Measuring the Power Consumption on eZ430-RF2480

By Magnus Wines & Morten Braathen

Keywords

- *ZigBee*
- *ZigBee Coprocessor*
- *ZigBee Accelerator*
- *CC2480*
- *eZ430-RF2480*
- *MSP430*
- *Application Acknowledgement*
- *Periodic Transmission*
- *Current Consumption*
- *Battery Lifetime*

1 Introduction

CC2480 is a ZigBee network coprocessor that greatly simplifies getting up and running with ZigBee. The remote procedure call interface of the CC2480, running over a serial interface, provides a powerful and simple way of controlling the ZigBee device. With just a few commands required to operate the chip, the developer can stay focused on the application rather than the network protocol.

eZ430-RF2480 is a miniaturized hardware platform demonstrating ZigBee embedded networking and low power consumption for a node with an MSP430F2274 and a CC2480.

The MSP430 runs the ZigBee Accelerator Sample Application (ZASA) which implements a manufacturer specific ZigBee application profile with two end points: Source and Sink. The Sink is mapped to the logical device type Coordinator and the Source is mapped to the logical device type End Device or Router. The Sink receives temperature and voltage samples sent periodically from the Sources. In between the data transmissions, the End Device goes to sleep to save power.

This document will present power consumption measurements and lifetime calculations based on the eZ430-RF2480, where the CC2480 is configured as an End Device. We will also discuss some of the factors and configuration options in ZASA that influence the battery lifetime of the system.

All of the measurements are done on the eZ430-RF2480, with only one Sink and one Source in the ZigBee network.

Note that there are many factors that influence the overall power consumption and that the results presented in this document should only be regarded as indicative for what is possible to achieve in systems with similar hardware.

Table of Contents

2 Abbreviations and Acronyms

ACK	Acknowledgement
ADC	Analog to Digital Converter
APS	Application Support Sub-Layer
CSMA-CA	Carrier Sense Multiple Access, Collision Avoidance
DCO	Digitally Controlled Oscillator
GPIO	General Purpose IO
IO	Input Output
MAC	Medium Access
RX	Receive
TX	Transmit
USB	Universal Serial Bus
VCP	Virtual Communication Port
VLO	Very Low Power Oscillator
ZASA	ZigBee Accelerator Sample Application
I	Current (Ampere, A)
U	Voltage (Volt, V)
R	Resistance (Ohm, Ω)
mA	Milliampere (10e-3 A)
uA	Microampere (10e-6 A)
nA	Nanoampere (10e-9 A)

3 System Overview

3.1 CC2480

The CC2480 is a ZigBee Network (co) Processor. It has built in the complete TI ZigBee Stack on chip and can be operated as a smart transceiver using a highly efficient remote procedure call interface. This allows the software designer to focus on the Network Application rather than the ZigBee stack. CC2480 is using the ZigBee 2006 stack profile.

3.2 CC2480 Target Board

The CC2480 Target Board is part of the low cost eZ430-RF2480 Demonstration Kit. The kit is used to demonstrate the ZigBee processor in combination with an MSP430F2274 acting as the application processor.

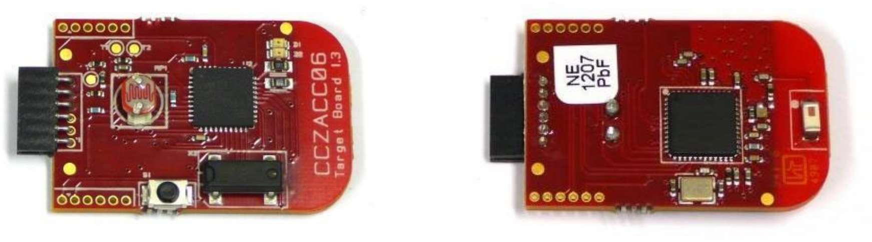


Figure 1 - CC2480 Target Board, front and rear

The eZ430-RF2480 platform provides debugging capabilities of the MSP430 and a Virtual Communication Port (VCP) over the USB interface. The key features of the platform are:

- MSP430 debugging capability
- Virtual serial communications port over USB
- Exposed MSP430 General Purpose IO for peripheral connectivity
- Exposed CC2480 signaling for monitoring purposes.
- User Interface capabilities (LEDs, button and light sensor)

The CC2480 Target Board can either be connected to the USB emulator (for programming and UART communication) or it can be connected to the supplied battery boards.

For more specific information, please refer to the eZ430-RF2480 User Guide.

3.3 Software

The software used in the following experiments and analyses is available in source code on the eZ430-RF2480 product web page. For an in-depth description of the software functionality, please consult the eZ430-RF2480 User's Guide.

The software can be modified and debugged using the free version of IAR Embedded Workbench for MSP430, also known as the Kickstart version.

4 Measurement Setup and Theory

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. The set up is illustrated in the figure below.

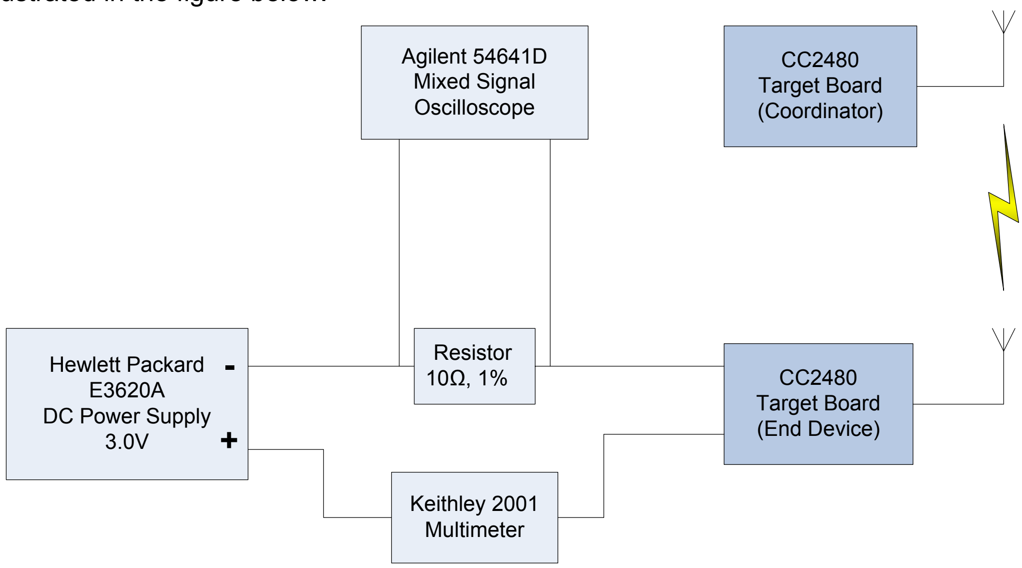


Figure 2 - Measurement Setup

The oscilloscope will provide a graphical representation of the voltage drop over the resistor. Since there is a linear relationship between the voltage and current, the same graphical representation will illustrate the current consumed by the system.

In addition to the oscilloscope, we will use a multi meter to measure e.g. the current during sleep and other static states. The multi meter is connected to the CC2480 Target Board such that we can measure the current drawn by the MSP430 alone, the CC2480 alone or both.

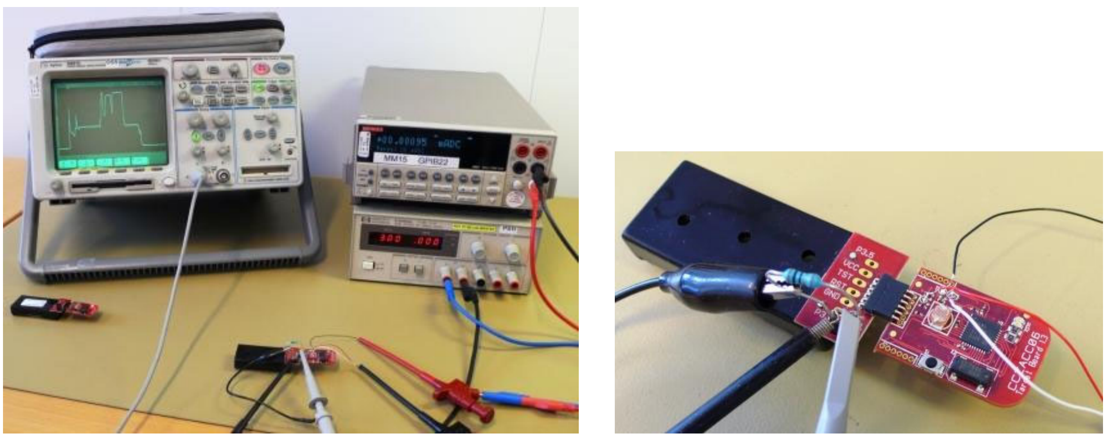


Figure 3 - Picture of test setup and instrumentation

4.2 Calculation of the average current consumption

The calculation of the current is based on the well known relation

U = RI

Where U is the voltage, R is the resistance and I is the current. By measuring at the power supply side, the test system will observe the current consumed by the complete target board.

Note that R should not be too large, since it will reduce the effective voltage over the target board itself, i.e.

$U_{TARGET\ BOARD} = U_{POWER\ SUPPLY} - RI$

As long as I is just in the range of some tens of mA and R is relatively small, the overall effect of the voltage drop is negligible. Generally, we can say that

$U_{POWER\ SUPPLY} \approx U_{TARGET\ BOARD}$ if and only if $RI \ll U_{POWER\ SUPPLY}$

$RI \ll U_{POWER\ SUPPLY}$ if and only if $U_{POWER\ SUPPLY} > 10 \times RI$.

Using R = 10Ω we will be able to say that the above is true as long as I < 30 mA. In the subsequent sections, we will see that we are pushing the limit saying that the effect of R is negligible, since the peak current sometimes exceeds 30 mA. However, the current consumption of both the MSP430 and CC2480 are almost independent of the input voltage. In addition, if we choose a smaller value for R, the oscilloscope would need to capture accurately even lower voltage levels and the uncertainty in the measured values would increase. In order to keep the measurement system simple and easy to reproduce, we will accept the error introduced by the resistor.

Once the current is determined, we can proceed by finding the overall average current consumption. The general formula below can be used.

$$I_{avg} = \sum_{i=0}^{n}\left(\frac{d_i}{P_i}\cdot I_i\right) + \left(1 - \sum_{i=0}^{n}\left(\frac{d_i}{P_i}\right)\right)\cdot I_{SLEEP} \qquad \text{(EQ 1)}$$

i	This is the id of an active sequence
d_i	This is the duration of i
P_i	This is the period of i
I_i	This is the average current consumption during i

Knowing I_i, d_i and I_{SLEEP}, we can find I_{avg} based on the period of the active sequences.

As a final step, we can calculate the total lifetime of the system, knowing that

$$\textit{Battery Capacity}\ [mAh]\ /\ \textit{Average Current}\ [mA] = \textit{Lifetime}\ [h] \qquad \text{(EQ 2)}$$

The battery capacity will differ from one battery type to another. For normal AAA batteries available in stores, the capacity is typically 1200 mAh. Using AA batteries, capacities of up to 3000 mAh is possible.

In the following chapters, we will assume that two AAA batteries are used.

4.3 Accuracy

To get a picture of the accuracy of the measurements, the system was set up to switch between three states consuming different amount of current. An ampere meter was used to measure the static current draw in the states. The oscilloscope was used to measure the amplitude or difference between the states.

The CC2480 Target Board was programmed such that the CC2480 was set in Low Power Mode 3. The MSP430 toggled either the RED LED (State 2) or both LEDS (State 3) once every 5 seconds, using the VLO as clock source for an internal timer. The MSP enters LPM3 after setting the LEDs.

With a calibrated ampere meter, we measured:

State 1: 655 nA
State 2: 3.316 mA
State 3: 5.835 mA

Using the Oscilloscope, the voltage difference between the two states, averaging over 8 samples, was measured to be

Amplitude 1: 36.2 mV
Amplitude 2: 63.7 mV

Knowing that the voltage measurements were done over a 10 Ohm resistor with an accuracy of 1%, we can calculate the current. The table below summarizes the calculations.

Description	Ampere meter	Oscilloscope	Diff MAX	Diff MIN	% MAX	% MIN
State 1	655 nA	-	-	-	-	-
State 2	3.316 mA	3.62 +/- 0.036 mA	0.340 mA	0.268 mA	10.3 %	8.1 %
State 3	5.835 mA	6.37 +/- 0.064 mA	0.599 mA	0.471 mA	10.3 %	8.1 %

We see there is an offset in the range of 8.1% to 10.3% for the values captured with the oscilloscope compared to the ampere meter. In the following paragraphs, we will take this into account. However, when dimensioning a system, it is always a good rule of thumb to add some slack to make sure that the values used will meet or exceed the final requirements. Thus, we have rounded off many of the values to get approximate, yet representative numbers.

5 Current Consumption Measurements

5.1 Default Configuration of ZASA

The default configuration of the End Device, operating as the data source in the sample application, is as follows:

Behavior	Corresponding Constant	Value
No blinking of LEDs	APP_BLINK_LEDS	FALSE
Application layer acknowledgement is turned off	APP_DATA_CNF	FALSE
Send an application layer packet every 10 seconds	APP_REPORT_INTERVAL	10
Data polling is turned off	APP_POLL_INTERVAL	0

When the nodes are up and running in the small network, a packet sniffer can be used to visualize the packets going over the air. The figure below shows two data packets being sent. Each packet is acknowledged with a MAC ACK packet.

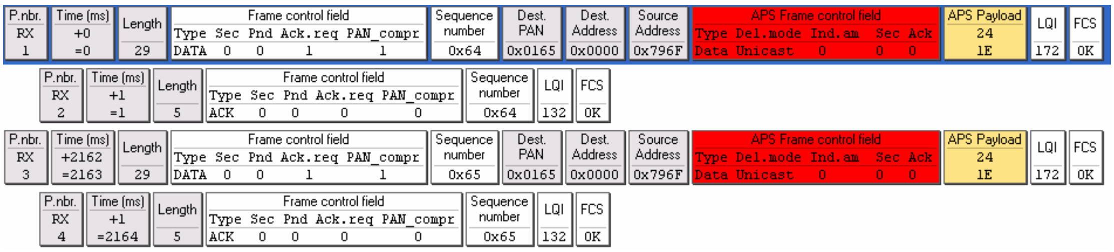


Figure 4 - Transmission of data without Application Acknowledgement (2 packets)

Using the oscilloscope, we can get a picture of the dynamic power consumption of the system during transmission of one of these packets.

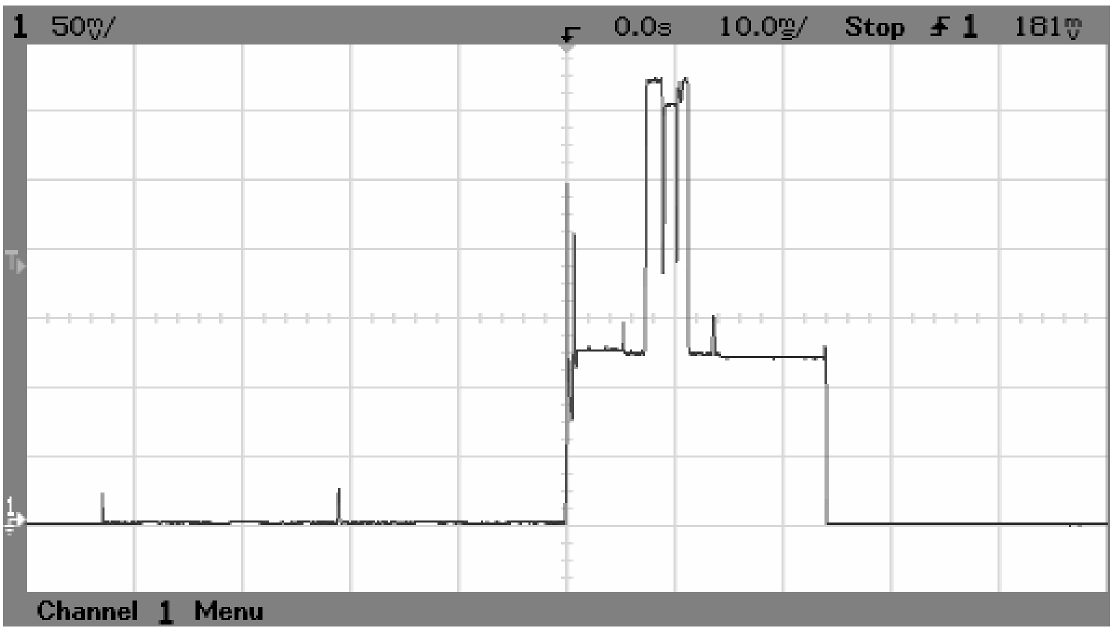


Figure 5 - Transmission of data without Application Acknowledgement

We see several events in the plot above.

The two small peaks before the large peak in the centre of the figure is the MSP430 waking up, starting the ADC for doing some measurements. First, the temperature is measured. Next the on-board voltage level is measured. The time between the measurements is determined by the worst case start-up and settling time of the reference voltage for the ADC. This has been set to 20 ms in ZASA. The measurement itself only takes a couple of milliseconds. Note that once the ADC is turned on, the MSP430 core will be shut off, reducing power consumption to a minimum. The average current consumption during the 44 milliseconds used for measurements can be determined based on numbers from the MSP430F2274 datasheet.

The large peaks in the centre of the picture represent the CC2480 waking up, starting its local oscillators, crystals and timers. Next, the packet is prepared and sent over the air. At the end, the CC2480 prepares for going back down to deep sleep.

The major events, with approximate current and duration based on averaging values from several measurements, can be summarized in the table below. All of the numbers are typical. Some of the events, like CSMA-CA at the beginning of phase 3, might take a very short time (down to 320 us) or a long time (several tens of ms) - depending on how much traffic there is on the selected channel. For details, see chapter 7.

Table 1 - Measured Current Consumption during TX without APS ACK (typical values)

Id	Description	Duration [ms]	Current [mA]	[ms*mA]
1	ADC conversion (average for two samples, including wake up)	44.0	0.170	7.48
2	Synchronizing clocks and processing of data packet	8.0	12.0	96.0
3	CSMA-CA (RX), transmit packet and receive MAC ACK	3.5	30.5	106.75
4	Post processing and preparing for deep sleep	11.0	12.0	132.0
	Sum	**66.5**		**342.2**
	Average		**5.15**	

To calculate the average current consumption for the system over time, we also need to find the current consumption when the system is in deep sleep mode. Using an ampere meter, the sleep current of the system was measured to be 655 nA.

We can now proceed and find the total average current consumption, based on (EQ1). Substituting in the formula, we get

(66.5/10000)*5.15 + (1 – (66.5/10000))*0.00066 mA = **0.0349 mA**

(EQ2) can now be used to calculate the expected lifetime of the system:

1200 mAh / 0.0349 mA = 34384 h = **1433 days**

We have just seen that if the end device is configured to transmit one packet every 10 seconds, with no application acknowledgement and no data polling, the board can operate for 1433 days (3.9 years) with two AAA batteries.

In the following sections we will see how the various configuration options in ZASA influence the power consumption.

5.2 Application Acknowledgment

Depending on the reliability requirements to the system, application acknowledgement (APS ACK) can be enabled or not. An APS ACK is the reply from the receiver of an application layer packet, saying that the packet was received by the recipient with no errors. Using APS ACK is the only way a ZigBee node can make sure that the packet it sent was received by the intended recipient. The MAC ACK, on the other hand, only indicates that the raw MAC packet was received correctly by a node one hop away in the network. In alarm and security systems or sensor networks monitoring critical parameters, it might be crucial that all transmitted packets are received by the recipient. On the other hand, in non-critical systems, losing a packet might not matter.

ZASA has the option to turn on and off application acknowledgement with the constant APP_DATA_CNF in sample_app.c. Setting it to TRUE will turn APS ACK on.

Note that, in ZASA, setting APP_DATA_CNF to true will also enable data polling. Polling can be turned off by setting APP_POLL_INTERVAL to 0.

In the following, we will see the effect this has on the total system power consumption.

In chapter 5, we discussed the current consumption of the system when application acknowledgement was turned off. Turning it on will change things quite a bit, since the active period will look quite different. In this case, the end device will have to request the acknowledgement packet from the receiver after having transmitted the packet. It must also wait to send the request until the recipient has actually received the packet. The time to wait will depend heavily on the system and number of hops. The default wait time is set to 100ms in ZASA. After receiving the acknowledgement (as a response to the request), the end device will send a last data request in order to check whether additional packets are on its way to the node. If no packets are received, the device will enter deep sleep.

The sequence described above can be visualized in a packet sniffer by capturing the packets going over the air. As many as 8 packets are involved.

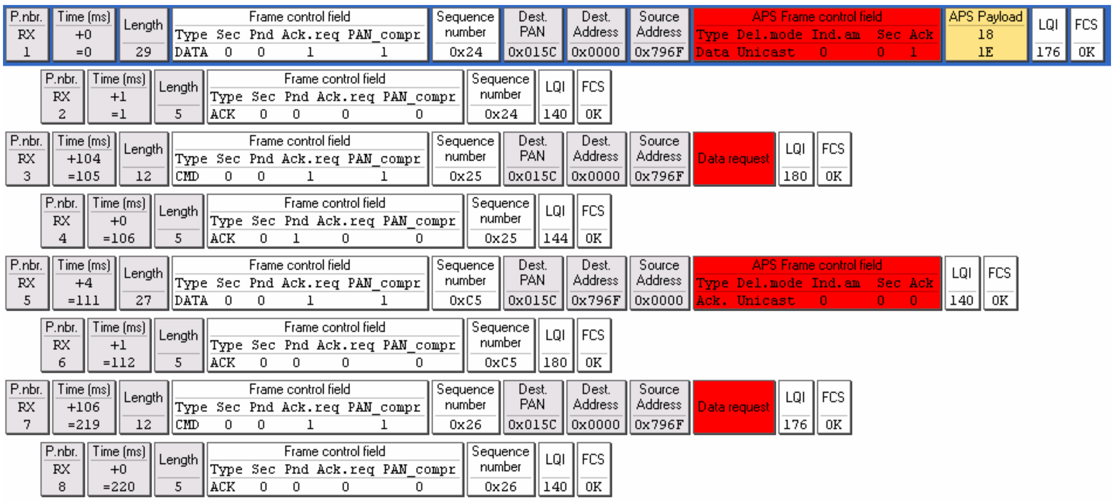


Figure 6 - Transmission of data with Application Acknowledgement (1 packet)

The current consumption profile of the sequence is shown in the figure below.

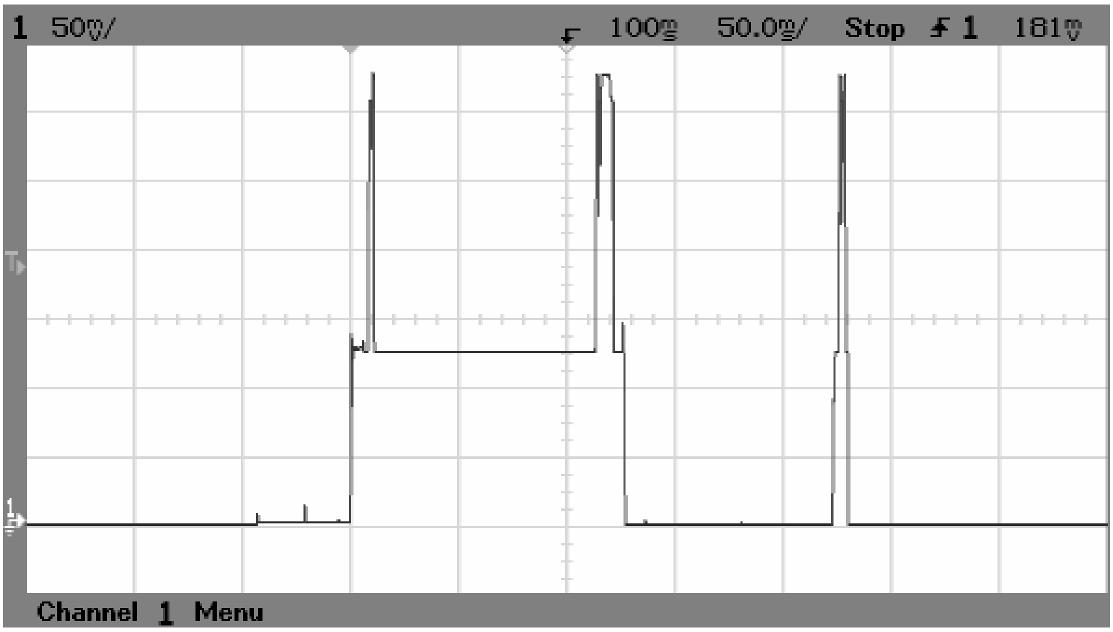


Figure 7 - Transmission of data with Application Acknowledgement

Based on the graph above, we can recalculate the average current consumption for the active period. Using a simplified approach, we can divide the sequence in different phases and measure the approximate current and duration for each phase. A more detailed description can be found in chapter 7.

Table 2 - Measured Current Consumption during TX with APS ACK (typical values)

Id	Description	Duration [ms]	Current [mA]	[ms*mA]
1	Acquire sample (ADC on)	44.0	0.170	7.48
2	Startup	8.0	12.0	96.0
3	Transmit packet	3.5	30.5	106.75
4	Wait before requesting APS ACK	100.0	12.0	1200.0
5	Request and receive APS ACK	9.0	30.5	274.5
6	Post processing of packet	4.5	12.0	54.0
7	Enter LPM2	100.0	0.00097	0.097
8	Request data (single poll)[1]	7.7	20.4	157.2
	Sum	**276.7**		**1896.0**
	Average Current		**6.85**	

The sleep current in this case will still be 655 nA. Using (EQ1), we can find the average current consumption:

(276.7/10000)*6.85 mA + (1 – (276.7/10000))*0.00066 mA = **0.190 mA**

The lifetime of the device is now

1200 mAh / 0.190 mA = 6316 h = **263 days**

[1] See chapter 5.3

The value above can be compared with the 1433 days when APS ACK is turned off.

Note that in some scenarios, it might make sense to turn on APS ACK just for every n^{th} packet. Then the node will make sure that the packet goes all the way through the network, and the node can realign with the network if some of the routers or devices in the network are down.

Based on EQ1 and the measurements from chapter 5, it is relatively easy to calculate the effect of only using application acknowledgement for a few packets. As an example, we will do the calculation for a system where every 10^{th} packet requires an acknowledgement. We get

	(1/10)*(276.7/10000)*6.85 mA	// TX with APS ACK
+	(9/10)*(66.5/10000)*5.15 mA	// TX without APS ACK
+	(1 – (1/10)*(276.7/10000) – (9/10)*(66.5/10000))*0.00066 mA	
=	**0.050 mA**	

The expected lifetime is now

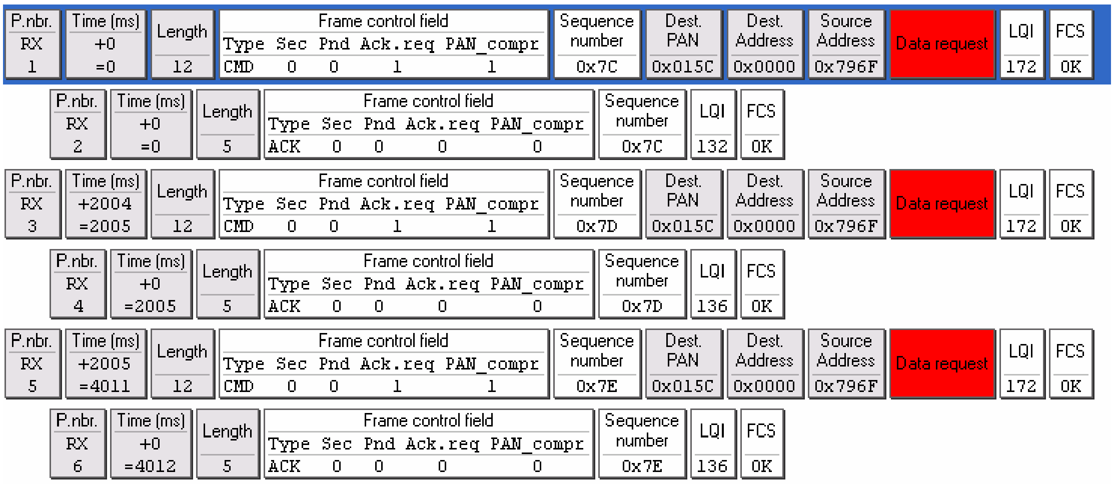
1200 mAh / 0.050 mA = 24000 h = **1000 days**

5.3 Data polling

The only way an end device will be able to receive data from another node is to periodically send data requests to the associated device. If there are packets destined for the end device, the packet can be sent once the end device asks for them. In some systems, where the end device is never expected to receive any application data, polling can be disabled.

CC2480 can be configured to automatically poll for data (sending data requests) without involving the host MCU. The MCU will only be notified when data is received. In ZASA, turn polling on by setting APP_POLL_INTERVAL to a non-zero value. The value given will be the poll interval in milliseconds. Setting it to 0 will turn polling off.

The figure below shows the data request packets from the packet sniffer.



P.nbr.	Time (ms)	Length	Frame control field Type	Sec	Pnd	Ack.req	PAN_compr	Sequence number	Dest. PAN	Dest. Address	Source Address		LQI	FCS
RX 1	+0 =0	12	CMD	0	0	1	1	0x7C	0x015C	0x0000	0x796F	Data request	172	OK
RX 2	+0 =0	5	ACK	0	0	0	0	0x7C					132	OK
RX 3	+2004 =2005	12	CMD	0	0	1	1	0x7D	0x015C	0x0000	0x796F	Data request	172	OK
RX 4	+0 =2005	5	ACK	0	0	0	0	0x7D					136	OK
RX 5	+2005 =4011	12	CMD	0	0	1	1	0x7E	0x015C	0x0000	0x796F	Data request	172	OK
RX 6	+0 =4012	5	ACK	0	0	0	0	0x7E					136	OK

Figure 8 - Periodic Data Requests (three data requests)

The active sequence of the data request is represented with the following current profile on the oscilloscope:

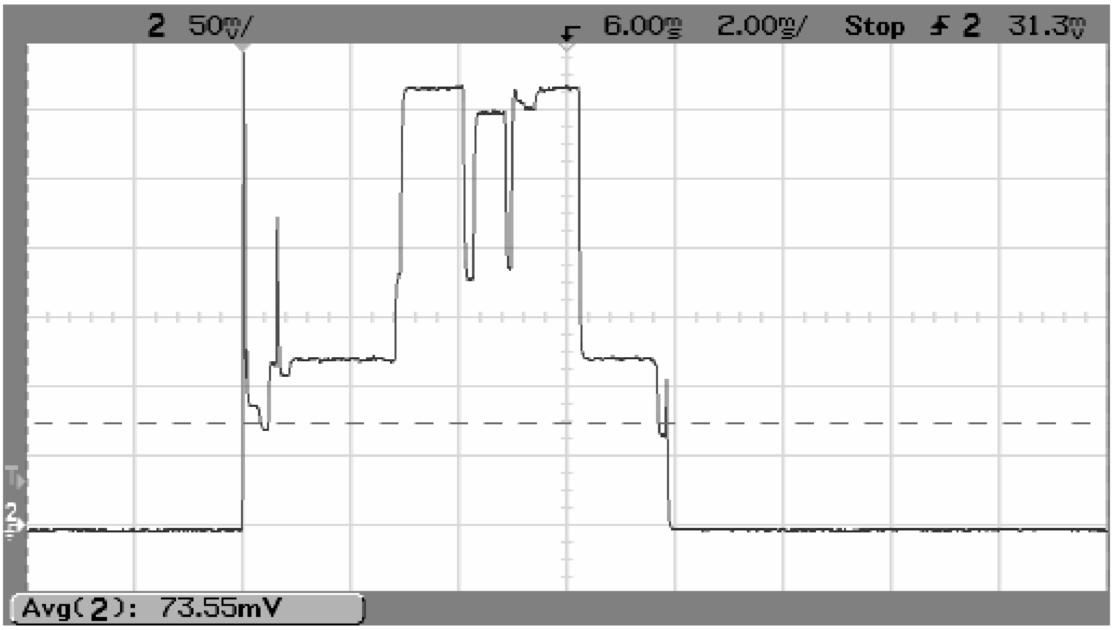


Figure 9 - Periodic Data Request

Looking at multiple sequences of the data request, we can measure the approximate current and duration for the major events. Once again, the numbers are based on averaging results from several measurements. See further details in chapter 7.

Table 3 - Measured Current Consumption during Data Poll (typical values)

Id	Description	Duration [ms]	Current [mA]	[ms*mA]
1	Wake up	2.6	12.0	31.2
2	CSMA-CA (RX), transmit data request and receive MAC ACK.	3.5	30.5	106.75
3	Prepare for sleep (LPM2)	1.6	12.0	19.2
	Sum	**7.7**		**157.2**
	Average		**20.4**	

In this case, the sleep current was measured to be 970 nA. The increase in the sleep current is caused by a timer on the CC2480 that has to be up and running to handle the timing of the periodic data requests.

Given a polling period, we can calculate the average current consumption using EQ1. For this example, we set the poll period to 1 second (1000 ms). Thus:

(7.7/1000)*20.4 mA + (1 – (7.7/1000))*0.00097 mA = **0.158 mA**

The lifetime of the device is now

1200 mAh / 0.158 mA = 7595 h = **316 days**

The calculations above do not take into account the scenario where the polling device actually receives a packet. We could say that we briefly touched the subject for the end device requesting an application acknowledgement. That scenario is a bit different, though, since the recipient of the APS ACK will always end the transaction with a MAC ACK. For the general case, the recipient would have to parse the packet and figure out whether it should prepare an APS ACK and send back to the original transmitter. Only if no APS ACK is required, the device can go to sleep after sending the MAC ACK.

Thus, as for the transmitter, we would have to investigate the three different cases

1. Data request with no data
2. Data request with data, no APS ACK required
3. Data request with data, reply with APS ACK

The idea of ZASA is that the end devices just transmits data and never really receives anything other than the MAC ACK and APS ACK if enabled. As a consequence, we will not investigate case 2 and 3 above.

For additional information about a polling end device in a ZigBee network, please consult the document “AN053 – Measuring Power Consumption with CC2430 and Z-Stack” (swra144).

5.4 Combining Transmission and Polling

In the previous sections, we have looked at the average current consumption for three different scenarios:

1. Transmission of data without data acknowledgement
2. Transmission of data with data acknowledgement
3. Automatic data requests

We have also seen that the sleep current will be different depending on whether polling is enabled or not.

To get the full picture of the current consumption, we can now combine the various configurations and tune the system to get the required lifetime. Using the numbers from the previous paragraphs, we can give a graphical representation of the system lifetime as a function of the transmit period and poll period.

In Figure 10 below, we have shown the difference between two systems, where one is always using data acknowledgement, whereas the other does not. By combining ACK on and off for every n[th] packet, any lifetime between the two graphs can be achieved.

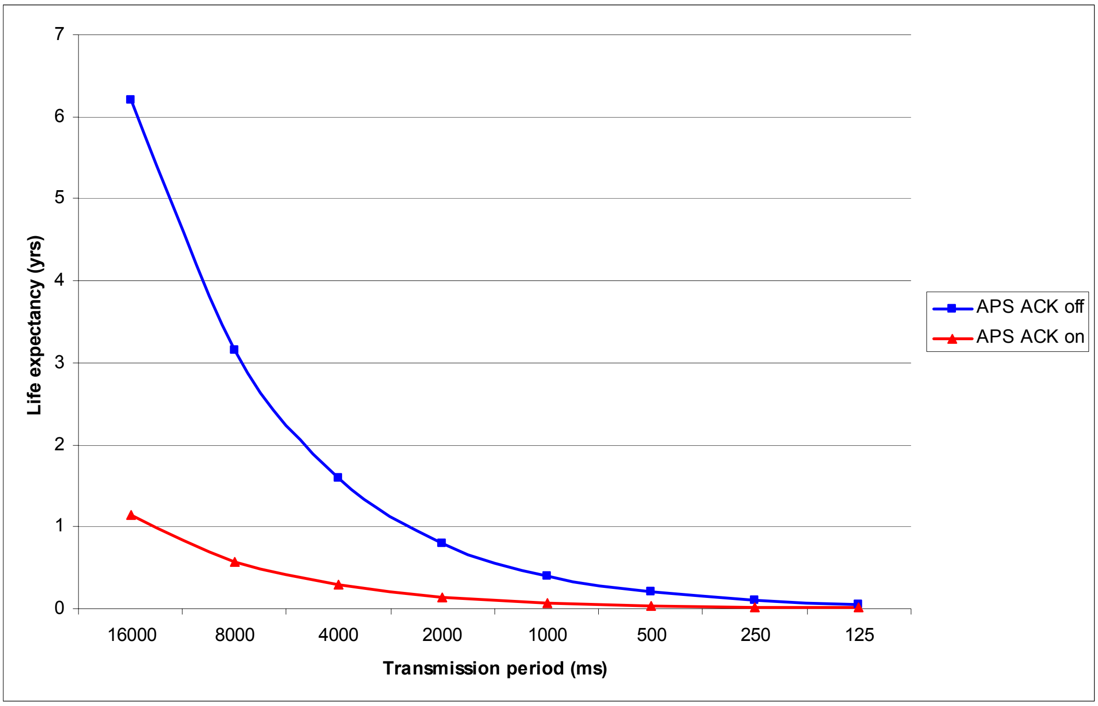


Figure 10 - Years of Operation vs. Transmission Period

In Figure 11, we have shown how the two parameters poll period and transmission period together give the total lifetime of the system. In the example, we use numbers for transmission where application acknowledgement is turned off.

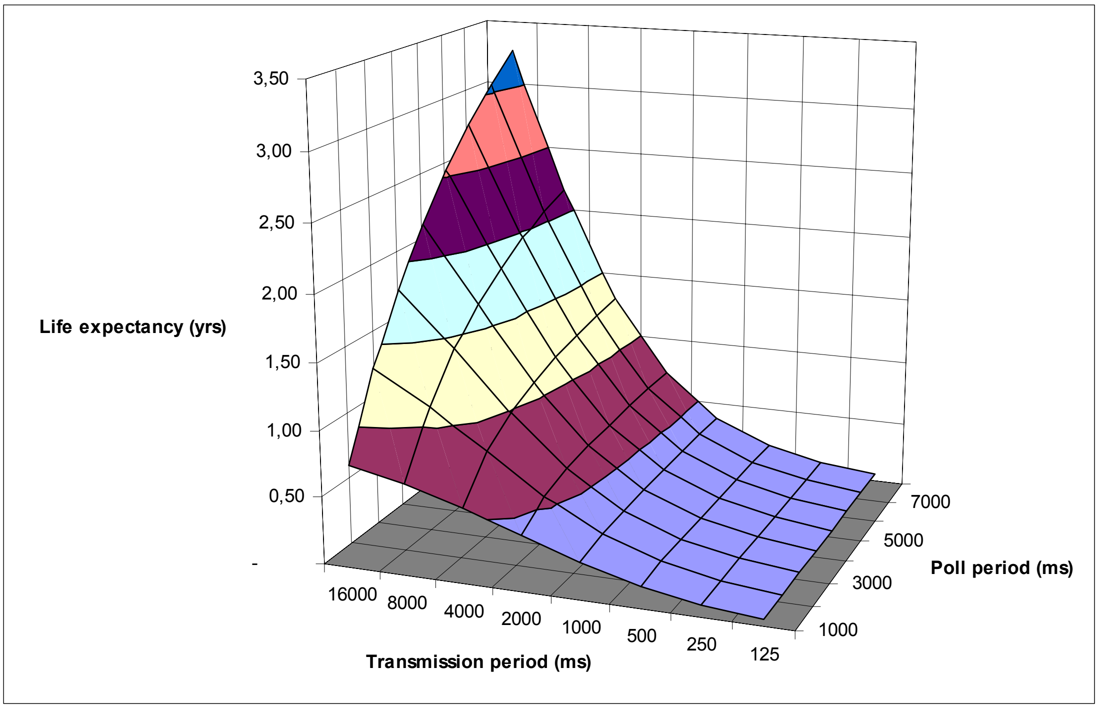


Figure 11 - Years of Operation vs. Transmission and Poll Period

From the figure, we see for instance, that if the required lifetime of the system is 1 year, it is possible to set it up to transmit a packet every 4 seconds and to poll every 3 seconds. By reducing the transmit rate to every 8 seconds and the poll rate to every 6 seconds, the lifetime is almost doubled.

5.5 Retransmission of packets

Retransmission of packets that are not being acknowledged, either on the MAC or the APS layer, will have a direct impact on the battery lifetime. There are many reasons for losing packets: noise or interference on the RF channel, poor RF design reducing sensitivity or stability of the radio, operating close to sensitivity limit of radio (out of range), not proper operation of the application etc.

It is possible to do calculations similar to the ones in section 5.2, looking at the probability of packet loss and the number of retries. To some extent, it can be compared to increasing the overall packet rate. However, the time the node needs to stay in RX waiting for the ACK to arrive, must be taken into account.

As a consequence, the following parameters must be determined for a full analysis:

1. Probability of packet loss
2. Timeout waiting for missing MAC ACK
3. Timeout waiting for missing APS ACK
4. Number of retries
5. Retry period

This will not be discussed further in this application note.

6 Other contributors to the total current consumption

We have currently only looked at how we can change the current consumption by modifying the functional configuration of the end device. However, there are some other elements in the system that should be taken into account.

6.1 Battery type

As briefly mentioned at the end of chapter 4, the battery type will have a direct impact on the lifetime of the system. Typical, low cost household batteries are AA and AAA Alkaline batteries. The average capacity of one AA battery is 2500 mAh and 1200 mAh for one AAA battery. The capacity determines for how many hours the battery will be able to supply a specific current. If your application drains 1 mA in average, the system can stay alive for 1200 hours using AAA batteries. The lifetime is doubled when using AA batteries.

Note that there are many other criteria than the capacity that should be carefully considered before selecting a battery type. Some of them are:

- Voltage
- Peak current
- Rechargeable
- Price
- Availability
- Size
- Material
- Self discharge current
- Environmentally friendly

6.2 Taking advantage of the MSP430

The MSP430F2274 used on the CC2480 Target Board is a good fit for the CC2480. The MCU has a very short wake-up time and both the operating current and sleep current are low. Using the many low power features of the MSP430, it is possible to develop systems that consumes just a fraction of the current other MCUs would consume.

It is still important that the MSP430 is used correctly in order to ensure as low average current consumption as possible.

Since the CC2480 Target Board does not have any 32 kHz crystals connected to the MSP430, there are no traditional clock sources that can be used for an ultra low power sleep timer. For many systems, the only solution would be to use the internal high speed RC oscillator of the MCU as clock source. The current consumption in such “sleep” modes can be several hundred uA – wasting a lot of precious power when the system is idle.

The MSP430F2274 has a very low power, low frequency oscillator called the VLO. It is operational down to low power mode 3 and draws only a couple of hundred nA. By calibrating the VLO with the DCO during system initialization, the accuracy can be improved significantly. The VLO can then be used as the clock source for a sleep timer if the requirements to accuracy are within the limits possible to achieve using the VLO.

Thus, even without a crystal, the system can perform periodic tasks and have a sleep current of just a few hundred nA. Using an ampere meter, we measured the sleep current on the MSP430F2274 in LPM3 with the VLO enabled to be 316 nA.

Another parameter to have in mind is the operating frequency of the MCU core when it is active. The MSP430 makes it easy to set up the DCO to any frequency up to 16 MHz. However, the current consumption increases as the frequency increases. The system should be carefully designed such that the tradeoff between speed and current consumption is taken into account. The slower the clock, the longer time the transactions take and the longer the MSP430 has to stay awake. However, the higher the clock rate, the more energy you burn when the system is active.

For the application running on the eZ430-RF2480, having a high clock rate makes sense, since there are a lot of data being transferred over the SPI bus between the CC2480 and the MSP430. For both parties, it is beneficial that these transactions (or really remote procedure calls) are effectuated as quickly as possible, such that each party can go to sleep immediately when its part of the transaction is finished. Thus, ZASA has been set up to operate at 8 MHz in active mode.

Note that the selected DCO frequency sets a limit to the lowest operating voltage of the system. To safely operate at 8 MHz, the on board voltage has to be well above 2.2V. As a consequence, ZASA will measure the on board voltage before setting up the DCO. If the voltage is below the given threshold, the standard DCO frequency of 1 MHz is used.

The software should in addition take advantage of the fast transition times of the MSP430 from sleep to active mode. Whenever the software is waiting for an event, the software should implement delay routines that sets the MCU in deep sleep and have interrupts to wake up the system - either by using timed periodic polling or by having a hardware trigger on the actual event. The use of spinning loops should be limited to extremely short delays (several microseconds or less).

In addition to the clock, it is also important to make sure that all peripherals and GPIOs on the MSP430 are configured correctly to reduce the current consumption. Most peripherals are automatically turned off when the MSP430 enters a low power mode, but some things must be turned off by the user. For instance, the internal reference voltage for the ADC must be turned off by writing to the appropriate register when the ADC has been disabled.

The GPIOs can cause extensive current leakage if not set correctly. A typical example would be the case where one GPIO is configured as an input and the external line is floating, causing a leakage of potentially several hundred uA on that IO pad. If two signals are driving against each other (both are set as outputs), the current draw can be several mA.

It is recommended to go over the schematics of the system carefully to make sure that all of the pins on the MSP430 are configured for the lowest possible current consumption.

7 Deciphering the current consumption profile

The figures above only used average values and a simplified approach to calculate the average current consumption during an active period of the device. In this chapter, we will go into the details and explain what is happening on the device during a data transmission when application acknowledgment is enabled

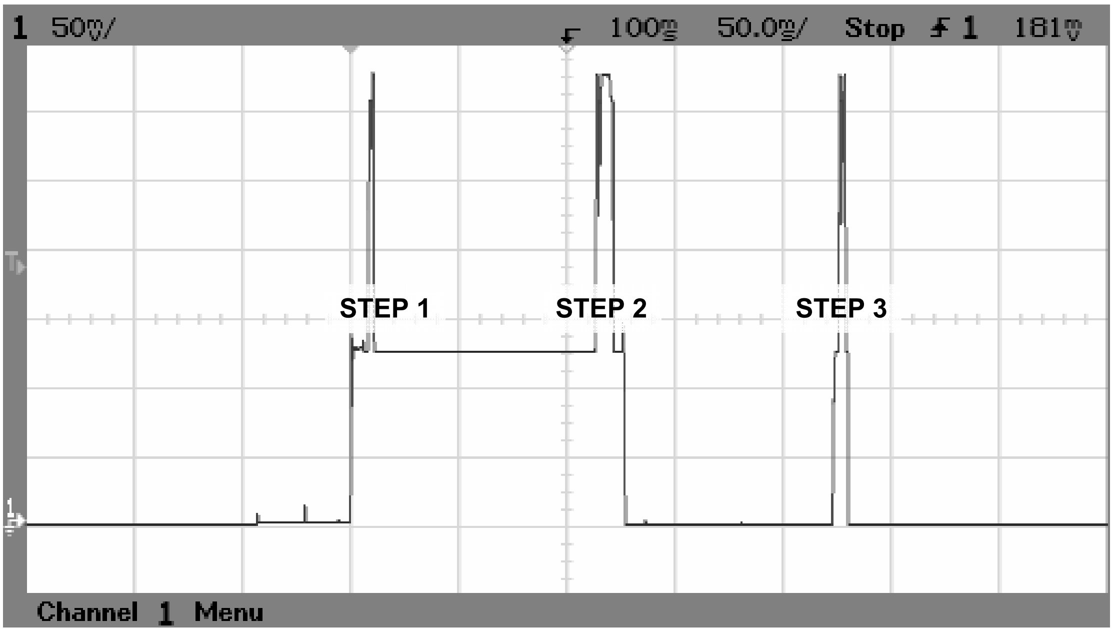


Figure 12 - Transmission with APS ACK enabled

The figure above gives the overview of the events during the transmission, but it is difficult to see and understand what is happening. By zooming in on the three peaks, it is easier to see and understand what is happening.

The three subsequent chapters will give a detailed description of each of the three steps.

> Please note that the values shown for the currents and durations are based on approximate measurements using the oscilloscope and to some extent a multi meter. The average value for several measurements has been used.
>
> The values should only be regarded as indicative for the given hardware and software.

7.1 Step 1 – Transmit the Packet

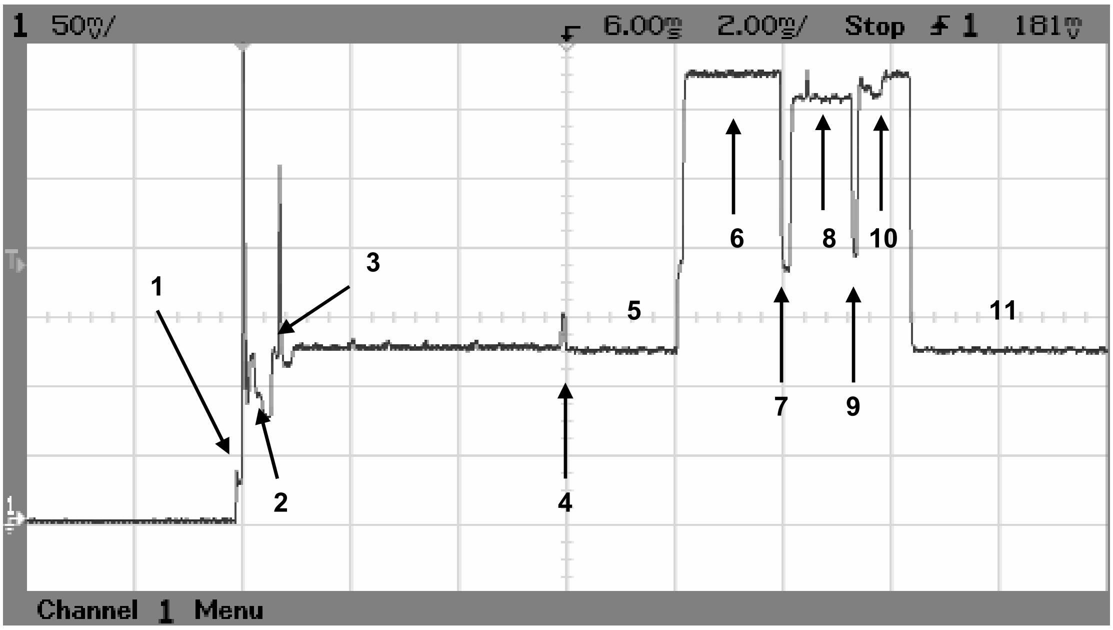


Figure 13 - Transmit the Packet

Event	Description	[ms]	[mA]
1	The MSP430 has finished the measurements and wakes up CC2480 to send the packet. The initial peak is caused by the startup of the internal 16 MHz RC oscillator and the 32 kHz crystal.	0.7	7.0
2	The MSP430 sends the data and appropriate command over to the CC2480 so that it can start the transmission. The MSP430 will immediately go to sleep, waiting for the CC2480 to signal that it has processed the command.	0.1	12.0
3	CC2480 starts the 32 MHz crystal and sets it up as the core clock. The command from the MSP430 is processed and the CC2480 starts synchronizing an internal timer in order to be able to transmit the packet.	5.5	12.0
4	CC2480 wakes up the MSP430, such that it can read out the return value for the command that was invoked. The MSP has finished the transmission step and goes back to sleep.	0.1	14.7
5	CC2480 sets up the radio and churns the packet through the ZigBee stack, preparing it for transmission.	2.0	12.0
6	CC2480 starts the Carrier Sense Multiple Access Collision Avoidance (CSMA-CA) algorithm. The radio is set in RX and CC2480 will assess whether the channel is clear for transmission. The duration of this step will vary if the channel is noisy. This is due to the random interval the device will defer from sending when the medium is sensed to be busy or noisy. The average duration over 20 observations is approximately 1 millisecond. The duration in the plot is 2 milliseconds.	1.0	31.0
7	Switch from RX to TX.	0.2	18.0
8	The packet is sent over the air.	1.0	29.0
9	Switch from TX to RX.	0.2	18.0
10	CC2480 receives the MAC ACK from the associated device in the network, indicating that the packet was received with no errors. Note that this does not mean that the application data was received by the intended recipient, only that the packet was transmitted and acknowledged on the MAC layer by the network node that was associated with the transmitter.	1.0	31.0
11	CC2480 enters an IDLE state, waiting to request the APS ACK from the recipient. The duration of this step will depend on the number of nodes and how many hops there are between the sender and transmitter. ZASA sets this duration to 100ms. **Note:** The normal case would be that CC2480 enters low power mode 2 in this state. Sometimes, however, it only enters the IDLE state. This is errata 2 in the CC2480 Errata Note.	100.0	12.0

7.2 Step 2 – Request and Receive APS ACK

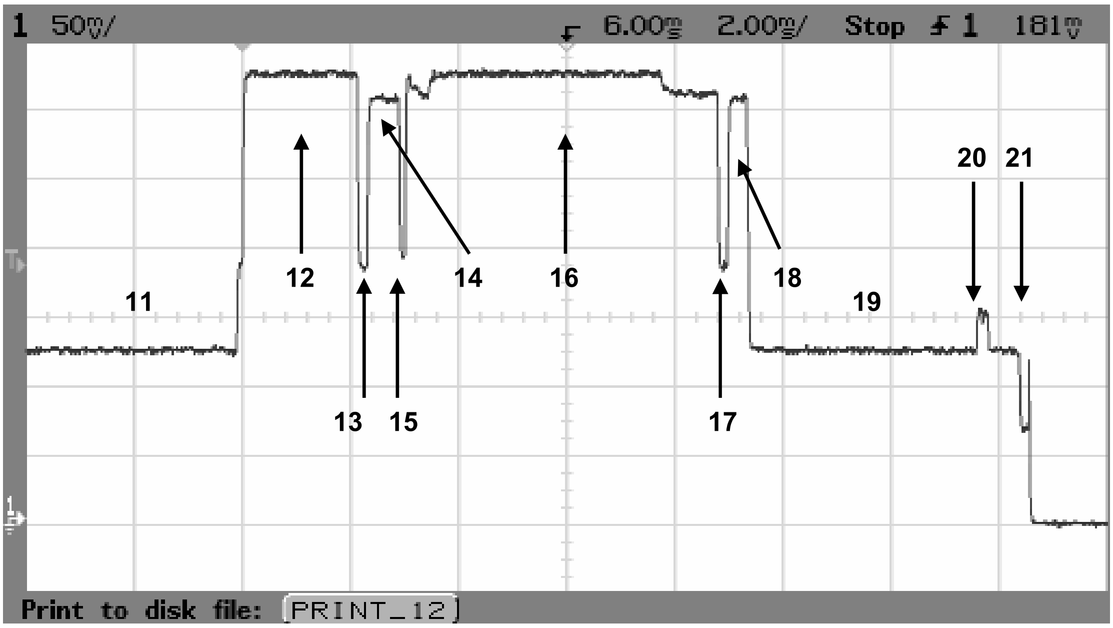


Figure 14 - Request and Receive APS ACK

Event	Description	[ms]	[mA]
12	CC2480 wakes up and begins a new transmission sequence. The node has to request the APS ACK from the recipient. As in step 6, it runs the CSMA-CA algorithm.	1.0	31.0
13	Switch from RX to TX.	0.2	18.0
14	The data request packet is sent over the air.	0.8	29.0
15	Switch from TX to RX.	0.2	18.0
16	The node receives first the MAC ACK for the data request packet, then it stays in RX until it receives the APS ACK.	6.0	31.0
17	Switch from RX to TX.	0.2	18.0
18	The node sends a MAC ACK to confirm that it has received the APS ACK.	0.4	29.0
19	CC2480 processes the incoming APS ACK and prepares a response to the host (MSP430)	4.0	12.0
20	CC2480 wakes up the MSP430 to indicate that the packet was sent without failure to the recipient.	0.4	14.7
21	CC2480 prepares for sleep and turns off the 32 MHz crystal.	0.2	7.0

7.3 Step 3 – Request Additional Data

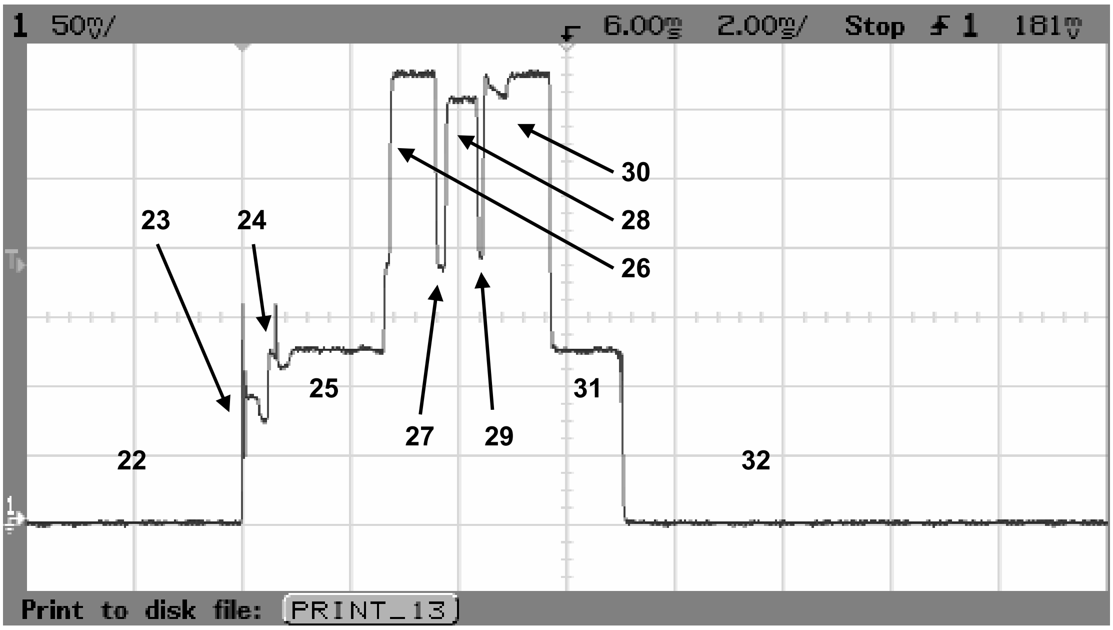


Figure 15 - Request Additional Data

Event	Description	[ms]	[mA]
22	CC2480 is sleeping, but has only entered LPM2, as it needs to wake up one last time. ZASA has configured CC2480 such that this duration is 100 ms.	100.0	0.00097
23	CC2480 wakes up from LPM2. Similar to step 1. The node will transmit a new data request just to make sure that there are no more pending data destined for the node before it enters deep sleep. Note that the MSP430 is not involved in this transaction.	0.7	7.0
24	CC2480 starts the 32 MHz clock.	0.1	12.0
25	Internal processing and preparing for transmit.	1.8	12.0
26	CSMA-CA	1.0	31.0
27	Switch from RX to TX.	0.2	18.0
28	The data request packet is sent over the air.	0.8	29.0
29	Switch from TX to RX.	0.2	18.0
30	Receive MAC ACK.	1.2	31.0
31	Post processing and prepare for deep sleep.	1.6	12.0
32	Deep sleep.	-	0.00066

8 Conclusion

In this document, we have gone through the basics of performing current consumption measurements and estimating the expected lifetime of a battery operated system. Then we looked more closely at the eZ430-RF2480 platform and performed a series of measurements on the CC2480 Target Board to determine the lifetime of the system depending on various configuration options in the software.

In a two chip solution, it is clear that the two devices must operate “in harmony” with each other to get the best out of the system in terms of low power. Both devices must be used and configured so that they meet the functional requirements and at the same time do not waste power by staying awake when they could have been in a sleep state. We have seen that by using the ultra low power MSP430 microcontroller and the CC2480 ZigBee coprocessor, it is possible to achieve several years of operation using common off-the-shelf batteries.

Two factors that influence the power consumption strongly are the transmission period and the poll period. Effectively increasing the transmission period can be done by making sure that data is only sent when it must and then send as much data as possible. It is e.g. better to transmit one packet with two samples rather than two packets with one sample each.

Another factor that will influence the power consumption is the condition of the RF channel. With much traffic on the channel, the duration of the CSMA-CA phase will be longer and have a direct impact on the expected lifetime of the system. In addition, if a packet is lost due to interference, the packet may have to be retransmitted, which in turn will translate into shorter transmit periods and thus shorter lifetime.

As a consequence, it is difficult to calculate the exact lifetime, since the model of the system rapidly becomes very complex. However, by having a basic understanding of the major contributors to the lifetime and then combining the various elements, it is possible to get good estimates that can be used as basis for the final system design.

9 References

[1] CC2480 Datasheet (**swrs074.pdf**)

[2] CC2480 Interface Specification (**swra175.pdf**)

[3] CC2480 Errata Note (**swrz027.pdf**)

[4] MSP430F2274 Datasheet (**slas504.pdf**)

[5] eZ430-RF2480 User’s Guide (**swru151.pdf**)

[6] Measuring Power Consumption with CC2430 and Z-Stack (**swra144.pdf**)

10 Document History

Revision	Date	Description/Changes
SWRA177	2008-04-29	Initial version.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265