Tx-to-Rx)	140	13	1828.806
State 9 (Rx)	140	14	2020.802
State 10 (Rx-to-Tx)	640	9.1013	5824.832
State 11 (Tx)	420	10	4341.582
State 12 (Tx-to-Rx)	80	13	1024.8
State 13 (Rx)	160	14	2212.8
State 14 (post-processing)	2220	8.9751	19924.72
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
			67819.16
Total time of connection event	7280		
Average Current draw during connection event (mA):		9.315818956	
Average current draw accounting for sleep (mA):			0.679119
<b>Average current draw during connection (mA):</b>	<b>0.67911882</b>		
<b>Expected battery life (hours):</b>	<b>338.674166</b>		
<b>Expected battery life (days):</b>	<b>14.11142358</b>		

Table 3-6 Battery life of an advertising event of Freedom development board

3.5.5 Connection

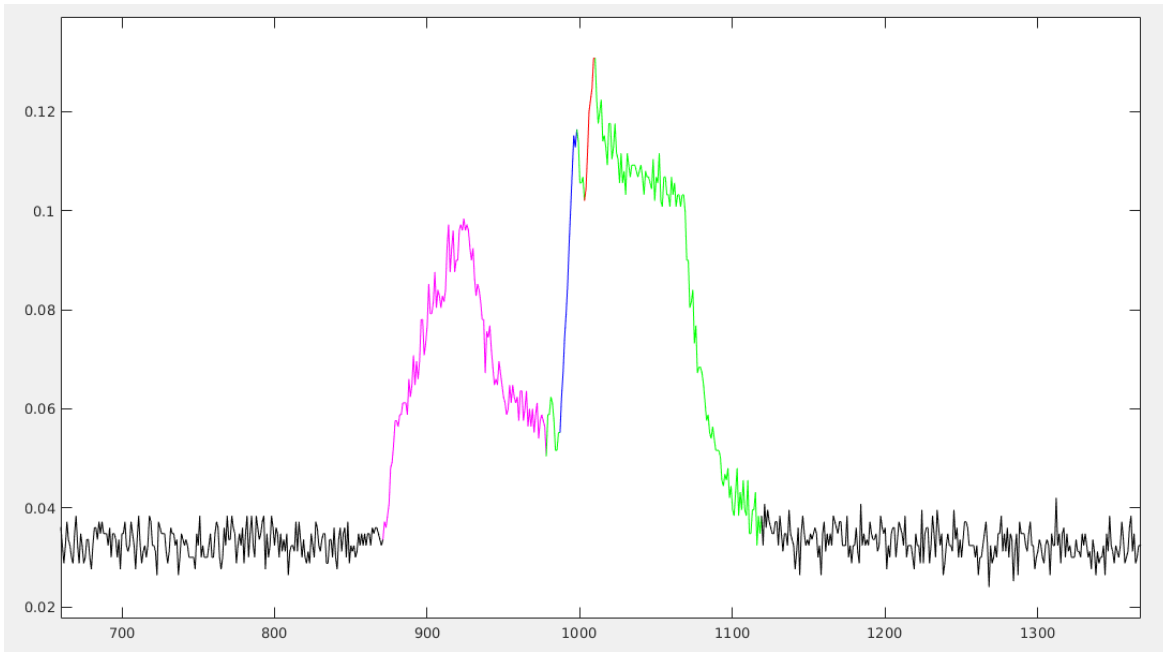


Fig. 3-22 Connection event on Freedom board

<b>Status:</b>	<b>OK</b>			
Battery capacity (mAh):	230			
Connection Interval (ms):	100			
Sleep Current with timer running (mA)	0.001			
	Case 1			
	Time (us)	Current (mA)	Percent of events	
			100	
State 1 (wake-up)	2,160	7.0256		15175.3
State 2 (pre-processing)	200	5.628		1125.6
State 3 (Rx)	240	8.92		2140.8
State 4 (Rx-to-Tx)	120	10.84		1300.8
State 5 (Tx)	160	11.835		1893.6
State 6 (post-processing)	2,200	8.3105		18283.1
State 7	0	0		0
State 8	0	0		0
State 9	0	0		0
State 10	0	0		0
State 11	0	0		0
State 12	0	0		0
State 13	0	0		0
State 14	0	0		0
State 15	0	0		0
State 16	0	0		0
State 17	0	0		0
State 18	0	0		0
State 19	0	0		0
State 20	0	0		0
Total time of connection event	5080			39919.2
Average Current draw during connection event (mA):		7.8581094		
Average current draw accounting for sleep (mA):				0.400141
<b>Average current draw during connection (mA):</b>	<b>0.40014116</b>			
<b>Expected battery life (hours):</b>	<b>574.7971541</b>			
<b>Expected battery life (days):</b>	<b>23.94988142</b>			

Table 3-7 Battery life of an advertising event of Freedom development board



### 3. Power Measurements

#### 3.6. CY8CKIT-042-BLE-A PIONEER KIT

##### 3.6.1 Schematics and measurement point

From the Pioneer board schematics, we know that it comes with built-in test points designed for taking power measurements. We have a jumper connected to both ends of the R44 resistor (see Fig. 3-23 below). The jumper can be removed and then use the available pins to measure the voltage drop across the resistor. Fig. 3-23 shows how the voltage from the battery or the USB port goes to the BLE submodule with the PSoC 4 BLE chip as the main component.

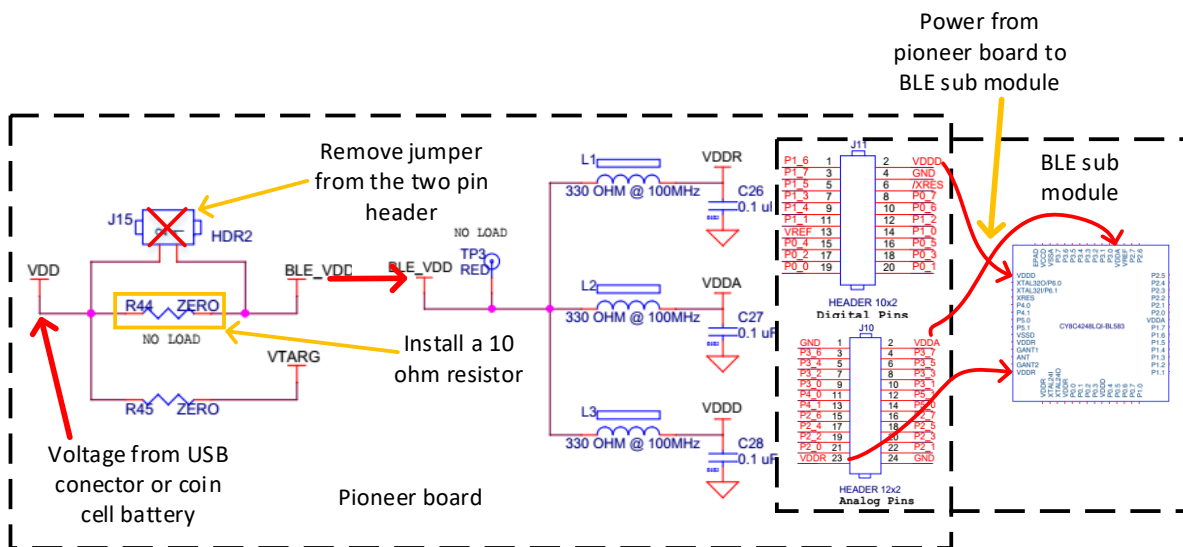


Fig. 3-23 Power supply connection for Pioneer board. Schematics modified from [Cypress-16] and [Cypress-16b].

### 3.6.2 Modifications for measurements

Fig. 3-24 Resistor replaced for power measurements on the Cypress Pioneer kit. Fig. 3-24 shows what part of the Pioneer board the R44 resistor should be located. The only rework necessary in this board is to add the R44 resistor. The Pioneer board could also use the same approach as the Freedom board explained in 3.5.2 by using J15 jumper located next to R44.

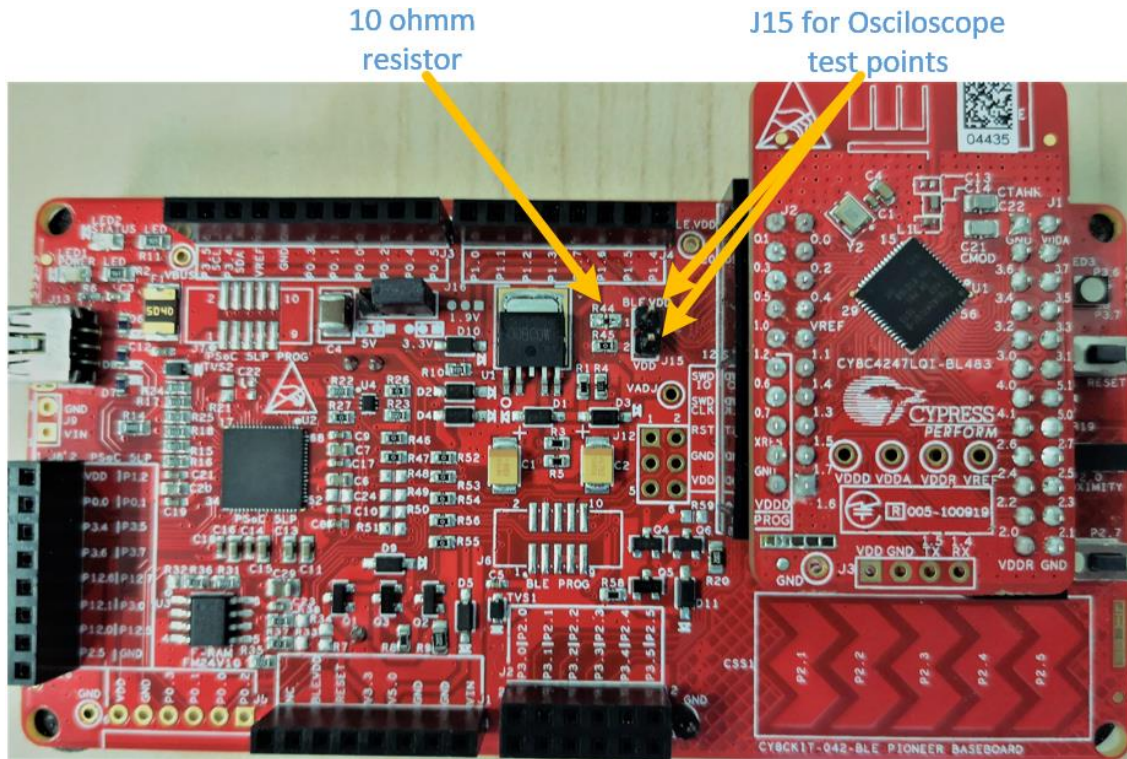


Fig. 3-24 Resistor replaced for power measurements on the Cypress Pioneer kit.

### 3. Power Measurements

The measurement setup in Fig. 3-25 shows an advertising event captured on the oscilloscope and the probes connected on J15.

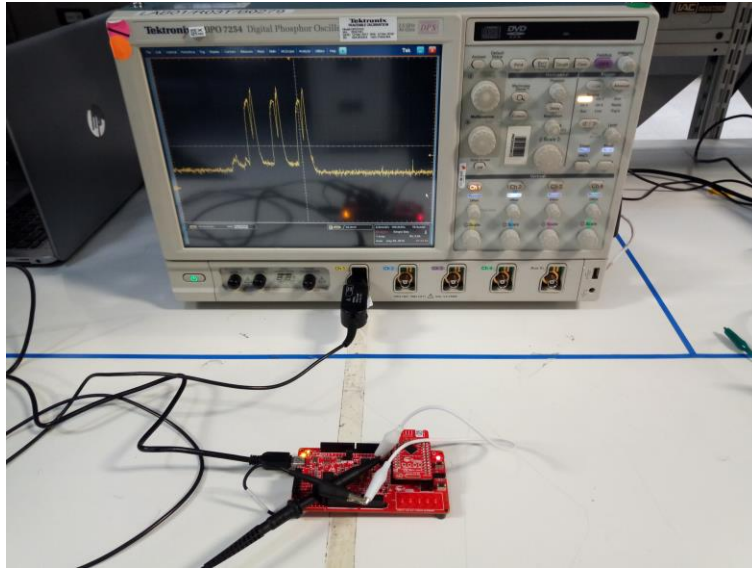


Fig. 3-25 Measurements setup on the Cypress pioneer kit

### 3.6.3 Measurement results

### 3.6.4 Advertising

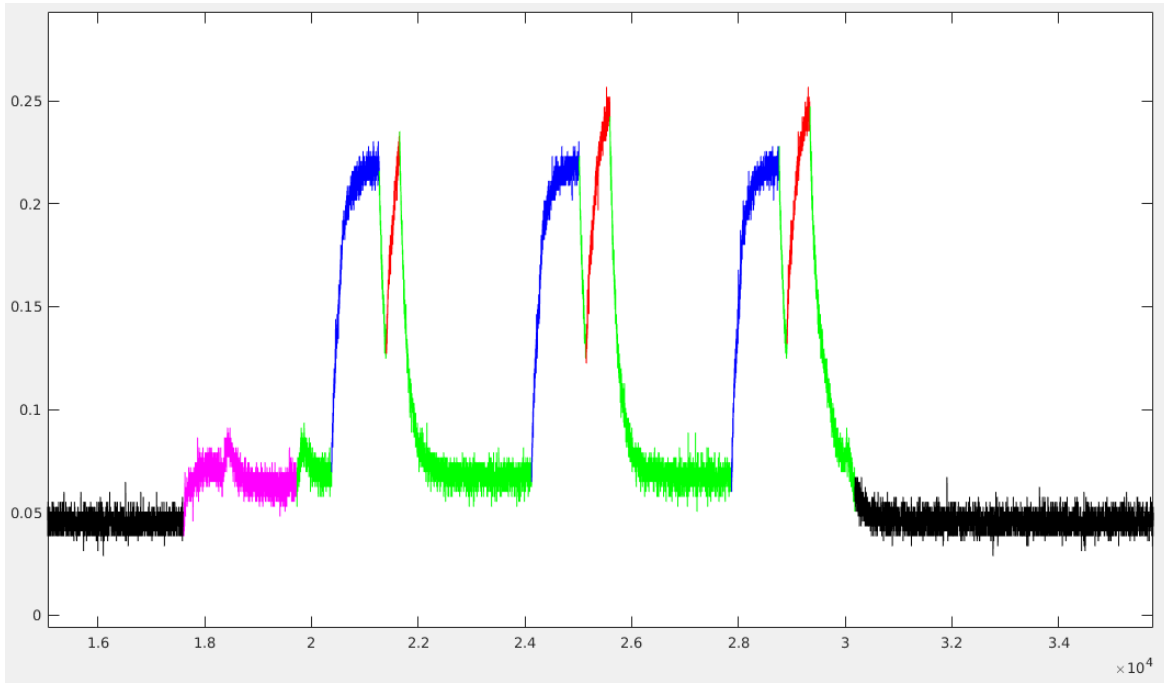


Fig. 3-26 Advertising event on Cypress pioneer kit

### 3. Power Measurements

Status:	OK			
Battery capacity (mAh):	230			
Connection Interval (ms):	100			
Sleep Current with timer running (mA)	0			
	Case 1			
	Time (us)	Current (mA)	Percent of events	
			100	
State 1 (wake-up)	1,060	6.6978		7099.668
State 2 (pre-processing)	324	7.2592		2351.981
State 3 (Tx)	446	19		8611.368
State 4 (Tx-to-Rx)	66	16		1073.16
State 5 (Rx)	127	19		2397.963
State 6 (Rx-to-Tx)	1,236	7.7313		9555.887
State 7 (Tx)	444	19		8572.752
State 8 (Tx-to-Rx)	66	17		1100.108
State 9 (Rx)	223	21		4644.979
State 10 (Rx-to-Tx)	1,141	7.9165		9032.727
State 11 (Tx)	446	19		8585.857
State 12 (Tx-to-Rx)	73	16		1182.301
State 13 (Rx)	216	21		4570.301
State 14 (post-processing)	420	11.0691		4649.022
State 15	0	0		0
State 16	0	0		0
State 17	0	0		0
State 18	0	0		0
State 19	0	0		0
State 20	0	0		0
				73428.07
Total time of connection event	6288			
Average Current draw during connection event (mA):		11.67749235		
Average current draw accounting for sleep (mA):				0.734281
<b>Average current draw during connection (mA):</b>	<b>0.734280719</b>			
<b>Expected battery life (hours):</b>	<b>313.2317029</b>			
<b>Expected battery life (days):</b>	<b>13.05132095</b>			

Table 3-8 Battery life of an advertising event of Cypress Pioneer kit

### 3.6.5 Connection

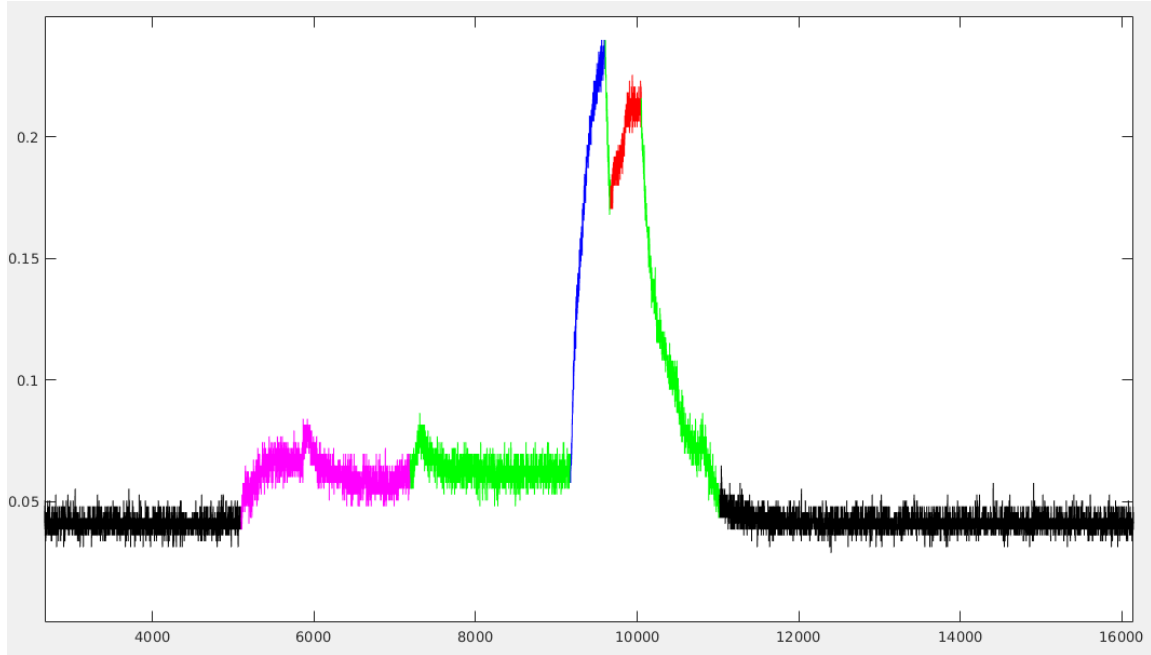


Fig. 3-27 Connection event on Cypress pioneer kit

### 3. Power Measurements

<b>Status:</b>	<b>OK</b>		
Battery capacity (mAh):	230		
Connection Interval (ms):	100		
Sleep Current with timer running (mA)	0.001		
	Case 1		
	Time (us)	Current (mA)	Percent of events
			100
State 1 (wake-up)	1,045	6.1818	6459.981
State 2 (pre-processing)	993	6.3523	6307.834
State 3 (Rx)	215	17.9064	3849.876
State 4 (Rx-to-Tx)	33	19.8873	656.2809
State 5 (Tx)	191	20.0513	3829.798
State 6 (post-processing)	485	9.7898	4748.053
State 7	0	0	0
State 8	0	0	0
State 9	0	0	0
State 10	0	0	0
State 11	0	0	0
State 12	0	0	0
State 13	0	0	0
State 14	0	0	0
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
Total time of connection event	2962		25851.82
Average Current draw during connection event (mA):		8.7278268	
Average current draw accounting for sleep (mA):			0.259489
<b>Average current draw during connection (mA):</b>	<b>0.259488611</b>		
<b>Expected battery life (hours):</b>	<b>886.3587466</b>		
<b>Expected battery life (days):</b>	<b>36.93161444</b>		

Table 3-9 Battery life of a connection event of Cypress Pioneer kit

### 3.7. Power consumption summary

The summary of the measurements presents the battery life estimation for a 230mAh coin cell battery using different connection intervals and Advertising delays in Table 3.7.1. The battery life estimation for Advertising events will give an idea of how long (in days) a beacon device can work (see Table 3-11). The battery life estimation for connection events will give an idea of how long a slave device can work, this calculation leaves out the consumption for the advertising before a connection was established (Table 3-10).

A summary of the power consumption of an advertising event is shown in Table 3-12 and the power consumption of a connection event is shown in Table 3-13. This will give an idea of how each one of the states affects the total power consumption for a given event in 3.7.2.

#### 3.7.1 Battery life estimation

Board	Connection interval 10ms	Connection interval 100ms	Connection interval 1000ms
CC2540	4 days	37 days	357 days
Arduino/Genuino101 (Arduino BLE stack)	4 days	43 days	422 days
Arduino/Genuino101 (zephyr BLE stack)	4 days	42 days	409 days
FRDM-KW41Z	2 days	23 days	234 days
CY8KIT-042-BLE	3 days	36 days	256 days

Table 3-10 Battery life summary of all the development boards for connection events



### 3. Power Measurements

Board	Advertising Interval 100ms	Advertising Interval 1s	Advertising Interval 10s
CC2540	19 days	190 days	1617 days
Arduino/Genuino101 (Arduino BLE stack)	20 days	200 days	1688 days
Arduino/Genuino101 (zephyr BLE stack)	19 days	189 days	1610 days
FRDM-KW41Z	13 days	130 days	1305 days
CY8KIT-042-BLE	14 days	139 days	1231 days

Table 3-11 Battery life summary of all the development boards for advertising events

### 3.7.2 Power consumption

State/board	CC2540	Arduino/Genuino 101 (Arduino BLE stack)	Arduino/Genuino 101 (zephyr BLE stack)	FRDM-KW41Z	CY8KIT-042-BLE
Wake-up	0.360	0.355	1.079	0.562	0.453
Pre-process	0.747	0.713	0.842	0.490	0.530
Tx1	7.005	1.904	2.107	1.339	3.856
Tx-to-Rx	1.969	1.723	2.213	1.675	2.708
Rx1	3.510	2.238	3.053	2.011	3.639
Rx-to-Tx	1.108	0.883	2.378	0.962	0.664
Tx2	6.874	1.944	2.546	1.091	3.860
Tx-to-Rx	2.167	1.744	2.173	1.708	2.854
Rx2	3.478	2.206	2.734	2.089	4.445
Rx-to-Tx	1.165	0.866	1.940	0.879	0.709
Tx3	6.729	1.960	2.439	1.125	3.843
Tx-to-Rx	1.952	1.752	2.183	1.644	2.694
Rx3	3.478	2.236	2.693	1.921	4.567
Post-process	0.634	0.880	1.085	0.924	1.401

Table 3-12 Power consumption in mW of all the development boards for advertising event states/sections

### 3. Power Measurements

State/board	CC2540	Arduino/Genuino 101 (Arduino BLE stack)	Arduino/Genuino 101 (zephyr BLE stack)	FRDM-KW41Z	CY8KIT-042-BLE
Wake-up	0.359	0.358	0.792	0.516	0.386
Pre-process	0.812	0.607	0.856	0.318	0.406
Rx1	3.799	2.016	2.068	0.838	3.439
Rx-to-Tx	1.246	1.920	2.351	1.177	3.990
Tx1	5.891	2.000	2.422	1.411	4.037
Post-process	0.661	0.592	0.919	0.781	1.092

Table 3-13 Power consumption in mW of all the development boards for connection events states/sections

## ***Conclusions***

*In this work, 4 commercial BLE development boards were used to carry out power measurements specifically on advertising and connection events. A fifth board, Raspberry-Pi3, was planned to be used due to the fact that the board itself is a high power computer running at GHz speeds and it's not considered a low power device. It was decided not to perform measurements together with the unavailability of full board schematics, which made impossible to find a measurement point like the rest of the boards.*

*The manufacturer of the Pioneer board added a placeholder for a resistor and a jumper intended for BLE power measurements. For this board, the rework was minimal and physical connections for the measurements were very straightforward.*

*For the Freedom board a jumper used to isolate the battery from the USB voltage input, according to the user guide, was helpful to carry out relatively simple the connections for the power measurements on this board.*

*For Genuino/Arduino 101, finding a measurement point wasn't as straightforward as the Pioneer board. Here the schematics were reviewed and found that the power inputs to the SoC were split in different input ports. There was the option of doing a rework on the USB cable but it was decided to drop it and go for the rework presented in section 3.4.2.*

*The summary of measurements shows that Arduino/Genuino 101 with Arduino's BLE stacks gives the longest battery life, although the purpose of this work is not finding the lowest power consumption, but to present a guided way of performing the measurement so that it can be replicated, do all the power optimizations on the application software and being able to validate the enhancement using the presented data for reference.*

*Formulas ( 3-7 ) and ( 3-8 ) are a contribution of this work for the calculations. These two formulas are a good representation of the measurements and are more reliable than the guessed average presented in TI's application note [Kamath-10].*

*The laboratory experiments and results of this work were used and supported publishing a paper on MDPI which is a pioneer in scholarly open access publishing [MDIPI-18]. The paper "Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications" and can be found in MDPI's website [Espinosa-18].*





## References

- [Kamath-10] S. Kamath, J. Lindh, "*Measuring Bluetooth Low Energy Power Consumption*", Application Note AN092, Texas Instruments, Dallas, TX, USA, 2010.
- [Afonso-16] J. A. Afonso, A. Maio, R. Simoes, "*Performance Evaluation of Bluetooth Low Energy for High Data Rate Body Area Network*", Springer Science+Business Media, New York 2016
- [Schrader-16] R. Schrader, T. Ax, C. Röhrig, C. Fühner, "*Advertising Power Consumption of Bluetooth Low Energy Systems*", IEEE International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems, Offenburg, Germany, September 2016
- [Tei-15] R. Tei, H. Yamazawa and T. Shimizu, "*BLE Power Consumption Estimation and Its Applications to Smart Manufacturing*", SICE Annual Conference, Hangzhou, China, July 2015
- [Gomez-12] C. Gomez, J. Oller and J. Paradells, "*Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology*", Sensors,
- [Kashi-16] M. Kashi, S. Vangent, "*Arduino101-REV4Schematic*", 2016.
- [Intel-16] Intel, "*Intel Quark SE Datasheet*", August 2016.
- [Zephyr-17] Zephyr Project, "*supported boards*", <https://docs.zephyrproject.org/latest/boards/boards.html>
- [NXP-18] NXP Semiconductors, "*MKW41Z Datasheet: technical Data Rev 4*", March 2018.
- [Quiros-16] A. Quiros, Shawn, "*FRDM-KW41Z Schematics*", August 2016.
- [Cypress-16] Cypress Semiconductor, "*BLE Pioneer Board Schematic*", March 2016.
- [Cypress16b] Cypress Semiconductor, "*PSoc 4 Module Schematic*", March 2016.
- [TI-13] Texas Instruments, "*cc2540\_schematics*", Texas, 2013.
- [Tektronix-09] Tektronix, "*Digital Phosphor Oscilloscopes DPO/DSA70000 Series Data Sheet*" <https://www.tek.com/oscilloscope/dpo7000-digital-phosphor-oscilloscope>, 2009
- [Tektronix-18] Tektronix, "*Passive probes*" <https://www.tek.com/passive-probe>, 2018
- [TI-18] Texas Instruments, "*CC2540*" <http://www.ti.com/product/CC2540>, 2018.
- [TI-13b] Texas Instruments, "*CC2540/41 mini Development kit user's guide*" Texas, 2013.
- [TI-15] Texas Instruments, "*SWRS084F*" Texas, 2013.
- [Arduino-18] Arduino, "*Arduino 101*" <https://store.arduino.cc/usa/arduino-101>, 2018.

- [NXP-18b] NXP Semiconductor, "*FRDM-KW41Z: Freedom Development Kit for Kinetis®KW41Z/31Z/21Z MCUs*" <https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/w-serieswireless-conn.m0-plus-m4/freedom-development-kit-for-kinetis-kw41z-31z-21z-mcus:FRDM-KW41Z>, 2018.
- [Cypress-16c] Cypress Semiconductor, "*CY8CKIT-143A PSoC® 4 BLE 256KB Module with Bluetooth 4.2 Radio*", <http://www.cypress.com/documentation/development-kitsboards/cy8ckit-143a-psoc-4-ble-256kb-module-bluetooth-42-radio>
- [Cypress-16d] Cypress Semiconductor, "*PSoC 4XX7\_BLE Family Datasheet*", 2017.
- [BLESIG-13]Specification of the Bluetooth System, "*Covered Core Package, Version: 4.1*"; The Bluetooth Special Interest Group: Kirkland, WA, USA, 2013
- [BLESIG-10]Specification of the Bluetooth System, "*Covered Core Package, Version: 4.0*"; The Bluetooth Special Interest Group: Kirkland, WA, USA, 2010
- [Adafruit-18] Adafruit "*Raspberry documentation*" <https://www.raspberrypi.org/documentation/>, 2018.
- [MDIPI-18] MDPI, "about MDPI", <https://www.mdpi.com/about>, 2018.
- [Espinosa-18] E. Garcia-Espinosa, O. Longoria-Gandara, I. Pegueros-Lepe, A. Veloz-Guerrero, "*Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications*", <https://www.mdpi.com/2079-9292/7/12/386/htm>, MDPI, December 2018.



Fig. 1-1	BLE stack.....	8
Fig. 1-2	GATT profile hierarchy.....	10
Fig. 1-3	Link layer state machine states.....	11
Fig. 1-4	Advertising event in Bluetooth low energy.....	14
Fig. 1-5	Scan request and response in the middle of an advertising event.....	14
Fig. 1-6	Timing between advertising events.....	15
Fig. 1-7	Connection creation in Bluetooth low energy.....	16
Fig. 1-8	Timing between connection events.....	17
Fig. 1-9	from advertising to connection event sequence.....	17
Fig. 2-1	Tektronix DPO7252 Oscilloscope used for measurements. Image taken from [Tektronix-09].....	18
Fig. 2-2	Tektronix passive probes used for measurements. Image taken from [Tektronix-18].	19
Fig. 2-3	CC2540 MINI development kit main elements. Image taken from [TI-13b].....	19
Fig. 2-4	CC2540 SoC subsystems. Image taken from [TI-15].	20
Fig. 2-5	Genuino 101 development board. Image downloaded from [Arduino-18].....	21
Fig. 2-6	Intel Curie block diagram. Image taken from [Intel-16].....	22
Fig. 2-7	NXP freedom development board. Image downloaded from [NXP-18b].	24
Fig. 2-8	MKW41Z SoC subsystems block diagram. Image taken from [NXP-18].....	25
Fig. 2-9	Cypress Pioneer development board. Image downloaded from [Cypress-16c].	26
Fig. 2-10	PSoc4 SoC subsystems. Image taken from [Cypress-16d].	27
Fig. 2-11	Raspberry Pi 3 development board. Image downloaded from [Adafruit-18].	28
Fig. 3-1	High level diagram of current measurement Setup.....	29
Fig. 3-2	Connection event TI CC2540. Image taken from [Kamath-10].....	30
Fig. 3-3	Advertising event TI CC2540. Image taken from [Kamath-10].	30
Fig. 3-4	battery life estimation spreadsheet TI CC2540. Image taken from [Kamath-10].	32
Fig. 3-5	example of sections found in a connection event waveform.....	33
Fig. 3-6	Power supply connection for CC2540. Schematics modified from [TI-13].	35
Fig. 3-7	Modification on CC2540 according to TI's application notes.....	36
Fig. 3-8	Measurement Setup on CC2540 board.....	37
Fig. 3-9	Advertising event on CC2540.....	40
Fig. 3-10	Connection event on CC2540.....	41
Fig. 3-11	Power supply connection for BLE submodule in Genuino 101. Schematic taken from [Kashi-16].	42
Fig. 3-12	Resistor replaced for power measurements on Genuino 101.....	43
Fig. 3-13	Measurements setup on Genuino 101.....	43
Fig. 3-14	Connection event with Arduino's BLE stack.....	44
Fig. 3-15	Connection event with Zephyr's BLE stack.....	45
Fig. 3-16	Advertising event with Arduino's BLE stack.....	47
Fig. 3-17	Advertising event with Zephyr's BLE stack.....	48
Fig. 3-18	Power supply connection for MKW41Z. Schematic modified from [NXP-18].	50
Fig. 3-19	Resistor placed for power measurements on Freedom development board.....	51
Fig. 3-20	Measurements setup on Freedom development board.....	52
Fig. 3-21	Advertising event on Freedom board.....	53
Fig. 3-22	Connection event on Freedom board.....	55

Fig. 3-23 Power supply connection for Pioneer board. Schematics modified from [Cypress-16] and [Cypress-16b].....	57
Fig. 3-24 Resistor replaced for power measurements on the Cypress Pioneer kit. ....	58
Fig. 3-25           Measurements setup on the Cypress pioneer kit .....	59
Fig. 3-26 Advertising event on Cypress pioneer kit .....	60
Fig. 3-27 Connection event on Cypress pioneer kit.....	62