Measuring Power Consumption of CC2530 With Z-Stack

By Chester Kim

Keywords

- *ZigBee®*
- *Z-Stack™*
- *CC2530*
- *Polling*

- *Power Consumption*
- *Battery Lifetime*
- *HA Sample Application*

1   Introduction

The CC2530 is a true system-on-chip (SoC) solution for IEEE 802.15.4, ZigBee and RF4CE applications. It enables building of robust network nodes with very low total bill-of-material (BOM) costs. The CC2530 combines the excellent performance of a leading RF transceiver with an industry-standard enhanced 8051 MCU, in-system programmable flash memory, 8-kB RAM, and many other powerful features [1], [2].

The CC2530 has various operating modes, making it highly suited for systems in which ultralow power consumption is required. Short transition times between operating modes further ensure low energy consumption.

Combined with the industry-leading and golden-unit-status ZigBee protocol stack (Z-Stack) from Texas Instruments, the CC2530F256 provides a robust and complete ZigBee solution [3].

This application note describes how to set up power consumption measurements for a CC2530EM node running Z-Stack. The application used for these measurements is the sample application for the use of the ZigBee Home Automation profile, which is included in the Z-Stack 2.5.1a release (downloaded from [3]) and it is described in [4].

This application note uses the ZigBee PRO stack profile.

For measurements in combination with the ZigBee RF4CE compliant RemoTI™ stack [5], the reader is referred to application note AN073 [6].

The measurement setup consists of a ZigBee End Device (ZED) and a ZigBee Coordinator (ZC). The ZED periodically polls the ZC for data, and between the different poll messages the ZED goes to sleep (using power mode 2) to save power.

This document first describes which hardware and software (see Chapter 3) was used for the measurement setup, which is described in Chapter 4. The obtained results are shown and discussed in Chapter 5.

Many factors influence the overall power consumption; the results presented in this document are indicative of what is possible to achieve in systems with similar hardware.

For more information about the use of the CC2530, the reader is referred to the CC253x/4x User's Guide [7]. More details regarding the power management using Z-Stack on the CC2530 can be found in [8].

Table of Contents

2   Abbreviations and Acronyms

ACK	Acknowledgement
APS	Application Support Sub-Layer
CSMA-CA	Carrier Sense Multiple Access, Collision Avoidance
BB	Battery Board
DB	Development Board
EB	Evaluation Board
EW	Embedded Workbench
I	Current (Ampere, A)
mA	Milliampere (10^{-3} A)
MAC	Medium Access
OSAL	OS Abstraction Layer
PM2	Power Mode 2
R	Resistance (Ohm, Ω)
RAM	Random Access Memory
RX	Receive
TX	Transmit
PHY	Physical Layer
Px.y	Pin y on Port x
µA	Microampere (10^{-6} A)
V	Voltage (Volt)
ZC	ZigBee Coordinator
ZDK	ZigBee Development Kit
ZED	ZigBee End Device
ZR	ZigBee Router

3   System Overview

This chapter describes the hardware and software used for the measurement setup described in Chapter 4.

3.1   Hardware - CC2530ZDK

The hardware used in this application note (two SmartRF05EB, one CC2531-USB-dongle, two CC2530EMs, cabling, and two antennas) is taken from the CC2530 ZigBee Development Kit (CC2530ZDK); however, one could also use the CC2530DK. Figure 1 shows the content of the complete CC2530ZDK (see [9] for more details).

Figure 1: CC2530ZDK – Hardware Included in the CC2530 ZigBee Development Kit

Additionally, a DC power supply, a digital real-time oscilloscope, a 10 Ω (ohm) resistor, and some cabling are used. The detailed measurement setup is shown in Chapter 4.

3.2   Software – Z-Stack

The following subsections describe the software that was used to perform the measurements.

3.2.1   Z-Stack Development Environment

The measurement setup shown in Chapter 4 is based on the Home Automation sample application, which comes as part of the Z-Stack 2.5.1a release [3] and is described in [4].

To compile the application and load onto the CC2530EMs, one must install the correct version of the IAR 8051 Embedded Workbench (IAR EW8051 8.10), which can be obtained at http://www.iar.com/ti_zigbee (evaluation version) or from www.iar.com/ew8051.

4 Measurement Setup

This chapter explains how to setup and configure the HW and SW described in Chapter 3 in order to produce the measurements shown and discussed in Chapter 5.

4.1   Instrumentation

The general idea of the current consumption measurement is to visualize the current profile on an oscilloscope by measuring the voltage drop over a fixed resistor. Figure 2 shows the measurement setup.

Figure 2 - Measurement Setup

The oscilloscope provides a graphical representation of the voltage drop over the resistor. Because there is a linear relationship between voltage and current (Ohm's Law), the same graphical representation illustrates the current consumed by the system (see also Section 4.2.4). The EM is connected as shown in Figure 2 through two pins (GND on EM_P.1.1 and VDD on EM_P2.7). The power supply is set up to provide a constant voltage of 3.0 V.

Note:
This measurement method may influence the result because of the voltage drop over the resistor and the stray (cable) resistance. Ideally the measurement setup should have been verified with a high-precision ampere-meter. The results are also read from the oscilloscope using the eye; which will also influence the accuracy.
Keep in mind that the results shown in this application note are indicative and that you would have to perform your own measurements on your own hardware to determine true power consumption.

The very low current consumption during sleep mode (PM2) has not been measured using the above setup because this setup does not allow measuring a small current in the magnitude of µA due to measurement accuracy. Instead, this current was measured to approximately 1 µA with an ampere-meter (replacing the resistor and oscilloscope with the ampere-meter in Figure 2), which is in accordance to the value stated in the CC2530 data sheet [1].

4.2   Software Setup

This section describes the software application setup for the measurements (obtained with the measurement setup shown in Figure 2) and how to reproduce it.

The Home Automation sample applications *SampleSwitch* and *SampleLight* using ZigBee Pro are programmed on the CC2530EM modules. Both sample applications are included in the Z-Stack 2.5.1a release for the CC2530 [3] and described in [4]. They are part of the sample application demonstrating the use of the ZigBee Home Automation profile. Further details about the use of the Z-Stack and features like binding are in the ZigBee Developer's Guide [13].

The idea is to bind the light device (SampleLight – EndDeviceEB) to the switch (SampleSwitch – CoordinatorEB) using the binding command ZDP_EndDeviceBindReq as explained in Section 4.2.4, ZC Node. Then the ZED polls its parent (the ZC) periodically to determine whether there are messages pending. This polling operation is referred to as **Operation1** in the following when there is no data pending. The second scenario (**Operation2**) describes the situation in which a message (*toggle the light*) is pending, received, processed, and acknowledged by the light application.

Note:
In the described setup, the ZED (CC2530EM) is running the SampleLight application and the ZC (CC2530EM + SmartRF05EB) is running the SampleSwitch application, even though it would seem more realistic with the switch as a battery-powered ZED instead of the light. The setup in these measurements is not meant as a specific real application scenario, but as an easy-to-set up way of measuring current consumption in a generic battery-powered ZED running Z-stack and CC2530; which is polling its parent and receiving messages and sending replies.

4.2.1   Programming

The following two sections describe how to set up, compile and download firmware to the ZC node (see Section 4.2.2) and the ZED node (see Section 4.2.3). The default settings for the *SampleLight* project have power saving disabled; hence, the following instructions describe how to set up the project with power saving enabled on the ZED to allow it to go into sleep mode (PM2), which consumes approximately 1 µA.

To best understand the following detailed description for the measurement setup one must have the correct software installed (Z-Stack and IAR 8051 EW); for details see Section 3.2.1, Z-Stack Development Environment.

In the following, the boards (CC2530EM) that run the application are programmed by simply connecting them to a Smart05EB that is connected to the PC through a USB cable included in the kit. When connected in such a way, the debug feature in the IAR EW can be used to program the devices because the EW automatically programs the device with the current project open in the EW for debugging it (after compiling it if needed). Then one can stop the debugging process and the firmware remains on the CC2530EM. For more details, see the Z-Stack User's Guide [14].

4.2.2   ZC Node

Open the workspace file *SampleSwitch.eww* with the correct version of the IAR EW 8051.
The project file is found in the following folder after installing the Z-Stack-CC2530-2.5.1a:

C:\Texas Instruments\ZStack-CC2530-2.5.1a\Projects\zstack\HomeAutomation\SampleSwitch\CC2530DB\

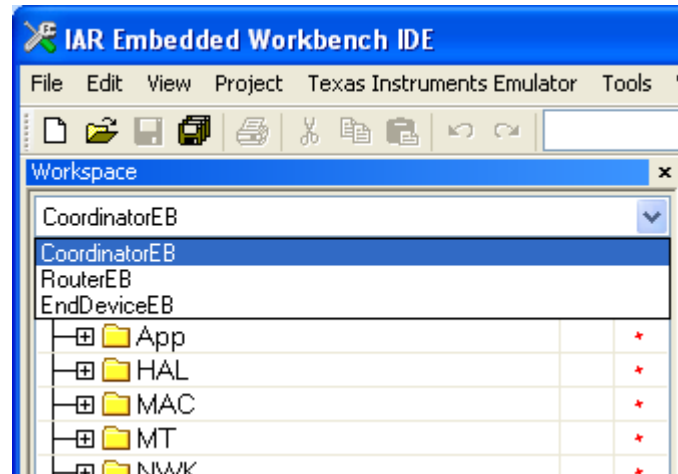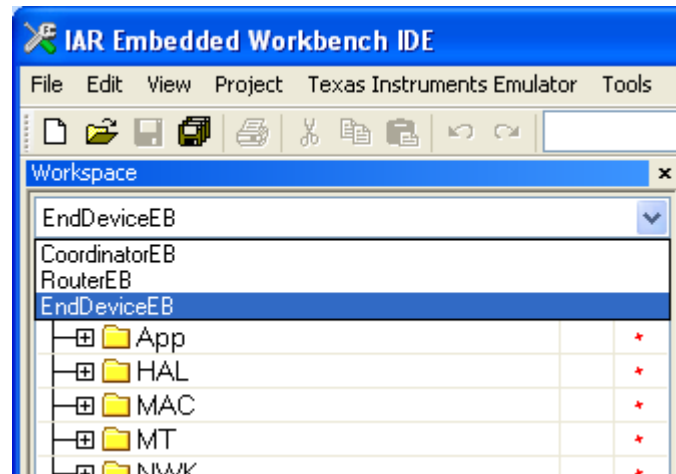Next, choose the CoordinatorEB configuration.

Figure 3 – Choosing Workspace

Section 4.2.4, Binding, explains how to establish the binding between the light and the switch. To enable the features to store and restore the binding information, one has to add the NV_RESTORE compile option to the project, as shown in Figure 4. For more information about the NV_RESTORE compile option see [13] and other Z-Stack documentation.

When programming a board with the NV_RESTORE option (that enables the node to save it current network state, bindings, and so forth, such that they are still there after a power toggle or failure on the board) it is important to erase the flash to ensure that all old data are erased; this prevents a freshly programmed node from behaving strangely based on old NV data. The erase flash option must be enabled in the project options, as shown in Figure 5.

As the final step, start the debug process Project→ Debug (or CTRL+D). Now the CC2530EM board is programmed with the *SampleSwitch* application and acts as ZC during the measurements. Stop the debugger (CTRL+SHIFT+D) and the CC2530EM is ready to be used. To start the ZC, toggle the power on the board.

For the measurements the programmed CC2530 EM board was plugged into a SmartRF05EB board that is either powered through USB or batteries (see [11] for details).
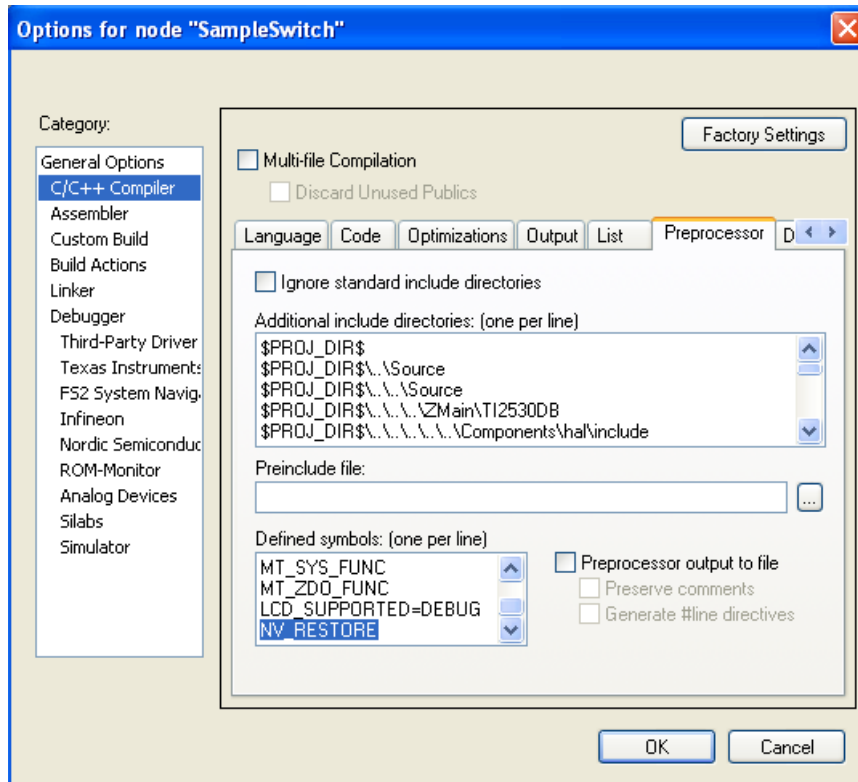
Figure 4 – Setting NV_RESTORE Compile Option
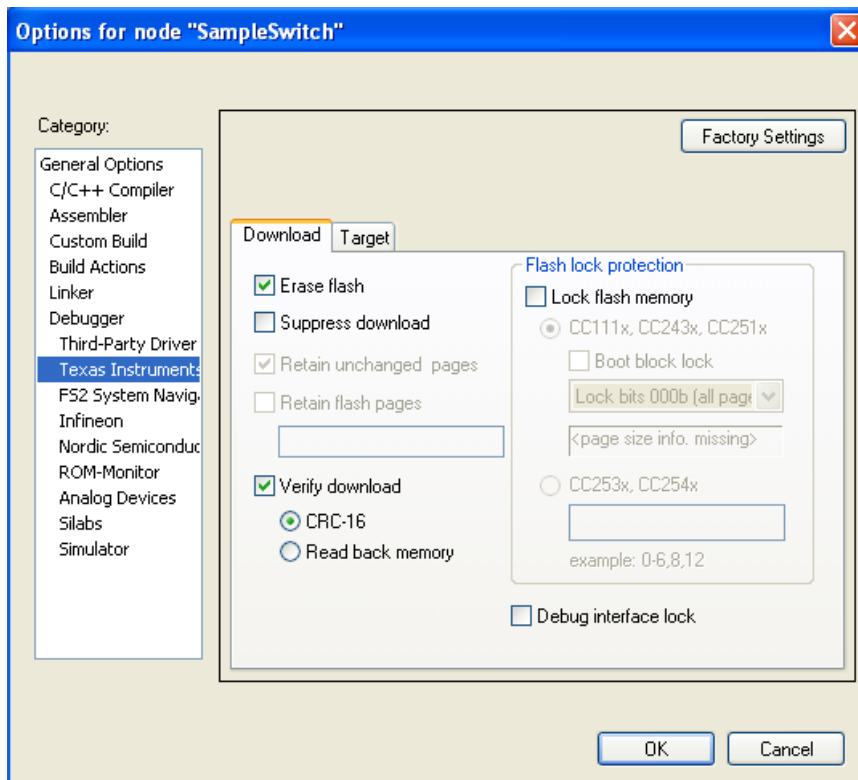
Figure 5 – Setting Erase Flash Option

4.2.3 ZED Node

Open the workspace file *SampleLight.eww* with the correct version of the IAR EW 8051.
The project file is found in the following folder after installing the Z-Stack-CC2530-2.5.1a:

C:\Texas Instruments\ZStack-CC2530-2.5.1a\Projects\zstack\HomeAutomation\SampleLight\CC2530DB

Next, choose the EndDeviceEB configuration.

Figure 6 – Choosing workspace

The power saving features (implemented in the Z-Stack for ZEDs) are not enabled by default (as they would hinder the debugging in the development phase). To enable the power-saving feature for the measurement setup, simply set the correct compile options by setting the Defined Symbols in the IAR EW. For all the details about the power-saving functionality, see [8].

As already mentioned in Section 4.2.3, ZC Node, it is also important to add the NV_RESTORE compile option; see Section 4.2.4 for more details, see Section 4.2.4, Binding as it is explains how to establish the binding between the light and the switch.

Go to Project → Options (ALT+F7) and find the C/C++ Compiler->Preprocessor and make sure that the following values are included to enable network polling, power saving, and remembering the binding (see also):
        NWK_AUTO_POLL
        xLCD_SUPPORTED=DEBUG
        POWER_SAVING
        NV_RESTORE

Also, ensure that the ZED sets the erase flash option as shown in Figure 5 for the ZC.
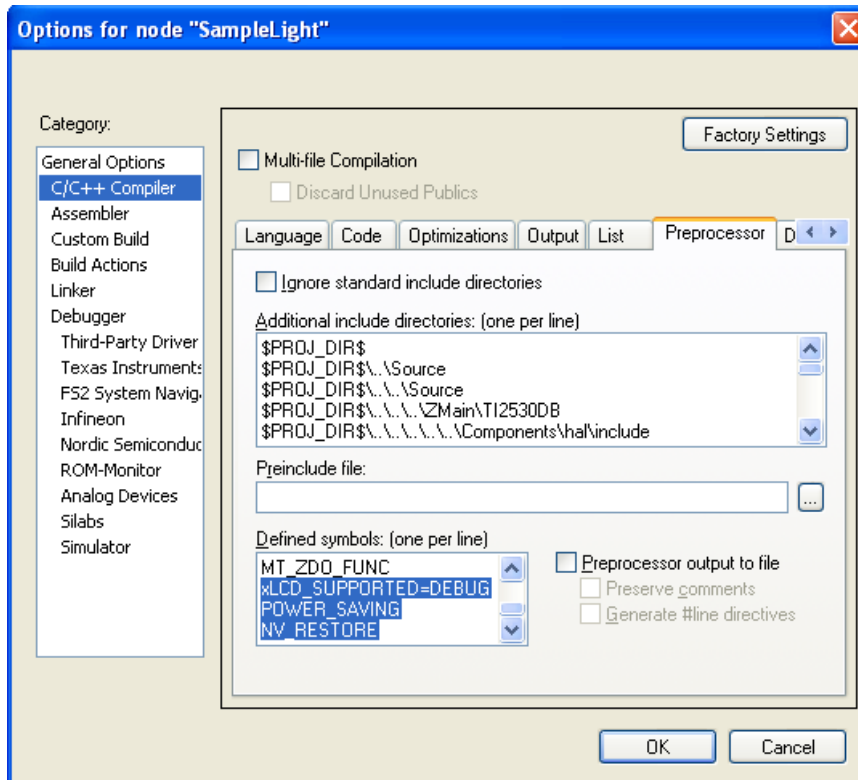
Figure 7 – Setting Compile Options for the ZED

To more easily catch the measurements on the oscilloscope, adjust the polling rate in the ZED, which determines how often the device wakes up from sleep (PM2) and sends a data request to its parent to poll for pending data (queued messages). In this application note, a polling rate of 5 seconds (-DPOLL_RATE=5000) has been used:
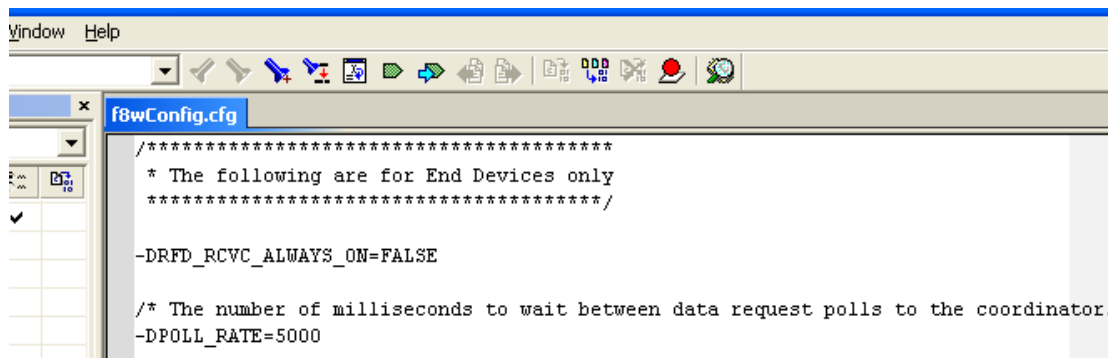
Figure 8 – Setting Up Poll Rate

Remark: To save power one might want to increase the polling rate even further; however, one should be aware that there are basically two limiting factors for the polling rate:
1) the length of the timer that is used to implement this (maximum 65 seconds) and
2) in a ZigBee network, parents keep only the data buffered for their children for a certain time (hence a ZED that sleeps too long would miss messages as they time-out in its parent.
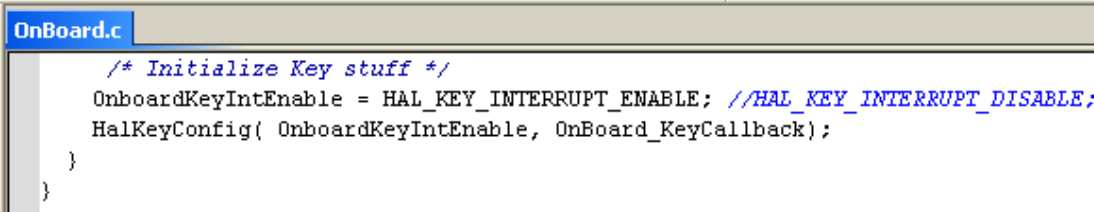
One could, of course, disable polling completely (if the device is not supposed to receive data); however, this might lead to high latency as the device (when waking up) first must poll and if the parent is gone rejoin or orphan join the network again before it can send its data. The latter can take a long time or become a problem if the network in the meantime changed channel or security key. For more details, see the ZigBee specification.

The sample application is by default programmed to check for user input (for example, pressing buttons or using the joystick). This also periodically wakes up the MCU and consumes power. To turn this off as it is too application board specific and hence not relevant for this application note, one must turn off the key polling feature. This is done by changing the following line in the function `InitBoard(byte level)` that can be found in the file *Onboard.c*

From: `OnboardKeyIntEnable = HAL_KEY_INTERRUPT_DISABLE;`
To: `OnboardKeyIntEnable = HAL_KEY_INTERRUPT_ENABLE;`

```
OnBoard.c

    /* Initialize Key stuff */
    OnboardKeyIntEnable = HAL_KEY_INTERRUPT_ENABLE; //HAL_KEY_INTERRUPT_DISABLE;
    HalKeyConfig( OnboardKeyIntEnable, OnBoard_KeyCallback);
  }
}
```

Figure 9 – Disabling Key Polling

As the final step, start the debug process Project→ Debug (or CTRL+D). Now the CC2530 EM board is programmed with the SampleLight application with power savings enabled and act as ZED during the measurements.

Stop the debugger (CTRL+SHIFT+D) and follow the instructions in Section 4.2.4 to establish the correct binding, before detaching the CC2530EM from the SmartRF05EB board (that was used to program it and establish the binding), and then connect it to the wiring, as shown in Figure 2.

4.2.4 Binding

After following the above steps place each freshly programmed CC2530EM (ZC and ZED) on an EB board (one could also use a BB board) to perform the binding based on the ZDP_EndDeviceBindReq.

1. Turn on the ZC to start the network. The LCD display on the EB (to which the ZC is attached) should indicate when the network is established by displaying "ZigBee Coord" and "Newtwork ID: <PANID>".
2. As soon as the network is started, turn on the ZED and wait until its yellow LED is solid to indicate that it successfully joined the ZC.
3. As soon as the network is established, press the joystick on both boards to the right (seen from the lower side of the board when it is oriented such that one can read the silk screen). Then the ZED issues a ZDP_EndDeviceBindReq, which is answered by the ZC with a ZDP_EndDeviceBindRsp to indicate that the light is now bound to the switch. The success of this procedure is indicated by a solid green LED1 on the ZC board.
4. Toggle the light (green LED1 on the board to which the ZED is attached) by pressing the joystick up (toward the mounted CC2530 EM).

For more details, see the Home Automation sample application description in [4]. At this stage, the ZED can be turned off and taken off from the board it was plugged into in order to connect it through wires (plugged into the EM connectors), as shown in Figure 2.

Also, the ZC can now be turned off (just like the ZED) as both were compiled with the NV_RESTORE compile options, which saves the established binding, PAN ID, and short address (next to other parameters).

From now on, one can always turn them on again and the network forms itself again (the ZED performs an orphan join to the ZC). From that point, the ZED starts again to poll the ZC for data. If the joystick on the ZC board is pressed up, the toggle command is picked up by the ZED.

The operation where the ZED polls its parent (the ZC) periodically to see whether there are messages for it pending is in the following referred to as:
- **Operation1** when there is no data pending
- **Operation2** when there is a message (*toggle the light*) from the switch (ZC) pending, received, processed, and acknowledged by the light application (ZED).

4.3   Calculation of the Average Current Consumption

The calculation of the current is based on the well known relation (Ohm's Law)

V = IR

where V, I, and R represent the voltage, the current, and the resistance, respectively.

By measuring at the power supply side, the test system observes the current consumed by the CC2530EM. FAQ 5 of Chapter 11 in the SmartRF05EB User Guide [11] also describes a method to measure the current consumption while the CC2530EM is plugged into the SmartRF05EB; however, to avoid any leakage current due to components on the EB, the measurement setup described in Figure 2 was chosen here.

As explained in AN057 [12], the value of R should not be too large; a value that is too large reduces the effective voltage over the evaluation module (EM) itself, for example:

$V_{\text{TARGET BOARD}} = V_{\text{POWER SUPPLY}} - IR$

To keep the measurement system simple and easy to reproduce, we will accept the error introduced by the resistor is acceptable and a value of 10 Ω (ohm) is chosen in accordance with AN057.

5   Measurements

This chapter shows and explains the results that were obtained with the setup explained in the previous chapters. In the measurement setups, the TX packets are sent out at 0-dBm power. Because a 10-Ω resistor is used, 10 mV in the measurement is translated into 1 mA consumption, and 50 mV is equivalent to 5 mA, accordingly.

5.1   Measurements for Operation1

In Operation1, the ZED polls the ZC, sending MAC Data Req command, and it is assumed that there is MAC ACK from the ZC with Frame Pending subfield of 0, which means the ZC doesn't have any message to send to the ZED and therefore the ZED doesn't need to wait for a message and can go to sleep right away.

MAC Data Req packet is 18 bytes long including 6-byte PHY overhead, which takes 576 μs (18 byte x 8 bit/byte x 4 μs/bit) over-the-air. In the same manner, MAC ACK is 11 bytes-long and takes 352 μs over-the-air. MAC Data Req and MAC ACK are plotted at Point 5 to 6 and Point 7 to 8, respectively, in Figure 10.

Time period length of each unit operation section is constant except CSMA/CA (Point 3 to 4) time, which depends on the channel condition. 1.2025 ms for the unit operation CSMA/CA in Table 1 is the value averaged from 20 measurement samples.

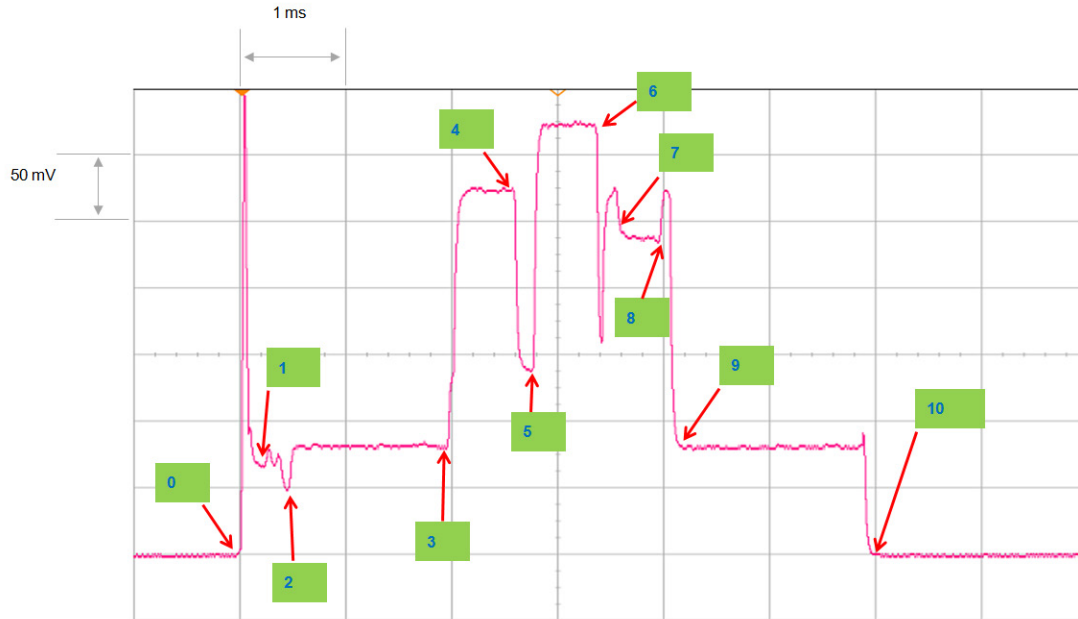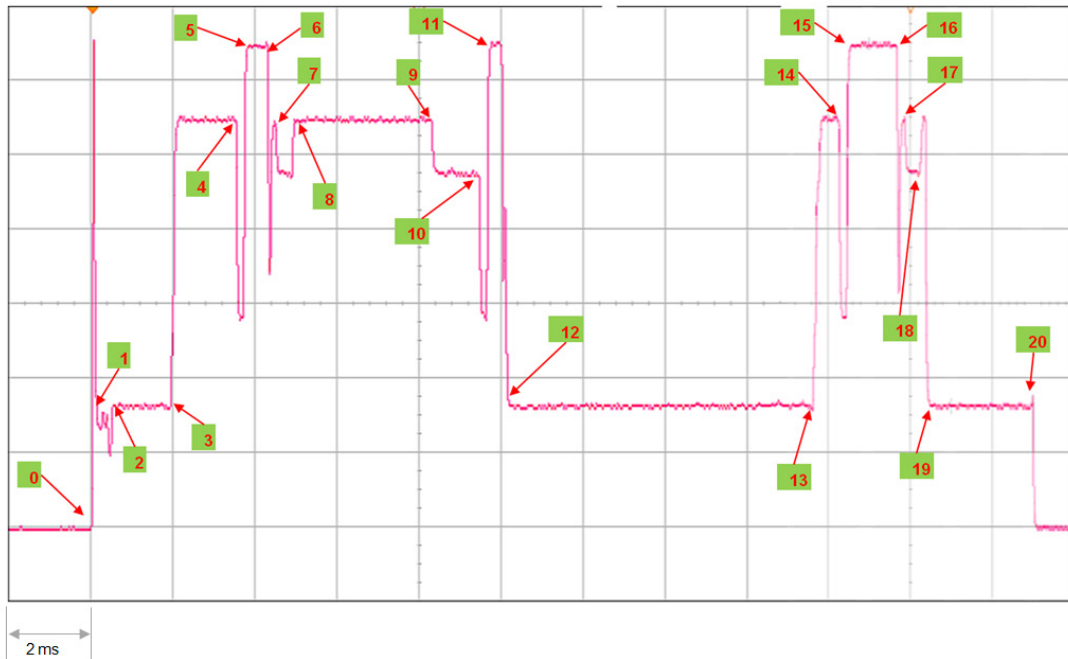**Figure 10 – Current Consumption Measurement Plot on Operation1**

Section	Unit Operation Description	Voltage (mV)	Current (mA)	Time (ms)	Consumption (mA*ms)
Before 0	Power Mode 2		0.001		
Point 0 to 1	Power mode start-up sequence.	120	12	0.2	2.4
Point 1 to 2	MCU in active mode running on 16-MHz clock	60	6	0.25	1.5
Point 2 to 3	MCU running on 32-MHz clock	75	7.5	1.7	12.75
Point 3 to 4	CMSA/CA algorithm. Radio in RX mode	270	27	1.2025	32.4675
Point 4 to 5	Switch from RX to TX	140	14	0.2	2.8
Point 5 to 6	Transmitting MAC data request. Radio in TX mode	320	32	0.58	18.56
Point 6 to 7	Switch from TX to RX	250	25	0.2	5
Point 7 to 8	Receiving MAC ACK from coordinator	235	23.5	0.35	8.225
Point 8 to 9	Radio remaining in RX mode and processing the MAC ACK	250	25	0.15	3.75
Point 9 to 10	Processing and shut down	75	7.5	1.75	13.125
After 10	Power Mode 2		0.001		
				6.5825	100.5775

Table 1 – Current Consumption Measurement Breakdown on Operation1

5.2 Measurements for Operation2

In Operation2, the ZED polls the ZC, sending MAC Data Req command, and the ZC responds to the ZED with MAC ACK where Frame Pending subfield is 1, which means the ZC has a message heading to the ZED. Upon the reception of the MAC ACK with Frame Pending subfield of 1, the ZED waits for a message (Toggle command in this example) that the ZC is sending soon. After receiving the message, the ZED responds to the ZC with sending Default Response command.

A Toggle command packet is 36 bytes long, which takes 1.152 ms, and a Default Response command packet is 38 bytes long, which takes 1.216 ms. The Receiving Toggle command and sending Default Response command are plotted at Point 9 to 10 and Point 15 to16, respectively, in Figure 11.

Time period length of each unit operation section is constant except Point 3 to 4, Point 8 to 9, and Point 13 to 14 which depend on the channel condition. The time values for those three unit operation sections in Table 2 are the averaged values from 20 measurement samples.

Figure 11 – Current Consumption Measurement Plot on Operation2

Section	Unit Operation Description	Voltage (mV)	Current (mA)	Time (ms)	Consumption (mA*ms)
Before 0	Power Mode 2		0.001		
Point 0 to 1	Power mode start-up sequence.	120	12	0.2	2.4
Point 1 to 2	MCU in active mode running on 16-MHz clock	60	6	0.25	1.5
Point 2 to 3	MCU running on 32-MHz clock	75	7.5	1.7	12.75
Point 3 to 4	CMSA/CA algorithm. Radio in RX mode	270	27	1.068	28.836
Point 4 to 5	Switch from RX to TX	140	14	0.2	2.8
Point 5 to 6	Transmitting MAC data request. Radio in TX mode	320	32	0.58	18.56
Point 6 to 7	Switch from TX to RX	250	25	0.2	5
Point 7 to 8	Receiving MAC ACK from coordinator	235	23.5	0.35	8.225
Point 8 to 9	Radio in RX mode (processing MAC ACK and then waiting for the packet)	250	25	4.1915	104.7875
Point 9 to 10	Receiving Toggle command	230	23	1.2	27.6
Point 10 to 11	Switch from RX to TX	140	14	0.2	2.8
Point 11 to 12	Transmitting MAC Acknowledgement. Radio in TX mode	320	32	0.35	11.2
Point 12 to 13	Processing incoming Toggle command (for example, toggling the light)	75	7.5	7.8	58.5
Point 13 to 14	CMSA/CA algorithm. Radio in RX mode	270	27	0.955	25.785
Point 14 to 15	Switch from RX to TX	140	14	0.2	2.8
Point 15 to 16	Transmitting Default Response command. Radio in TX mode	320	32	1.22	39.04
Point 16 to 17	Switch from TX to RX	250	25	0.2	5
Point 17 to 18	Receiving MAC ACK from coordinator	230	23	0.35	8.05
Point 18 to 19	Radio remaining in RX mode and processing the MAC ACK	270	27	0.15	4.05
Point 19 to 20	Processing and shut down	75	7.5	2.5	18.75
After 20	Power Mode 2		0.001		
				23.865	**388.4335**

Table 2 – Current Consumption Measurement Breakdown on Operation2

6   Application to a Practical Use Case

In the discussions in Chapter 4 and 5, ZC switch and ZED light model was used for ease of measurement setup, excluding any possible CC2530-external current consumption. However the typical and realistic use case is a combination of a switch as a battery-powered sleeping ZED and a light as a mains-powered ZC or ZR. With this configuration, it is assumed that the ZED switch sends Toggle command and polls for receiving Default Response from the ZC light. Though the measurement on that configuration has not been done in this document, it can be calculated from the unit operation-wise results obtained in Chapter 5.

6.1   Power Consumption Estimation per Operation

6.1.1   Operation3 — Toggle Command TX

Transmission of Toggle a command has the same timing pattern as Table 1 except the unit operation of Point 5 to 6. The length of Point 5 to 6 can be replaced with that of Point 9 to 10 of Table 2. Consequently, the current consumption estimation of Operation3 will be as Table 3.

Section	Unit Operation Description	Voltage (mV)	Current (mA)	Time (ms)	Consumption (mA*ms)
Before 0	Power Mode 2		0.001		
Point 0 to 1	Power mode start-up sequence.	120	12	0.2	2.4
Point 1 to 2	MCU in active mode running on 16-MHz clock	60	6	0.25	1.5
Point 2 to 3	MCU running on 32-MHz clock	75	7.5	1.7	12.75
Point 3 to 4	CMSA/CA algorithm. Radio in RX mode	270	27	1.2025	32.4675
Point 4 to 5	Switch from RX to TX	140	14	0.2	2.8
Point 5 to 6	Transmitting Toggle command. Radio in TX mode	320	32	1.2	38.4
Point 6 to 7	Switch from TX to RX	250	25	0.2	5
Point 7 to 8	Receiving MAC ACK from coordinator	235	23.5	0.35	8.225
Point 8 to 9	Radio remaining in RX mode and processing the MAC ACK	250	25	0.15	3.75
Point 9 to 10	Processing and shut down	75	7.5	1.75	13.125
After 10	Power Mode 2		0.001		
				7.2025	**120.4175**

Table 3 – Current Consumption Estimation Breakdown on Operation3

6.1.2 Operation4 — Polling Followed by Default Response RX

Operations of polling and receiving a Default Response command have the same timing pattern as Point 0 to 12 of Table 2 except the unit operation of Point 9 to 10. The length of Point 9 to 10 can be replaced with that of Point 15 to 16 of Table 2. In addition to those, Operation4 will have the unit operation of processing and shutdown as Point 19 to 20 of Table 2. Consequently, the current consumption estimation of Operation4 will be as Table 4.

Section	Unit Operation Description	Voltage (mV)	Current (mA)	Time (ms)	Consumption (mA*ms)
Before 0	Power Mode 2		0.001		
Point 0 to 1	Power mode start-up sequence.	120	12	0.2	2.4
Point 1 to 2	MCU in active mode running on 16-MHz clock	60	6	0.25	1.5
Point 2 to 3	MCU running on 32-MHz clock	75	7.5	1.7	12.75
Point 3 to 4	CMSA/CA algorithm. Radio in RX mode	270	27	1.068	28.836
Point 4 to 5	Switch from RX to TX	140	14	0.2	2.8
Point 5 to 6	Transmitting MAC data request. Radio in TX mode	320	32	0.58	18.56
Point 6 to 7	Switch from TX to RX	250	25	0.2	5
Point 7 to 8	Receiving MAC ACK from coordinator	235	23.5	0.35	8.225
Point 8 to 9	Radio in RX mode (processing MAC ACK and then waiting for the packet)	250	25	4.1915	104.7875
Point 9 to 10	Receiving Default Response command	230	23	1.22	28.06
Point 10 to 11	Switch from RX to TX	140	14	0.2	2.8
Point 11 to 12	Transmitting MAC ACK. Radio in TX mode	320	32	0.35	11.2
Point 13 to 14	Processing and shut down	75	7.5	2.5	18.75
After 14	Power Mode 2		0.001		
				13.01	245.669

Table 4 – Current Consumption Estimation Breakdown on Operation4

6.2 Estimation for Usage Scenario

In this section, we estimate per-day current consumption amounts based on daily usage scenarios, and then calculate the battery life for each scenario. It is assumed that one AA battery with a capacity of 3000 mAh is used.

For ease of calculation, we define the constant CC_{SLEEP}, per-day current consumption in sleep (PM2) mode. Because the switch is supposed to sleep in most of time and the time period where it is awake is relatively negligible, assume the light is consuming sleep (PM2) current all the time. CC_{SLEEP} is calculated as following:

$$CC_{SLEEP} = 0.001 \text{ mA} \times 1000 \text{ ms} \times 60 \times 60 \times 24 = 86400 \text{ mAms}$$

Two example usage scenarios are considered to present how to estimate per-day current consumption in various actual usages.

- Usage Scenario 1
  In Scenario 1, assume the switch polls the light every 5 seconds and toggles the light 20 times a day. Assuming there is no communication failure, this scenario has (60 × 60 × 24 / 5 – 20) = 17260 times of Operation1, 20 times of Operation3, and 20 times of Operation4. Therefore the total per-day current consumption CC_{TOTAL} is calculated as:

$$CC_{TOTAL} = CC_{SLEEP} + CC_{Operation1} \times 17260 + CC_{Operation3} \times 20 + CC_{Operation4} \times 20$$
$$= 86400 + 100.5775 \times 17260 + 120.4175 \times 20 + 245.669 \times 20 \text{ (mAms)}$$
$$= 1829689 \text{ mAms} = 0.508 \text{ mAh}$$

And the battery life therefore can be calculated as:

$$\text{Battery Life} = 3000 / 0.508 = 5906 \text{ days} = 16.18 \text{ years}$$

- Usage Scenario 2
  In Scenario 2, assume the switch polls the light every second and toggles the light 10 times a day. Assuming there is no communication failure, this scenario has (60 × 60 × 24 / 1 – 10) = 86390 times of Operation1, 10 times of Operation3, and 10 times of Operation4. Therefore the total per-day current consumption CC_{TOTAL} is calculated as:

$$CC_{TOTAL} = CC_{SLEEP} + CC_{Operation1} \times 86390 + CC_{Operation3} \times 10 + CC_{Operation4} \times 10$$
$$= 86400 + 100.5775 \times 86390 + 120.4175 \times 10 + 245.669 \times 10 \text{ (mAms)}$$
$$= 8778951 \text{ mAms} = 2.439 \text{ mAh}$$

And the battery life therefore can be calculated as:

Battery Life = 3000 / 2.439 = 1230 days = 3.37 years

For more usage scenarios, the corresponding results are obtained using the Usage Scenario sheet of the accompanying spread sheet [15].

7   Conclusion

This document covers the basics of performing current consumption measurements using the Z-Stack and the CC2530. We measured current consumptions on two cases to obtain the current consumption of each unit operation, and then applied the result to practical usage scenarios.

Referring to this application note, one can make one's own measurement setup and take measurement. Also, one can calculate daily current consumption based on scenarios and finally estimate the battery life. All of these calculation and estimation formulas are provided in a spread sheet separately.

8 References

[1] <u>CC2530 Datasheet</u> (*swrs081b.pdf*)

[2] <u>CC2530 Product Page</u> (*http://www.ti.com/product/cc2530*)

[3] <u>TI's ZigBee Stack – Z-Stack</u> (*http://www.ti.com/tool/z-stack*)

[4] <u>Home Automation Sample Application description</u> (*Chapter 5 in the Z-Stack Sample Applications.pdf located at C:\Texas Instruments\ZStack-CC2530-2.5.1a\Documents*)

[5] <u>TI's ZigBee RF4CE Stack – RemoTI</u> (*http://www.ti.com/tool/remoti*)

[6] <u>AN073 - RemoTI™ Power Consumption</u> (*swra263a.pdf* and *swra263a.zip*)

[7] <u>CC253x/4x User's Guide</u> (*swru191c.pdf*)

[8] <u>Power Management For The CC2530.pdf</u> (*C:\Texas Instruments\ZStack-CC2530-2.5.1a\Documents*)

[9] <u>CC2530 ZigBee Development Kit</u> (*http://www.ti.com/tool/cc2530zdk*)

[10] <u>Texas Instruments Packet Sniffer</u> (*http://www.ti.com/tool/packet-sniffer*)

[11] <u>SmartRF05EB User's Guide</u> (*swru210a.pdf*)

[12] <u>Application Note AN057 - Measuring the Power Consumption on eZ430-RF2480</u> (*swra177.pdf*)

[13] <u>Z-Stack Developer's Guide.pdf pdf</u> (*C:\Texas Instruments\ZStack-CC2530-2.5.1a\Documents*)

[14] <u>Z-Stack User's Guide - CC2530DB.pdf pdf</u> (*C:\Texas Instruments\ZStack-CC2530-2.5.1a\Documents*)

[15] <u>CC2530_measurements_Z-Stack_2.5.1a.xls</u>

9 Document History

Revision	Date	Description/Changes
SWRA292	2012-09-25	Initial version.

IMPORTANT NOTICE

Products		Applications	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		