Battery life estimation for Thread and Zigbee SEDs

nWP-039

White Paper

1159720_177 / 2021-10-27

NORDIC SEMICONDUCTOR

Contents

Revision history

Date	Description
2021-10-28	First release

1 Introduction

This white paper gives an overview of the estimated energy consumption for Thread and Zigbee Sleepy End Devices (SED) and the battery life you can expect.

The goal is to provide a simple mathematical model based on measurements to determine the battery life for any application in a few representative scenarios.

2 SED power consumption model

Radio activity, base consumption, and application activity determine total power consumption for a SED.

Radio activity is made up of several factors that affect its power consumption: network commissioning and link maintenance logic, application-specific data transfer, and environment (for example, the availability of the radio channel).

Base consumption is affected by power consumption in the sleep state (powered peripherals, RAM).

Application activity is affected by the CPU utilization and the peripherals in use.

When a device first starts up, it is in a detached state not connected to a network. Because a SED is not able to form a network on its own, it must connect to an already existing network. To join an existing network, the device needs to obtain and store the network credentials in its persistent memory. Once the device connects to a network it transitions between the following states:

- Sleep state - this is where the device waits for events
- Radio operation - here the device transmits or receives data
- Application activity - the device communicates with connected wired peripherals and performs operations required for the application it is running

The following figure shows the relationship between the states in more detail.

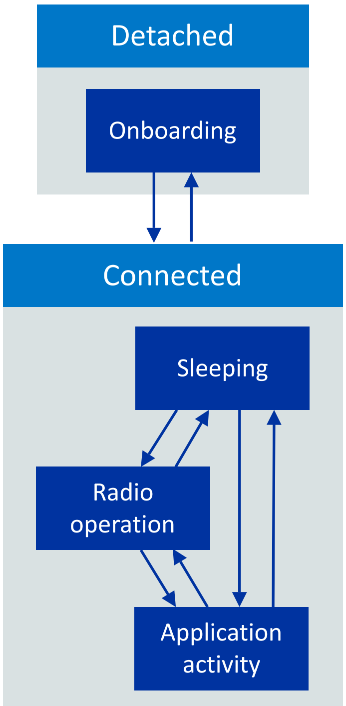

Figure 1: Sleepy End Device states

2.1 Onboarding

At the beginning of its life cycle, a new Thread or Zigbee Sleepy End Device needs to securely obtain network credentials, which allow it to communicate with other devices in the network. This process is called commissioning and is performed only once per device factory reset.

Once the device starts up, it sends a message indicating that it is looking for a router to attach to. It then gains security information from the network, which allows it to join the network as a child device. This works the same for Thread and Zigbee SEDs. The impact commissioning has on the battery life is very small and will not be analyzed in this document.

After the device is commissioned, it attaches to the mesh network by establishing communication with the neighboring router. It then starts acting as an end node. This process is repeated usually once per device power cycle.

The processes of commissioning and establishing communication with the router are not frequent operations, and were not included in the analysis. To measure the effect of these operations, perform the following steps:

1. Measure the charge per single operation.
2. Multiply by the expected number of occurrences during the device lifetime.
3. Reduce the effective battery charge by the value obtained in the previous step.

2.2 Sleep state

The IEEE 802.15.4 standard supports sleeping devices, which spend most of the time with the radio peripheral disabled. This functionality is used for efficient device power management in both Thread and Zigbee protocols.

To conserve battery life, the SED remains in sleep state until an active state is needed.

The sleep state reduces energy consumption through any of the following ways:

- Turning off all unnecessary modules and peripherals
- Putting the CPU in a low power state or disabling it
- Using dedicated low power peripherals when needed
- Pulling up unused GPIO pins to avoid current leakage through unconfigured pins

The amount of current consumed by a device in sleep state is three to four orders of magnitude smaller than while active.

2.3 Radio activity - data requests

The device’s radio transceiver is disabled while the device is sleeping. In order to communicate, it must be in an active state.

A SED can transmit at any time, as its parent keeps the transceiver permanently on. However, to receive data, the SED must initiate the process by using a data polling mechanism.

A data poll mechanism is initiated periodically through a data request frame, asking the parent if there is any data to receive. The parent replies with an ACK packet, including a frame pending bit set to `1` if there are pending frames to receive. Depending on the reply, a SED can stay in the reception state and get data, or go to sleep state to save energy, as shown in the following figure. The parent buffers the data meant for a SED and sends it as a response to the data request message.

The interval between when the SED sends data request packets is known as the poll period. You can adjust its value according to your application needs. This feature enables a poll period suited to its role, creating an energy efficient device.

Shorter interval increases the responsiveness, but also negatively impacts the power consumption.

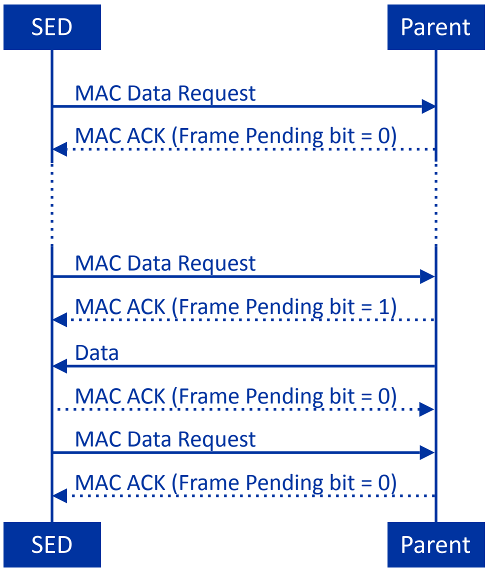

Figure 2: IEEE802.15.4 SED communication scheme

2.4 Radio activity - application data transfer

Data transfer patterns differ according to the application's functionality needs, the network load, and application protocols.

One application might send small amounts of data frequently, while another might stay asleep for a longer time before sending a larger packet. Some applications might require the recipient to confirm data receipt, while others might not.

The amount of energy consumed through data transfer is one of the most application dependent areas of device power consumption. This is because packet size is not constant like in a data request or ACK packet, but is dependent on the payload size. The larger the data packets are, the more time the device needs to send them, and the more energy it consumes.

Total packet length is not equal to data size, because the packet also contains overhead added by various layers.

2.4.1 Thread

Thread is an IPv6 based protocol that allows enabling a wide range of application layer protocols.

One of the application layer protocols enabled by Thread is Constrained Application Protocol (CoAP), which is useful for low power devices that don't have a lot of memory. The CoAP message header has a fixed size of 4 B, which is equal to the minimum size for sending an empty message. In most cases, a packet contains other fields, increasing the size.

A typical CoAP packet includes these fields:

- Protocol version (fixed 2 b)
- Message type (fixed 2 b)
- Token length (fixed 4 b)
- Message code (fixed 1 B)
- Message ID (fixed 2 B)
- Token (0 B to 8 B, where 8 B is used here)
- Optional (minimum 1 B, where 5 B is used here)
  - Unique Resource Identifier (URI)
  - End of options marker

- Application payload

The CoAP specification does not clearly define maximum packet payload size, but it does define maximum message size. This should be small enough to fit in a single packet to avoid IP fragmentation. The IP maximum transmission unit is 1280 B. We recommend using 1152 B as the upper limit for the message and 1024 B for the payload. CoAP frames that do not fit within a single 127-byte IEEE 802.15.4 data frame are fragmented and sent in multiple radio transmissions, which increases the power consumption. The measurement section focuses only on the cases without fragmentation.

The CoAP protocol uses the UDP transport protocol in Connectionless mode. To improve reliability, CoAP enables two types of messages.

- Confirmable (CON) - requires confirming each message with CoAP ACK packet. The message is retransmitted if an ACK is not received.
- Non-confirmable (NON) - does not use acknowledgments. This messaging type is less reliable than CON.

These message types provide flexibility for the application, allowing for confirmation of a received package using CON messaging, and conserving power on NON messages.

These communication types are shown in the following diagram.

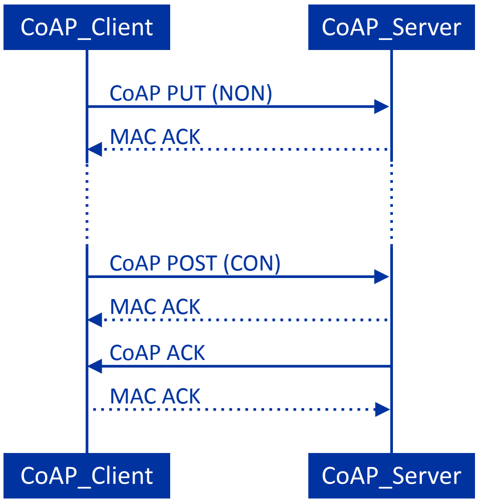

Figure 3: CoAP communication scheme

2.4.2 Zigbee

The Zigbee protocol uses the Zigbee Cluster Library (ZCL) as an application layer (APL) to organize data. It relies on the application support sub-layer (APS), which provides an interface between the network (NWK) and application layers.

The APS layer provides end-to-end reliability, with acknowledgment and packet fragmentation for packets that don't fit into a single IEEE 802.15.4 frame.

ZCL defines a catalog of clusters. A cluster specifies a logical function for a device. The *Zigbee Cluster Library Specification* defines attributes, commands, and behaviors. The On/Off cluster defines the OnOff attribute with a readable value, and commands **`On`**, **`Off`**, and **`Toggle`**.

Zigbee devices communicate with each other by sending commands. The communication scheme depends on the device type and packets being sent. The following is an example process:

- The bulb device stores the state of the physical bulb in the OnOff attribute of the On/Off cluster. The device checks if the physical bulb is on or off and, based on the bulb's state, it sets its own state in the ZCL cluster's attribute.
- To turn the bulb off, the switch sends an **`Off`** command to the bulb, with APS acknowledgment requested and default response disabled.

- Upon receiving the command, the bulb changes state of a physical bulb and the attribute value. If the received frame is valid, the bulb responds to the switch by sending an APS acknowledgment.

The diagram below shows how a Zigbee Sleepy End Device switch turns off a Zigbee bulb.

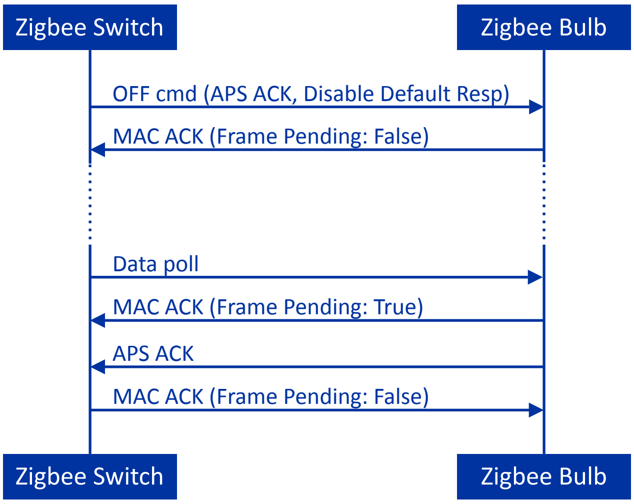

Figure 4: Simple Zigbee communication

3 Mathematical model

The application functionality and the amount of time a device is expected to spend in each state is defined here.

Once you know the battery characteristics, you can calculate the battery life. The following table defines the symbols used to characterize properties for each state.

Variable	Description
Q_{total}	Battery charge
Q_{poll}	Average charge of a single data poll operation
Q_{app}	Average charge of a single application operation
Q_{net}	Average charge of application data over network transfer
T_{total}	Lifetime, battery life
T_{poll}	Average period for data request action
T_{app}	Average period for application activity
T_{net}	Average period for sending application data over the network
t_{poll}	Average time for a single data request action
t_{app}	Average application activity time
t_{net}	Average time spent on the single application data over network transfer
t_{active}	Total time spent in an active state
I_{poll}	Average current for a single data request action
I_{app}	Average current for a single application activity
I_{net}	Average current for sending application data
I_{sleep}	Average sleep current

Table 1: State properties

Total time spent in active states is given in the following equation.

$$t_{active} = \frac{t_{poll} * T_{total}}{T_{poll}} + \frac{t_{app} * T_{total}}{T_{app}} + \frac{t_{net} * T_{total}}{T_{net}}$$

Total sleep time is given by the following equation.

$$t_{sleep} = T_{total} - t_{active} = T_{total} - T_{total} * \left(\frac{t_{poll}}{T_{poll}} + \frac{t_{app}}{T_{app}} + \frac{t_{net}}{T_{net}}\right) = T_{total} * \left(1 - \frac{t_{poll}}{T_{poll}} - \frac{t_{app}}{T_{app}} - \frac{t_{net}}{T_{net}}\right)$$

Total charge consumed by data requests is given in the following equation.

$$Q_{poll} = \frac{I_{poll} * t_{poll} * T_{total}}{T_{poll}}$$

The following equation shows the total charge that is consumed by application activity.

$$Q_{app} = \frac{I_{app} * t_{app} * T_{total}}{T_{app}}$$

Total charge consumed by an application data network transfer is shown in the following equation.

$$Q_{net} = \frac{I_{net} * t_{net} * T_{total}}{T_{net}}$$

Total charge consumed in sleep state is given in the following equation.

$$Q_{sleep} = I_{sleep} * t_{sleep}$$

The following equation calculates the total charge consumed by all components.

$$Q_{total} = Q_{poll} + Q_{app} + Q_{net} + Q_{sleep}$$

The following equation gives the total charge represented by the current and time characteristics.

$$Q_{total} = T_{total} * \left(\frac{I_{poll} * t_{poll}}{T_{poll}} + \frac{I_{app} * t_{app}}{T_{app}} + \frac{I_{net} * t_{net}}{T_{net}} + I_{sleep} * \left(1 - \frac{t_{poll}}{T_{poll}} - \frac{t_{app}}{T_{app}} - \frac{t_{net}}{T_{net}} \right) \right)$$

Battery life as a ratio of total consumed charge (or battery charge) over the average current of the system is shown in the following equation.

$$T_{total} = \frac{Q_{total}}{\left(\frac{I_{poll} * t_{poll}}{T_{poll}} + \frac{I_{app} * t_{app}}{T_{app}} + \frac{I_{net} * t_{net}}{T_{net}} + I_{sleep} * \left(1 - \frac{t_{poll}}{T_{poll}} - \frac{t_{app}}{T_{app}} - \frac{t_{net}}{T_{net}} \right) \right)}$$

Because the time of an active operation is very short compared to its period and the three orders of magnitude difference between the sleep current and active operations current, we can approximate the battery life expression with current values equation as follows.

$$T_{total} = \frac{Q_{total}}{\left(\frac{I_{poll} * t_{poll}}{T_{poll}} + \frac{I_{app} * t_{app}}{T_{app}} + \frac{I_{net} * t_{net}}{T_{net}} + I_{sleep} \right)}$$

Working with current values for short events is more difficult than operating on charge values. The charge value is equal to the operation current integrated over the time duration as presented in the following figure showing a simplified battery life expression.

$$T_{total} = \frac{Q_{total}}{\left(\frac{Q_{poll}}{T_{poll}} + \frac{Q_{app}}{T_{app}} + \frac{Q_{net}}{T_{net}} + I_{sleep}\right)}$$

With this approximation, the battery life can be estimated for any application by measuring the average charge for active operations and their period. Ratios of these together with the average sleep current give the total application average current. Dividing the battery charge by the application average current results in the estimated battery life.

4 Measuring current consumption

This section presents setup, settings, and methodology for measuring current consumption in Thread and Zigbee devices.

When selecting a Nordic Semiconductor development kit for measuring power consumption, choose one that has power configuration switches to disable additional on-board modules, such as the SEGGER J-Link programmer. These are usually installed on the development kit and must be disconnected to prevent an increase in current consumption beyond what you're trying to measure.

Current consumption measurements differ between boards. A board with additional components, such as the nRF52840 Development Kit, consumes more current in sleep mode. According to our measurements, the basic EBYTE E73 module showed 1 µA lower sleep current than the nRF52840 DK. Sleep current also differs between Development Kits of the same revision. Board temperature and other environmental factors can also affect measurements.

4.1 Hardware setup

The power measurement setup consists of three boards:

- Power Profiler Kit (PCA63511)
- nRF52840 DK (PCA10056 v2.0.1)
- nRF52 DK (PCA10040)

> **Note:** In place of the nRF52 DK, you can use another Nordic Development Kit or SEGGER J-Link.

In addition to these boards, you also need two micro USB cables.

To set up the boards, follow these steps:

1. Plug the PPK into the pins on top of the nRF52 DK.
2. Connect two jump wires from **P16** external DUT on the PPK to **P21** external supply on the nRF52840 DK.
3. Connect the micro USB cables to the **J2** micro USB connectors located on both the nRF52840 DK and nRF52 DK boards. Connect the other end of the cables to the USB hub. This minimizes current leakage.

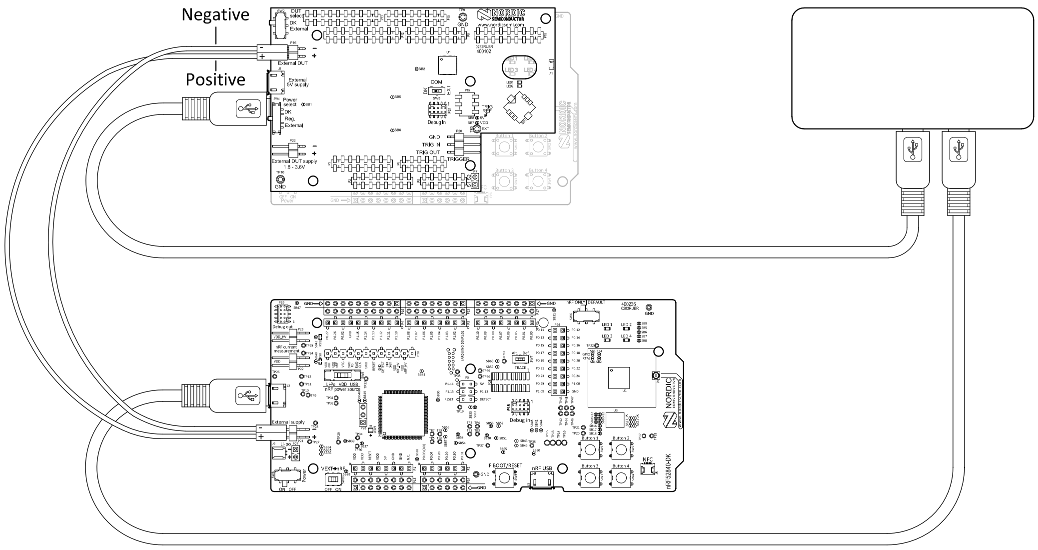

Figure 5: Hardware configuration

Measurements were taken with the Power Profiler v3.0.0-alpha.0 application. The PPK voltage regulator was set to 3 V.

Set the switches to the positions indicated in the following table.

Board	PCB name	Switch description and position
PPK	SW1	Power select ≥ nRF52 DK
PPK	SW2	nRF52840 DK select ≥ External
PPK	SW3	COM ≥ nRF52 DK
nRF52840 DK	SW6	nRF ONLY|DEFAULT ≥ nRF ONLY
nRF52840 DK	SW8	Power ≥ ON
nRF52840 DK	SW9	nRF power source ≥ VDD
nRF52840 DK	SW10	VEXT > nRF ≥ ON

Table 2: Description of jumpers and switches

4.2 Measurement configuration examples

This section focuses on radio activity with chosen topologies, message schemes, and their format for Thread and Zigbee protocols.

Battery life is shorter when the application performs activities like computation, driving LEDs, or communication with connected sensors.

Another factor impacting performance is the radio characteristic of the environment. If it is lossy, some frames won't reach the destination. You can avoid this by increasing the transmit power or enabling acknowledgments on the application layer. Keep in mind that both of these actions increase energy consumption, which lowers battery life.

4.2.1 Thread

In Thread applications, two nodes communicate in the test network. The first is the Full Thread Device (FTD) acting as a leader running a CoAP server sample. The second is the Device Under Test (DUT), which runs a modified CoAP client sample to send different CoAP type and length messages.

We tested TX power values of 0 dBm and 8 dBm, with Non-confirmable and Confirmable CoAP message types, followed by reception of the acknowledgment on the CoAP layer. The Confirmable CoAP message requires the device to enable the receiver for a longer period of time to receive an acknowledgment. This results in an over 2.5 times increase of the charge for each sent CoAP frame. After sending the CoAP CON, the SED sends the parent the data request. The set frame pending bit instructs the SED to keep the transceiver enabled longer to receive a 56-byte acknowledgment frame.

Message	Operation	CoAP code	Token size	Options size	Payload size	IEEE802.15.4 frame size
IEEE802.15.4 Data Request	TX	N/A	N/A	N/A	N/A	20 B
Small CoAP	TX	PUT	8 B	0	0	57 B
Large CoAP	TX	PUT	8 B	5 B (URI 3 B)	64 B	125 B
Data Request + CoAP ACK	RX	Valid	8 B	N/A	0 B	56 B (CoAP ACK)

Table 3: Thread radio frames at 0 dBm and 8 dBm

4.2.2 Zigbee

For Zigbee tests, two nodes communicate with each other. They consist of a Zigbee Router and a Zigbee SED (with an additional Coordinator node used only for the network formation and association process).

A modified Light Bulb sample was used as a Zigbee router. The Zigbee shell library was enabled for the sample to handle custom Zigbee Ping commands, implemented as a custom Zigbee cluster in the Zigbee shell library.

A modified Light Switch sample was used as a Zigbee SED. The Zigbee shell library was enabled for the sample to handle Zigbee Ping commands, and the sample was modified to send Ping commands with an interval.

Two TX power values have been tested: 0 dBm and 8 dBm. Zigbee Ping commands were used that allowed for testing frames of different length, with and without APS acknowledgment, and with a Default Response command disabled or not.

For power consumption measurements, ping command response was disabled by default. The frames in the following table were tested.

The difference in length of the IEEE802.15.4 Data Request between Thread and Zigbee is also notable. The Zigbee protocol has security disabled at the MAC layer, which shortens the frame length to 10 B, compared to 20 B for Thread.

Message	ZCL payload length	IEEE802.15.4 frame length
IEEE802.15.4 Data Request	N/A	10 B
Short ZCL frame	7 B	69 B
Long ZCL frame	79 B	125 B
Data Request + APS ACK	N/A	59 B (APS ACK)

Table 4: Zigbee radio frames at 0 dBm and 8 dBm

5 Battery life estimation

Measured scenarios differ depending on protocol. This section shows the most common data exchange patterns and their impact on battery life.

Two lithium batteries were used for testing, as shown in the following table.

Battery name	Attribute	Value
CR123A	Standard capacity	1550 mAh
	Yearly loss	2%
	15 year self-discharge loss	465 mAh
	Effective battery capacity	1085 mAh
	Effective battery charge	3906 C
CR2032	Standard capacity	235 mAh
	Yearly loss	2%
	3 year self-discharge loss	14.1 mAh
	Effective battery capacity	220.9 mAh
	Effective battery charge	795.24 C

Table 5: Battery information

A CR2032 battery is often used in small, size constrained devices. For testing purposes, the PCA10056 Development Kit comes with a CR2032 battery holder, so the board can be powered directly by the battery. For larger devices, the CR123A battery is used for device lifetime estimation.

5.1 Thread operations

The following scenarios were evaluated in the active connected state, as shown in the table. No additional application activity or data exchange occurs.

State	CoAP acknowledgment	Battery
Idle SED connected to the parent	No	Useful in predicting base battery life
Short CoAP NON frame sent every 30 min to parent	No	Estimate battery life for the simple light switch application
Short CoAP CON frame sent every 30 min to parent	Yes	Base estimation for the simple sensor
Long CoAP NON frame sent every 10 min to the parent	No	Base estimation for the more advanced switch
Long CoAP CON frame sent every 10 min to the parent	Yes	Base estimation for the more advanced sensor
Long CoAP CON frame sent every 1 min to the parent	Yes	Base estimation for the more advanced sensor Shows impact of increasing messaging frequency

Table 6: Thread operations

5.1.1 Thread transmit

The following scenarios have been measured for 0 dBm and 8 dBm power setting.

TX power	Operation	Measurement
	Sleep current	3.74 µA
0 dBm	Data request	15.97 µC
	CoAP NON 57 B	27.96 µC
	CoAP NON 125 B	43.71 µC
	Data request + CoAP ACK	136.7 µC
8 dBm	Data request	26.37 µC
	CoAP NON 57 B	51.37 µC
	CoAP NON 125 B	90.97 µC
	Data request + CoAP ACK	170.03 µC

Table 7: Thread operations measured on nRF52840 v2.0.0 Development Kit

Figure 6: Data request operation charge measurement

A simplified device lifetime formula with charge values is used in the following calculations.

Two assumptions were made in the calculation:

1. User application activity is limited to the absolute minimum.

- The application sleeps the majority of time, and wakes up only to send a frame with fixed data.
- The logger module is disabled.

Therefore, the application activity charge (Q_{app}) is close to 0 and can be omitted. The application activity period (T_{app}) is irrelevant in this case.

2. The network transfer charge is set as a sum of CoAP NON and data request + CoAP ACK transaction charges. This value contains the following assumptions:

- The same data request frame is counted twice, once for typical periodic data polling and again to pull the CoAP ACK frame. The counting takes place on a condition that the CoAP ACK frame is received by the parent Router before the next periodic data polling happens. If this condition is not met, the Q_{net} should be reduced proportionally by the periods ratio multiplied by the Q_{poll}.
- No retransmissions occur. In the case of a lossy channel, DUT does not receive a CoAP ACK and needs to retransmit the frame.

Thread TX examples using 0 dBm

The following scenario is the device connected and sleeping at 0 dBm.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	15.97 µC
	T_{poll}	3 s
	Q_{net}	0 µC
	T_{net}	30 s
CR123A battery	Battery life	13.67 y
CR2032 battery	Battery life	2.78 y

Table 8: Thread Scenario 1 at 0 dBm

This scenario is of the device transmitting a CoAP short NON frame every 30 minutes at 0 dBm.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	15.97 µC
	T_{poll}	3 s
	Q_{net}	27.96 µC
	T_{net}	1800 s
CR123A battery	Battery life	13.64 y
CR2032 battery	Battery life	2.78 y

Table 9: Thread Scenario 2 at 0 dBm

The following scenario is the device transmitting a CoAP short CON frame every 30 minutes at 0 dBm.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	15.97 µC
	T_{poll}	3 s
	Q_{net}	164.66 µC
	T_{net}	1800 s
CR123A battery	Battery life	13.53 y
CR2032 battery	Battery life	2.75 y

Table 10: Thread Scenario 3 at 0 dBm

The device transmits a CoAP long NON every 10 minutes at 0 dBm in the following scenario.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	15.97 µC
	T_{poll}	3 s
	Q_{net}	43.71 µC
	T_{net}	600 s
CR123A battery	Battery life	13.56 y
CR2032 battery	Battery life	2.76 y

Table 11: Thread Scenario 4 at 0 dBm

The following scenario shows a CoAP long CON every 10 minutes at 0 dBm.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	15.97 µC
	T_{poll}	3 s
	Q_{net}	180.41 µC
	T_{net}	600 s
CR123A battery	Battery life	13.23 y
CR2032 battery	Battery life	2.69 y

Table 12: Thread Scenario 5 at 0 dBm

The device transmits a CoAP long CON every minute at 0 dBm in the following scenario.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	15.97 µC
	T_{poll}	3 s
	Q_{net}	180.41 µC
	T_{net}	60 s
CR123A battery	Battery life	10.26 y
CR2032 battery	Battery life	2.09 y

Table 13: Thread Scenario 6 at 0 dBm

Thread TX examples using 8 dBm

The following scenario shows the device connected and sleeping at 8 dBm.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	26.37 µC
	T_{poll}	3 s
	Q_{net}	0 µC
	T_{net}	30 s
CR123A battery	Battery life	9.88 y
CR2032 battery	Battery life	2.01 y

Table 14: Thread Scenario 7 at 8 dBm

This scenario is of the device transmitting a CoAP short NON frame every 30 minutes at 8 dBm.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	26.37 µC
	T_{poll}	3 s
	Q_{net}	51.37 µC
	T_{net}	1800 s
CR123A battery	Battery life	9.86 y
CR2032 battery	Battery life	2.01 y

Table 15: Thread Scenario 8 at 8 dBm

The following scenario is the device transmitting a CoAP short CON frame every 30 minutes at 8 dBm.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	26.37 µC
	T_{poll}	3 s
	Q_{net}	221.40 µC
	T_{net}	1800 s
CR123A battery	Battery life	9.79 y
CR2032 battery	Battery life	1.99 y

Table 16: Thread Scenario 9 at 8 dBm

The device transmits a CoAP long NON every 10 minutes at 8 dBm in the following scenario.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	26.37 µC
	T_{poll}	3 s
	Q_{net}	90.97 µC
	T_{net}	600 s
CR123A battery	Battery life	9.77 y
CR2032 battery	Battery life	1.99 y

Table 17: Thread Scenario 10 at 8 dBm

The following scenario shows a CoAP long CON every 10 minutes at 8 dBm.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	26.37 µC
	T_{poll}	3 s
	Q_{net}	261 µC
	T_{net}	600 s
CR123A battery	Battery life	9.55 y
CR2032 battery	Battery life	1.94 y

Table 18: Thread Scenario 11 at 8 dBm

The device transmits a CoAP long CON every minute at 8 dBm in the following scenario.

		Value
Operation parameters	I_{sleep}	3.74 µA
	Q_{poll}	26.37 µC
	T_{poll}	3 s
	Q_{net}	261 µC
	T_{net}	60 s
CR123A battery	Battery life	7.34 y
CR2032 battery	Battery life	1.49 y

Table 19: Thread Scenario 12 at 8 dBm

5.2 Zigbee operations

The following scenarios were evaluated in the active connected state, as shown in the table. No additional application activity or data exchange occurs.

Operation	APS Acknowledgment	Battery
Idle SED connected to the parent	No	Useful in predicting base battery life
Short ZCL frame sent every 30 min to parent	No	Estimate battery life for the simple light switch application
Short ZCL frame with APS-ACK sent every 30 min to parent	Yes	Good estimation for a more reliable light switch application Good estimation for a sensor application reporting attribute size 4 B
Long ZCL frame sent every 10 min to the parent	No	Base estimation for the more advanced switch
Long ZCL frame with APS-ACK sent every 10 min to the parent	Yes	Base estimation for the more advanced sensor reporting multiple attributes
Long ZCL frame with APS-ACK sent every 1 min to the parent	Yes	Base estimation for the more advanced sensor that reports multiple attributes frequently

Table 20: Zigbee operations

The ZBOSS stack features an adaptive poll period mechanism called Turbo poll. This mechanism makes devices more responsive for a short period of time after the Zigbee frame has been transmitted or received. This shortens the data poll period so the device can respond faster to incoming frames. After the last transaction, the polling period is incrementally extended back to its default. This feature is optional and is enabled by default.

This responsiveness requires greater power consumption. Turning this feature off may extend battery life for SEDs that frequently send or receive frames. As shown in Table 27: Zigbee Scenario 6 at 0 dBm on page 26 it can extend battery life by about 22%, but only about 1% as shown in Table 24: Zigbee Scenario 3 at 0 dBm on page 25. With the default data poll period set to 3 seconds, the device sends a total of 10 data poll packets within 6.5 seconds, before polling its parent again every 3 seconds. This additional power consumption is added to the cost of radio operations when estimating battery life.

This added cost is calculated as the difference between power consumption of data request packets sent when the adaptive poll mechanism is not running, and power consumption of data request packets sent within the time the adaptive poll mechanism is running.

5.2.1 Zigbee transmit

The following scenarios were performed on the nRF52840 v2.0.0 Development kit using a 0 dBm and 8 dBm power setting.

TX power	Operation	Measurement
	Sleep current	3.18 µA
0 dBm	Data Poll period	3 s
	Data Poll	19.33 µC
	Data Poll with APS ACK	64.636 µC
	Short ZCL frame	37.571 µC
	Long ZCL frame	49.201 µC
	Adaptive poll period added charge	155.409 µC
8 dBm	Data Poll	25.91 µC
	Data Poll with APS ACK	82.3 µC
	Short ZCL frame	68.931 µC
	Long ZCL frame	101.081 µC
	Adaptive poll period added charge	202.949 µC

Table 21: Zigbee operations measured on nRF52840 v2.0.0 Development Kit

To perform the calculations the simplified device lifetime formula with charge values has been used.

Two approximations were taken:

1. User application activity is limited to the absolute minimum.

    - The application sleeps for most of the time and wakes up only to send a frame that has fixed data
    - Logger module is disabled

    Therefore, application activity charge (Q_{app}) is set to 0 in the model. The application activity period (T_{app}) becomes irrelevant in this case.
2. The network transfer charge is set as a sum of ZCL frame, Data Poll, and an APS ACK (if APS acknowledgment is requested) transaction charges and added charge of Adaptive poll period behavior. This value carries the following assumptions:

    - Packets with APS acknowledgment requested are sent to nearby devices. The more nodes the packet needs to go through, the longer the originator of the packet waits for the acknowledgment. It is assumed that APS ACK was received by the device's parent before the device sent its first data request after sending the packet.
    - The amount of time the device keeps the radio turned on to receive packet from its parent is minimal - the longer it takes for the device’s parent to send a packet, the more energy the device is using.
    - No retransmissions occur. A lossy channel may lead to failing transmissions which results in frame retransmissions, causing increased power consumption.

Zigbee TX examples using 0 dBm

The following scenario is the device connected and sleeping at 0 dBm.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	19.33 µC
	T_{poll}	3 s
	Q_{net}	0 µC
	T_{net}	30 s
CR123A battery	Battery life	12.87 y
CR2032 battery	Battery life	2.62 y

Table 22: Zigbee Scenario 1 at 0 dBm

This scenario is of the device transmitting a short ZCL frame every 30 minutes at 0 dBm.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	19.33 µC
	T_{poll}	3 s
	Q_{net}	192.98 µC
	T_{net}	1800 s
CR123A battery	Battery life	12.73 y
CR2032 battery	Battery life	2.59 y

Table 23: Zigbee Scenario 2 at 0 dBm

The following scenario is the device transmitting a short ZCL + APS ACK every 30 minutes at 0 dBm.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	19.33 µC
	T_{poll}	3 s
	Q_{net}	257.616 µC
	T_{net}	1800 s
CR1232A battery	Battery life	12.68 y
CR2032 battery	Battery life	2.58 y

Table 24: Zigbee Scenario 3 at 0 dBm

The device transmits a long ZCL frame every 10 minutes at 0 dBm in the following scenario.

		Value
Operation parameters	I_{sleep}	3.18 μA
	Q_{poll}	19.33 μC
	T_{poll}	3 s
	Q_{net}	204.61 μC
	T_{net}	600 s
CR123A battery	Battery life	12.43 y
CR2032 battery	Battery life	2.53 y

Table 25: Zigbee Scenario 4 at 0 dBm

The following scenario shows a long ZCL frame + APS ACK every 10 minutes at 0 dBm.

		Value
Operation parameters	I_{sleep}	3.18 μA
	Q_{poll}	19.33 μC
	T_{poll}	3 s
	Q_{net}	269.25 μC
	T_{net}	600 s
CR123A battery	Battery life	12.30 y
CR2032 battery	Battery life	2.50 y

Table 26: Zigbee Scenario 5 at 0 dBm

The device transmits a long ZCL frame + APS ACK every minute at 0 dBm in the following scenario.

		Value
Operation parameters	I_{sleep}	3.18 μA
	Q_{poll}	19.33 μC
	T_{poll}	3 s
	Q_{net}	269.25 μC
	T_{net}	60 s
CR123A battery	Battery life	8.78 y
CR2032 battery	Battery life	1.79 y

Table 27: Zigbee Scenario 6 at 0 dBm

Zigbee TX examples using 8 dBm

The following scenario shows the device connected and sleeping at 8 dBm.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	25.91 µC
	T_{poll}	3 s
	Q_{net}	0 µC
	T_{net}	30 s
CR123A battery	Battery life	10.48 y
CR2032 battery	Battery life	2.13 y

Table 28: Zigbee Scenario 7 at 8 dBm

This scenario is of the device transmitting a short ZCL frame every 30 minutes at 8 dBm.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	25.91 µC
	T_{poll}	3 s
	Q_{net}	271.88 µC
	T_{net}	1800 s
CR123A battery	Battery life	10.35 y
CR2032 battery	Battery life	2.10 y

Table 29: Zigbee Scenario 8 at 8 dBm

The following scenario is the device transmitting a short ZCL + APS ACK every 30 minutes at 8 dBm.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	25.91 µC
	T_{poll}	3 s
	Q_{net}	354.18 µC
	T_{net}	1800 s
CR123A battery	Battery life	10.31 y
CR2032 battery	Battery life	2.10 y

Table 30: Zigbee Scenario 9 at 8 dBm

The device transmits a long ZCL frame every 10 minutes at 8 dBm in the following scenario.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	25.91 µC
	T_{poll}	3 s
	Q_{net}	304.03 µC
	T_{net}	600 s
CR123A battery	Battery life	10.05 y
CR2032 battery	Battery life	2.05 y

Table 31: Zigbee Scenario 10 at 8 dBm

The following scenario shows a long ZCL frame + APS ACK every 10 minutes at 8 dBm.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	25.91 µC
	T_{poll}	3 s
	Q_{net}	386.33 µC
	T_{net}	600 s
CR123A battery	Battery life	9.94 y
CR2032 battery	Battery life	2.02 y

Table 32: Zigbee Scenario 11 at 8 dBm

The device transmits a long ZCL frame + APS ACK every minute at 8 dBm in the following scenario.

		Value
Operation parameters	I_{sleep}	3.18 µA
	Q_{poll}	25.91 µC
	T_{poll}	3 s
	Q_{net}	386.33 µC
	T_{net}	60 s
CR123A battery	Battery life	6.78 y
CR2032 battery	Battery life	1.38 y

Table 33: Zigbee Scenario 12 at 8 dBm

6 Conclusion

In this document, we presented the states in the life cycle of a Sleepy End Device, and a method to estimate battery life.

Particular focus has been given to radio activity, and the typical application layers for Thread, Zigbee, and IEEE 802.15.4. We have given a mathematical model, that when combined with the measurement methodology presented, can be applied to any application to calculate battery life. The Power Profiler Kit measures sleep current and charge of dynamic radio events. You can apply the mathematical model, combined with the measurement methodology presented, and calculate expected battery life for your device and application using known sleep current, radio, application event periods, and their average charge values.

Even in the worst-case radio scenario, we have shown that the small CR2032 battery can power a SED sending a large packet every minute for up to a year and a half. In some of the scenarios we tested, the CR123A battery can provide up to 10 years of battery operation.

Legal notices

By using this documentation you agree to our terms and conditions of use. Nordic Semiconductor may change these terms and conditions at any time without notice.

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function, or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

Nordic Semiconductor ASA does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. If there are any discrepancies, ambiguities or conflicts in Nordic Semiconductor's documentation, the Product Specification prevails.

Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice.

Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

RoHS and REACH statement

Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

Trademarks

All trademarks, service marks, trade names, product names, and logos appearing in this documentation are the property of their respective owners.

Copyright notice

© 2021 Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

QUALITY SYSTEM CERTIFICATION

DNV

ISO 9001