



July 13, 2015

This Thread Technical white paper is provided for reference purposes only.

The full technical specification is available to Thread Group Members. To join and gain access, please follow this link: <http://threadgroup.org/Join.aspx>.

If you are already a member, the full specification is available in the Thread Group Portal: <http://portal.threadgroup.org>.

If there are questions or comments on these technical papers, please send them to help@threadgroup.org.

This document and the information contained herein is provided on an "AS IS" basis and THE THREAD GROUP DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NONINFRINGEMENT.

IN NO EVENT WILL THE THREAD GROUP BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

Copyright © 2015 Thread Group, Inc. All rights reserved.



Battery-Operated Devices

July 2015

Revision History

Revision	Date	Comments
1.0	July 13, 2015	Public Release

Contents

Introduction **2**

Battery-Operated Device Summary **2**

IEEE 802.15.4 Operation of End Devices **2**

 Structure of a Wake Sleep Cycle 3

 Size and Formatting of Sleeping Child Messages..... 4

 Time Required for Sending Messages 5

References **6**



Introduction

The goal of the Thread Group is to design, deploy, and promote an open standard for reliable, cost-effective, low-power, wireless D2D (device-to-device) communication within the home. (For a full and complete overview, refer to the “Thread Stack Fundamentals” white paper.) The low-power part of this goal is delivered by operation of battery-powered devices. The Thread Network is built with always on (powered) Routers and battery-operated devices use these Routers as their Parent and to hold and route messages.

This white paper explains the basic operation of these battery-operated devices and compares their performance.

Battery-Operated Device Summary

Battery-operated devices either wake and poll their Parent for any messages, or wake and send data. The efficiency of these operations determines the energy usage and ultimately the battery life of these devices. It is critical to optimize these basic operations to ensure suitable battery life for devices.

IEEE 802.15.4 Operation of End Devices

Battery-operated devices have two basic behaviors: waking to send data or waking to retrieve data that has been sent to them. Because data cannot be sent directly to a sleepy device, the Parent of that device holds the data for a finite time and the sleeping device can request it (Figure 1).

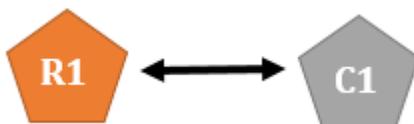


Figure 1. Router Acting as a Parent for Child End Device

For the battery-operated device to wake and send data, it simply wakes up, forms a packet with the appropriate data, and sends it to its Parent which is then responsible for forwarding it through the Thread Network to the destination. A simple case of this is when a window sensor is activated and this interrupt wakes the processor to send a message indicating the window is now open.



The IEEE 802.15.4-2006 standard [\[IEEE802154\]](#) prescribes how battery-operated devices can request data from their Parent. Specifically Section 5.3.4 details a Data Request MAC Command frame that is used to poll a Parent to see if it has data. The Parent responds with an acknowledgement frame with the frame pending bit set to '1' if there is data for the device and '0' if there is no data for the device. A simple case of this data flow is if a user wants to change the configuration of a sleeping door sensor to wake more frequently to report status and battery life. This message is routed to the Parent of the sleeping device and held there for the sleeping device to wake and poll for the message.

Structure of a Wake Sleep Cycle

Figure 2 illustrates a normal wake sleep cycle.

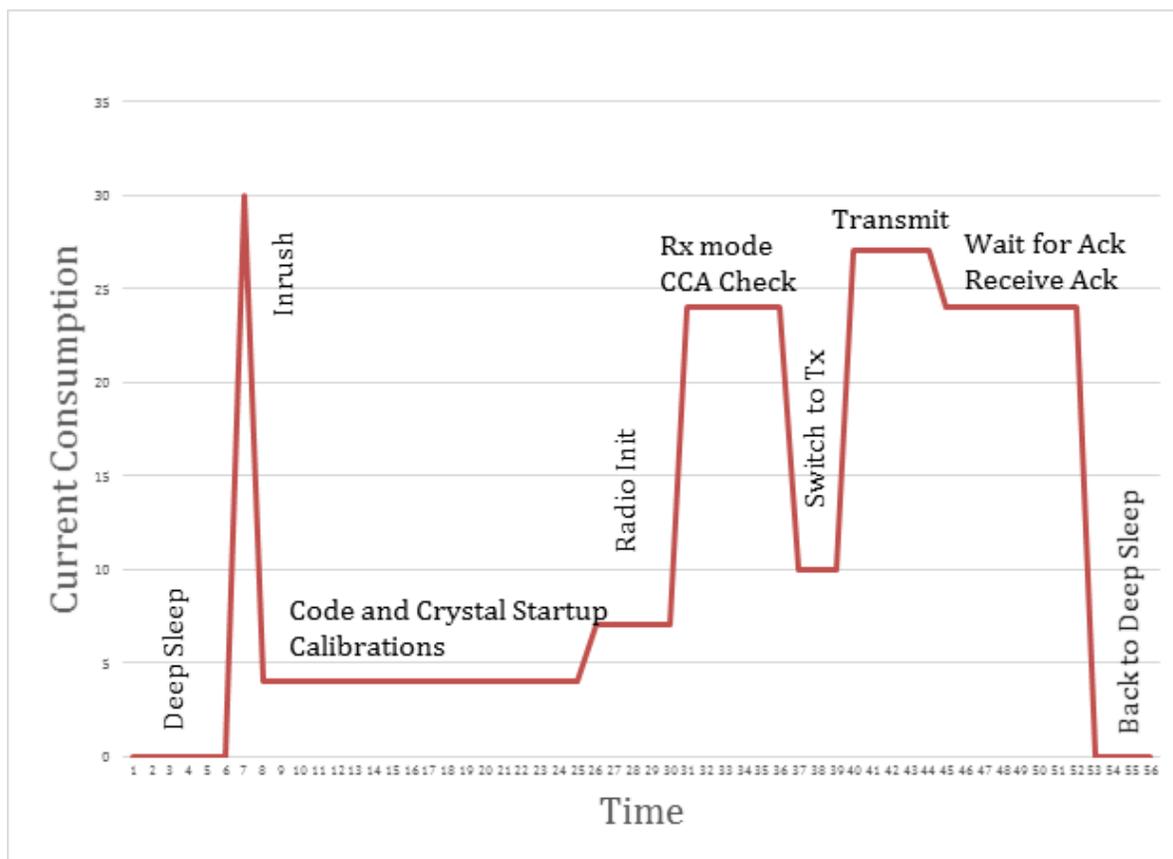


Figure 2. Typical Wake Sleep Cycle



The radio and MCU (Microcontroller) go through the following steps:

1. Wake from deep sleep.
2. Complete code and crystal startup sequence.
3. Radio initialization.
4. Enter receive mode and check for clear air.
5. Switch to transmit.
6. Transmit packet.
7. Wait for and receive acknowledgement.
8. Return to deep sleep.

There are a number of variables that impact the actual power consumption during this wake sleep cycle. Some important factors are:

- Time for the device to wake from deep sleep and be ready with the radio (implementation-specific).
- Time on the air transmitting (size of the packet).
- Time waiting for the ACK packet (defined in the 802.15.4 specification)
- Active currents during the various operations (implementation-specific).

An implementer can optimize some of the times and the radio and MCU design impact the timing and current consumption. The time on the air and time waiting for the ACK is set by the packet size and 802.15.4 specification timing and do not vary between implementations.

Size and Formatting of Sleeping Child Messages

There are two packet formats that impact the time on-the-air and therefore battery life of sleeping devices: the Data Poll MAC Command Frame and the normal Data Frame used for sending messages.

Figure 3 illustrates the IEEE 802.15.4 MAC Command Frame used for the data poll.

Octets: 2	1	variable	0/5/6/10/14	1	variable	2
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	Command Frame Identifier	Command Payload	FCS
MHR				MAC Payload		MFR

Figure 3. IEEE 802.15.4 MAC Command Frame



In Thread the Addressing fields include a 2 byte PANID, the 2byte Short Destination Address and the 2 byte long Source Address. There is no command payload. Thread also secures these data polls so there is a 6-byte security header and a 4-byte MIC (Message Integrity Code). This means the data poll command is 22 bytes long.

Figure 4 illustrates the overall structure of a complete Thread data packet.

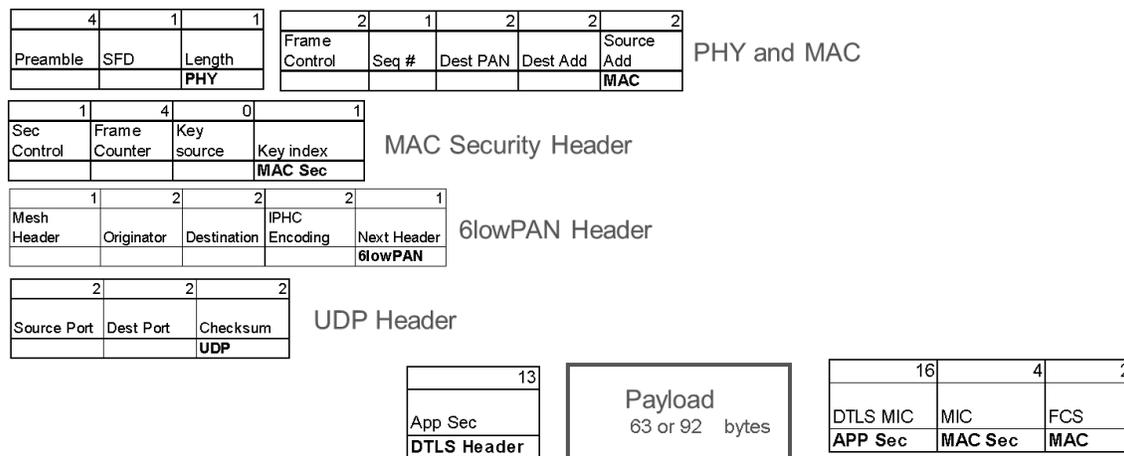


Figure 4. Complete Thread Data Packet

This packet results in an overhead of 35 bytes without application security and 64 bytes if DTLS (Datagram Transport Layer Security) security is added to the application (note this is optional). These packet formats determine the transmit time for a radio sending the data.

Time Required for Sending Messages

The discussion above details the basis for battery operated device behavior. However, actual measurements are required to validate how long a radio and software implementation takes to perform these operations.

Table 1 shows the time in milliseconds for Thread transactions from a battery-operated end device.



Table 1. Thread Wake Sleep Transaction Time

Event	Minimum Time (milliseconds)	Maximum Time (milliseconds)	Notes
Poll No Data	4.03	6.31	22-byte packet, with security
Send 10 bytes	4.41	8.2	
Send 50 bytes	8.38	9.94	97 byte packet with MAC security
Send 100 bytes	12.78	16.78	

All packets are using MAC security, including the data poll.

The difference between the minimum time and the maximum is the variation in the clear channel assessment timing.

The actual current consumption during these operations depend on the radio and MCU used and its active current profile.

For example on the Poll No Data the energy used averages 100 μ Coulombs. This means a device with a CR2032 battery waking every 10 seconds would last approximately 2.6 years.

References

Document	Title
[IEEE802154]	Wireless Personal Area Networks

