AN713

MEASURING EMBER® EM35X POWER CONSUMPTION FOR SLEEPY END DEVICES USING EMBERZNET PRO

(Formerly document 120-5068-000)

This document describes a technique for making detailed measurements of the power consumption of sleepy end devices running EmberZNet PRO on the Ember® EM35x SoC platform. It includes information about a test setup to use, some common usage scenarios, and details about what operations are observed in the measurements.

New in This Revision

Updated current consumption measurements to be more accurate for current software and reduce spurious current draw. Also updated test procedure.

Contents

1 Terminology and Basic Formulas

Energy is the capacity to do work and is measured in Joules. ***Power*** is the rate of doing work and is measured in Watts.

$$P = E / t$$

We are interested in characterizing the energy consumed by a ZigBee node as it performs certain operations. Energy can be represented as ***charge*** (Q, measured in Coulombs) at a specific ***voltage***:

$$E = Q * V$$

Charge itself is ***current*** flowing for a period of time:

$$Q = i * t$$

The charge in one Coulomb is equal to one Amp flowing for one second. For our low-current, short operations we will refer to our measurements in micro Coulombs (µC), the equivalent of one milliamp for one millisecond.

We can easily isolate the current flowing into our device using techniques described in the Test Setup section that follows. We will not directly measure this current but rather the voltage that falls across a resistor that the current flows through. Current can then be calculated with Ohm’s law:

$$I = V / R$$

We can sample the voltage across a resistor, convert the voltage to current and multiply the resulting current by the sample period to determine the charge in each sample. We can then integrate these charge values to determine the total charge for the entire duration of the operation. Later in this document we describe several ways to use an oscilloscope to perform this sampling.

This document focuses on measuring quantities of voltage and time in order to determine charge. We also show how convert charge to energy. We do not perform the final conversion from energy to power. Since specific active events (such as sending a data poll, as described later) have varying rates of energy consumption over a known duration, we choose to describe the total energy they consume. Power is more appropriate for steady-state conditions that persist for certain durations, such as deep sleep.

2 Test Setup

The power measurements described in this document can be taken using the following equipment:

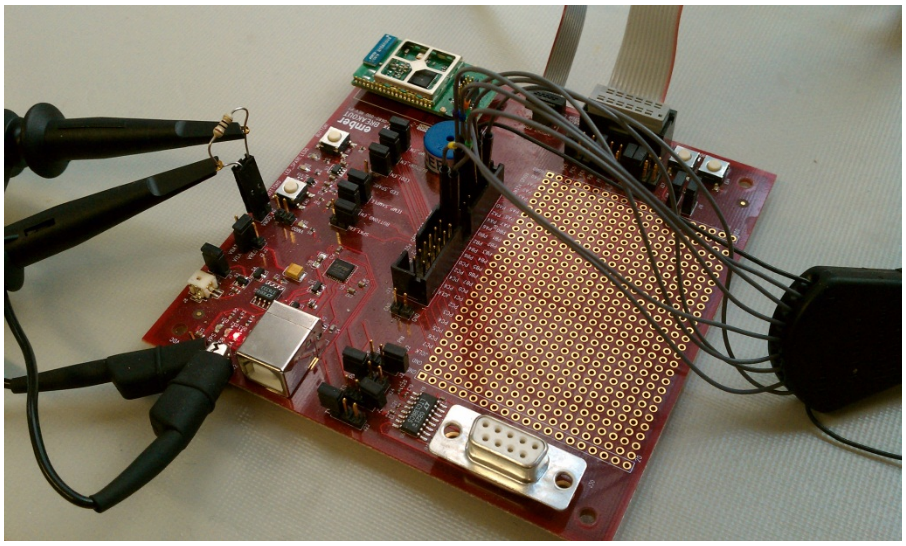
- EM35x breakout board with EM357 radio module (x2)
- Dual channel oscilloscope with math functionality
- 10 Ω resistor

One of the EM357 modules is configured as a sink node and the other as a sleepy sensor node, using the sample sink and sleepy sensor applications provided with Ember Desktop. For instructions on setting up and loading sample applications on the EM357, see the Ember EM35x Development Kit User Guide (UG110).

Replace the VMOD_EN jumper on the breakout board of the sleepy sensor with the 10 Ω resistor. This jumper supplies power to the radio module, and allows us to measure the current consumed by only the radio module without observing current consumed by other peripherals on the breakout board. The jumper going to the VMOD LED should also be removed so that its current consumption is not observed.

The oscilloscope is then set up with one probe on each side of the resistor. The oscilloscope is configured in a mode that subtracts the voltage observed on the module side of the resistor from the voltage observed on the power supply side. This configuration allows the voltage drop across the resistor to be measured directly and plotted over time.

This setup is shown in Figure 1. The photo also shows logic analyzer probes connected to some of the GPIO on the breakout board. These are not required, but are useful for observing PacketTrace and other timing signals on the same timebase as the current consumption profile.



3 Figure 1. Hardware Test Setup Measurement Procedure

The first step in measuring the charge in a given ZigBee event is to configure the oscilloscope to trigger at the start of the event with enough pre-trigger time so as to see the lead-up to the start of the event on the screen. The events that we are interested in all start with the node waking from deep sleep. At this time a significant regulator inrush current produces a large voltage drop across the resistor. The oscilloscope can be configured to trigger on the falling voltage on the node side of the resistor. Some oscilloscopes may also allow triggering on the math signal itself, in which case the trigger would be a rising voltage. If your oscilloscope has digital or unused analog inputs, it is also possible to trigger on the PacketTrace frame signal on GPIO port A pin 4, again with enough pre-trigger time so as to see the inrush current spike on the screen.

The timebase of the oscilloscope should be configured so that the entire event fits on the screen. The event is considered complete when the node has returned to deep sleep as indicated by the differential voltage signal going back to its quiescent (effectively zero) level. While the voltage samples before and after the event should be zero, and therefore should not contribute to the overall charge calculation, this is not always the case. Due to common-mode voltage effects, limited dynamic range, and other characteristics oscilloscopes are sometimes inaccurate at very low signal levels. To avoid including too many inaccurate low-level samples in the overall charge calculation the horizontal timebase of the oscilloscope should be configured to be as short as possible while still allowing the entire event to be captured on the screen. It is also advisable to position cursors at the start and end of the event and perform any subsequent measurements with cursor gating activated.

Once the voltage samples have been acquired for the event of interest, the next step is to integrate them over the duration of the event. The basic idea is to calculate the product of each voltage sample with the sample period. The sum total of these products is a quantity of Volt-seconds, which can be converted to Amp-seconds (or Coulombs) using Ohm’s law.

Some oscilloscopes (Tektronix in particular) can perform this integration directly through an “area” measurement function. Other oscilloscopes may be able to calculate and display an integral signal on the screen, which can then be measured with a cursor at the end of the event to determine the value that we are interested in. Other oscilloscopes may be able to calculate and display the mean of the voltage signal, which can be multiplied by the duration of the event to arrive at the same result.

If your oscilloscope does not support any of these functions you should be able to export the raw sample data to a file which can then be imported into a spreadsheet where you can perform the calculations described above.

4 Scenarios

The sleepy-sensor sample application is capable of presenting two common sleepy end device scenarios:

- Data Poll: Wake from deep sleep, send a data poll, and return to deep sleep as fast as possible. This is a typical operation often performed by a sleepy end device in order to query its parent for any new data.
- APS-Encrypted Packet: Wake from deep sleep, send an APS-encrypted packet with 27 bytes of application payload, and return to deep sleep as fast as possible. This scenario allows observation of the impact of encryption and larger packet sizes on overall current consumption.

For the purpose of this analysis the random CSMA backoff in both of these scenarios is forced to zero. It is assumed that in many situations the timing of an individual end device waking from deep sleep is generally random compared to the operation of the larger network of which it is a part, so the initial random backoff is not necessary. This may not be the case for all application and network designs.

These scenarios do not cover many other application behaviors, but they can be measured using the same technique.

5 Test Application

The sleepy-sensor application has a special test mode which facilitates analysis of these scenarios. This mode can be enabled with the ‘T’ command prior to joining a network. Activating this mode causes the application to change its behavior in the following ways:

- Disable all serial printouts, which would delay going back to sleep.
- Disable all use of LEDs, which would consume excess current.
- Shorten the deep sleep period, to allow easier oscilloscope triggering.
- Force initial backoff to 0 when waking from deep sleep.
- Change application behavior to:
    1. Wake up, send a data poll, return to deep sleep.
    2. Wake up again, send an APS-encrypted packet, return to deep sleep.
    3. Wake up again, send a data poll, receive APS ACK for secure packet, return to deep sleep.
    4. Start over from step 1.

Note: With most applications, it is better to combine steps 2 and 3 into a single operation without going back to deep sleep in between. For the purpose of analyzing the secure packet scenario independently, these were made separate.

6 Energy Consumption Analysis

The following oscilloscope captures were obtained with the sleepy-sensor application running EmberZNet PRO 4.5 using the test setup described earlier in this document. A Tektronix MSO2024 oscilloscope was configured to calculate:

1. The difference between the voltages on each side of the 10 Ω resistor. This is red trace labeled “M”.
2. The area under the “M” trace, labeled “M Area”.

An oscilloscope screenshot of the Data Poll scenario is shown in Figure 2.

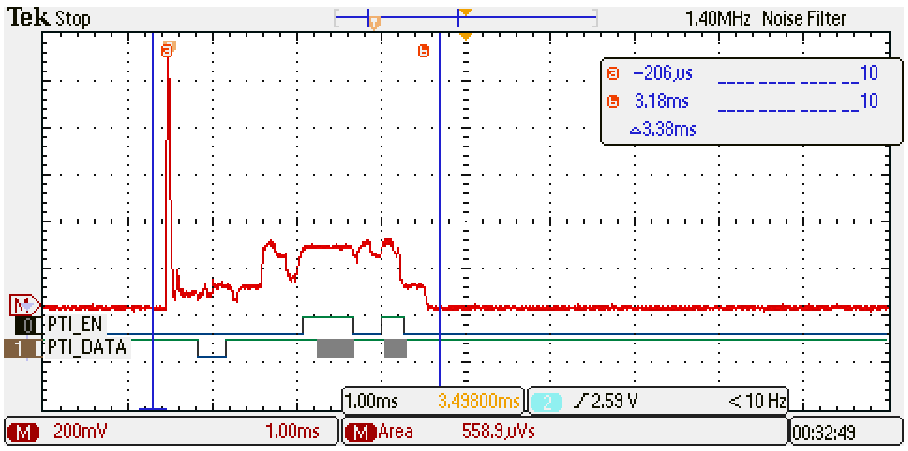


Figure 2. Data Poll Current Profile

The total charge for the data poll can be calculated by applying Ohm's law to the "M Area" quantity. The result is that the charge consumed by this particular data poll is 55.89 µC. Since the voltage supply to the node is 3.3 V, we can calculate the energy consumed in this operation to be 184.4 µJ.

An oscilloscope screenshot of the APS-Encrypted Packet scenario is shown in Figure 3.

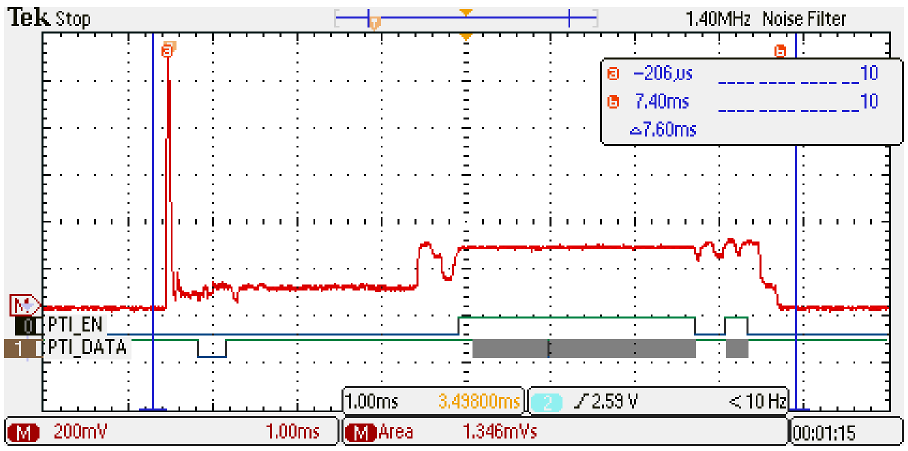


Figure 3. APS Encrypted Packet Current Profile

The total charge for the APS-Encrypted Packet can be calculated by applying Ohm's law for the "M Area" quantity. The result is that the charge consumed by this particular APS-Encrypted Packet is 134.6 µC. Since the voltage supply to the node is 3.3 V, we can calculate the energy consumed in this operation to be 444.2 µJ.

The Data Poll is just over 3 ms long, and is made up of a series of operations that are described in detail in Figure 4.

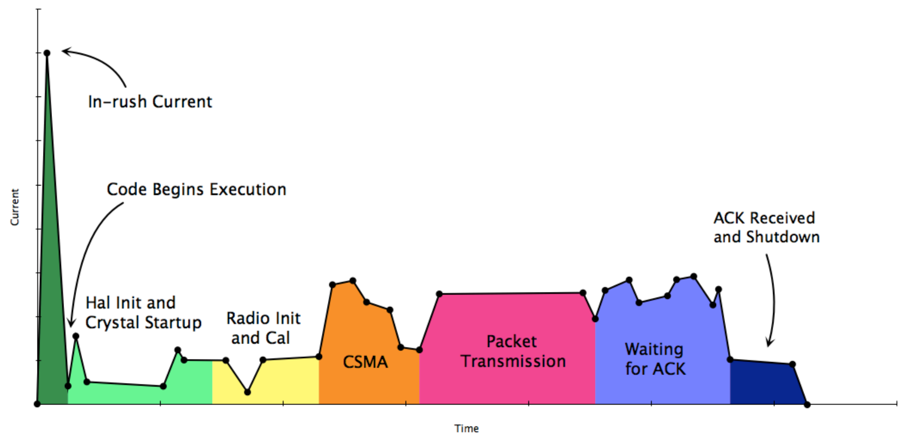


Figure 4. Operations Performed in Data Poll Scenario

The APS-Encrypted Packet is ~7.25 ms long. Its duration is longer than the Data Poll due to three factors:

- Time spent performing APS encryption after radio init, but before transmit preparation.
- Additional time spent performing encryption as part of transmit preparation.
- More time spent on the air transmitting the longer packet.

NOTES:

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Please visit the Silicon Labs Technical Support web page for ZigBee products:
www.silabs.com/zigbee-support and register to submit a technical support request

Patent Notice
Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog-intensive mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories, Silicon Labs, and Ember are registered trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.