

Measuring CC13xx and CC26xx current consumption

Joakim Lindh, Christin Lee, Marie Hernes and Siri Johnsrud

ABSTRACT

This application report describes the setup and procedures to measure power consumption on the CC13xx and C26xx devices. It describes how this can be done both using a DC Power Analyzer or EnergyTraceTM. Steps to analyze both a Bluetooth Low Energy peripheral and a device running the proprietary rfPacketTx example are included.

Contents

1	Introduction	3
2	Standby	4
3	Understanding Bluetooth Low Energy Power Metrics	7
4	SimpleLink Bluetooth Low Energy Wireless MCUs	8
5	Power Measurement Setup – Preparing the DUT	9
6	Measuring Power Consumption With a DC Power Analyzer	15
7	EnergyTrace	27
8	References	36

List of Figures

1	VDDR Recharge	4
2	Change of Recharge Interval Based on Standby Interval and Time From Reset	. 4
3	Measuring Standby Current During Advertisement (CC26x0)	5
4	Measuring Standby Current During Advertisement (CC26x2)	5
5	Measuring Standby Current During Connection (CC26x0)	6
6	Measuring Standby Current During Connection (CC26x2)	6
7	Current Consumption vs. Time During a Bluetooth Low Energy Connection	. 7
8	Device Under Test	9
9	CC2652R LaunchPad Jumper Removal	11
10	BTool Serial Port Settings	12
11	BTool Connection Settings	13
12	BTool Scan	13
13	BTool Scan Results	14
14	BTool Establish Link	14
15	BTool Connected Device	15
16	Agilent N6705B DC Power Analyzer	15
17	DUT Test Setup	16
18	Agilent 14585A Control and Analysis Software, Start-Up	17
19	Agilent 14585A Control and Analysis Software, Connect	17
20	Agilent 14585A Control and Analysis Software, Connected	18
21	Agilent 14585A Control and Analysis Software, Source Settings	19
22	Connecting to the CC26x2r1 LaunchPad	20
23	Connected DUT to Agilent 14585A	20
24	Agilent 14585A Control and Analysis Software, Scope	21



25	Agilent 14585A Control and Analysis Software, Scope Setup	21
26	Agilent 14585A Control and Analysis Software, Instrument Range	21
27	Agilent 14585A Control and Analysis Software, Advertisement Capture (CC26x2)	22
28	Connectable Advertising Event, Capture	23
29	Beacon Event, Capture	24
30	Connection Event, Marker #1 Placement	25
31	Average Current Consumption After Establishing a Connection	25
32	Current Consumption versus Time During a Single Connection Event	26
33	rfPacketTx in Resource Explorer	27
34	Jumper Settings	28
35	Preferences	28
36	EnergyTrace Technology Configuration	29
37	EnergyTrace Button	30
38	Set Measurement Duration	30
39	Start Trace Collection	31
40	Current Profile for rfPacketTx (without modifications)	31
41	Using SmartRF Studio to Find New Settings	32
42	Modifying the smartrf_settings.c File	32
43	Current Profile of TX (EnergyTrace)	33
44	Current Profile of TX (DC Power Analyzer)	33
45	Recharge Pulses	34
46	Average Current Consumption When Device Enters Standby	34
47	IDLE State Between Packets	35
48	Average Current Consumption When Device Does not Enter Standby	35

List of Tables

1	Acronyms Used in This Document	3
2	CC2652R Supply Voltage	8
3	Advertising Event, State Analysis	23
4	Beacon Event, State Analysis	24
5	Connection Event, State Analysis	26

Trademarks

2

EnergyTrace, Code Composer Studio, SimpleLink, LaunchPad are trademarks of Texas Instruments. Arm, Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. IAR Embedded Workbench is a trademark of IAR Systems AB. All other trademarks are the property of their respective owners.



1 Introduction

Power consumption measurements are presented and battery life time is calculated for an example application. An accompanying Power Calculation Tool is provided so that you can estimate your battery life based on your own custom usage scenario.

Note that the results presented in this document are intended as guidelines and measurement results presented in this application report may not be up to date with the latest software optimizations. A variety of factors will influence the battery life of a Bluetooth Low Energy product. Measurements should be performed on hardware in a controlled environment and under the target application scenario.

It is assumed the reader of this document has some knowledge of the Bluetooth Low Energy standard, as well as the Texas Instruments SimpleLink[™] Bluetooth Low Energy wireless MCUs with the Software Development Kit BLE-Stack. In addition, it is assumed that the reader has some knowledge of basic electrical engineering concepts and understands how to use laboratory test equipment such as an oscilloscope and DC power supply.

1.1 Acronyms

Acronym	Description
ADC	Analog to Digital Converter
BLE	Bluetooth Low Energy
CCS	Code Composer Studio
CM3	Cortex-M3
CPU	Central Processing Unit
CSV	Comma-Separated Values
DC	Direct Current
DK	Development Kit
DUT	Device under Test
GAP	Generic Access Profile
GPIO	General-Purpose Input/Output
MCU	Micro Controller Unit
PC	Personal Computer
RAM	Random Access Memory
RF	Radio Frequency
RTC	Real Time Clock
RTOS	Real Time Operating System
RX	Receive
SCA	Sleep Crystal Accuracy
SPI	Serial Peripheral Interface
ТХ	Transmit

Table 1. Acronyms Used in This Document

Introduction



Standby

2 Standby

Before we start looking into how we can measure current consumption, it is important to understand the Standby mode of the CC13xx and CC26xx devices. Standby is the lowest power mode where the CC13xx and CC26xx devices still have functionality other than maintaining I/O output pins. Standby is normally the power mode used between radio events if no other parts of the system are active. Current consumption in Standby mode consists of two parts: a recharge current pulse, used to charge up the VDDR capacitor, and the current consumption between the recharges. The latter is around 70 nA, almost too small to measure. It is the average power consumption during Standby including recharge that is defined as the Standby current, approximately 1 μ A, as stated in the data manual for the given device (see [4] through [13]). Figure 1 shows what a recharge pulse looks like.



Figure 1. VDDR Recharge

While the CC13x2 and CC26x2 have a built-in comparator that gives the optimal recharge interval at any time (and at any temperature), the recharge pulses are dynamically adapted based on the required time in Standby for the CC13x0 and CC26x0 devices. For the latter devices, the recharge interval will also depend on when in time the measurements are done with respect to the last reset of the DUT. This is illustrated in Figure 2.



Figure 2. Change of Recharge Interval Based on Standby Interval and Time From Reset



In the first case, the Standby intervals (A, B, C, ...) are short and there is only one recharge pulse between each wakeup. In this case, the Standby current will be higher than the 1 μ A stated in the data manuals.

In the second case, the Standby intervals are longer (A*, B*, C*), and there is room for several recharge pulses within one Standby interval. In this case, the recharge interval will get a little bit longer for every recharge pulse. When starting a new Standby interval (B*), you will not get back to the minimum recharge interval, but start where you ended up in the previous Standby period (A*) (with some margins). Because of this, you will end up with the max recharge interval after a while (M*) (if your Standby intervals are long enough) and your Standby current will get down to 1 μ A. In Figure 3, a CC26x0 is advertising with a 100 ms interval and there is one recharge in between the advertising events; in this case, the resulting Standby current is 1.57 μ A. The Standby current will not go lower than this when advertising with a 100 ms connection interval. For CC26x2, you will not have any recharge pulses for this setup (see Figure 4) and the current consumption is below 60 nA. Recharge pulses will be observed if increasing the advertisement interval sufficiently.



Figure 3. Measuring Standby Current During Advertisement (CC26x0)



Figure 4. Measuring Standby Current During Advertisement (CC26x2)



When a connection has been established as described in Section 6.3.2, similar measurements can be done (still using CC26x0), resulting in a Standby current of 0.88 μ A due to the long connection interval (1 s). In this case the recharge interval has increased and there are only 2 recharge pulses during the Standby period (see Figure 5). Measuring Standby current on the CC26x2 with the same connection interval results in a Standby current of 0.90 μ A and there is only one recharge pulse during Standby (see Figure 6).



Figure 5. Measuring Standby Current During Connection (CC26x0)



Figure 6. Measuring Standby Current During Connection (CC26x2)



3 Understanding Bluetooth Low Energy Power Metrics

A Bluetooth Low Energy device achieves low power consumption by keeping radio activity short and allowing the device to reside in Standby or Power Down mode most of the operating time.

The operation of a Bluetooth Low Energy device is typically static in the sense that it's staying in a certain mode for a certain amount of time, for example, when advertising or maintaining a connection. These modes are based on re-occurring events that can easily be used to estimated average power consumption. Each of these modes can be quantified into states for future estimations based on added data throughput or reduced latency (through higher connection interval, as an example).

The primary metric is the average current for the advertising and connected mode. It is these values that can be used to determine the battery life of a Bluetooth Low Energy device.

For a wireless MCU it is important to understand that the device is typically not only running the Bluetooth Low Energy protocol stack, but also profile services and an application. The application may also be using peripherals on the chip, such as serial peripheral interface (SPI) or analog-to-digital converter (ADC). In addition, other devices on the circuit board, aside from the device running the Bluetooth Low Energy protocol stack, may be drawing current as well.

There are three main components of a Bluetooth Low Energy application that together sum up the average power consumption: *Standby*, *Protocol events* and *Application events*. Depending on the use case of the Bluetooth Low Energy device, these components will consume different amounts of power.

Figure 7 shows the current profile for a connected Bluetooth Low Energy device. The device spends most of the time in Standby, where the average current consumption is around 1 µA (see [4] through [13]).



Figure 7. Current Consumption vs. Time During a Bluetooth Low Energy Connection

From Standby, the device only wakes up based on either external interrupts or scheduled events/interrupts from the RTC. Standby also includes the recharge, which is described in Section 2.

The Protocol event is where communication over the Bluetooth Low Energy protocol occurs. For a Bluetooth Low Energy device, these events can be either Advertising events or Connection events. There are multiple roles featured that allow a Bluetooth Low Energy device to enter Observer role and scan as well but they are not covered in this application report

The Application event is the application-specific implementation, for example, a periodic event, serial communication, or running algorithms based on sensor inputs. Depending on the amount of activity, the application event can increase power consumption significantly, hence, always aim to optimize processing usage. The Application events typically occur between protocol events, which mean that a longer advertising or connection interval gives longer time slots for processing.



SimpleLink Bluetooth Low Energy Wireless MCUs

4 SimpleLink Bluetooth Low Energy Wireless MCUs

There are several Bluetooth Low Energy Solutions provided by Texas Instruments. These cover a wide range of solutions, from simple broadcaster only to advanced multiple-role Real Time Operating System (RTOS) featured solutions. An overview of TI's Bluetooth Low energy offering can be found here: http://www.ti.com/wireless-connectivity/simplelink-solutions/bluetooth-low-energy/products.html.

In the first part of this application report, measurements are done on the CC2652R [13], but everything discussed here regarding how to measure the current consumption is also valid for the other CC26xx and CC13xx devices.

The CC2652R is a Multi-Standard Wireless MCU providing a complete solution on a single chip. The application processor is an Arm® Cortex®-M4F and it is used for running the Bluetooth Low Energy Profiles along with any user defined functionality.

The RF core ensures that all timing regarding the Bluetooth Low Energy protocol is being configured and handled properly. An Arm Cortex-M0 is dedicated for the radio operations and runs the Bluetooth Low Energy Radio Firmware from its own dedicated ROM.

The CC2652R can be powered by two supply ranges, as presented in Table 2. To enable the 1.8 V system, both hardware and software modifications are required, which is documented in the CC13x2, CC26x2 SimpleLink[™] Wireless MCU Technical Reference Manual [3]. For the CC13x0/CC26x0 device, see the CC13x0, CC26x0 SimpleLink[™] Wireless MCU Technical Reference Manual [2].

Supply Voltage	Internal DCDC	Minimum	Maximum
1.8 V System (External Regulator Mode)	No	1.7 V	1.95 V
3.3 V System	Optional	1.80 V	3.80 V

Table 2. CC2652R Supply Voltage

For more information about Supply Voltage, see the device-specific user's manual (Section 8).



5 Power Measurement Setup – Preparing the DUT

Before measurement and analysis can be performed, the device under test (DUT) must be prepared both from a hardware and software perspective. A peer device can also be configured in order to establish a connection. In this application report, a device running the example project HostTest [15] is used to establish the connection. This project can be run on any BLE-enabled development board and we have used a CC26x2R LaunchPad[™] [19].



Figure 8. Device Under Test

5.1 Requirements

To measure average power consumption for Bluetooth Low Energy on CC26xx, the following hardware from Texas Instruments can be used:

- CC26x2R LaunchPad
- CC2650/CC2640R2/CC1352R/CC1352P LaunchPad [17]/[18]/[20]/[21]
- A device running the HostTest project Optional

The above mentioned hardware can be can be purchased at the TI Store [14].

In terms of software resources, the following are required:

- BLE-Stack [15]
- IAR EWARM [16]

or

CCS Integrated Development Environment [22]

Make sure you are using the IAR and CCS version recommended for your version of the SDK/BLE-Stack. For more details, see the device-specific software release notes.

5.2 Embedded Software

The BLE-stack is either a stand-alone deliverable or provided as part of the Software Development Kit (SDK), depending on which device you use. The software package includes the full Bluetooth Low Energy protocol stack along with sample applications. The protocol stack is provided as a pre-qualified library component and the complete system is operated by an RTOS that introduces a threaded environment with full power management. The power management is maintained by the Power Driver automatically and the application can constrain tasks or disallow certain power modes, if required. The current consumption for the different power modes can be found in the device-specific data manual (Section 8).

The generic sample application simple_peripheral that is included with the BLE-Stack is ideal to use in order to analyze power consumption for the sole Bluetooth Low Energy protocol running on a wireless MCU. Depending on what version of simple_peripheral you are using, the application does one or more of the following:

- Advertise with legacy advertisements on LE 1M PHY
- Advertise with extended advertisements on LE Coded PHY
- Start a clock that calls a periodic event
- Print messages on UART display

For the CC2650 LaunchPad, the external flash is not turned off by default, hence you will measure an extra current consumption of about 7 μ A. The flash can be turned off by calling the ExtFlash_open(); followed by ExtFlash_close(); (functions found in ExtFlash.h/ExtFlash.c). To get a power measurement that is easy to interpret, you should disable all application behaviors described above except legacy advertising, and make sure the external flash on your board is shut down.

For more information including instructions on how to program the CC26x0, see the CC2640 and CC2650 SimpleLink[™] Bluetooth[®] low energy Software Stack 2.2.1 Developer's Guide [1]. For CC13x2/CC26x2, see the BLE5-Stack User's Guide [23].



5.3 Hardware

5.3.1 CC26x2R LaunchPad

To get a clean current measurement, the jumpers on the LaunchPad should be removed. The CC2652R Launchpad with all jumpers removed is shown in Figure 9. Note that when the JTAG jumpers are removed, the programming and debug capabilities of the chip become unavailable. This is also applicable for the other CC26xx and CC13xx LaunchPads ([17] - [21]).



Figure 9. CC2652R LaunchPad Jumper Removal

5.4 BTool (Optional)

The BLE-Stack also includes BTool along with drivers and firmware. BTool can be used to emulate a Bluetooth Low Energy application from a PC environment. BTool is used to create a connection with the DUT. If the intention is to measure power consumption of the DUT when being in advertising or beacon mode, BTool is not required.

To connect to the DUT using BTool, a device running a Bluetooth Low Energy wireless network processor image named HostTest is required. This can be found with the other example projects in the BLE-Stack.

With a Launchpad running the HostTest application connected to the PC, open BTool and select the COM port used by the application (see Figure 10).

Nerial Port	t Settings	×
Port:	COM5 - XDS110 Class Application/User UART 🗸	
Baud:	115200 ~	
Flow:	None ~	
Parity:	None ~	
Stop Bits:	One ~	
Data Bits:	8 ~	
	OK Cancel	

Figure 10. BTool Serial Port Settings

Press OK and there should be an initialization process that is observed in the log window.

Before forming the connection, the proper connection parameters should be used. This is dependent on the application that is being considered. The supervision timeout setting should not affect the power measurements. A connection interval of one second, with zero slave latency, is used in this document. Therefore, use the values as shown in Figure 11. Be sure to select the "Set" button after entering in the values. Setting up the connection parameters needs to be done before a connection is established.



BTool - Bluetooth Low Energy Application - v1.42.13	(BLE5)								- 0	6 ×
Device Options View About										
Setta Device Operation	Convertient Conver	5 Servery Perki Scan Scan Man Co Man Co Sup Sup Sup Sup Sup Sup Sup Sup Sup Sup	ande: 0	Dustion: 500 Man Records: 40 Sour & Concel terrel (5200): 600 © (1000 Obm) terrel (5200): 000 © (2000m) exer (1020 0bm) (449): 00 © (2000m) exer (1020 0bm) © (2000m) Establin Lnk A Torc Cancel Lisk Option			Description 000000144:FE 000000144:FE 000000144:FE 000000144:FE 000000144:FE 000000144:FE 000000144:FE 00000144:FE 00000144:FE 00000144:FE 00000144:FE 0000144:FE 0000144:FE	22 01 06		*
	ConHnd	Handle	Uuid	Uuid Description	Value	Value Description	Properties			
Hardware Connected - (Devices = 1)									TEXAS INSTE	RUMENTS

Figure 11. BTool Connection Settings

With the connection parameters set as needed, setup is completed.

At this point, BTool is ready to discover the DUT. If you left the SimpleBLEPeripheral application running on your DUT, you should be ready to use BTool. As long as the device running SimpleBLEPeripheral is powered up and not connected to anything, it should be in discoverable (advertising) mode.

In the Discovery section, press the "Scan" button, as shown in Figure 12.

Discovery			
Period:	0	Duration:	500
Devices Found:	0	Max Records:	40
Scan	S	ican & Connect	Cancel

Figure 12. BTool Scan



Power Measurement Setup – Preparing the DUT

www.ti.com

BTool will begin searching for Bluetooth Low Energy devices. When the discovery process finishes, the address of any scanned devices will appear in the "Slave BDA" section, as shown in Figure 13.



Figure 13. BTool Scan Results

To establish a connection with the peripheral device, select the address of the device to connect with and click the "Establish" button, as shown in Figure 14.

		Establish Link		
	Slave BDA:	00:81:F9:4A:4D:F5	\sim	
Establish			Cancel	

Figure 14. BTool Establish Link



As long as the peripheral is powered-up and still in discoverable mode, a connection should be established immediately. Once a connection is established, the message window will return a "GAP_EstablishLink" event message with a "Status" value of "0x00 (Success)". In BTool, you can see your connected peripheral device in the Device Information field, as shown in Figure 15.



Figure 15. BTool Connected Device

6 Measuring Power Consumption With a DC Power Analyzer

The most accurate way of measuring power consumption is to use a DC Power Analyzer (since the power consumption varies over time, a simple multimeter will not be sufficient). An oscilloscope can be used as well, as long as the sampling rate and bandwidth is good enough. For the purpose of this application report, an Agilent N6705B DC Power Analyzer is used (see Figure 16). The internal module is a N6781A, a 2-quadrant source and measure unit for battery drain analysis.



Figure 16. Agilent N6705B DC Power Analyzer



6.1 Test Setup

Make sure that the system is set up properly and review the steps described in Section 5. For reference, the full overview is illustrated in Figure 17. VDD is connected to the 3V3 pin on the LaunchPad.

When the DUT is correctly connected, the power supply is enabled by pressing the "On" button within the Agilent 14585A Control and Analysis Software. The power consumption measurements can be done by two separate functions: Scope or Data Logger. The Data Logger provides an average power consumption measurement over longer time, for example, minutes and hours, although the resolution is not as good as using Scope. This document focuses on doing measurements by using the Scope feature.



Figure 17. DUT Test Setup

The Agilent N6705B powers the DUT as well as performs the current measurement.



Measuring Power Consumption With a DC Power Analyzer

www.ti.com

6.1.1 Analysis Software Setup

All measurements and analysis can be done directly with the Agilent N6705B interface, but in this application report, a PC Tool called "14585A Control and Analysis Software for Advanced Power Supplies" (v2.0.2.1) is used to control the Agilent N6705B. The software is available from http://www.keysight.com/ (in 2014, Agilent electronics instruments division was acquired by Keysight Technologies).

When the PC Tool is started, no external equipment is connected, which is observed in the "Instrument Control Tab", as shown in Figure 18.



Figure 18. Agilent 14585A Control and Analysis Software, Start-Up

To connect the Agilent N6705B, make sure that it is connected via USB and that it is powered. Use the bottom left "Connect" button to select the connected hardware, as shown in Figure 19.



Figure 19. Agilent 14585A Control and Analysis Software, Connect



When the hardware has been successfully connected, it is fully controlled from the PC Tool, which is verified by the "Instrument Control" tab, as shown in Figure 20.



Figure 20. Agilent 14585A Control and Analysis Software, Connected



Note that the Output may be "On" per default (observed by the lit "on" button). If so, turn the Output off since the actual output parameters have not been configured yet. The next step is to configure the output. In the "Instrument Control" tab, click the "Settings" button to bring up the Source Settings for Output 1. Depending on the module within the Agilent N6705B, the options may be limited. Select "2 Quadrant Power Supply" and set the "Voltage" to 3.0 V.

Output 1 - Source Settings ** Mode Emulating 2 Quadrant Power Supply Operating In Voltage Priority **	Instrument Con
Voltage 3.00000 V Range 6.12 V € + Current Limit 3.060000 A ± Tracking Limits - Current Limit -3.060000 A ✓ Tracking Limits Resistance 0.0000 Ω Ω Delay Close	trol Error Log
Meter Properties Menu Settings Select Output	0
On On On On On	

Figure 21. Agilent 14585A Control and Analysis Software, Source Settings



Measuring Power Consumption With a DC Power Analyzer

Connect the instrument probes to the DUT. For the LaunchPads, the VDD line should be connected to the 3V3 pin. The GND can be connected to any GND pin. Connecting to the CC26x2R1 LaunchPad is shown in Figure 22.



Figure 22. Connecting to the CC26x2r1 LaunchPad

When the DUT is correctly connected, the power supply is enabled by pressing the "On" button within the Agilent 14585A Control and Analysis Software. The power consumption measurements can be done by two separate functions: Scope or Data Logger. The Data Logger provides an average power consumption measurement over longer time, for example, minutes and hours, although the resolution is not as good as using Scope. This document focuses on doing measurements by using the Scope feature.



Figure 23. Connected DUT to Agilent 14585A



6.2 Measurement Using Scope

When the instrument has been correctly setup and configured, make sure that Scope has been selected, as shown in Figure 24.

File Edit Iools Scope Help 🕕 🔨 Scope	Data Logger 🛛 🙀 CCDF	M ARB – 🗖	×
--------------------------------------	----------------------	-----------	---

Figure 24. Agilent 14585A Control and Analysis Software, Scope

The scope mode allows that measurement be ran over a short amount of time. In order to maximize the amount of data, use the following measurement setup: (see Figure 25).

- Time/div: 200 ms/
- Points: 512k
- Trigger: Scope Run Button
- Mode: Single
- Slope: Rising Edge

Stopped		Ň	Aarkers 8	t Measu	reme	nts 🕟					6	Ranges	AUTO SCALE	E.	Ð	Q
200 ms / 🔻	Offset:	0 s	Points:	512k		Period: 5 µs	Trigger	Scope Run Button	Mode:	Single		Slope:	Level:			

Figure 25. Agilent 14585A Control and Analysis Software, Scope Setup

Next, make sure that the "Ranges..." is setup to Auto, as shown in Figure 26.

Output 1 Rar	iges			
Voltage:	Auto	 Current	Auto	
Output 2 Rar	iges			
Voltage:	20 V	Current	3 A	
Output 3 Rar	nges			
Output 4 Rar	nges			
		-		

Figure 26. Agilent 14585A Control and Analysis Software, Instrument Range

The instrument should now be setup properly and the measurement can start. Click the Play button in the bottom right corner \rightarrow allow the instrument to start the measurement.



6.3 Analysis

Depending on what the DUT is setup to do, the result will vary. If no interaction has been made with the DUT, it will be sending out periodic advertisements each 100 ms (see Figure 27).



Figure 27. Agilent 14585A Control and Analysis Software, Advertisement Capture (CC26x2)

The approximately 2.6 s measurement includes 26 advertising events. There are no recharge pulses in Standby, as the measurement is done on CC26x2 (see Section 2).

There is functionality to do detailed measurements of the acquired waveform. Select "Markers & Measurements" to enable the markers. There are two approaches of using the markers:

- Measure the average power consumption from a symmetric point of the measurement, (for example, from the start of an event to the point where the next event starts). This will give an approximation of the overall power consumption over time because of the reoccurring symmetry.
- Break down the events into states to be used for various use case studies and estimations. This is very useful in order to analyze the resulting power consumption when intervals are changed

If the objective is to simply obtain a power consumption figure of the DUT, the first option is fast and reliable.

6.3.1 Advertising Event

An advertising event is where the (Bluetooth Low Energy) peripheral device broadcasts information in order to either share information or become connected to a (Bluetooth Low Energy Ready) Central device, such as a smart phone. The device wakes up and broadcasts packets on three separate channels and listens on each of these channels for Scan Requests or Connection Requests. Scan Requests is a way for a Central device to obtain more information about the device before connecting, because the advertising data is typically chosen to be very short to minimize power consumption. Based on advertising data or the scan response data, connection requests can be sent, which initiates a connection between the Peripheral and the Central.

With connectable advertising packet format, the base time of data transmitting is 144 μ s, which contains a pilot tone plus 1 byte preamble, 4 bytes Access Address, 2 bytes PDU, 3 bytes CRC and 6 bytes AdvA in the payload. For every additional transmitted bit, 1 μ s should be added to the TX time.



The Agilent 14585A Control and Analysis Software "Markers & Measurements" functions are used to quantify a single advertising event, which is visualized in Figure 28 and summarized in Table 3.



Figure 28. Connectable Advertising Event, Capture

Number	State	Comments
1	Pre-processing	RTOS wake-up, radio setup, XTAL guard time
2	Radio preparation	Radio is turned on and in transition to TX
3	ТХ	The radio transmits an advertisement packets with 3 bytes data on Channel 37. Time is dependent on amount of transmitted data
4	TX to RX transition	TX to RX transition
5	RX	Time depends on advertising interval and Sleep Crystal Accuracy (SCA)
6	RX to TX transition	RX to TX transition
7	ТХ	The radio transmits an advertisement packets with 3 bytes data on Channel 38. Time is dependent on amount of transmitted data
8	TX to RX transition	TX to RX transition
9	RX	Time depends on advertising interval and SCA
10	RX to TX transition	RX to TX transition
11	ТХ	The radio transmits an advertisement packets with 3 bytes data on Channel 39. Time is dependent on amount of transmitted data
12	TX to RX transition	TX to RX transition
13	RX	Time depends on advertising interval and SCA
14	Post-processing and going to Standby	Bluetooth Low Energy protocol stack processes the received packets and sets up the sleep timer in preparation for the next event, and then goes to Standby

Table 3. Advertising Event, State Analysis

This is also the event occurring when a device is in beacon mode. For a non-connectable beacon, there are no RX states during the advertising event, reducing the average current consumption.



Figure 29. Beacon Event, Capture

Table 4. Deacon Event, State Analysis	Table 4.	Beacon	Event,	State	Analysis
---------------------------------------	----------	--------	--------	-------	----------

Number	State	Comments
1	Pre-processing	RTOS wake-up, radio setup, XTAL guard time
2	Radio preparation	Radio is turned on and in transition to TX
3	ТХ	The radio transmits an advertisement packets with 3 bytes data on Channel 37. Time is dependent on amount of transmitted data
4	TX-to-TX transition	TX to TX transition
5	ТХ	The radio transmits an advertisement packets with 3 bytes data on Channel 38. Time is dependent on amount of transmitted data
6	TX-to-TX transition	TX to TX transition
7	ТХ	The radio transmits an advertisement packets with 3 bytes data on Channel 39. Time is dependent on amount of transmitted data
8	Post-processing and going to Standby	Bluetooth Low Energy protocol stack sets up the sleep timer in preparation for the next event, and then goes to Standby

6.3.2 Connection Event

When a connection has been established between a Peripheral and a Central device, they communicate during connection events. The Central device operates as the master and the Peripheral device as the slave.

All communication between two connected devices occurs on these connection events. They are periodically with a configurable connection interval, ranging from 7.5 ms to 4 s.

Each event occurs on one of the 37 data channels and the master always initiates the event, with the slave listening. They can continue to communicate back and forth as much as they want during one connection event.

Connection events occur even if neither side has data to send. This ensures that the link is still valid. If a specified number of connection events occur without acknowledgment, the connection will be considered lost.



In order to measure the current consumption during a Connection event, the DUT must be connected to a peer device. By using BTool as described in Section 5.4, a connection with 1 second connection interval is established.

To measure the average current consumption after a connection has been established (Figure 6), you should identify two Connection events. Place Marker 1 right after a Connection event, as shown in Figure 30, and the second marker after the following Connection event. The average current is in this case approximately $10.2 \mu A$, as shown in Figure 31.



Figure 30. Connection Event, Marker #1 Placement



Figure 31. Average Current Consumption After Establishing a Connection



The Connection event can also be analyzed (like what was done with the Connectable Advertising event in Figure 28) by selecting "Markers & Measurements". This is shown in Figure 32 and summarized in Table 5.



Figure 32. Current Consumption versus Time During a Single Connection Event

Table 5. Connection Event, State Analysis

Number	State	Comment
1	Pre-processing	RTOS wake-up, radio setup, XTAL guard time
2	Radio preparation	Radio is turned on and in transition to RX
3	Recieve (RX)	The radio receiver listens for a packet from the master. Time depends on connection interval and SCA.
4	RX to TX transition	RX to TX transition
5	Transmit (TX)	The radio transmits a packet to the master on one of the 37 channels. Time is dependent on the amount of transmitted data
6	Post-processing and going to Standby	Bluetooth Low Energy protocol stack processes the received packets and sets up the sleep timer in preparation for the next event, and then goes to Standby

6.3.3 Power Consumption Calculator

The state analysis from advertising and connection states can be used to investigate how the battery life varies depending on connection interval. For that purpose, a Power Calculator Tool [24] is provided that can be used to perform calculations for your custom application.



7 EnergyTrace

EnergyTrace is available on all CC13x2 and CC26x2 LaunchPads. The tool can be used stand-alone, as a power profiling tool, or in EnergyTrace++ mode (4-pin JTAG required) within a debug session for state monitoring to help optimize the application for ultra-low-power consumption. This section focuses on the steps necessary to run EnergyTrace in stand-alone-mode in CCS. In this mode, the debugger is not active and the displayed current consumption is what to expect for the final application. Since the previous sections have focused on Bluetooth Low Energy, this section will use one of the proprietary examples to show how to use EnergyTrace. Please note that the CC13x2 Proprietary RF User's Guide [26] has a section with info regarding EnergyTrace.

EnergyTrace

The rfPacketTx example (available for our CC13x2 devices) was used running on the CC1352R1 LunchPad [20].

The example can be found and downloaded using Resource Explorer (see Figure 33).

Elle Edit View Navigate Project Bun Scripts Window Help					
🔁 • 🔄 🐘 🗳 • 🦉 • 🔦 • 🖉 🛷 • ি 🗎 😳 🗇 •	¢ *			Quick Access	3 🗊 🗞
🍘 🔁 Project Explorer 😫 📄 🛱 👻 📟 🗖	Resource Explorer 20			-	
V riPacketTx_CC1352R1_LAUNCHXL_tirtos_ccs [Active - Debug] S Includes	Q, Select a Device or Board	×	Search Q	△ 🕈 👁	= 0
> >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Device Documentation	erfPacketTx	rfPacketTx	105	
 gb supplicating gb supplicating gb supplicating gb cC1381, LARCK141, Into gb cC1381, LARCK141, LARCK141, Into gb cC1381, LARCK141, LARCK141, Into gb cC1381, LARCK141, LARCK141, LARCK141, LARCK141, Into gb cC1381, LARCK141, LARCK1	 Chick Downentation Sphare S			<u> </u>	
	 b) an should be be an analysis of the second second				
1 ADALAT: CONSIDE LAURICIAL Sites on			es t		

Figure 33. rfPacketTx in Resource Explorer



After building the example, it should be downloaded to the LaunchPad and the following steps should be done:

 Remove all jumpers on the LaunchPad between the XDS debugger and the device, except for the XDS110 Power, the 3V3 and RXD jumpers (see Figure 34). It is important that the XDS110 jumper is mounted in the "XDS110 Power" position when the LaunchPad is powered up, otherwise the calibration of EnergyTrace will fail. The UART driver in the SDK configures the UART RX pin without internal pull-up. To avoid current leakage in the input buffer, the pin must always be firmly pulled to a logic level. This can be achieved by keeping the RXD jumper on (connecting the debugger output to the UART RXD input).



Figure 34. Jumper Settings

- 2. If the jumpers were not set correctly BEFORE powering the board, trigger a re-calibration of EnergyTrace by power cycling the LaunchPad (disconnect and re-connect the micro-USB cable).
- 3. EnergyTrace requires some configurations the first time it is being used within a CCS workspace. Go to the menu `Window` and select `Preferences` (see Figure 35).



Figure 35. Preferences



4. Navigate to the `EnergyTrace Technology` window and configure it as shown in Figure 36.

pe filter text	EnergyTrace	' Technology	\$	• 🗇 •	
Code Composer Studio Advanced Tools Disk Usage EnergyTrace ^m Technology Source Line Reference Trace Viewer Build Debug Grace Products Help Install/Update Run/Debug Team Terminal	EnergyTace" ' determine the - available for all EnergyTrace+ " analysis tool th energy profile - for ultra-low p devices and sel Guide" for deta EnergyTrace++ energy-based and viewing th CPU and perjup consumption. devices and sel SimpleLink" U	echnology enables energy consumptio devices with select technology in add at is useful for mea and correlating with wer consumption. ected debuggers. P ils. Th technology in ad ode and peripheral a applications energi- neral states and opt Phis feature is avail acted debuggers. P ser's Guide" for det -Launch on target	analog energy measurement i n of an application. This featu ed debuggers. ition supports an energy-base suring and viewing the applica the devices CPU state and op This feature is available on MI lease check the "CCS for MSP- dition supports an enhanced analysis tool that is useful for yp profile and correlating with imizing it for ultra-low power uble on selected MSP430 and S lease check the "CCS for MSP- ails.	to re is ations timizing it SP432 432 User's measuring the device SimpleLink 430 or	
	EnergyTrac O EnergyTrac	:e :e+[CPU State]+[Pe	ripheral States]		
	Cell Type	CR2032		~	
	Cell voltage (/) 3.0			
	Cell capacity				
	Optional				
	Peak current	- continuous (mA)	0.0		
	Peak current	- pulse (mA)	0.0		
	Target lifetim	e (days)	0.0		
	Target Conne	ction			
	Connection	XDS110			
	Device	N45D422/DE422/C	C12/CC26		
	Device	WISP452/KF452/C	C15XX/CC20XX	~	
	Voltage (mV)	3300.0			
	ET-HDR Rang Low currer High currer	e Selector nt, narrower range H nt, wider range low	igher accuracy er accuracy		
	When save pr	ofile also save: o CSV file			

Figure 36. EnergyTrace Technology Configuration

If post-processing of the acquired data is wanted, select the 'Raw data to CSV file' checkbox. If this checkbox is selected, you can, after EnergyTrace is finished capturing data, select the 'Save current energy profile' button, to save a .csv file. The default location for this file will be under your project workspace.



EnergyTrace

5. Click the EnergyTrace Button (see Figure 37).

EnergyTrace - rfPacketTx_CC1352R1_LAUNCH0L_titrtos_ccs/main_titrtos.c - Ele file View Noticet, Delete Res Excite Window Hole	Code Composer Studio							-	σ	×
C + C / C = + C + C + O + N = C + C	• ¢ •							Quick Access	et 54	0
Project Explorer 12 EnergyTrace TM Technology	Resource Explorer	05.6 22								
 Biological Constant JulianCold, Biological Constant, Col Johnson, Col	<pre>42 / BOS header files '/ disclude city/selectEs.) 43 / Foundar/Board Header file 45 / Foundar/Board Header file 45 / Foundar/Board Header file 45 / Foundar/Board Header file 45 / Foundar/Board Header file 55 / Foundar files header 55 / Foundar files header 56 / Foundar files header 57 / Foundar files header 56 / Foundar files header 57 / Foundar files header 57 / Foundar files header 56 / Foundar files header 57 / Foundar files header 58 / Foundar files header 59 / Foundar files header 50 / Foundar fil</pre>	ho le: "/ id "arg0); 1024	<pre>ch0tate); bptAccsIIE(); edg, MALL();</pre>					۵ و	* •	
	Description	Resource	Path	Location	Туре					
						én				

Figure 37. EnergyTrace Button

A dialog with instructions on how to use the EnergyTrace Stand-alone Measurement Mode will pop-up. Click 'Proceed' to continue.

 Select how long you want to capture data by clicking the 'Set Measurement Duration' button (see Figure 38).



Figure 38. Set Measurement Duration



7. Click the green play button to start capturing data (see Figure 39). The red LED on the XDS110 debugger should be turned on, and will be so for the duration of the EnergyTrace capture.



Figure 39. Start Trace Collection

 When EnergyTrace is finished capturing data, review the application's power profile and have a closer look at the current graph [▶]Current.

Figure 40 shows the current profile for the rfPacketTX example taken over 1 s. From the plot, it is easy to identify the packet interval of 500 ms and to verify that the device enters Standby in-between packets. If you want to zoom in on the current graph, you can use the magnifying glass symbol.



Figure 40. Current Profile for rfPacketTx (without modifications)



EnergyTrace

7.1 Modifying the rfPacketTX Example

Using SmartRF[™] Studio [25], you can find the modifications that need to be done in smartrf_settings.c to change, for example, the output power to 10 dBm (see Figure 41).

CC1352R_PROP_REV11 (Preview) - Device Control Panel (offline)				– a ×
Eile Settings View Evaluation Board Help				
Packet TX 🔹 🕨 Start 🖩 Stop 🗹 Command View 🖓 RF Parameters				
				8 ×
Target Configuration				Carlo Daniel - Conside Filling - Characht
RF Design Based On: LAUNCHOL-CC1352R1 🔹 🥝	DC/DC Enable	Deta 0 0 0 000	Custo	mize
				Radio Operation Commands Value
Tuning Fattings				CMD_FS CMD_PROP_TX
Category Setting Name				CMD_PROP_RADIO_DIV_SETUP
Settings for 779 - 930 MHz band	,			status 0x0000
50 kbps, 2-GESK, 25 kHz deviation	·			pNextOp 0x0000000 startTime 0x00000000
5 kbps, SimpleLink Long Range (20 kchip/s, 2-GFSK, conv. FEC r=1/2 K+7, DSSS SF+2, Tx dev	:: 5 kHz, Rx BW: 34 kHz) (Release Candidate)			startTrigger 0x00
2.5 kbps, SimpleLink Long Range (20 kchip/s, 2-GFSK, conv. FEC r=1/2 K=7, DSSS SF=4, Tx d	ev.: 5 kHz, Rx BW: 34 kHz) (Release Candidate)			condition 0x01 modulation 0x0321
V Seconds for 451 - 527 Minz pairs				symboRate 0x0080000F
				ndbw 0x52 preamConf 0x04
•				formatConf 0x00A0
RF Parameters 🕗				config 0x0008 txPower 0x6EE1
Frequency Symbol Rate		Deviation		pRegOverride 0x0000000
868.00000 MHz 50.00000		kBaud 25.000		kHz centerFreq 0x0364 intFreg 0x8000
RX Filter BW TX Power		Whitening		IcDivider 0x05
98.0 • kHz	✓ dBm	No whitening		-
Continuous TX Continuous RX Packet TX Packet RX				
Packet Count 100 infinite				
Length Config: Variable 👻				
Length Length				5 L
Presmole Sync word (dec)	Packet Data		CAL	
	0116 40 00 80 00			4
Preamble Count: 4 Bytes V Sync Word Length: 32 Bits Address				
Preamble Mode: Send 0 as the first preamble bit				
Add Seq. Number			1 0	< >
Random a5 ea 5b b8 c1 b7 10 08 07 84 0e 20 c6 75 f9 69 14 53 45 10 b7 de 81 fe 46 65 95 0b				▼ Output
() Text				
() Hex			TX RX	
			Sent Packets: 0	
			Frequency: 868.0000	å MHZ
			Output Power: 10 dBm	
Advanced			Start Stop	
Not Conserted	Office M	via		Darlie State N.A.

Figure 41. Using SmartRF Studio to Find New Settings

After changing the output power by modifying the smartrf_setting.c file, as shown in Figure 42, the measurements were repeated.



Figure 42. Modifying the smartrf_settings.c File



Figure 43 shows the current profile for transmitting a packet achieved with using EnergyTrace. It shows a TX power consumption at +10 dBm (3.3 V) of 14.78 mA. Figure 44 shows the same profile, captured using the DC Power Analyzer. It shows a current consumption in TX of 14.4 mA. The data sheet numbers [7] for TX at +10 dBm is 14.3 mA.

Even if the numbers you get when using EnergyTrace is not as accurate as the ones obtained with the DC Power Analyzer, the numbers will be very useful when estimating current consumption for an application.



Figure 43. Current Profile of TX (EnergyTrace)



Figure 44. Current Profile of TX (DC Power Analyzer)

It is not possible to accurately measure the Standby current when using EnergyTrace, but you can still use it to verify that the device is in Standby.

As described in Section 2, the CC13x2/CC26x2 have a built-in comparator that gives the optimal recharge interval at any time. In the previous measurements, no re-charge pulses have been seen.

If the packet interval, and hence the time in Standby, is doubled the recharge pulses should appear. In the rfPacketTX example, this can easily be done by including the POWER_MEASUREMENT define in rfPacketTx.c.



EnergyTrace

www.ti.com

The packet interval is then changed from 500 ms to 5 s. The PACKET_INTERVAL also needs to change from 5 to 1.

With these modifications, the re-charge pulses are easy to identify during the time in Standby (see Figure 45).



Figure 45. Recharge Pulses

For the average current consumption of an application, it is very important that the device always enters the lowest possible power mode. If, for some reason, there were things in the rfPacketTX examples that prevented the device from entering Standby in-between packets, the average current consumption would increase significantly. For the current profile shown in the previous code example, the average current consumption is 0.1 mA (see Figure 46).

echnology 🔀 🛷 Search 🔤 Energy 🔤 Current
e
Live
5 sec
1.538 mJ
0.3384 mW
0.0000 mW
50.7862 mW
3.3000 V
0.1026 mA
0.0000 mA
15.3898 mA
CR2032: 89.4 day (est.)

Figure 46. Average Current Consumption When Device Enters Standby



EnergyTrace

To see how the current consumption will look like if the device is not entering Standby, a power constraint is set to disallow this power mode. This can be done with the following modification:



Figure 47 shows what the current profile looks like, with an average current consumption of 1 mA (see Figure 48).



Figure 47. IDLE State Between Packets

Ø EnergyTrace [™] 1	Fechnology 🔀 🔗 Search 🔛 Energy 🔛 Current
EnergyTrace™ Prof	ile
Name 🗸 System	Live
Time	5 sec
Energy	16.869 mJ
Power	
Mean	3.7128 mW
Min	1.8755 mW
Max	49.4523 mW
✓ Voltage	
Mean	3.3000 V
✓ Current	
Mean	1.1251 mA
Min	0.5683 mA
Max	14,9855 mA

Figure 48. Average Current Consumption When Device Does not Enter Standby

References



www.ti.com

8 References

- 1. Texas Instruments: CC2640 and CC2650 SimpleLink[™] Bluetooth[®] low energy Software Stack 2.2.1 Developer's Guide
- 2. Texas Instruments: CC13x0, CC26x0 SimpleLink™ Wireless MCU Technical Reference Manual
- 3. Texas Instruments: CC13x2, CC26x2 SimpleLink™ Wireless MCU Technical Reference Manual
- 4. Texas Instruments: CC1310 SimpleLink™ Ultra-Low-Power Sub-1 GHz Wireless MCU Data Manual
- 5. Texas Instruments: CC1312R SimpleLink™ High-Performance Sub-1 GHz Wireless MCU Data Manual
- 6. Texas Instruments: CC1350 SimpleLink™ Ultra-Low-Power Dual-Band Wireless MCU Data Manual
- 7. Texas Instruments: CC1352R SimpleLink™ High-Performance Dual-Band Wireless MCU Data Manual
- 8. Texas Instruments: CC1352P SimpleLink[™] High-Performance Dual-Band Wireless MCU With Integrated Power Amplifier Data Manual
- 9. Texas Instruments: CC2640 SimpleLink[™] Bluetooth® Wireless MCU Data Manual
- 10. Texas Instruments: CC2640R2F SimpleLink™ Bluetooth® low energy Wireless MCU Data Manual
- 11. Texas Instruments: CC2642R SimpleLink™ Bluetooth® 5 low energy Wireless MCU Data Manual
- 12. Texas Instruments: CC2650 SimpleLink[™] Multistandard Wireless MCU Data Manual
- 13. Texas Instruments: CC2652R SimpleLink™ Multiprotocol 2.4-GHz Wireless MCU Data Manual
- 14. TI Store: https://store.ti.com/
- 15. BLE-Stack[™]: www.ti.com/ble-stack
- 16. IAR Embedded Workbench[™] for Arm: http://www.iar.com/Products/IAR-Embedded-Workbench/ARM/
- 17. SimpleLink[™] CC2650 wireless MCU LaunchPad[™] Development Kit (http://www.ti.com/tool/launchXLcc2650)
- 18. SimpleLink[™] Bluetooth[®] low energy CC2640R2F wireless MCU LaunchPad[™] development kit (http://www.ti.com/tool/launchxl-cc2640r2)
- 19. SimpleLink[™] Multi-Standard CC26x2R Wireless MCU LaunchPad[™] Development Kit (http://www.ti.com/tool/launchxl-cc26x2r1)
- 20. SimpleLink[™] Multi-Band CC1352R Wireless MCU LaunchPad[™] Development Kit (http://www.ti.com/tool/launchxl-cc1352r1)
- 21. SimpleLink[™] Multi-Band CC1352P Wireless MCU LaunchPad[™] Development Kit (http://www.ti.com/tool/launchxl-cc1352p)
- 22. CCS Integrated Development Environment: (http://www.ti.com/tool/ccstudio)
- 23. BLE5-Stack User's Guide: (http://dev.ti.com/tirex/#/?link=Software%2FSimpleLink%20CC26X2%20SDK%2FDocuments%2FBLE5 -Stack%2FBLE5-Stack%20User's%20Guide)
- 24. Bluetooth Power Calculator Tool: (http://www.ti.com/tool/bt-power-calc)
- 25. SmartRF™ Studio: (http://www.ti.com/tool/smartrftm-studio)
- 26. CC13x2 Proprietary RF User's Guide: (http://dev.ti.com/tirex/#/?link=Software%2FSimpleLink%20CC13x2%20SDK%2FDocuments%2FProprietary%20RF%2FProprietary%20RF%20User's%20Guide)



Page

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from C Revision (January 2017) to D Revision

•	The complete document has been updated to cover the CC13x2/CC26x2 devices and references to the old hardware has been removed.
•	Added new Section 7 27

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2019, Texas Instruments Incorporated