WHITE PAPER

Rohit Kumar
Senior Applications Engineer
Cypress Semiconductor Corp.

Energy Harvesting and Bluetooth® Low Energy: Designing a Batteryless Beacon

Abstract

The market for Wireless Sensor Nodes and Beacons that help bring the Internet of Things to users is poised for explosive growth over the next five years. Ideally, these devices will be self-powered and able to be deployed anywhere, with minimal maintenance. This white paper looks at how to develop battery-less beacons using Energy Harvesting Power Management ICs (PMICs) and Bluetooth Low Energy technology.

Overview of Always-On, Zero Maintenance Data Transmitters

Smartphones have brought major changes to our day-to-day life. They are our gateway to on-demand information that relates to our health, environment, and even our purchases. Unfortunately, in most cases today, users have to initiate an action when they want data. This can be inefficient, since users may not know how to search for information (for example, offers on a product in a store).

The solution is to have a system that makes situation-specific data available to smartphones so that the right information can be presented to the user. This data may be from a sensor, or a unique identifier associated with an object or location. There is where beacons come into the picture.

In wireless terminology, a beacon is a device that broadcasts data that can be received by other devices in the vicinity. Ideally, the broadcast data can be received without user intervention, making the transfer seamless. Bluetooth® Low Energy has this capability, and thus has become the ubiquitous choice for beacon communication.

Bluetooth Low Energy (BLE) is widely used for low-power wireless communication in applications requiring the transfer of data within a relatively small radius, typically less than 10 meters. Consider a Wireless Sensor Node (WSN). Data can be collected by a WSN and sent to a smartphone. Figure 1 shows a typical application flow for these types of sensor nodes:

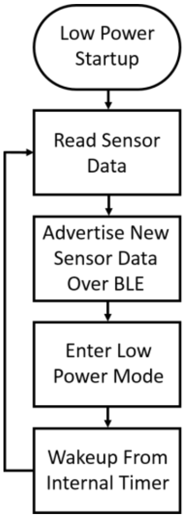

Low Power Startup
Read Sensor Data
Advertise New Sensor Data Over BLE
Enter Low Power Mode
Wakeup From Internal Timer

Figure 1. Typical Flow Diagram of BLE Sensor Devices

Beacons and sensors need to be powered from a source that allows them to function continuously while still maintaining the size and form factor of the overall device. Powering these devices from a wired source is rarely feasible because they are either positioned on a person's body or placed remotely. Therefore, any use case that requires wires for power typically does not make sense. Taking a battery-powered approach introduces issues such as limited operating life, the need to recharge batteries frequently, and a negative environmental impact arising from eventual battery disposal.

If we truly want beacons that do not require any kind of maintenance, then we need to utilize unharnessed energy from the surrounding environment such as light, motion, pressure, or heat. This will enable an "install and forget" approach where beacons and sensors can remain powered throughout the lifetime of the device.

Energy Harvesting

Energy harvesting is a method of collecting minute amounts of unharnessed energy from the surrounding environment and storing it. When enough energy has been saved, a sensor can perform tasks such as collecting data and transmitting it over BLE to another device.

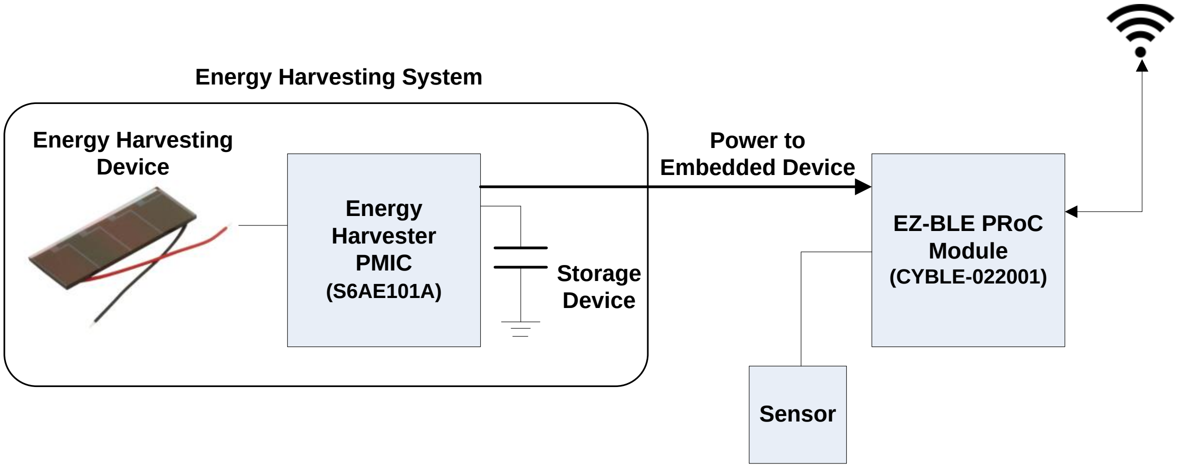


Figure 2. Energy Harvesting WSN Device Block Diagram

An Energy Harvesting System (EHS) is a circuit that includes an Energy Harvesting Device (EHD), an Energy Harvesting Power Management Integrated Circuit (PMIC), and an Energy Storage Device (ESD). The PMIC 'trickle' charges the Energy Storage Device (typically a capacitor) with energy provided by an EHD such as a solar cell, vibration sensor, or piezoelectric device. The EHS then uses this stored charge to provide energy to another embedded device. Depending on the state of the sensor, the EHS output power varies. When active, energy is consumed, and the voltage from the EHS begins to drop. When in a low-power state, the voltage from the EHS rises because the Energy Storage Device is being charged faster than it is being depleted. Figure 3 shows an example of an EHS output voltage varying with embedded device activity over a period of time.

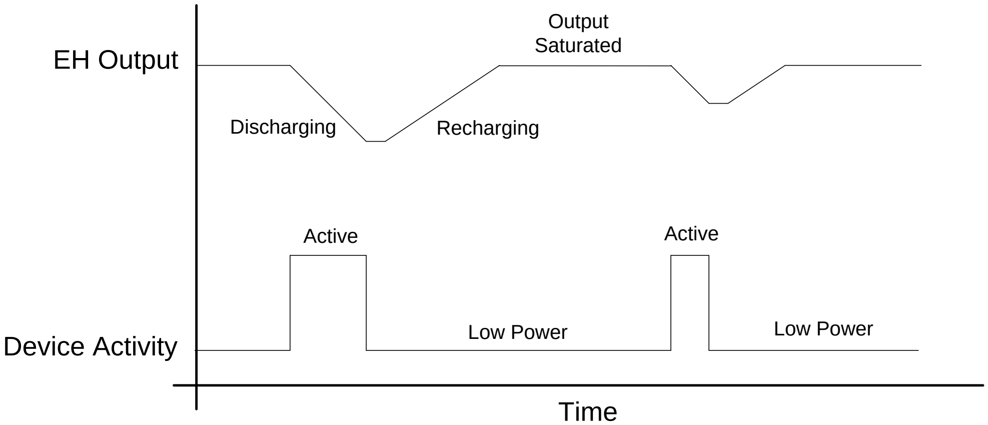


Figure 3. EH Output Variation Due to Device Activity over a Period of Time

For devices powered by an EHS, the energy consumed during an active state cannot exceed the available energy in the EHS. Figure 4 shows an EHS-powered system where the energy consumption during an active state is greater than the amount the EHS can provide. The output voltage of the EHS slowly drops due to consumption, eventually shutting down the output completely.

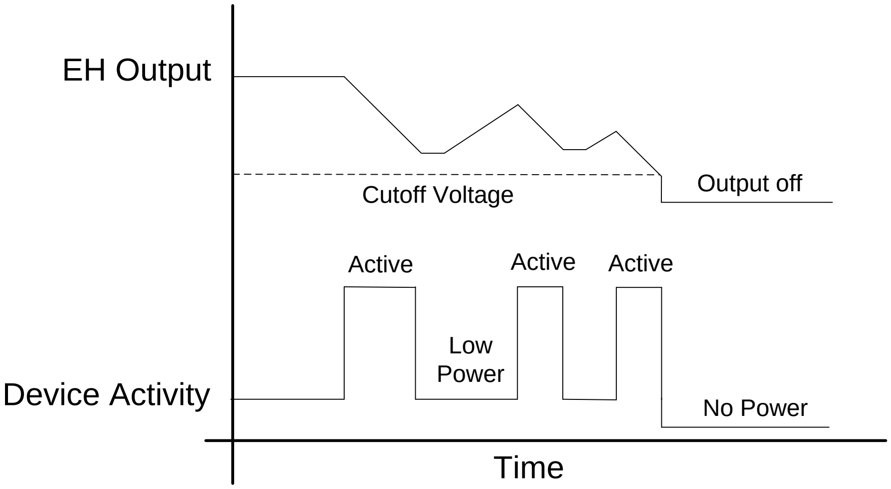


Figure 4. WSN Shutdown Due to Insufficient Power

To design a robust system powered by energy harvesting, every aspect of the embedded system needs to be optimized energy-wise so that it works seamlessly while being powered by the EHS. There are many subsystems in such a system that may be power-hungry and need to be optimized to ensure they do not pull down the output of the EH. Some of the key areas to target when optimizing power are:

1. CPU Clock Frequency

The system clock frequency decides how fast a particular routine is going to be processed and how much energy it is going to consume during that time. A faster clock means faster processing but higher current consumption. Also, each device has a certain minimum and maximum clock frequency requirement, which should not be violated.

For EHS-based designs, an optimized clock frequency has to be chosen with regards to the following two factors:

- Average current consumption
- Peak current consumption

The EHS capacity must consider both of these factors. The average current is the time average of current required during a particular active state. However, the peak current is the instantaneous maximum current requirement of the active state, and is often much higher than the average current. It may be possible that average current required is well within the capacity of the EHS, but that the peak current will cause the sudden energy depletion of the EHS, causing the voltage to fall below the cutoff voltage. Note that the processing time is part of the calculation of average current consumption.

Figure 5 shows the power versus time plot of a particular routine processed at a 48-MHz system frequency, and Figure 6 shows the power versus time plot of the same routine at 12 MHz.

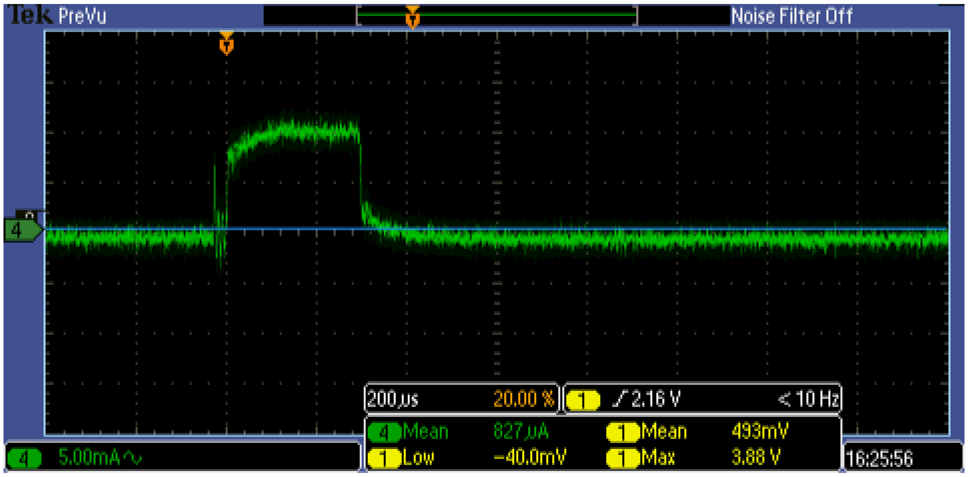


Figure 5. Current Consumption While Processing a Routine at 48 MHz

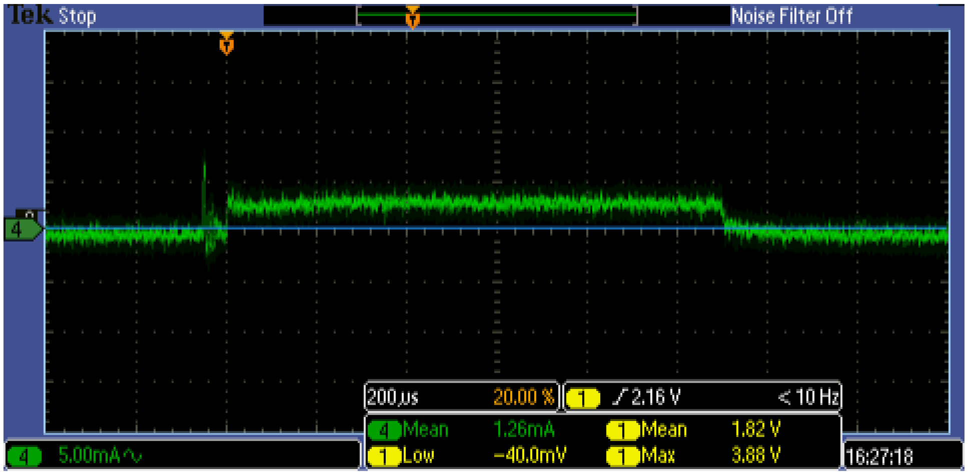


Figure 6. Current Consumption While Processing a Routine at 12 MHz

In this example, the 48-MHz-processed routine takes ~300 µs to complete and consumes about 10 mA peak current during this period. The 12-MHz-processed routine takes 1.1 ms to complete but consumes only 4 mA peak current. The average current consumed during the process is greater at 12 MHz but has a lower peak current requirement. Depending on the EHS capacity, one can go ahead with a short 48-MHz clock setup, a longer 12-MHz clock setup, or a mix of both, where the clock frequencies are switched from one process to another. Such current profiling should be taken into account while selecting the optimized system frequency.

2. Low-Power Device Bootup

When an embedded device is powered on, it must complete a bootup procedure before it can execute application code. A typical bootup sequence includes:

a. Initializing memory
b. Setting interrupt vectors
c. Configuring peripheral and common registers
d. Initializing external clocks, if any.

Each of these steps takes CPU processing time to complete, which in turns consumes energy. The amount of energy consumed depends on the type of device used, the system clock frequency, the size of the memory/register set being initialized, and the time it takes to set up external clocks. Thus, the bootup process is a power-intensive activity and should be optimized so that it does not consume an excessive amount of energy from the EH output. Factors that should be kept in mind while writing bootup code are:

a. Initialize only those sections of memory and registers which will be used. Leave others set to default values.

b. Most wireless systems need high-accuracy external clocks. These clocks, such as an External Clock Oscillator or a Watch Crystal Oscillator, have a large stabilization time after startup. Rather than wait in active mode for the clocks to stabilize, the system should be put in a low-power mode (sleep/deep sleep) and awakened only when the clocks are ready for use. Use an internal timer for this purpose.

3. Low-Power System Startup

Once the device begins executing application code, there is usually a need to start individual peripherals in the system. These peripherals could be internal to the device, such as an ADC, or may be external to the device, such as a sensor. While the starting time may not be large individually for peripherals, the combined overall set up time could require enough processing time to drain the stored energy in the EHS. First, calculate the startup time of individual peripherals for a given CPU frequency. Then determine if the energy budget for starting all peripherals together (faster) is feasible or whether the startup procedure needs to be executed in stages (slower).

4. Staged Application Processing

The device will have various application routines that require their own CPU bandwidth. These routines include tasks such as configuring a peripheral, receiving data from sensors, performing calculations, and managing events and interrupts. Ensure that the energy required for this processing does not exceed the capacity of the EHS. If it does, break routines into smaller subroutines and manage them in stages. This allows for the load on the EHS to be broken down into manageable current pulses that enable the EHS to recharge between active CPU processes.

Additionally, between each stage, place the system into a low-power mode with a wakeup source as an interrupt from a counter or watchdog timer. As the system will be in low-power modes most of the time, the current requirements during these modes must be as low as possible. The more efficient the system is in these modes, the less time the system will require to recharge between stages and the faster the system will be able to perform tasks.

5. Wireless Transmission

Once data has been collected, it has to be transmitted over BLE. This transmission can be done either on a BLE connection or BLE broadcast. Note that energy harvesting supported beaconing is limited to BLE advertisement. This is because it takes a larger amount of energy to set up a BLE connection before the connection can be used to transfer the data. In general, radio activity, either transmission (TX) or reception (RX), is the highest energy-consuming operation in a wireless device. Ensure that the BLE activity occurs as an independent process that it is grouped with other process only if the EH output can provide that much peak current.

Efficient Designs with Cypress PMICs and BLE solutions

Cypress's Energy Harvesting Power Management ICs (PMICs) enable a battery-free solution for wireless sensors and networks. Precise control over output power with efficient power conversion makes them an ideal choice for small, low-power BLE applications such as beacons and wireless sensor nodes. They can either be used in a battery-free solution, or in conjunction with a battery such as a Li-ion battery as a backup source of power. Optimized PMICs like the S6AE101A (solar or light EHD optimized) offer extremely low startup and quiescent power consumption, enabling the use of a tiny solar cell that minimizes the overall form factor. An EH PMIC such as the MB39C831 can start up from a low voltage and adapt to the power requirements of the application using a feature called Maximum Power Point Tracking (MPPT) functionality. MPPT allows the internal DC/DC converter to control the output charging by following the input power, thus maximizing the power output.

Cypress's Energy Harvesting PMICs fit a variety of applications. For example, the MB39C8xx family of PMICs supports solar-, vibration-, and thermal-based EH devices. For more complex systems, the S6AE10xA family of solar-optimized PMICs can control multiple outputs and storage devices.

Another aspect of a batteryless wireless beacon is the selection of an MCU. MCUs integrated as programmable systems, such as system-on-chip (SoC) devices with support for various low-power modes, are a good fit for such applications. For example, Cypress's Programmable System-on-Chip (PSoC®) and Programmable Radio-on-Chip (PRoC™) offers tight integration with various types of peripherals that can be used to interface to sensors. The Cypress PSoC 4 BLE and PRoC BLE, in particular, contain low-power peripherals along with a BLE radio and integrated BLE stack, enabling the design of a single-chip BLE sensor node. Also, the support for ultra-low-power modes allows systems to operate reliably using constrained power sources such as Energy Harvesters and coin cells. Such harvesters, along with PSoC, prove to be an optimized design for batteryless BLE sensor node applications.

Along with the low-power modes and multiple peripherals, another advantage offered by the PSoC 4 BLE and PRoC BLE devices is the versatile PSoC Creator™ Integrated Design Environment (IDE). PSoC Creator is free-of-cost and provides a complete ecosystem for seamless embedded solutions development by providing an intuitive GUI-based interface, easy-to-configure Components (free embedded ICs represented by an icon in the IDE), and integrated programmer/debugger tools to rapidly develop and test your applications. Cypress has simplified the BLE protocol stack and profile configuration into a royalty-free BLE Component that can be dragged and dropped into designs in PSoC Creator.

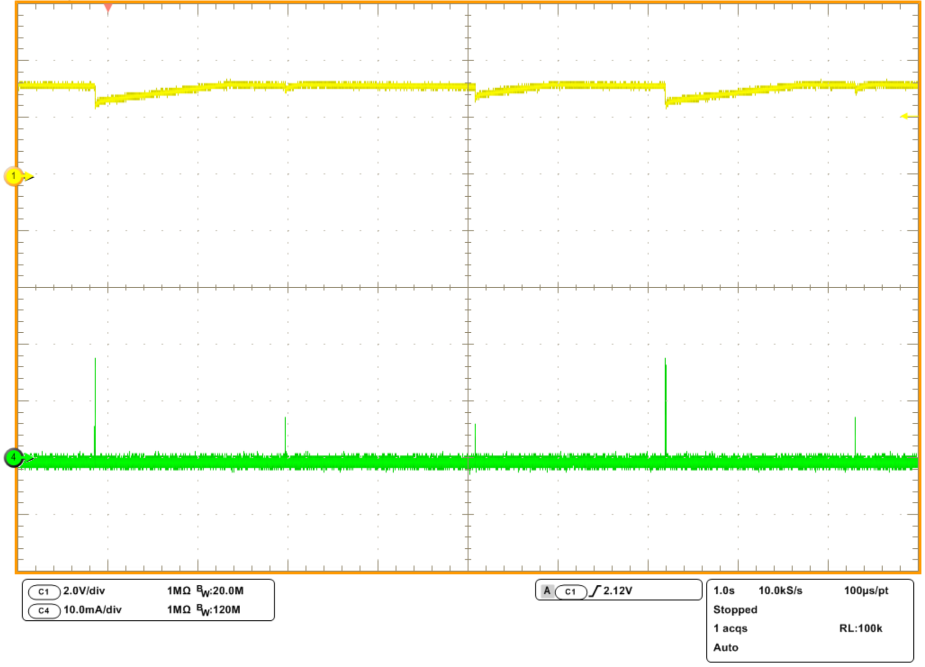
For BLE sensor nodes, applications range from simple beacons to managing multiple sensors and data transmissions. PSoC Creator streamlines development by allowing designers to drag in the Components they need for an application, configure them over the simple GUI, and write a few lines of code to set up the firmware. Everything else is taken care of by the tool. Additionally, Cypress offers fully-certified BLE Modules. The Cypress BLE solutions are backed by numerous code examples, application notes and hardware reference designs to further simplify application development.

To learn more about designing efficient wireless systems for energy harvesting applications, see the application note, *Getting Started with Bluetooth Low Energy*. For more detailed information on optimizing BLE subsystems for energy harvesting, see *Designing for Low Power and Estimating Battery Life for BLE-based Applications*. In addition, click here for an introduction to Power Management Integrated Circuits.

Appendix

A1: Oscilloscope screenshots of various processes in BLE sensor node powered from EH

This figure shows the EHS output voltage variation with respect to CPU processing over a period of time. The yellow signal is the EHS output voltage, while the green signal shows the current consumption by the embedded device. The green peaks are current consumption during the CPU active process. The flat signal occurs while in the device's low-power mode. Note that the EHS output voltage drops with every CPU activity (peaks in green signal) due to the energy consumed by the CPU. Also note that the voltage recovers during the low-power states as the EHS recharges the Energy Storage Device.



C1 2.0V/div 1MΩ BW:20.0M
C4 10.0mA/div 1MΩ BW:120M

A C1 ∫ 2.12V

1.0s 10.0kS/s 100µs/pt
Stopped
1 acqs RL:100k
Auto

Figure 7. EHS Output Voltage Variation with Respect to CPU Processing

This figure shows the EHS output voltage with respect to CPU activity without recharging the Energy Storage Device in the EHS. Note that as the energy is depleted, the voltage drops below the cutoff voltage, at which point the EHS output is shut down.

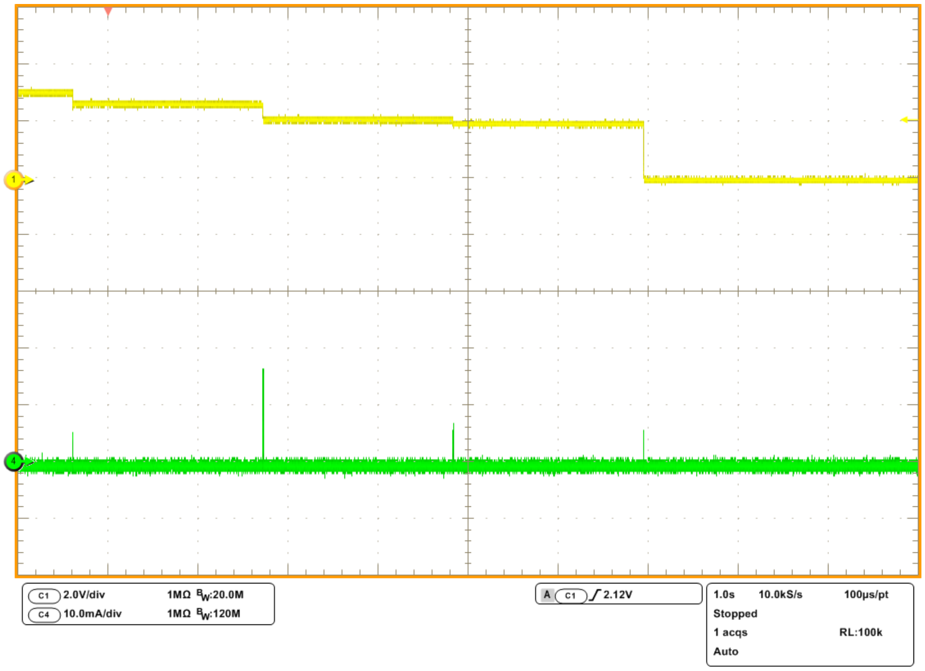


Figure 8. EHS Output Voltage with Respect to CPU Activity without Recharging the ESD

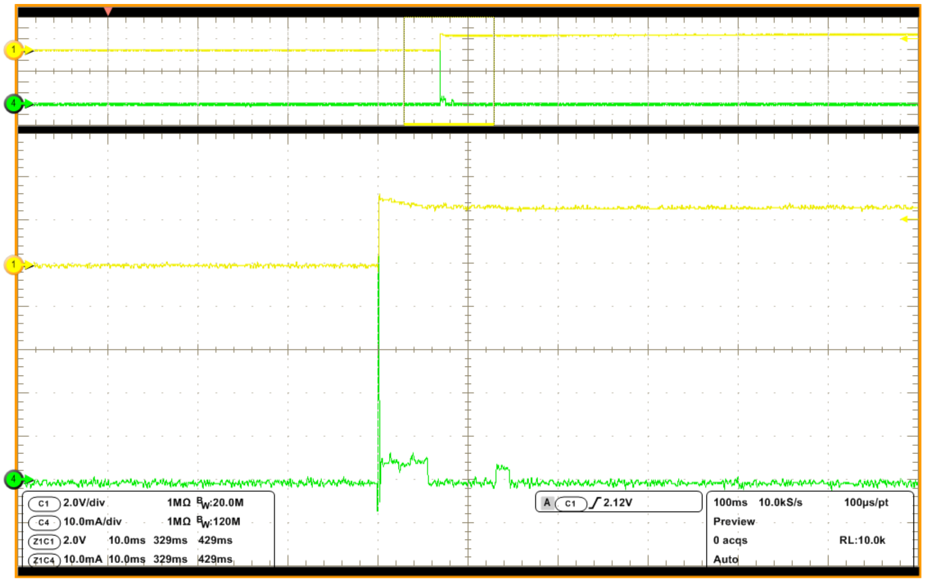


Figure 9. Current Consumption (Green Signal) at Device Bootup

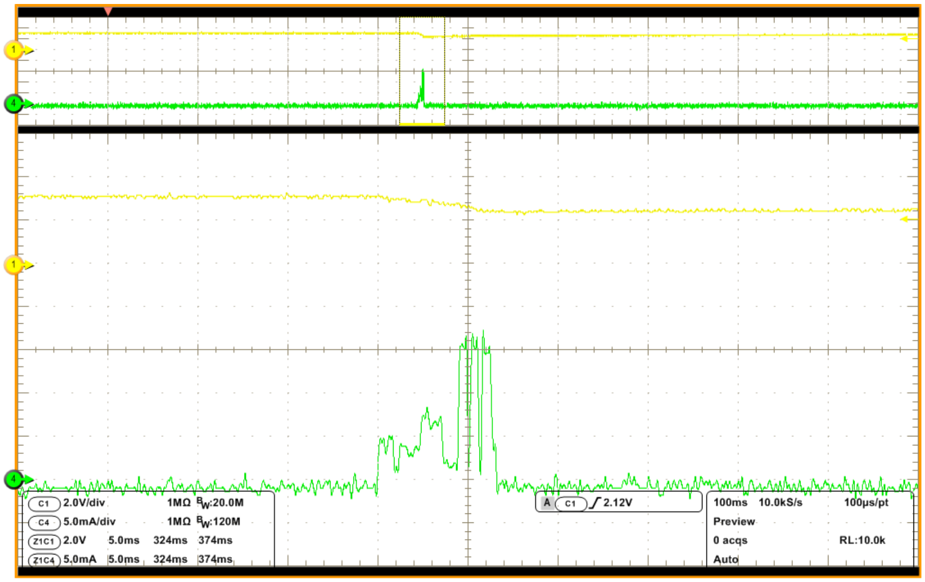


Figure 10. BLE Transmit Activity in a Beacon Powered by Energy Harvesting

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.