S6SAE101A00SA1002

Solar-Powered Internet of Things (IoT) Device Kit User Guide

Document No. 002-00297 Rev. **

Cypress Semiconductor

198 Champion Court

San Jose, CA 95134-1709

Phone (USA): 800.858.1810

Phone (Intnl): 408.943.2600

www.cypress.com

Preface

This manual explains how to use the evaluation board. Be sure to read this manual before using the product. For this product, please consult with sales representatives or support representatives.

Handling and Use

Handling and use of this product and notes regarding its safe use are described in the manuals.

Follow the instructions in the manuals to use this product.

Keep this manual at hand so that you can refer to it anytime during use of this product.

Notice on This Document

All information included in this document is current as of the date it is issued. Such information is subject to change without any prior notice.

Please confirm the latest relevant information with the sales representatives.

Caution of the Products Described in This Document

The following precautions apply to the product described in this manual.

⚠ **WARNING**	Indicates a potentially hazardous situation which could result in death or serious injury and/or a fault in the user's system if the product is not used correctly.

Electric shock, Damage	Before performing any operation described in this manual, turn off all the power supplies to the system. Performing such an operation with the power on may cause an electric shock or device fault.
Electric shock, Damage	Once the product has been turned on, do not touch any metal part of it. Doing so may cause an electric shock or device fault.

⚠ **CAUTION**	Indicates the presence of a hazard that may cause a minor or moderate injury, damages to this product or devices connected to it, or may cause the loss of software resources and other properties such as data, if the device is not used appropriately.

Cuts, Damage	Before moving the product, be sure to turn off all the power supplies and unplug the cables. Watch your step when carrying the product. Do not use the product in an unstable location such as a place exposed to strong vibration or a sloping surface. Doing so may cause the product to fall, resulting in an injury or fault.
Cuts	The product contains sharp edges that are left unavoidably exposed, such as jumper plugs. Handle the product with due care not to get injured with such pointed parts.
Damage	Do not place anything on the product or expose the product to physical shocks. Do not carry the product after the power has been turned on. Doing so may cause a malfunction due to overloading or shock.
Damage	Since the product contains many electronic components, keep it away from direct sunlight, high temperature, and high humidity to prevent condensation. Do not use or store the product where it is exposed to much dust or a strong magnetic or electric field for an extended period of time. Inappropriate operating or storage environments may cause a fault.
Damage	Use the product within the ranges given in the specifications. Operation over the specified ranges may cause a fault.
Damage	To prevent electrostatic breakdown, do not let your finger or other object come into contact with the metal parts of any of the connectors. Before handling the product, touch a metal object (such as a door knob) to discharge any static electricity from your body.
Damage	When turning the power on or off, follow the relevant procedure as described in this document. Before turning the power on, in particular, be sure to finish making all the required connections. Furthermore, be sure to configure and use the product by following the instructions given in this document. Using the product incorrectly or inappropriately may cause a fault.
Damage	Because the product has no casing, it is recommended that it be stored in the original packaging. Transporting the product may cause a damage or fault. Therefore, keep the packaging materials and use them when re-shipping the product.

Table of Contents

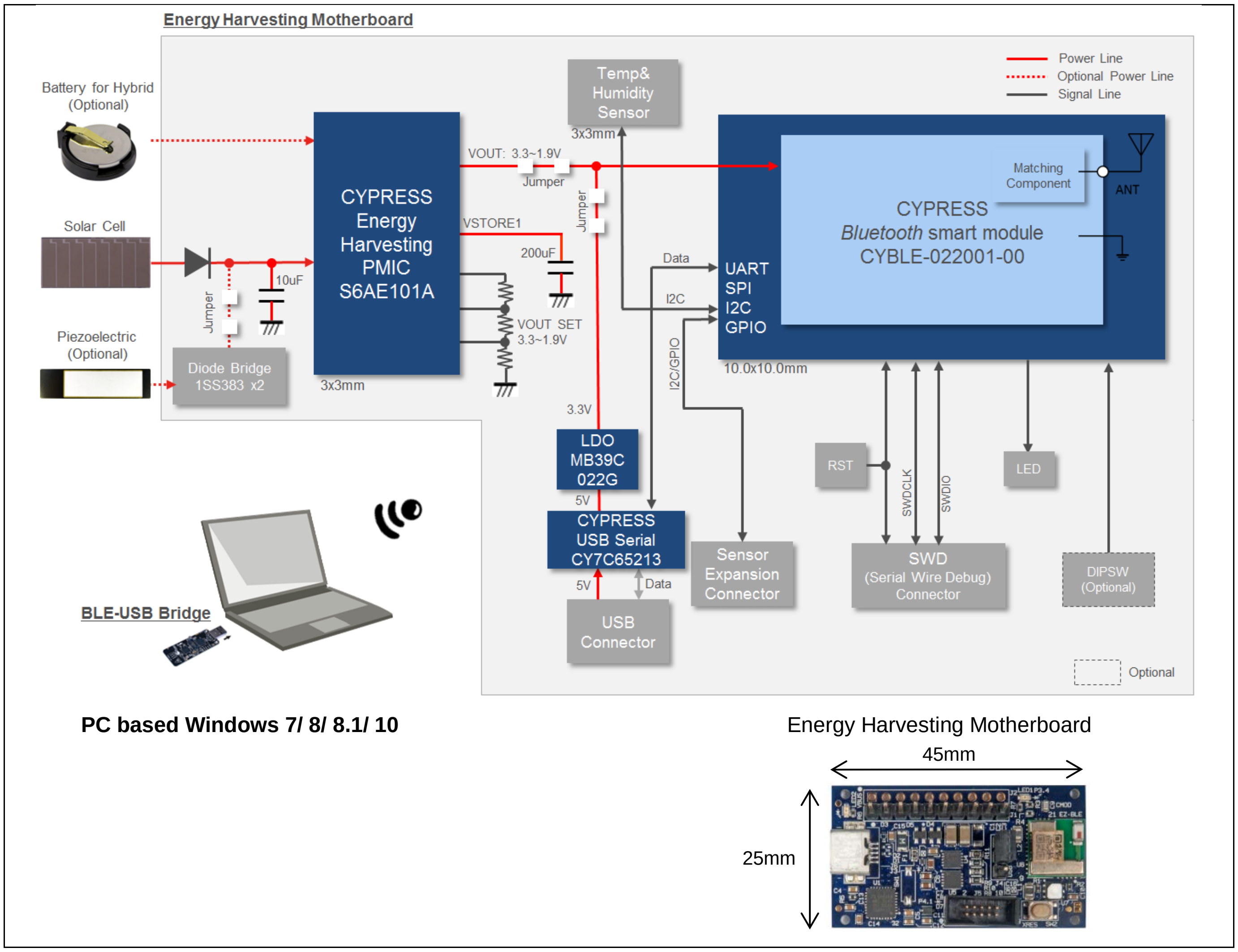
Figures

Tables

1. Description

The Solar-Powered IoT Device Kit provides an easy-to-use platform for the development of a solar-powered IoT device with BLE wireless connectivity. It includes the S6AE101A Energy Harvesting Power Management IC (PMIC) device, which is ideal for solar- or light-powered Energy Harvesting Systems (EHS) since it only consumes 250nA. The S6AE101A also supports a hybrid EHS that uses a solar cell Energy Harvesting Device (EHD) along with a coin cell battery, and an optional vibration EHD with external diode bridge. The output voltage from the S6AE101A is configurable from 1.1V to 5.2V, supporting a broad range of device components for an IoT device. Also included in the kit is Cypress' EZ-BLE™ PRoC™ Module (CYBLE-022001-00), a fully integrated Bluetooth Low Energy (BLE) module solution that offers high flexibility for a wide variety of IoT device uses. A USB port is provided by Cypress' USB-UART LP Bridge Controller device (CY7C65213).

Figure 1-1 Block Diagram of Development Kit



2. Features

The Solar Powered IoT Device Kit provides everything needed to develop a light-powered sensor node that transmits sensor data using BLE:

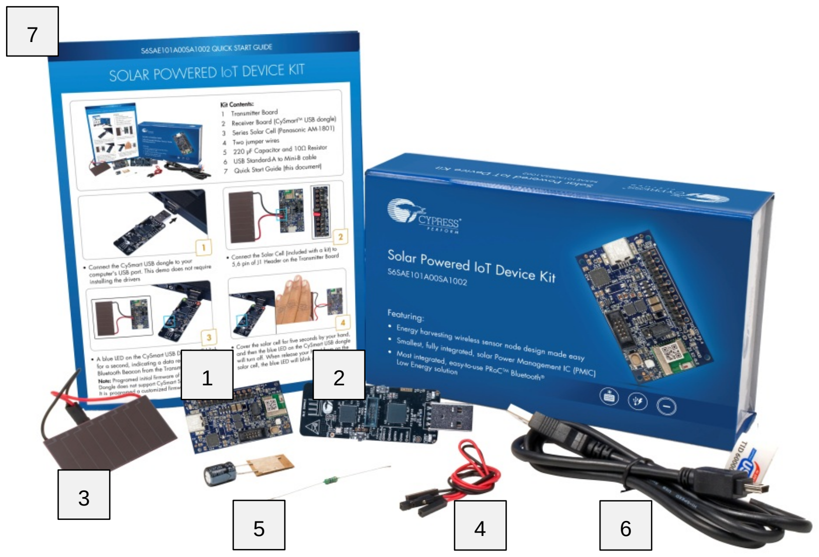
- Operates using light (>200 lux) energy harvested by the included solar cell
  - Supports the use of a vibration Energy Harvesting Device with an external diode bridge (not included)
  - Also supports the use of a coin cell battery (optional, not supplied)
- Supports BLE communication with a PC through the provided BLE-USB Bridge that is pre-programmed with custom firmware for this kit
- Includes firmware that supports the following applications:
  - Bluetooth Low Energy (BLE) Beacon, transmitting data at 1.5 sec intervals with ambient light as low as 200 lx
  - Wireless Sensor Node (WSN), transmitting data at 6 sec intervals with ambient light as low as 200 lx
- Includes an expandable terminal on the Motherboard that can support the following:
  - Reset button for EZ-BLE Module
  - JTAG header to debug EZ-BLE Module
  - Expandable sensor interface (I^2C/UART/SPI/GPIO)
  - DIP switch for future expansion (Not mounted)
  - LEDs for USB power and status
- Includes reference schematic, BOM list, and layout data for easy design
- Uses the following Cypress Devices:
  - S6AE101A ultra low power Energy Harvesting PMIC
  - CYBLE-022001-00 EZ-BLE PRoC Module
  - CY7C65213 USB-UART LP Bridge Controller
  - MB39C022G LDO

3. Applications

- Battery-less wireless sensor node (WSN)
- IoT device that monitors various sensors
- BLE Beacon
- Wearable device
- Building Energy Management System (BEMS)
- Home Energy Management System (HEMS)
- Factory Energy Management System (FEMS)
- Wireless lighting control
- Wireless HVAC sensor
- Security system

4. Kit Introduction

4.1 Contents



1. Energy Harvesting Motherboard
2. BLE-USB Bridge
3. Solar Module (Panasonic AM-1801)
4. Two jumper wires
5. 220 μF Capacitor and 10Ω Resistor[1]
6. USB Standard-A to Mini-B cable
7. Quick Start Guide

[1] The 220 μF capacitor is an additional output capacitor. The 10Ω resistor is for current measurement. Refer to "11.2 How to Use Extra Components" for detailed information.

5. Software Installation

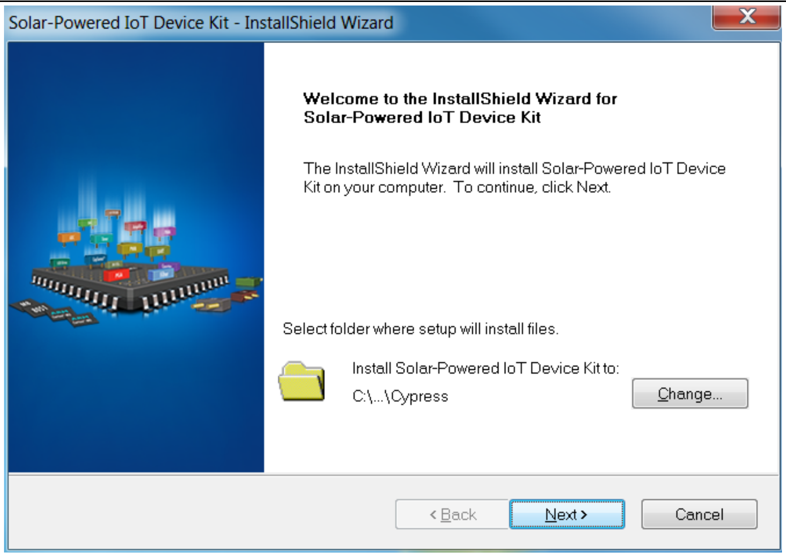
5.1 Install Software

Follow these steps to install the S6SAE101A00SA1002 Solar-Powered IoT Device Kit software:

1. Download and install the Solar-Powered IoT Device Kit software from www.cypress.com/energy-harvesting.
   The Solar-Powered IoT Device Kit software is available in two different formats for download:
   a. Solar-Powered IoT Device Kit Complete Setup: This installation package contains the files related to the kit. However, it does not include the Windows Installer or Microsoft .NET framework packages. If these packages are not on your computer, the installer directs you to download and install them from the Internet.
   b. Solar-Powered IoT Device Kit Only Package: This executable file installs only the kit contents, which include code examples, hardware files, and user documents. This package can be used if all the software prerequisites are installed on your computer.
   c. Solar-Powered IoT Device Kit ISO: This file is a complete package, stored in a CD-ROM image format that can be used to create a CD, or extract using ISO extraction programs, such as WinZip or WinRAR. This file includes all the required software, utilities, drivers, hardware files, and user documents.

2. Run Install Solar-Powered IoT Device Kit to start the installation, as shown below.

3. Select the folder to install the Solar-Powered IoT Device Kit-related files. Choose the directory and click Next.



Solar-Powered IoT Device Kit - InstallShield Wizard

Welcome to the InstallShield Wizard for Solar-Powered IoT Device Kit

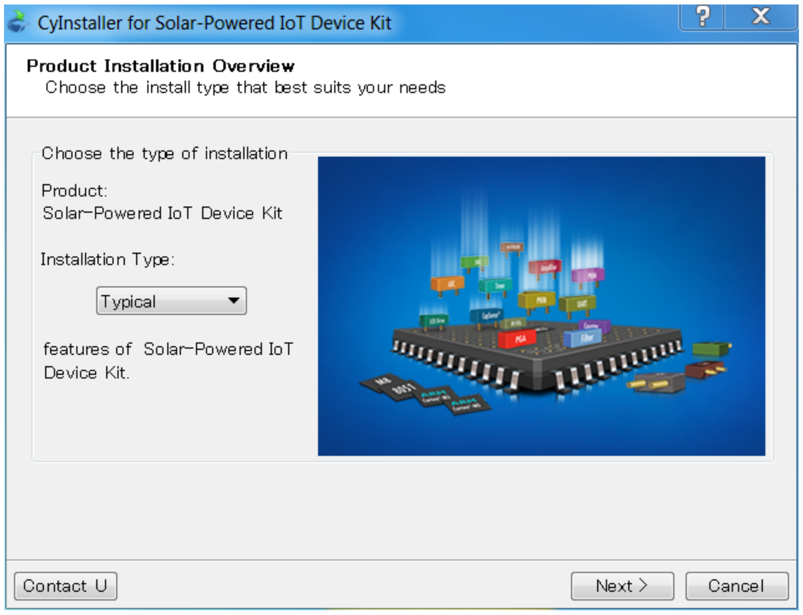
The InstallShield Wizard will install Solar-Powered IoT Device Kit on your computer. To continue, click Next.

Select folder where setup will install files.

Install Solar-Powered IoT Device Kit to:
C:\...\Cypress

4. The Solar-Powered IoT Device Kit ISO installer automatically installs the required software, if it is not present on your computer. The Solar-Powered IoT Device Kit Setup installer directs you to download the required software from the Internet.

5. Choose the Typical/Custom/Complete installation type in the Product Installation Overview window. Click Next after you select the installation type.



CyInstaller for Solar-Powered IoT Device Kit

Product Installation Overview
Choose the install type that best suits your needs

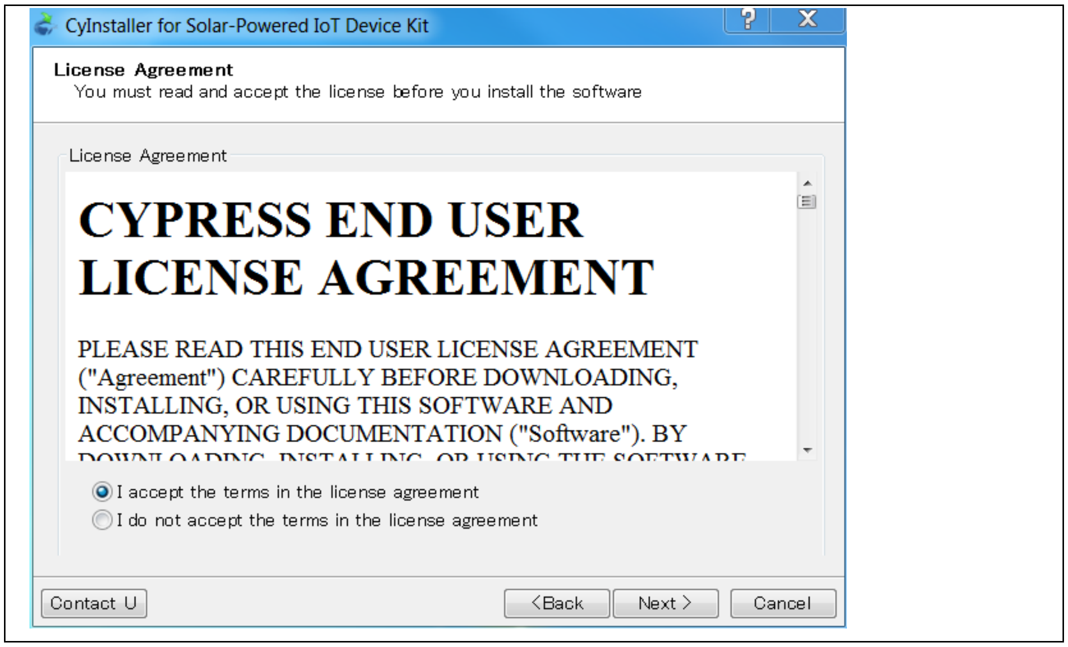
Choose the type of installation

Product:
Solar-Powered IoT Device Kit

Installation Type:
Typical

features of Solar-Powered IoT Device Kit.

6. Read the Cypress License Agreement and make a selection based on the terms of the license agreement. Click Next to continue the installation.


CyInstaller for Solar-Powered IoT Device Kit
License Agreement
You must read and accept the license before you install the software
License Agreement

CYPRESS END USER LICENSE AGREEMENT

PLEASE READ THIS END USER LICENSE AGREEMENT ("Agreement") CAREFULLY BEFORE DOWNLOADING, INSTALLING, OR USING THIS SOFTWARE AND ACCOMPANYING DOCUMENTATION ("Software"). BY

[✓ I accept the terms in the license agreement]
I do not accept the terms in the license agreement

Contact U
<Back
Next >
Cancel

7. When the installation begins, a list of packages appears on the installation page. A green check mark appears next to each package after successful installation.

8. Click Finish to complete the Solar-Powered IoT Device Kit installation.

9. Enter your contact information or select the Continue Without Contact Information check box. Click Finish to complete the Solar-Powered IoT Device Kit installation.

10. After the installation is complete, the kit contents are available at the following location:
<Install directory>\ Solar-Powered IoT Device Kit
Default location (Example. Windows 7):
64-bit: C:\Program Files (x86)\Cypress\ Solar-Powered IoT Device Kit
32-bit: C:\Program Files\Cypress\ Solar-Powered IoT Device Kit

5.2 Uninstall Software

You can uninstall the Solar-Powered IoT Device Kit software using one of the following methods:

Example. Windows 7

- Go to Start > All Programs > Cypress > Cypress Update Manager; click the Uninstall button.
- Go to Start > Control Panel > Programs and Features. Select the Solar-Powered IoT Device Kit program from the list and click the Uninstall/Change button.

5.3 PSoC Creator™

PSoC Creator[1] is a state-of-the-art, easy-to-use integrated design environment (IDE). It is a revolutionary hardware and software co-design environment, powered by a library of preverified and precharacterized PSoC Components™. With PSoC Creator, you can:

- Drag and drop PSoC Components to build a schematic of your custom design
- Automatically place and route components and configure GPIOs
- Develop and debug firmware using the included component APIs

PSoC Creator also enables you to tap into an entire tool ecosystem with integrated compiler chains and production programmers for PSoC devices.

PSoC Creator 3.2 SP1 or newer: Download the latest version from www.cypress.com/psoccreator.

For sample firmware information for this kit, refer to "8. Example Project".

[1] To develop firmware for the Solar-Powered IoT Device Kit, require PSoC Creator 3.2 SP1 or newer.

6. Getting Started

You will become familiar with the different components of the Solar-Powered IoT Device Kit by successfully establishing a BLE Beacon connection between the Energy Harvesting Motherboard operating as a WSN, and a PC with the BLE-USB Bridge. This will also confirm that the Motherboard, BLE-USB Bridge, and your PC are operating properly.

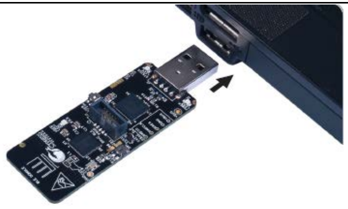
6.1 Solar-Powered BLE Beacon Operation

You will confirm that the Motherboard and BLE-USB Bridge are properly operating by establishing a BLE connection. The Motherboard contains all of the components of the WSN which are: the energy harvesting PMIC S6AE101A; capacitors for energy storage; an EZ-BLE PRoC Module for transmitting data; and an I^2C temperature and humidity sensor. A USB to serial device is also included on the Motherboard to allow the user to configure parameters such as the ID into the EZ-BLE module from a PC application. The Motherboard comes with pre-loaded firmware to operate as a BLE Beacon. By connecting the Solar Module to the Motherboard and exposing it to ambient light, it will power up and begin transmitting.

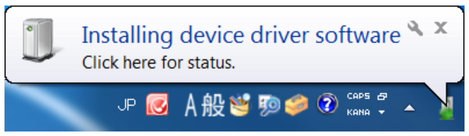
The BLE-USB Bridge is pre-configured to look for the transmission from the Motherboard operating as a BLE Beacon. By installing the BLE-USB Bridge on a Windows PC and using the provided software, you will be able to detect the Motherboard, and determine the distance between the Motherboard and the PC using BLE.

6.1.1 USB Driver Installation of the BLE-USB Bridge

1. Plug in the BLE-USB Bridge into your computer's USB port.

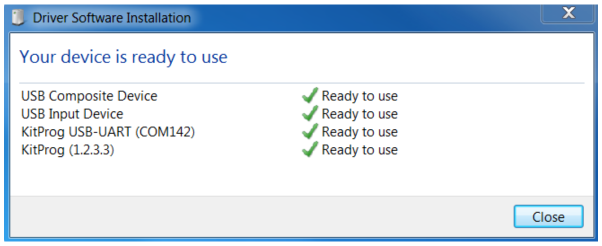


2. The driver Installation starts automatically and the following message window will appear. Click the message window for status.



Installing device driver software
Click here for status.

3. Confirm that the device driver installation has successfully completed (all components will be "Ready to use"). If the installation fails, do installation manually using a file in the "USB drivers" folder. Refer to "6.1.2 USB Driver Installation failed" below.



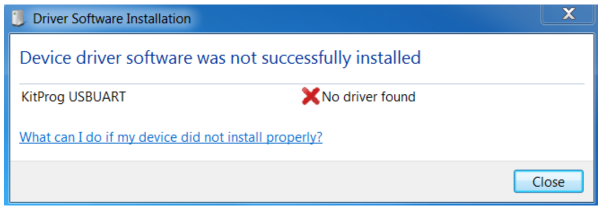
Driver Software Installation

Your device is ready to use

USB Composite Device	✓ Ready to use
USB Input Device	✓ Ready to use
KitProg USB-UART (COM142)	✓ Ready to use
KitProg (1.2.3.3)	✓ Ready to use

Completed

(Next to step4)



Driver Software Installation

Device driver software was not successfully installed

KitProg USBUART	✗ No driver found

What can I do if my device did not install properly?

Failed

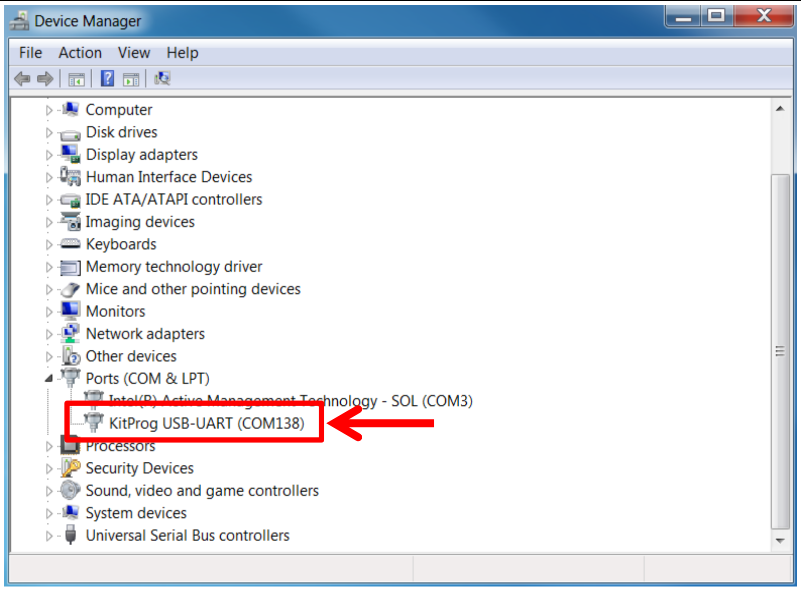
(Refer to "6.1.2 USB Driver Installation failed")

4. After successful device driver installation, confirm that a new COM port called KitProg USB-UART was added:

   a. Open the Device Manager:

Windows 7	Start Menu > Control Panel > Device Manager
Windows 8/8.1/10	Right Click Start Button > Device Manager

   b. Under Ports (COM & LPT), confirm that a COM port called KitProg USB-UART was added. Note the COM number (COMxx).



Device Manager

- Computer
- Disk drives
- Display adapters
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Imaging devices
- Keyboards
- Memory technology driver
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
- Ports (COM & LPT)
  - Intel(R) Active Management Technology - SOL (COM3)
  - KitProg USB-UART (COM138)
- Processors
- Security Devices
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers

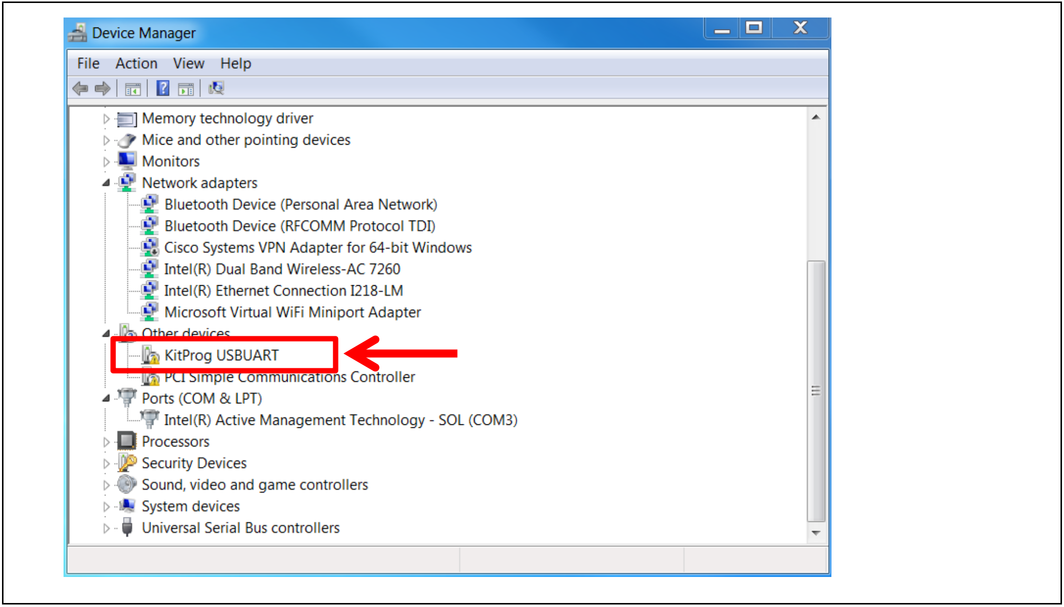
5. Skip the "5.1.2 USB Driver Installation failed", continue with the "6.1.3 USB Driver Installation for the Motherboard".

6.1.2 USB Driver Installation failed

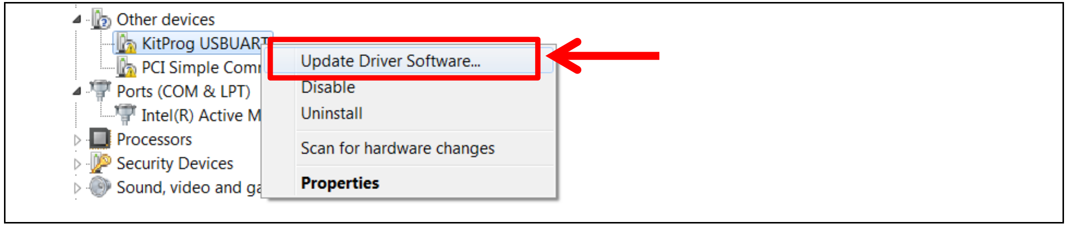
1. If the device driver installation fails, confirm that an unconfigured KitProg USB-UART appears in the device manager:
   a. Open the Device Manager:

Windows 7	Start Menu > Control Panel > Device Manager
Windows 8/8.1/10	Right Click Start Button > Device Manager

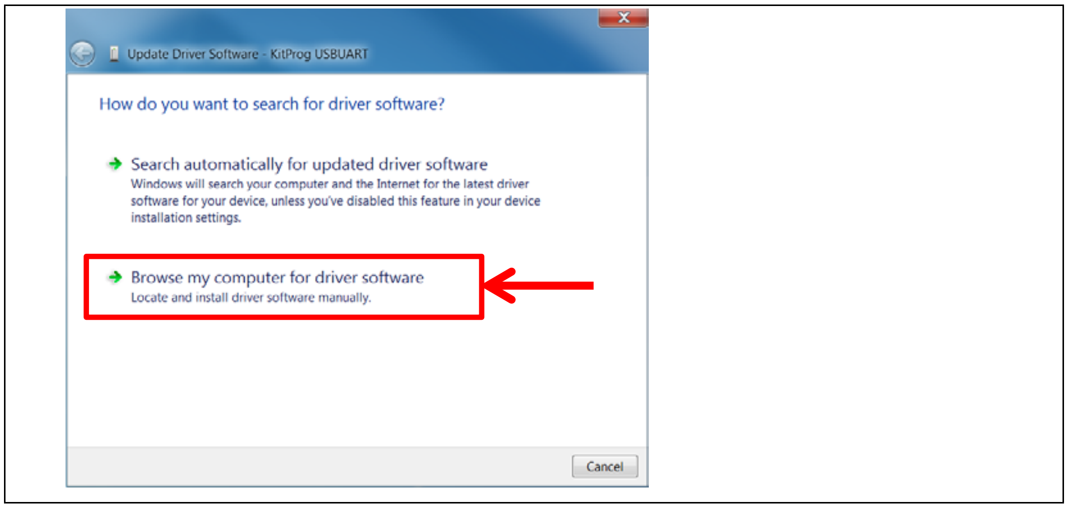
   b. Under Other devices, confirm that KitProg USB-UART appears with no associated COM port.


Device Manager
File Action View Help
Memory technology driver
Mice and other pointing devices
Monitors
Network adapters
Bluetooth Device (Personal Area Network)
Bluetooth Device (RFCOMM Protocol TDI)
Cisco Systems VPN Adapter for 64-bit Windows
Intel(R) Dual Band Wireless-AC 7260
Intel(R) Ethernet Connection I218-LM
Microsoft Virtual WiFi Miniport Adapter
Other devices
KitProg USBUART
PCI Simple Communications Controller
Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM3)
Processors
Security Devices
Sound, video and game controllers
System devices
Universal Serial Bus controllers

2. Update the USB driver software or the unconfigured KitProg USB-UART.
   a. Click the right mouse button on KitProg USB-UART
   b. Select Update Driver Software…



3. Select “Browse my computer for driver software”.

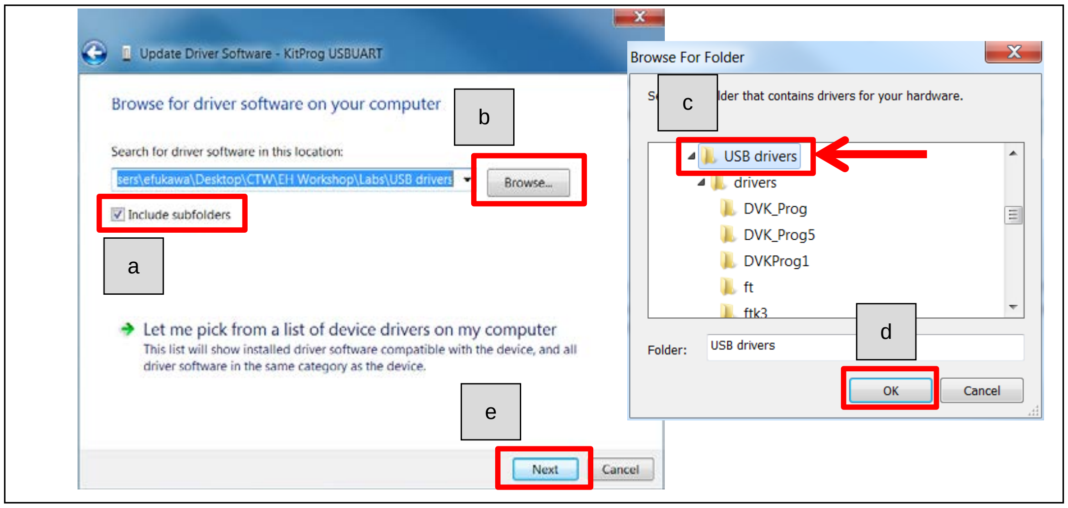

Update Driver Software - KitProg USBUART

How do you want to search for driver software?

Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software
Locate and install driver software manually.

4. Search for USB driver in the Lab files.
   a. Check the “Include subfolders” box
   b. Push the “Browse my computer for driver software” button
   c. Select the “USB drivers” folder in the kit files
      <Install directory>/Solar-Powered IoT Device Kit/1.0/USB drivers
   d. Push the “OK” button
   e. Push the “Next” button


Update Driver Software - KitProg USBUART

Browse for driver software on your computer

Search for driver software in this location:
sers\efukawa\Desktop\CTW\EH Workshop\Labs\USB drivers

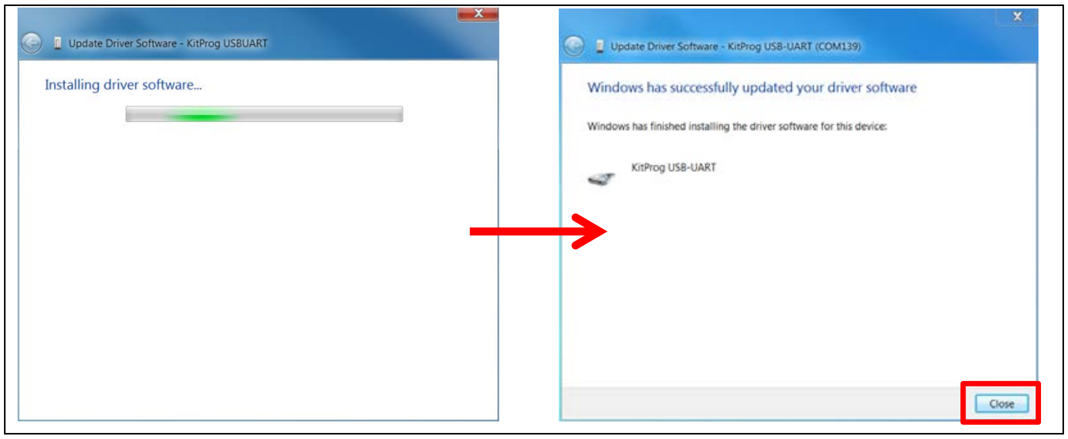
Include subfolders [✓]

Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Browse For Folder
Se[c]lder that contains drivers for your hardware.

Folder: USB drivers

5. Start installing USB driver. Push the “Close” button when the driver installation of the KitProg USB-UART finishes.

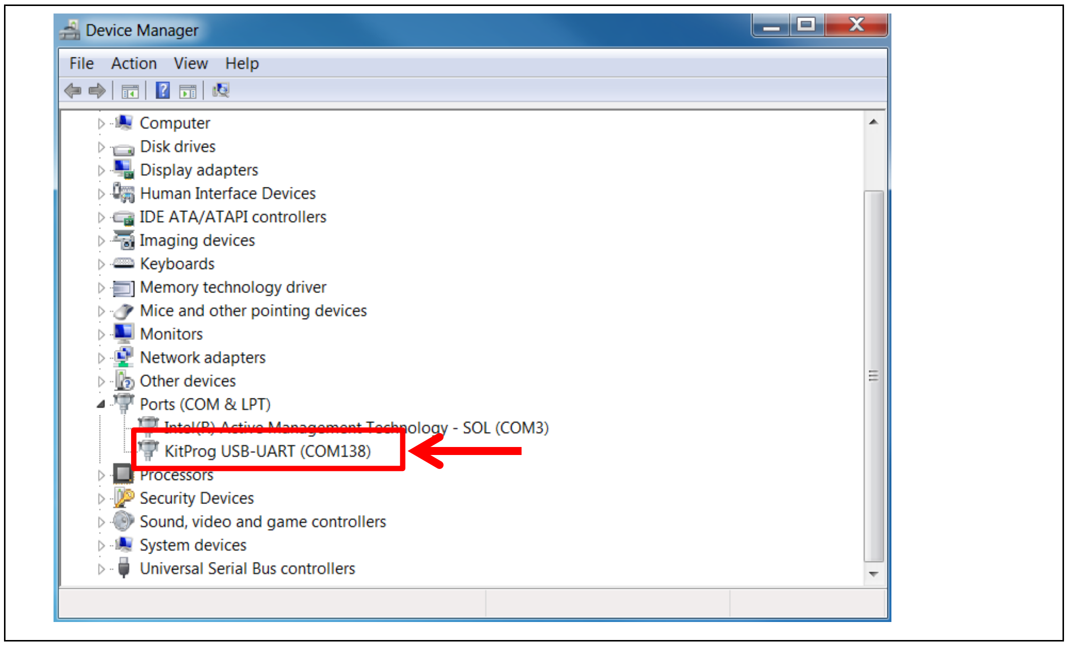

Update Driver Software - KitProg USBUART
Installing driver software...
Update Driver Software - KitProg USB-UART (COM139)
Close

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

KitProg USB-UART

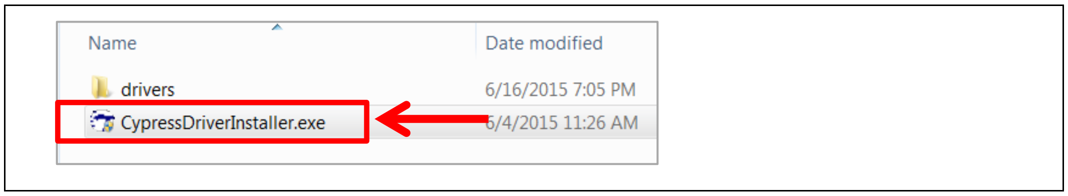
6. After successful device driver installation, confirm that a new COM port called KitProg USB-UART was added:
   a. Open the Device Manager.
   b. Under Ports (COM & LPT), confirm that a COM port called KitProg USB-UART was added. Note the COM number (COMxx).



Computer
Disk drives
Display adapters
Human Interface Devices
IDE ATA/ATAPI controllers
Imaging devices
Keyboards
Memory technology driver
Mice and other pointing devices
Monitors
Network adapters
Other devices
Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM3)
KitProg USB-UART (COM138)
Processors
Security Devices
Sound, video and game controllers
System devices
Universal Serial Bus controllers

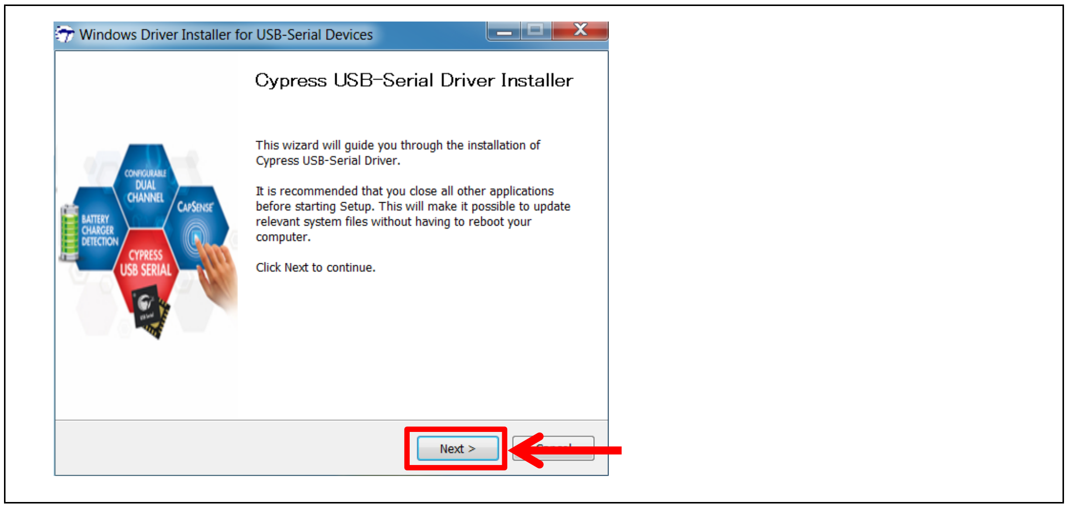
6.1.3 USB Driver Installation for the Motherboard

1. Run CypressDriverInstaller.exe, which is the installer for the USB serial device on the Motherboard. <Install directory>/Solar-Powered IoT Device Kit/1.0/USB drivers



Name	Date modified
drivers	6/16/2015 7:05 PM
CypressDriverInstaller.exe	6/4/2015 11:26 AM

2. Push the “Next >” button.



Windows Driver Installer for USB-Serial Devices

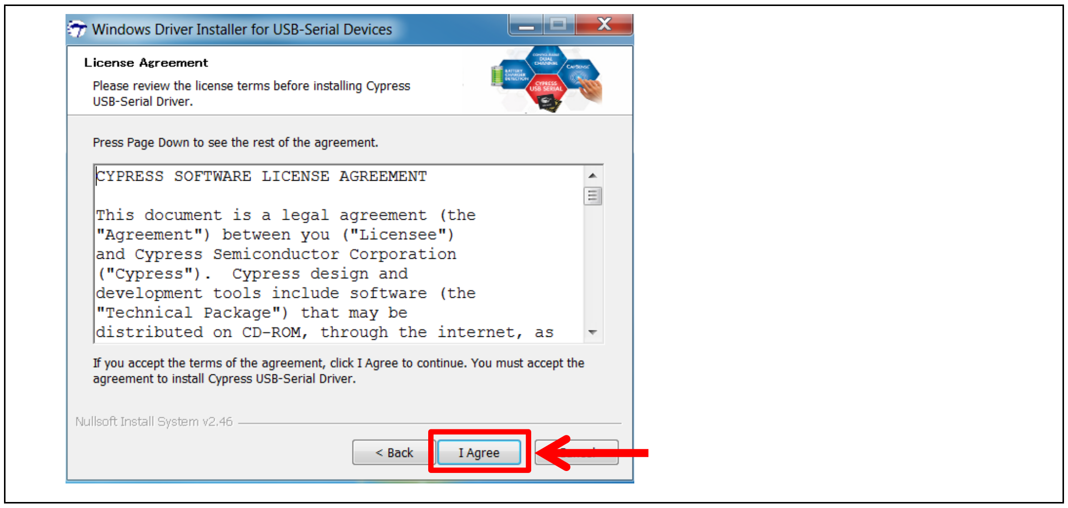
Cypress USB-Serial Driver Installer

This wizard will guide you through the installation of Cypress USB-Serial Driver.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

3. Push the “I Agree” button.



Windows Driver Installer for USB-Serial Devices

License Agreement

Please review the license terms before installing Cypress USB-Serial Driver.

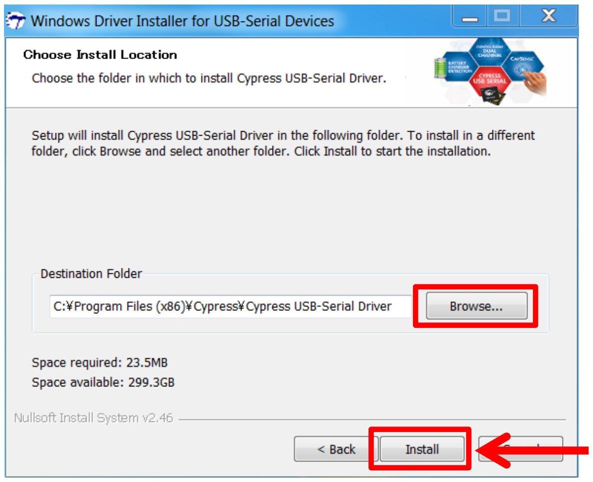
Press Page Down to see the rest of the agreement.

```
CYPRESS SOFTWARE LICENSE AGREEMENT

This document is a legal agreement (the
"Agreement") between you ("Licensee")
and Cypress Semiconductor Corporation
("Cypress").  Cypress design and
development tools include software (the
"Technical Package") that may be
distributed on CD-ROM, through the internet, as
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Cypress USB-Serial Driver.

Nullsoft Install System v2.46

4. Push the “Install” button. Use the “Browse…” button if you want to change the Destination Folder.



Windows Driver Installer for USB-Serial Devices

Choose Install Location

Choose the folder in which to install Cypress USB-Serial Driver.

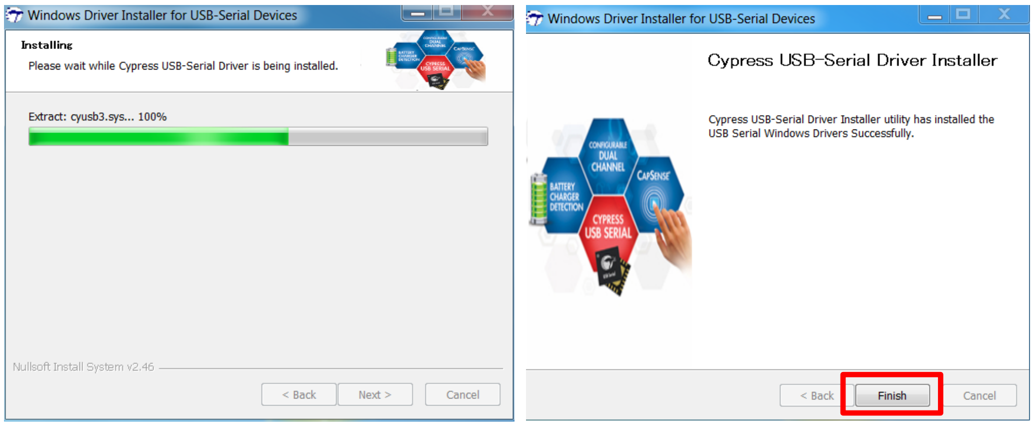
Setup will install Cypress USB-Serial Driver in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:¥Program Files (x86)¥Cypress¥Cypress USB-Serial Driver

Space required: 23.5MB
Space available: 299.3GB

Nullsoft Install System v2.46

5. Push the “Finish” button when the driver installation for the Motherboard finishes.



Windows Driver Installer for USB-Serial Devices

Installing

Please wait while Cypress USB-Serial Driver is being installed.

Extract: cyusb3.sys... 100%

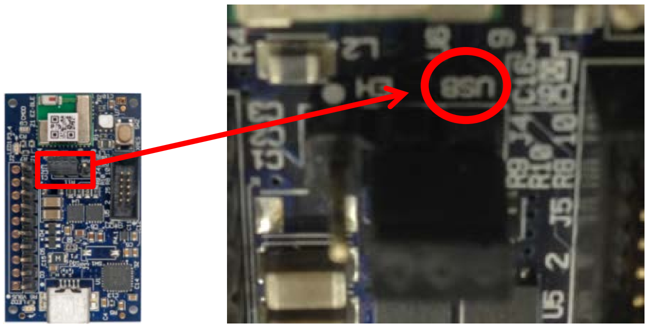
Nullsoft Install System v2.46

Windows Driver Installer for USB-Serial Devices

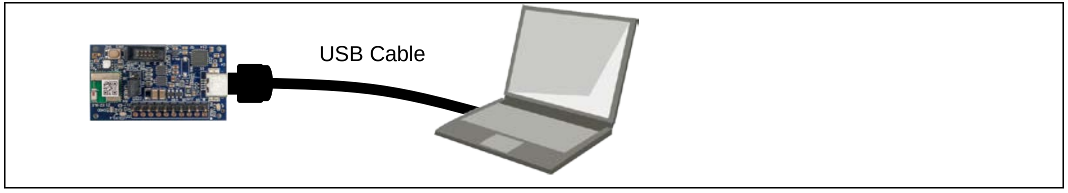
Cypress USB-Serial Driver Installer

Cypress USB-Serial Driver Installer utility has installed the USB Serial Windows Drivers Successfully.

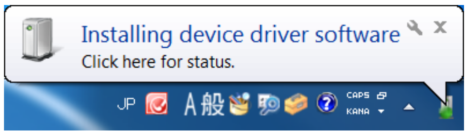
6. Configure the Motherboard to receive power from the USB port. Change jumper J4 to "USB" from "EH".



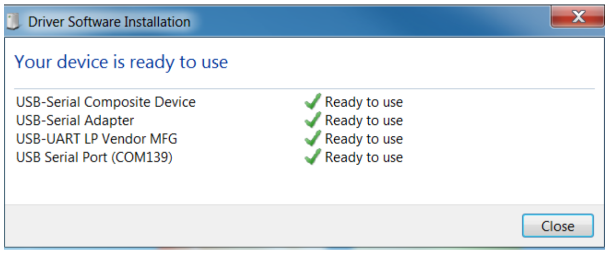
7. Connect the Motherboard to your computer using a USB cable.



8. The driver Installation starts automatically and the following message window will appear. Click the message window for status.


Installing device driver software
Click here for status.

9. Confirm that the device driver installation has successfully completed (all components will be "Ready to use").

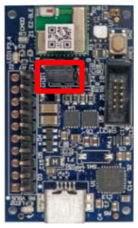

Driver Software Installation

Your device is ready to use

USB-Serial Composite Device	✓ Ready to use
USB-Serial Adapter	✓ Ready to use
USB-UART LP Vendor MFG	✓ Ready to use
USB Serial Port (COM139)	✓ Ready to use

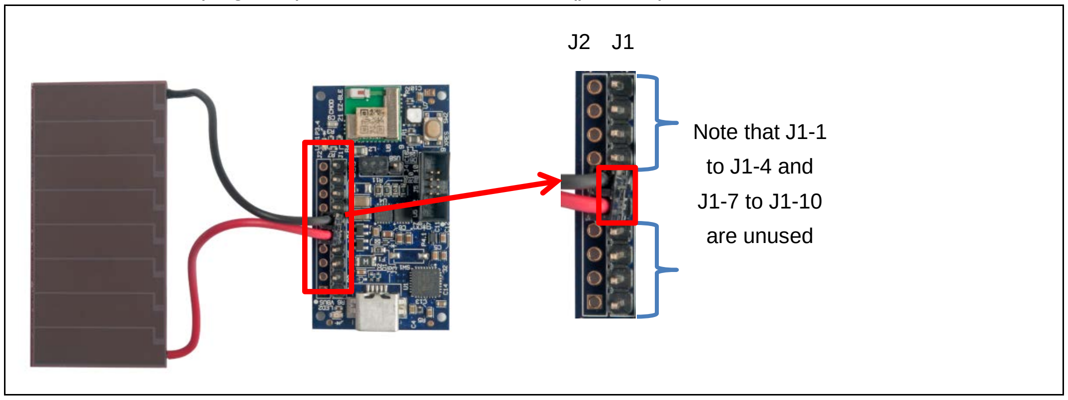
10. After successful device driver installation, confirm that a new COM port called USB Serial Port was added:
    a. Open the Device Manager.
    b. Under Ports (COM & LPT), confirm that a COM port called KitProg USB-UART was added. Note the COM number (COMxx).

11. Finally, disconnect the USB cable, then reset the jumper J4 set in step 6 back to "EH" from "USB" to supply power from the Solar Module.



6.1.4 Establishing BLE Connection

1. Connect the Solar Module (AM-1801, included in the kit) to the Energy Harvesting Motherboard. Plug the black wire (negative) to J1-6 and the red wire (positive) to J1-5 as shown below.



2. Place the Motherboard with Solar Module under an office light. The firmware to operate the Motherboard as a BLE Beacon is pre-loaded from the factory. After attaching the Solar Module and placing the Motherboard under a suitable light level (Refer to Table 5-1), it will automatically power up and begin transmitting.

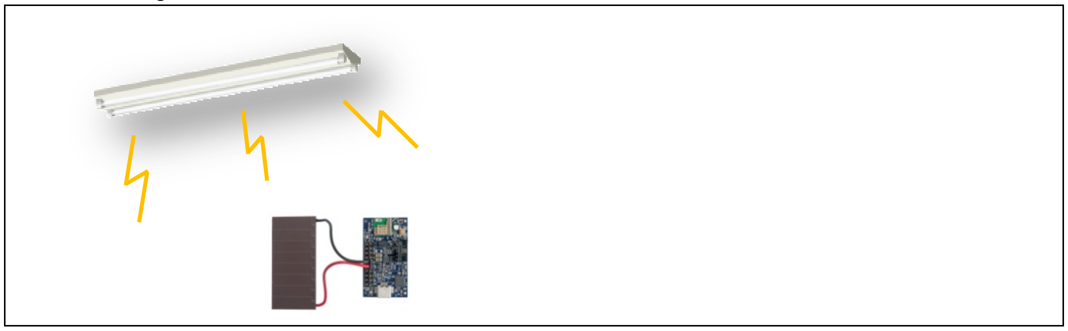


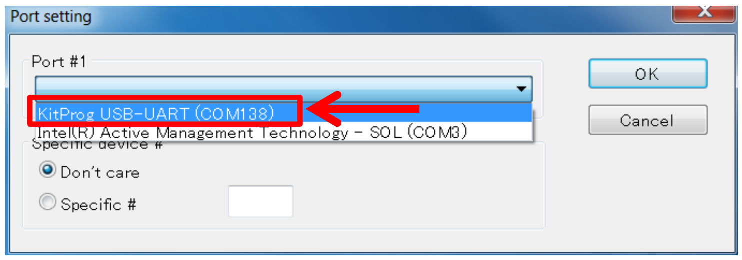
Table 6-1 Light level vs Time Interval

Typical Light Level	Environment	Time Interval of Beacon[1]
~1 lx	Moonlight	Does not work
50 lx~100 lx	Under street lighting	Does not work
200 lx~400 lx	At Museum	1.0 sec ~ 5.0 sec
400 lx~500 lx	Office lighting	0.6 sec ~ 1.0 sec
1000 lx	Shopping mall, Rainy day	0.4 sec ~ 0.6 sec

[1] The initial setting of time interval is set 1.5 sec. You need to configure an "ITRVL" command to change time interval. Refer to "6.3 Serial Command List".

3. Plug in the BLE-USB Bridge into your computer's USB port.

4. Run PMIC.exe, which is in the Windows application used to view data received from the Motherboard. It is located in the PMIC Software folder that you installed earlier
   <Install directory>/Solar-Powered IoT Device Kit/1.0/PMIC Software
   A dialog box will appear. Select KitProg USB-UART (COMxx) in the drop down menu under Port #1, where COMxx corresponds to the port that was confirmed in step 5. Leave the Specific Device # to "Don't care". Push the "OK" button.



Port setting

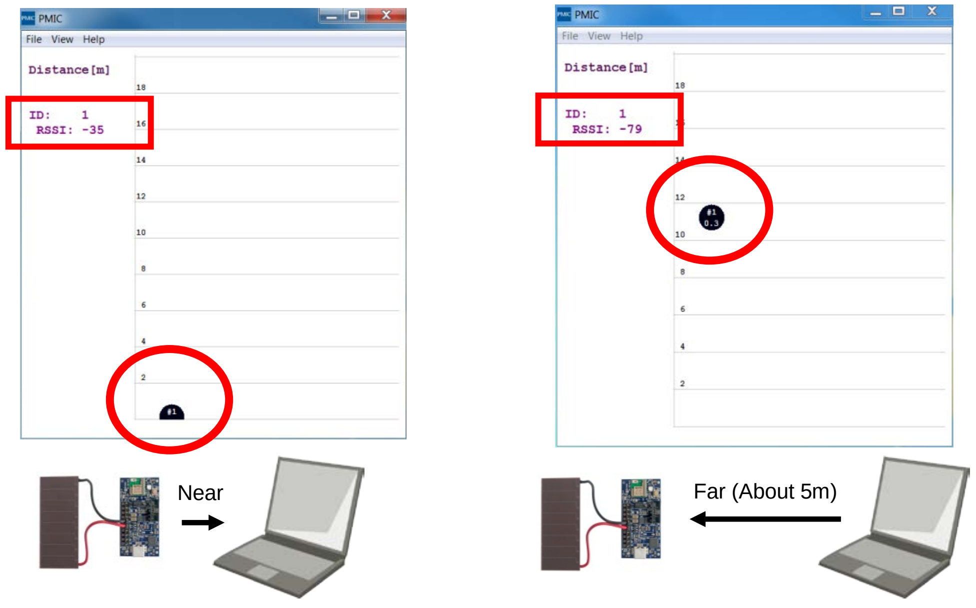
Port #1

KitProg USB-UART (COM138)
Intel(R) Active Management Technology - SOL (COM3)

Specific device #

(●) Don't care
( ) Specific #

5. Find your MAJOR number (Refer to "6.3 Serial Command List") of the Motherboard on the PMIC Software. Then move the Motherboard further away from your computer. The Received Signal Strength Indicator (RSSI) value will change and the graphic will be updated.



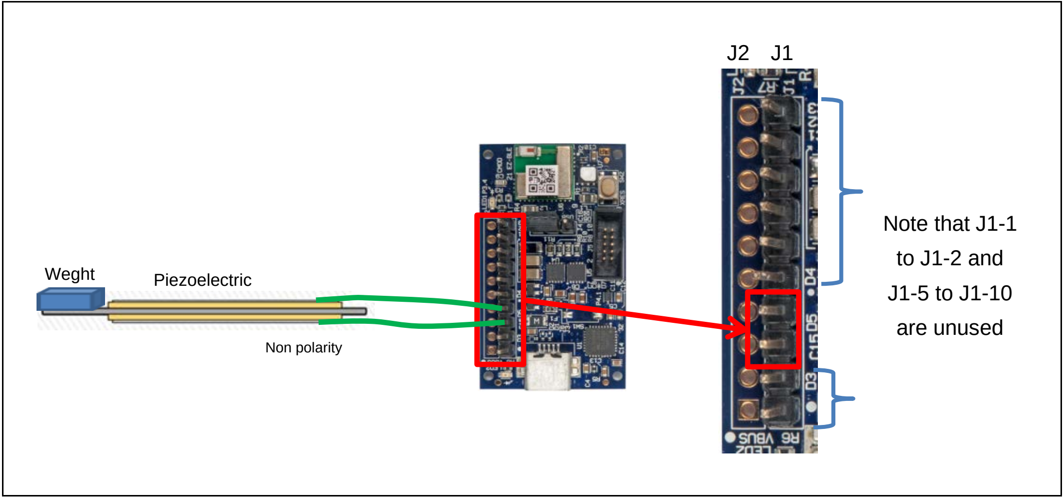
Distance[m]
ID: 1
RSSI: -35

Distance[m]
ID: 1
RSSI: -79

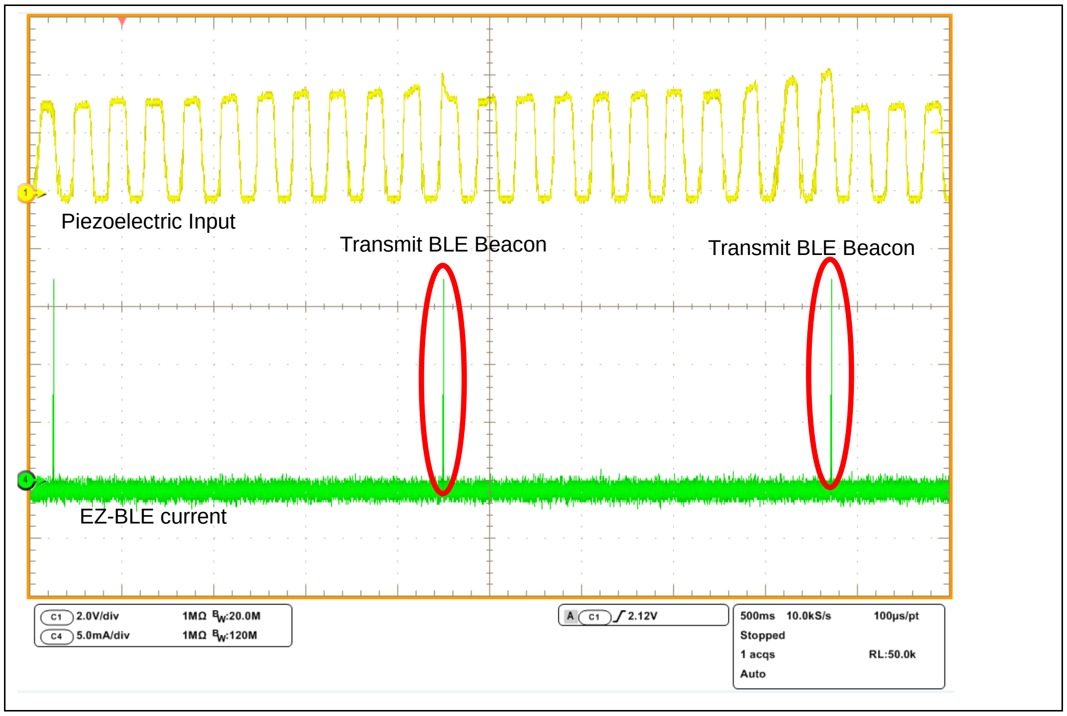
6.1.5 Vibration Energies Connection (Optional)

The Energy Harvesting Motherboard supports receiving AC voltage from piezoelectric or electro-magnetic Energy Harvesting Devices (EHDs) that harvest vibration energy. To confirm this operation, a piezoelectric or electro-magnetic EHD is required (not supplied with Kit).

1. Connect the piezoelectric or electro-magnetic EHD to the Motherboard. Plug the wires from the EHD to J1-3 and J1-4 as shown below. Note that there is no polarity.



2. Move the EHD to generate vibration energy.
3. Follow steps 3-5 of "6.1.4 Establishing BLE Connection" to confirm that the Energy Harvesting Motherboard is operating. Following is sample waveform for the operation of vibration energies. If the Motherboard is not operating, you may have to increase the vibration energy. Refer to the documentation for the EHD being used.



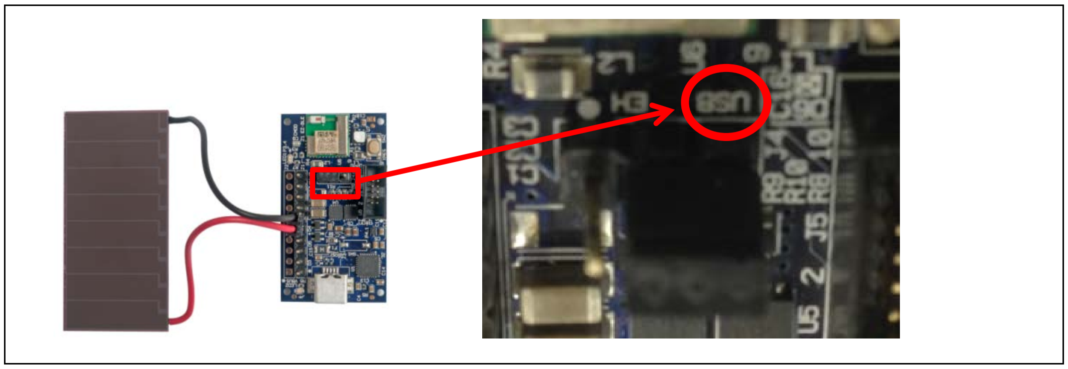
6.2 Solar-Powered Wireless Sensor Node (WSN) with BLE Beacon

You will configure the Motherboard as a WSN by turning on the temperature and humidity sensor. You will do this using a serial USB connection from your PC to send configuration commands to the Motherboard.

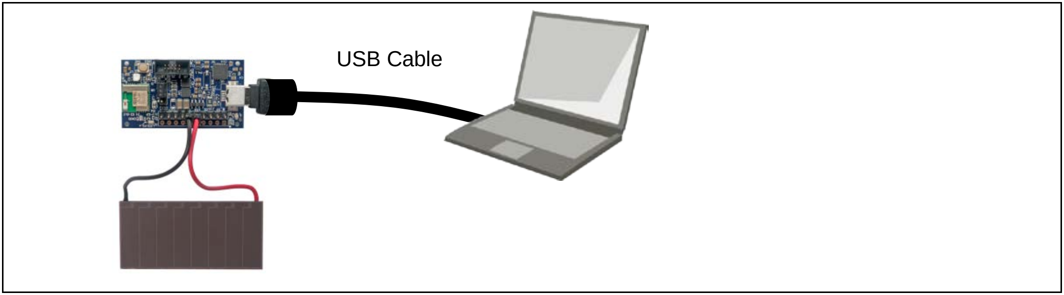
Finally, you will confirm that the Motherboard is operating as a WSN by using the provided software on your PC to detect temperature and humidity changes.

6.2.1 Configuring the Motherboard as a WSN

1. Configure the Motherboard to receive power from the USB port by changing jumper J4 to "USB" from "EH".



2. Connect the Energy Harvesting Motherboard to your computer using a USB cable.



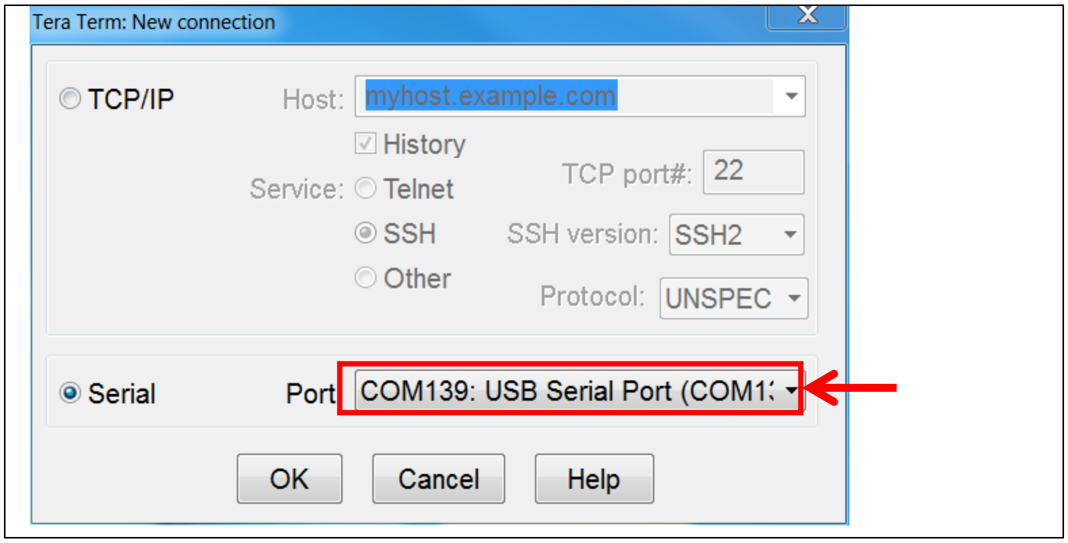
3. Confirm that a COM port (USB Serial Port) was added in the Windows Device Manager:
   a. Open the Device Manager:
   b. Under Ports (COM & LPT), confirm that a USB Serial Port was added. Note the COM number (COMxxx).

4. Install Tera Term from the following location:
   <Install directory>/Solar-Powered IoT Device Kit/1.0/PMIC Software/teraterm

5. After installed, run Tera Term:

Windows 7	Start Menu > All Programs > Tera Term
Windows 8/8.1	Ctrl + Tab keys > All Apps > Tera Term
Windows 10	Start Button > All Apps > Tera Term

6. In "Tera Term: File > New Connection" window, click Serial and select "COMxxx: USB Serial Port(COMxxx)", click OK.


Tera Term: New connection

TCP/IP Host: myhost.example.com

History

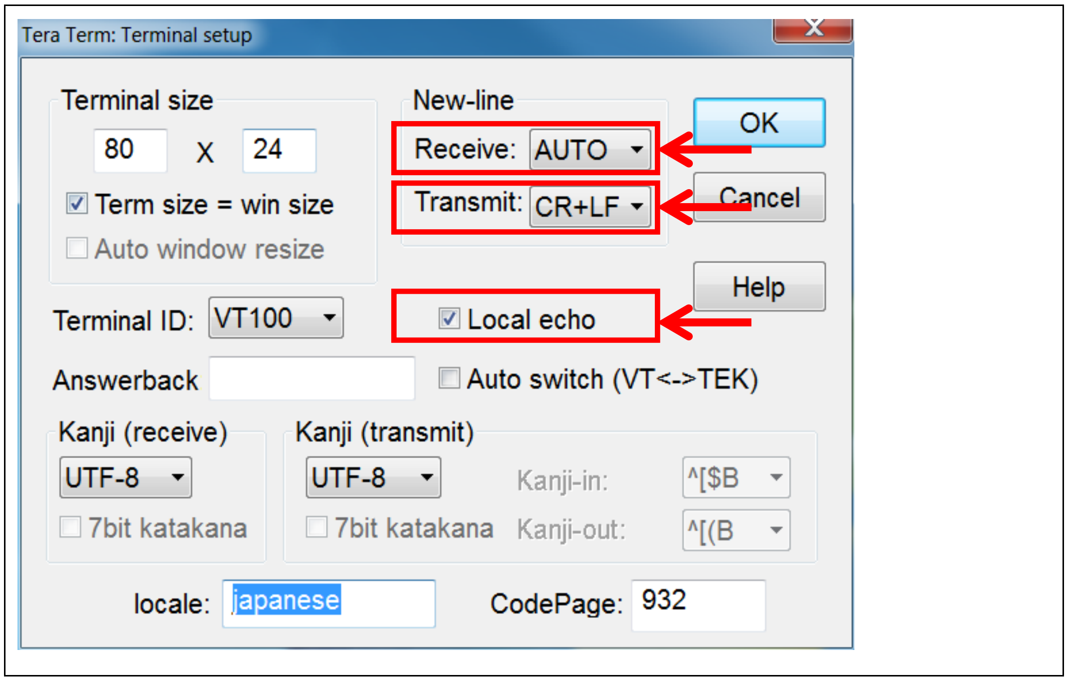
Service: Telnet TCP port#: 22

SSH SSH version: SSH2

Other Protocol: UNSPEC

Serial Port: COM139: USB Serial Port (COM1:

7. Configure Terminal setting (Setup > Terminal) as shown below, click OK.
   Receive: AUTO
   Transmit: CR+LF
   Local echo: Check
   Other settings: Default


Tera Term: Terminal setup

Terminal size: 80 X 24

[✓] Term size = win size

[ ] Auto window resize

Terminal ID: VT100

Answerback:

New-line

Receive: AUTO

Transmit: CR+LF

[✓] Local echo

[ ] Auto switch (VT<->TEK)

Kanji (receive): UTF-8

[ ] 7bit katakana

Kanji (transmit): UTF-8

[ ] 7bit katakana

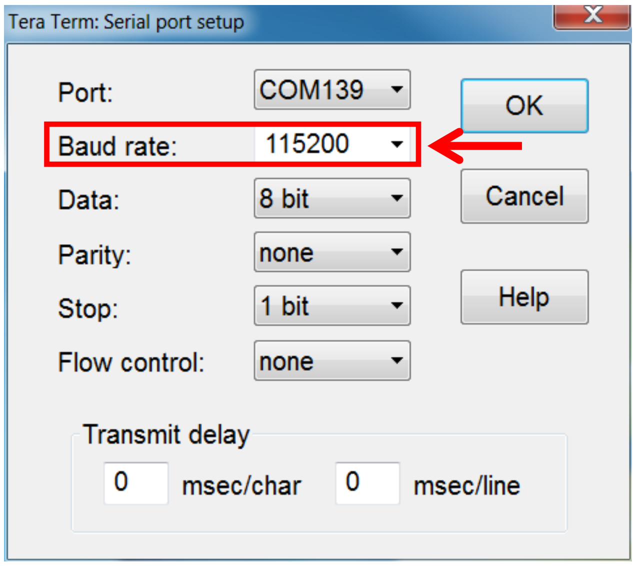
Kanji-in: ^[$B

Kanji-out: ^[(B

locale: japanese

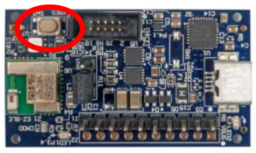
CodePage: 932

8. Configure Serial Port setting (Setup > Serial Port) as shown below, click OK.
   Baund rate: 115200
   Other settings: Default

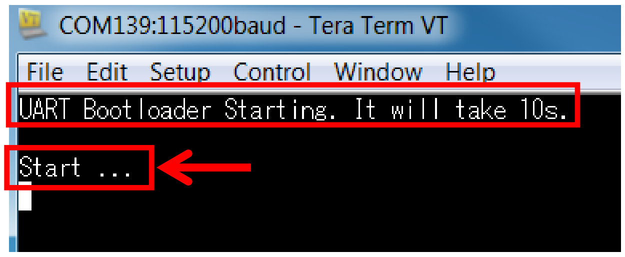


Port: COM139
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none
Transmit delay
0 msec/char 0 msec/line

9. Enable the sensor on the Motherboard by putting the Motherboard into command mode and sending it a command from the PC:
   a. Push and release the XRES button on the Motherboard. Note that when the Motherboard is in this mode, it stops transmitting BLE WSN data.

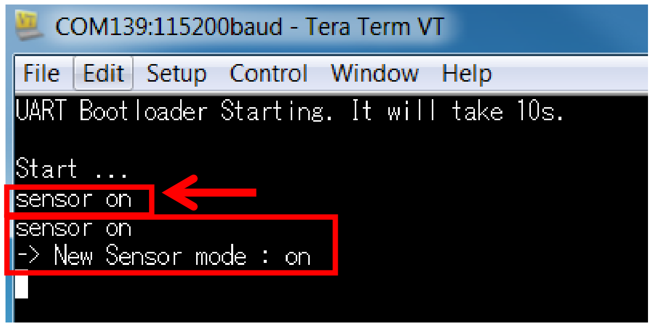


   b. Confirm that the Motherboard is in the command mode by observing that the "Start…." message appears on Tera Term. Your Motherboard is ready to receive commands.


COM139:115200baud - Tera Term VT
File Edit Setup Control Window Help

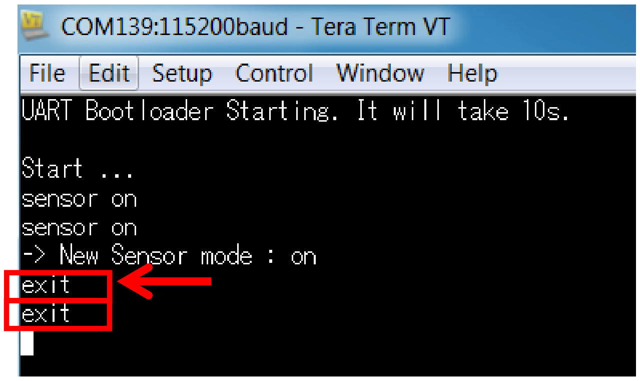
```
UART Bootloader Starting. It will take 10s.

Start ...
```

c. On Tera Term, type “sensor on⏎”. The Motherboard responds with a confirmation message as shown below. Refer to the " 6.3 Serial Command List" of the kit for a list of all commands.


COM139:115200baud - Tera Term VT
File Edit Setup Control Window Help

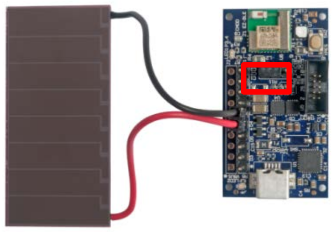
```
UART Bootloader Starting. It will take 10s.

Start ...
sensor on
sensor on
-> New Sensor mode : on
```

d. To finish the command waiting mode, type “exit⏎”, then the board will acknowledge the command by responding with “exit”. The board will retransmit the Bluetooth Beacon data.


COM139:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
UART Bootloader Starting. It will take 10s.

Start ...
sensor on
sensor on
-> New Sensor mode : on
exit
exit
```

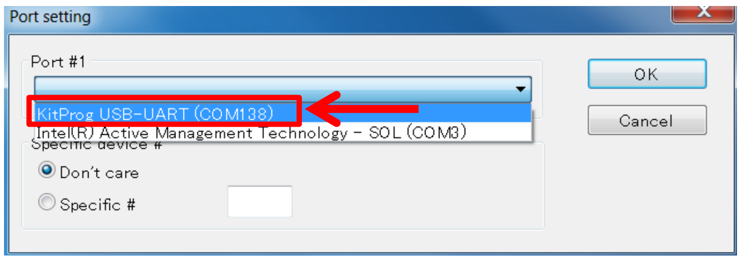
10. Finally, disconnect the USB cable, then reset the jumper J4 set in step 1 back to "EH" from "USB" to supply power from the Solar Module.



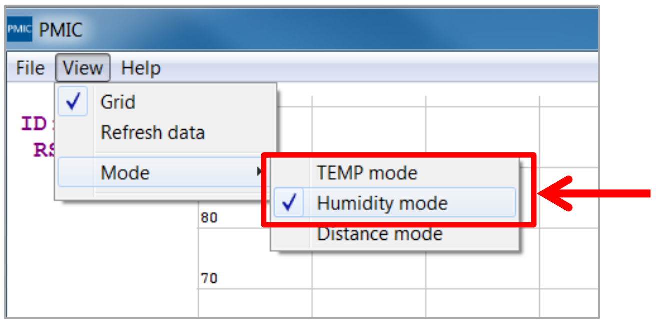
6.2.2 Confirm that your WSN is Operating

1. Connect the BLE-USB Bridge to your computer's USB port.

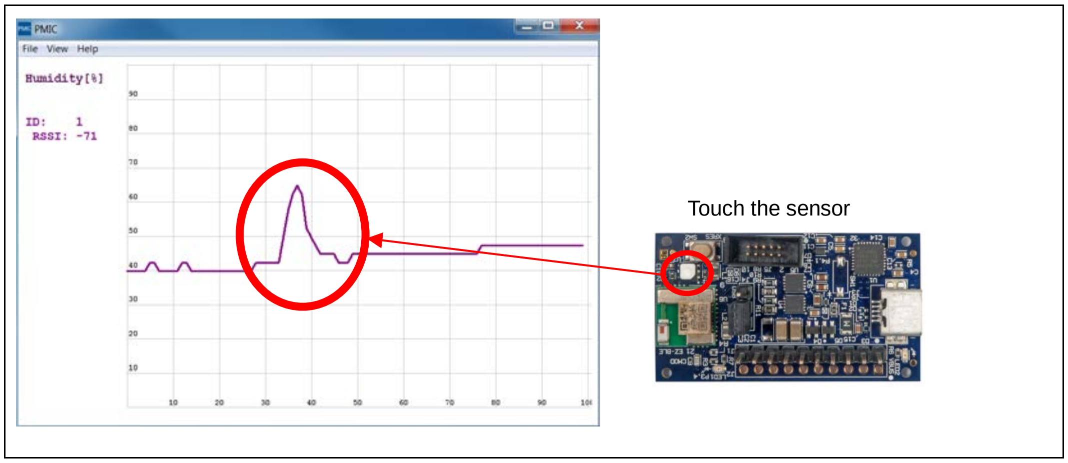
2. Run PMIC.exe (<Install directory>/Solar-Powered IoT Device Kit/1.0/PMIC Software).
A dialog box will appear. Select KitProg USB-UART (COMxx) in the drop down menu under Port #1, where COMxx corresponds to the port that was confirmed earlier. Leave the Specific Device # to "Don't care". Push the "OK" button.


Port setting
Port #1
KitProg USB-UART (COM138)
Intel(R) Active Management Technology - SOL (COM3)
Specific device #
Don't care
Specific #
OK
Cancel

3. Change the view Mode to "Humidity mode" or "TEMP mode".
View>Mode>Humidity mode or TEMP mode



4. Confirm that the WSN is operating by placing your finger on the sensor on the Motherboard. The temperature and humidity are raised from indoor environment condition by placing your finger You should see a corresponding change in humidity or temperature on your PC. When touching the board, please be careful of static electricity.



5. Refer to "6.1.5 Vibration Energies Connection (Optional)" if you are required the vibration operation.

6.3 Serial Command List

The following is the list of serial commands which can be used to control the kit from your computer using the USB serial interface. Refer to "6.2.1 Configuring the Motherboard as a WSN" for instructions on how to issue these commands via the USB serial interface. The commands are case sensitive.

Table 6-2 Commands List

No.	Commands Name	Description	Default
[1]	UUID	Read/Write of UUID[1]	00050001-0000-1000-8000-00805F9B0131 [hex]
[2]	MAJOR	Read/Write of MAJOR[1]	0x0001
[3]	MINOR	Read/Write of MINOR[1]	0x0001 (Note: sensor data is sent after it is overwritten in MINOR region when SENSOR command is ON)
[4]	TXPWR	Read/Write of Transmitter Power Strength	3 dBm
[5]	RSSI	Read/Write of Receiver Power Strength for distance 1m (RSSI)	-61 dBm
[6]	ITRVL	Read/Write of Advertise Interval for Beacon	1500 ms (Note: time interval is fixed at six seconds when the SENSOR command is on)
[7]	COID	Read/Write of Bluetooth Company	0x0131 (Cypress Semiconductor Corporation)
[8]	SENSOR	Overwrite sensor information in MINOR packet region and send it	OFF
[9]	ERASE	Default parameters	-
[10]	EXIT	Finish the command waiting mode, and then retransmit the Bluetooth beacon data	-
[11]	VER	Display Firmware Version	-

[1] Refer to "8.5 BLE Beacon Format" for detailed information.

[1] Read/Write of UUID

[1-1] Read

Read UUID data. Default: 00050001-0000-1000-8000-00805F9B0131

<Example>

```
UUID⏎
    (echo)    UUID
    (output) -> UUID: 00050001-0000-1000-8000-00805F9B0131
```

[1-2] Write

Write UUID data.

<Example>

```
UUID EEEEDDDD-CCCC-BBBB-AAAA-999988887777⏎
    (echo)    UUID EEEEDDDD-CCCC-BBBB-AAAA-999988887777
    (output) -> New UUID: EEEEDDDD-CCCC-BBBB-AAAA-999988887777
```

[2] Read/Write of MAJOR

[2-1] Read

```
  Read MAJOR.                       Default: 0x0001
<Example>
MAJOR⏎
 (echo)   MAJOR
 (output) -> MAJOR: 0001
```

[2-2] Write

```
  Write MAJOR.
<Example>
MAJOR 1A2F⏎                         <- Input HEX data
 (echo)   MAJOR 1A2F
 (output) -> New MAJOR: 1A2F
```

[3] Read/Write of MINOR

[3-1] Read

```
  Read MINOR.                       Default: 0x0001
<Example>
MINOR⏎
 (echo)   MINOR
 (output) -> MINOR: 0001
```

[3-2] Write

```
  Write MINOR.
<Example>
MINOR 2C3D⏎                         <- Input HEX data
 (echo)   MINOR 2C3D
 (output) -> New MINOR: 2C3D
```

[4] Read/Write of Transmitter Power Strength

[4-1] Read

```
  Read Power Strength.              Default: 3 dBm
<Example>
TXPWR⏎
 (echo)   TXPWR
 (output) -> TX power in dBm: 3
```

[4-2] Write

```
  Set Power Strength.               Set Value: -18, -12, -6, -3, -2, -1, 0, 3
<Example>
TXPWR -18⏎
 (echo)   TXPWR -18
 (output) -> New TX power in dBm: -18
```

[5] Read/Write of Receiver Power Strength for distance 1m (RSSI)

[5-1] Read

Read RSSI. Default: -61dBm

<Example>

```
RSSI⏎
 (echo)   RSSI
 (output) -> RSSI in dBm: -61
```

[5-2] Write

Set RSSI.

<Example>

```
RSSI -90⏎
 (echo)   RSSI -90
 (output) -> New RSSI in dBm: -90
```

[6] Read/Write of Advertise Interval

[6-1] Read

Read Advertise Interval. Default: 1500ms

<Example>

```
ITRVL⏎
 (echo)   ITRVL
 (output) -> Advertise Interval in msec:    1500
```

[6-2] Write

Set Advertise Interval. Set Value: 100~10240 ms

<Example>

```
ITRVL 10240⏎
 (echo)   ITRVL 10240
 (output) -> New Advertise Interval in msec: 10240
```

[7] Read/Write of Bluetooth Company

[7-1] Read

Read Bluetooth Company. Default: 0x0131 (Cypress Semiconductor Corporation)

<Example>

```
COID⏎
 (echo)   COID
 (output) -> Company ID: 0059
```

[7-2] Write

Write Bluetooth Company.

<Example>

```
COID 004C⏎                              <- Input HEX data
 (echo)   COID 004C
 (output) -> New Company ID: 004C
```

[8] Read/Write of sensor setting

[8-1] Read

Read sensor setting Default: OFF

<Example>

```
SENSOR⏎
    (echo)   SENSOR
    (output) -> Sensor mode: OFF
```

[8-2] Write

Change sensor setting set value: ON or OFF

<Example>

```
SENSOR ON⏎
(echo)   SENSOR ON
(output) -> New Sensor mode: ON
```

[9] ERASE

Erase the flash memory in MCU. After erase, all value will be default parameters.

<Example>

```
ERASE⏎
    (echo) ERASE
    (output) Erase completed!
```

[10] EXIT

Finish the command waiting mode, and then retransmit the Bluetooth beacon data.

<Example>

```
EXIT⏎
    (echo) EXIT
    (output) ---
```

[11] VER

Display Firmware Version.

<Example>

```
VER⏎
    (echo) VER
    (output) -> S6SAE101A00SA1002 Sample Firmware, Version 1.0
```

[*] Input another command (Error Handling)

```
TEST
    (echo)   TEST
    (output) Command format error!!
```

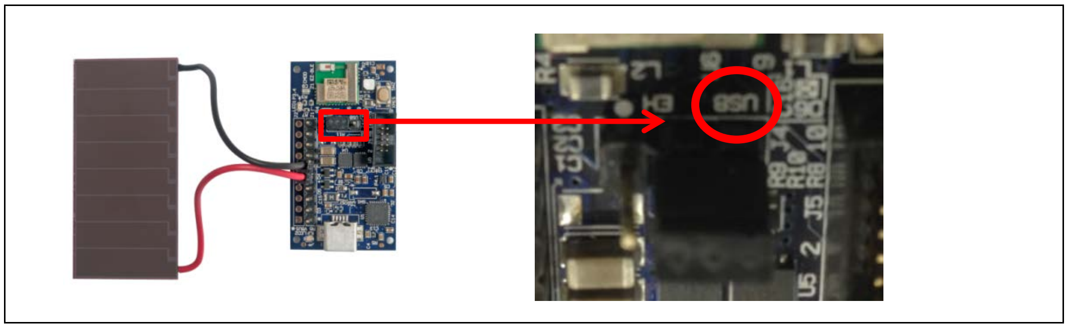
7. Program and Debug

The Solar-Powered IoT Device Kit can be programmed using UART Bootloader, and it can be programmed and debugged using PSoC Creater with MiniProg3. Before debugging the device, ensure that PSoC Creator is installed on the computer.

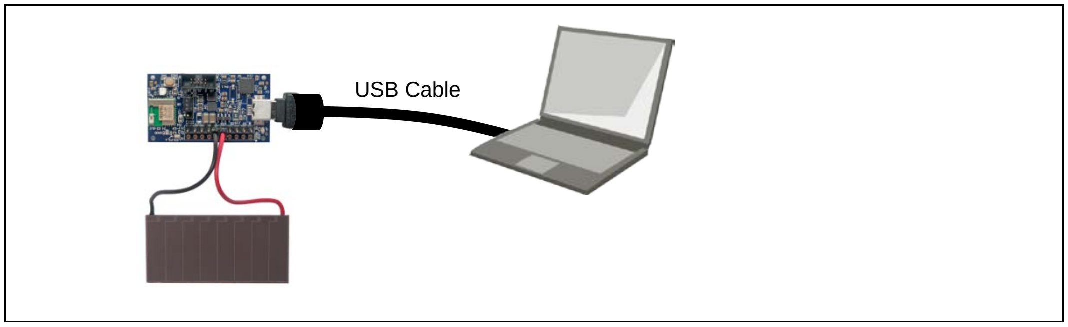
7.1 UART Bootloader (Program Only)

An UART Bootloader makes it possible for a product's firmware to be updated in the field. Bootloading is a process that allows you to upgrade your system firmware over UART. The UART Bootloader communicates with a host to get new application code or data, and writes it into the device's flash memory.

1. Configure the Motherboard to receive power from the USB port. Change jumper J4 to "USB" from "EH".

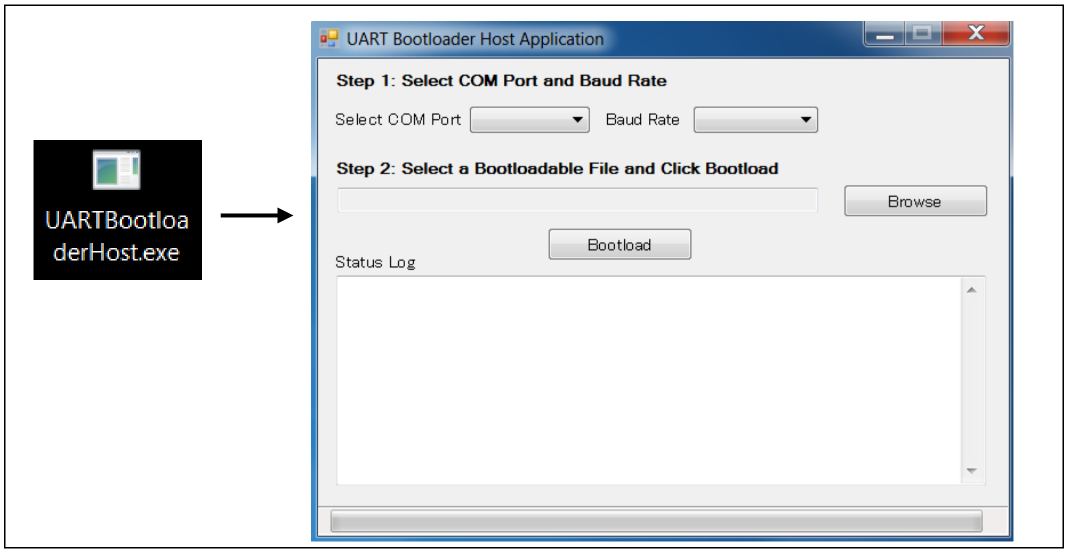


2. Connect the Energy Harvesting Motherboard to your computer using a USB cable.

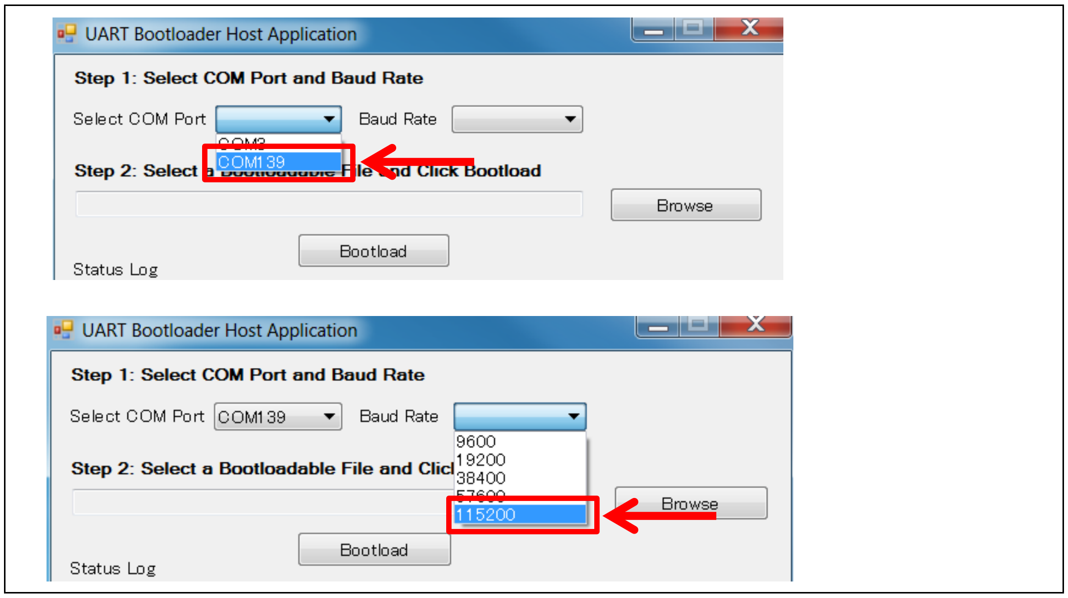


3. Confirm that a COM port called USB Serial Port was added:
   a. Open the Device Manager:
   b. Under Ports (COM & LPT), confirm that a COM port called USB Serial Port was added.
   Note the COM number (COMxx).

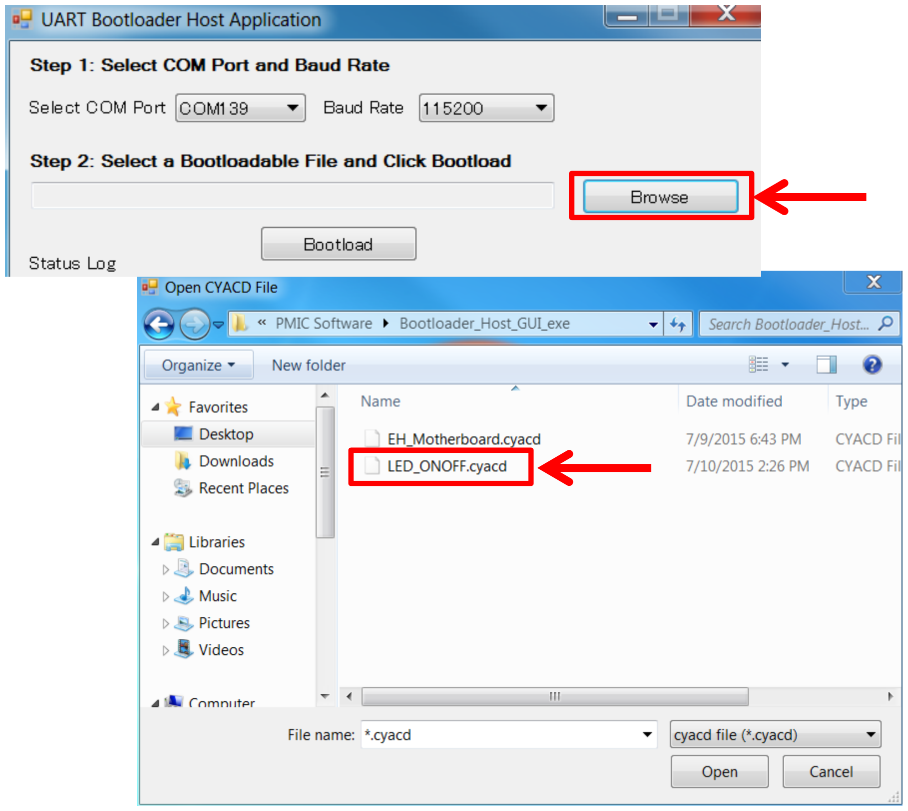
4. Run UARTBootloaderHost.exe. It is located in the PMIC Software folder that you installed earlier <Install directory>/Solar-Powered IoT Device Kit/1.0/PMIC Software/Bootloader_Host_GUI_exe.


UARTBootloaderHost.exe
UART Bootloader Host Application
Step 1: Select COM Port and Baud Rate
Select COM Port
Baud Rate
Step 2: Select a Bootloadable File and Click Bootload
Browse
Bootload
Status Log

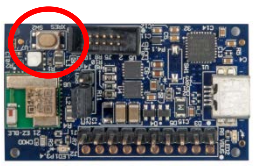
5. Select USB serial port where COMxx corresponds to the port that was confirmed in step 3 and select 115200 Baud Rate.

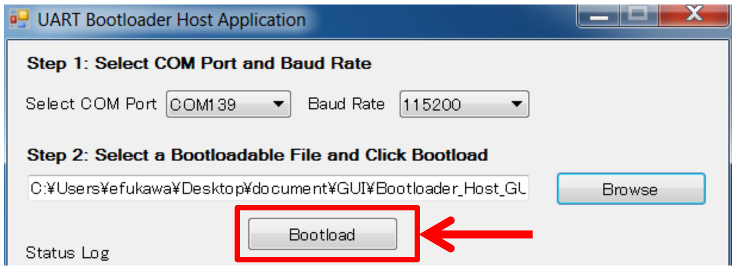

UART Bootloader Host Application
Step 1: Select COM Port and Baud Rate
Select COM Port
COM3
COM139
Baud Rate
Step 2: Select a Bootloadable File and Click Bootload
Browse
Bootload
Status Log
UART Bootloader Host Application
Step 1: Select COM Port and Baud Rate
Select COM Port
COM139
Baud Rate
9600
19200
38400
57600
115200
Step 2: Select a Bootloadable File and Click Bootload
Browse
Bootload
Status Log

6. Push the Browse button, then open "LED_ONOFF.cyacd" in the Bootloader_Host_GUI_exe folder <Install directory>/Solar-Powered IoT Device Kit/1.0/PMIC Software/Bootloader_Host_GUI_exe.
   If you have already developed your own firmware using PSoC Creator, the .cyacd file is generated in the project folder as shown below.
   [PSoC Creator Project Folder] /CortexM0/ARM_GCC_484/Debug (or Release)



7. Push and release the XRES button on the Energy Harvesting Motherboard and then push the "Bootload" button.





UART Bootloader Host Application

Step 1: Select COM Port and Baud Rate

Select COM Port: COM139 Baud Rate: 115200

Step 2: Select a Bootloadable File and Click Bootload

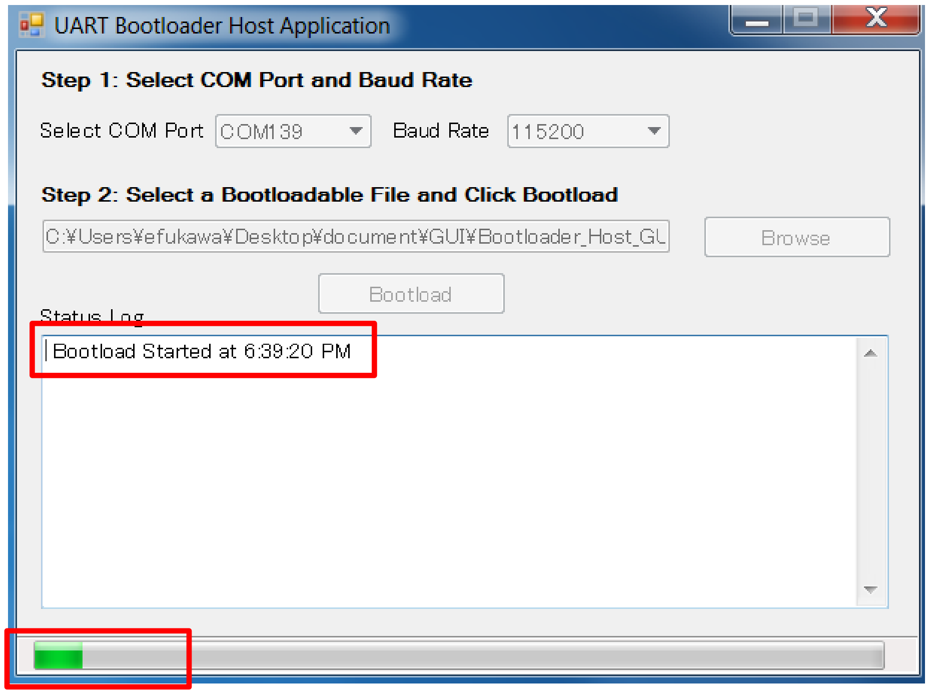
C:¥Users¥efukawa¥Desktop¥document¥GUI¥Bootloader_Host_GL

Browse

Bootload

Status Log

8. A "Bootload Started at X:XX:XX" message will appear in the Status Log window, and then an indicator will advance.



UART Bootloader Host Application

Step 1: Select COM Port and Baud Rate

Select COM Port: COM139 Baud Rate: 115200

Step 2: Select a Bootloadable File and Click Bootload

C:¥Users¥efukawa¥Desktop¥document¥GUI¥Bootloader_Host_GL

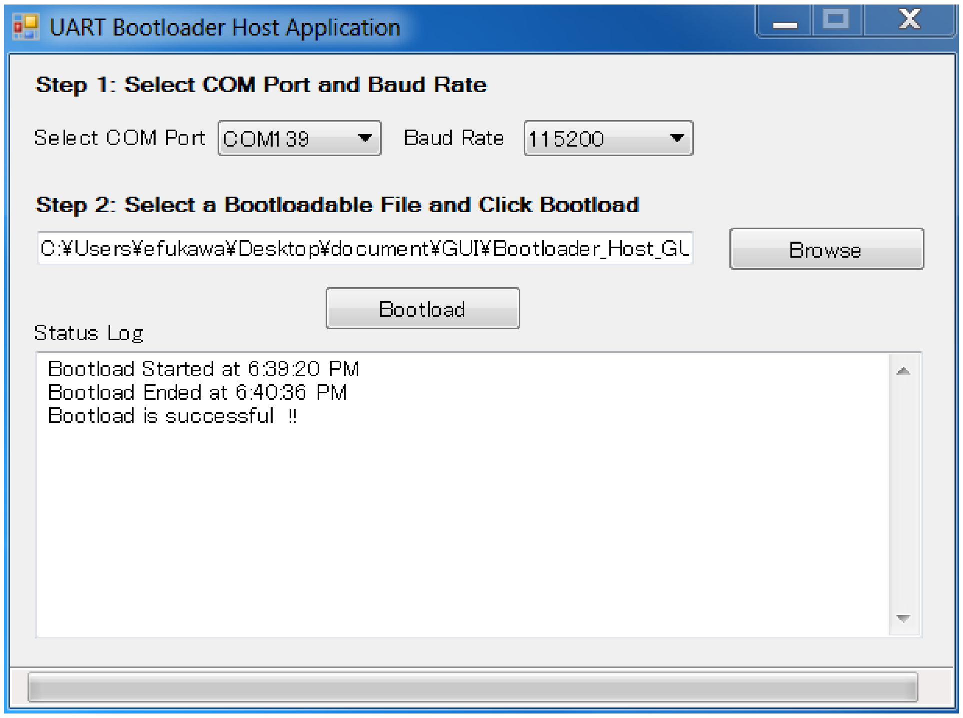
Browse

Bootload

Status Log

```
Bootload Started at 6:39:20 PM
```

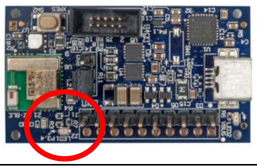
9. The programing is completed when a "Bootload in successful !!" message appears in the Status Log window.


UART Bootloader Host Application
Step 1: Select COM Port and Baud Rate
Step 2: Select a Bootloadable File and Click Bootload
Browse
Bootload
Status Log

Select COM Port COM139
Baud Rate 115200
C:¥Users¥efukawa¥Desktop¥document¥GUI¥Bootloader_Host_GU

```
Bootload Started at 6:39:20 PM
Bootload Ended at 6:40:36 PM
Bootload is successful  !!
```

10. Disconnect and re-connect the USB cable, and then the status LED will blink at 1 sec interval. This sample firmware only supports LED blinking control as a demonstration (Refer to "11.1.1 LED ONOFF Project" for detailed information), therefore please reprogram the "EH_Motherboard.cyacd" in the "Bootloader_Host_GUI_exe" folder to restore the BLE Beacon operation (Refer to steps 6 to 9 above).



7.2 PSoC Creator with MiniProg3 (Program and Debug)

The MiniProg3[1] (Not included in this kit) is the hardware/firmware block for onboard programming, debugging, and bridge functionality. It is a common reusable hardware/firmware block used across many Cypress kit platforms. The MiniProg3 communicates with PSoC Creator software to program/debug the target EZ-BLE PRoC Module over the SWD interface.

Requirement for preparation

■ Energy Harvesting Motherboard	1 pcs
■ MiniProg3 Program and Debug Kit[1] (Not included in this kit)	1 pcs
■ Windows PC for Programing and Debug	1 pcs
■ PSoC Creator 3.2 SP1 or newer[2] http://www.cypress.com/psoccreator/	1 pcs

[1] To debug this board, a MiniProg3 Program and Debug Kit is required. (www.cypress.com/?rID=38154)

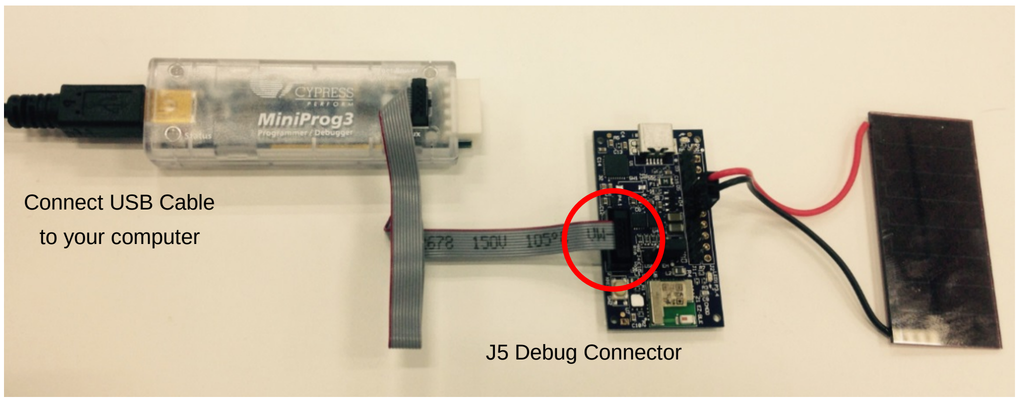
[2] The kit doesn't support PSoC Creator 3.2 or older. It needs to update PSoC Creator when you are using the 3.2 or older.

7.2.1 Program

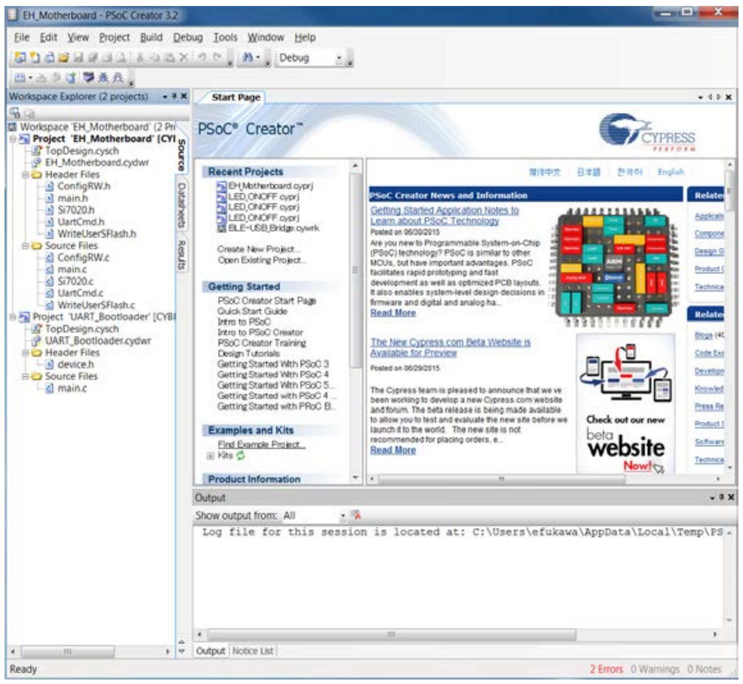
1. Disconnect USB cable from Energy Harvesting Motherboard.

2. Set the J4 jumper socket to "EH" side to supply the power from MiniProg3. A protection diode is mounted in between the Solar Module and the S6AE101A, therefore it is not necessary to remove the solar module when using the MiniProg3.

3. Connect the MiniProg3 to the Energy Harvesting Motherboard and your computer as shown below.

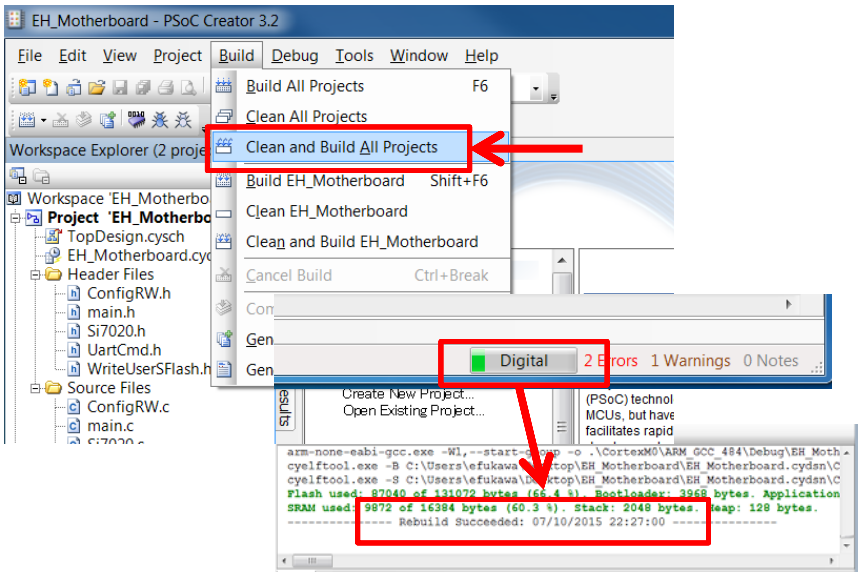


4. Open the sample project EH_Motherboard.cypj for this kit.
   <Install directory>/Solar-Powered IoT Device Kit/1.0/Firmware/EH_Motherboard /EH_Motherboard.cydsn/EH_Motherboard.cyprj


EH_Motherboard - PSoC Creator 3.2
File Edit View Project Build Debug Tools Window Help
Workspace Explorer (2 projects)
Start Page
PSoC® Creator™
Recent Projects
Getting Started
Examples and Kits
Product Information
Output
Show output from: All
Ready
2 Errors 0 Warnings 0 Notes

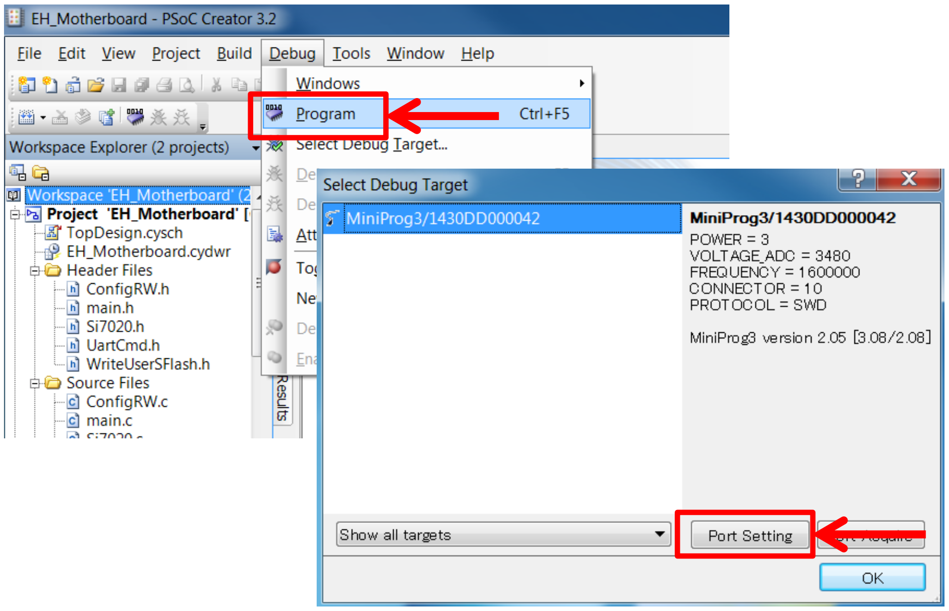
```
Log file for this session is located at: C:\Users\efukawa\AppData\Local\Temp\PS
```

5. Select the "Menu>Build>Clean and Build All Projects", and then the build status will appear on lower right side of PSoC Creator. The build process is finished when the "Rebuild succeeded" appears on massage window.


EH_Motherboard - PSoC Creator 3.2
File Edit View Project Build Debug Tools Window Help
Build All Projects F6
Clean All Projects
Clean and Build All Projects
Build EH_Motherboard Shift+F6
Clean EH_Motherboard
Clean and Build EH_Motherboard
Cancel Build Ctrl+Break
Workspace Explorer (2 proje
Digital
2 Errors 1 Warnings 0 Notes

```
arm-none-eabi-gcc.exe -Wl,--start-group -o .\CortexM0\ARM_GCC_484\Debug\EH_Moth
cyelftool.exe -B C:\Users\efukawa\Desktop\EH_Motherboard\EH_Motherboard.cydsn\C
cyelftool.exe -S C:\Users\efukawa\Desktop\EH_Motherboard\EH_Motherboard.cydsn\C
Flash used: 87040 of 131072 bytes (66.4 %). Bootloader: 3968 bytes. Application
SRAM used: 9872 of 16384 bytes (60.3 %). Stack: 2048 bytes. Heap: 128 bytes.
--------------- Rebuild Succeeded: 07/10/2015 22:27:00 ---------------
```

6. Select the "Menu>Debug>Program". If the Select Debug Target window appears as shown below, then perform the following.



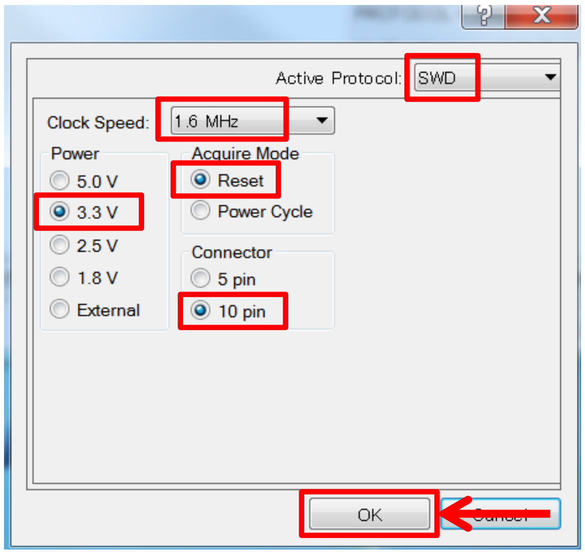
MiniProg3/1430DD000042
POWER = 3
VOLTAGE_ADC = 3480
FREQUENCY = 1600000
CONNECTOR = 10
PROTOCOL = SWD

MiniProg3 version 2.05 [3.08/2.08]

7. Click the "Port Setting" button, and then the following window will appear. The setting should be as below.

Active Protocol = SWD
Clock Speed = 1.6 MHz
Power = 3.3V
Acquire Mode = Reset
Connector = 10pin

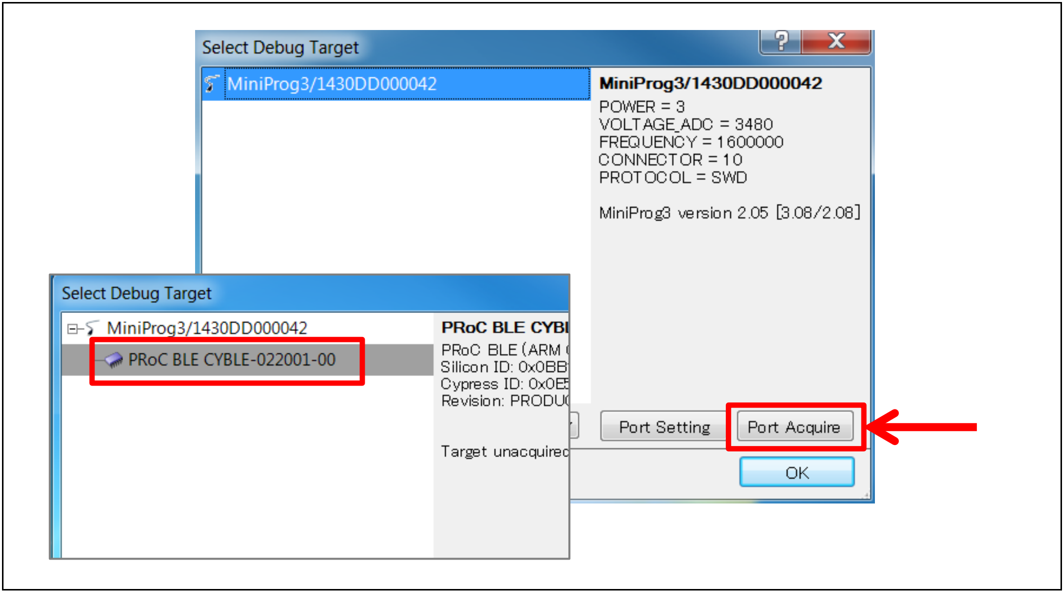
When finished all settings, push the OK button.



Active Protocol: SWD
Clock Speed: 1.6 MHz

Power: ( ) 5.0 V [✓ 3.3 V] ( ) 2.5 V ( ) 1.8 V ( ) External
Acquire Mode: [✓ Reset] ( ) Power Cycle
Connector: ( ) 5 pin [✓ 10 pin]

8. Click the "Port Acquire" button, and then "PRoC BLE CYBLE-022001-00" will appear.

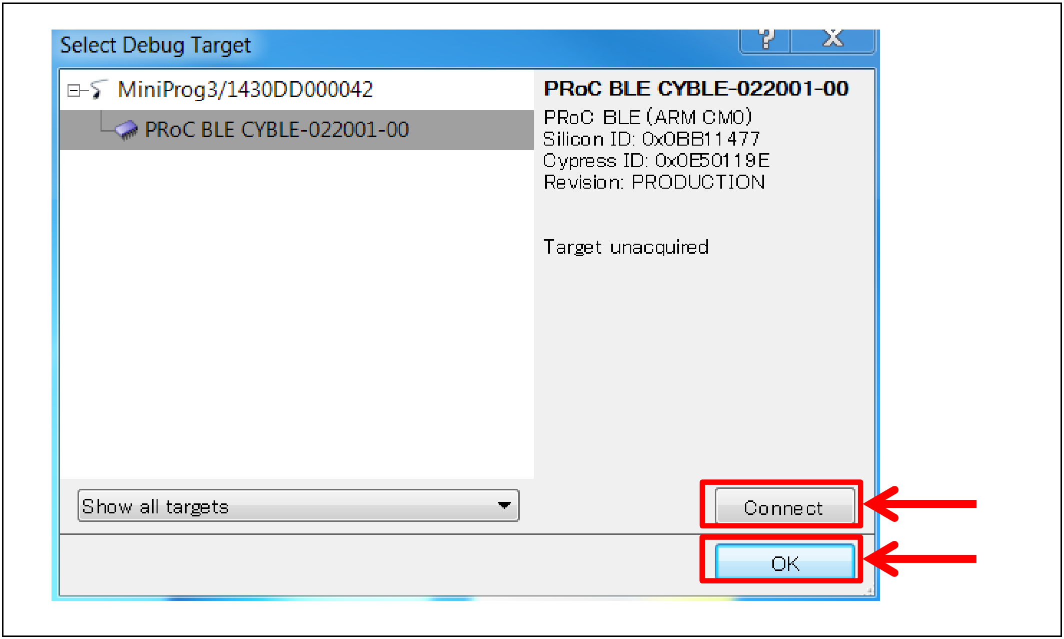


MiniProg3/1430DD000042
POWER = 3
VOLTAGE_ADC = 3480
FREQUENCY = 1600000
CONNECTOR = 10
PROTOCOL = SWD

MiniProg3 version 2.05 [3.08/2.08]

Target unacquired

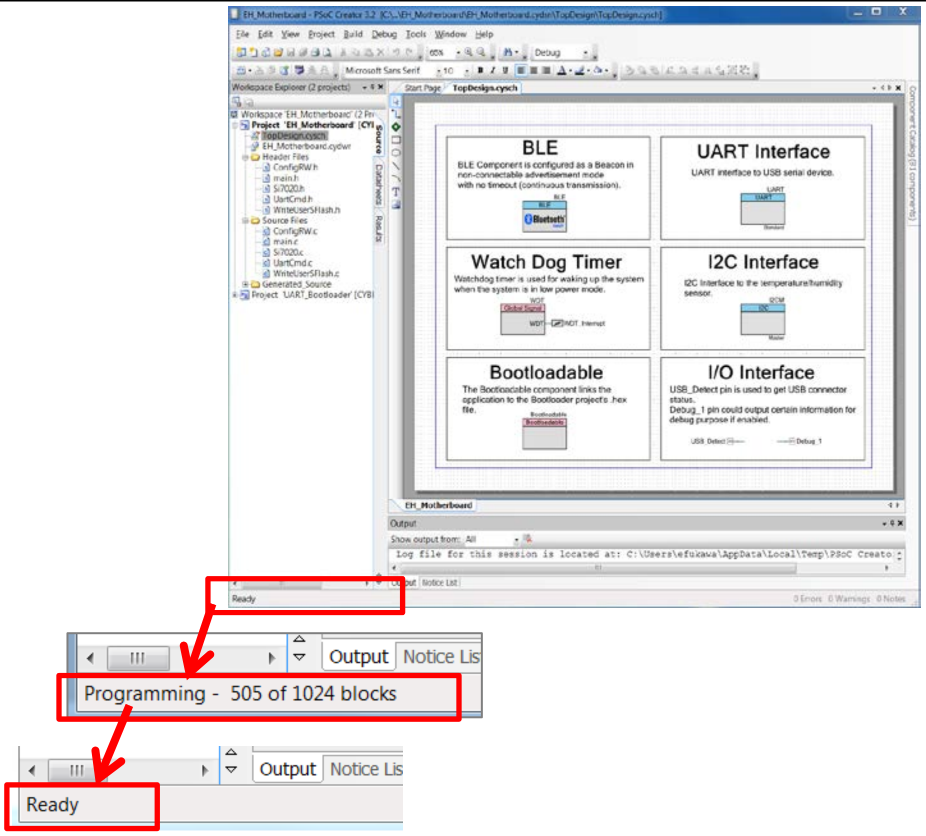
9. Click the "Connect" button, and then push OK button.



PRoC BLE CYBLE-022001-00
PRoC BLE (ARM CM0)
Silicon ID: 0x0BB11477
Cypress ID: 0x0E50119E
Revision: PRODUCTION

Target unacquired

10. A "Programming - XXX of 1024 blocks" message will appear on lower left side of PSoC Creator, and then "Ready" appears once programming has completed.



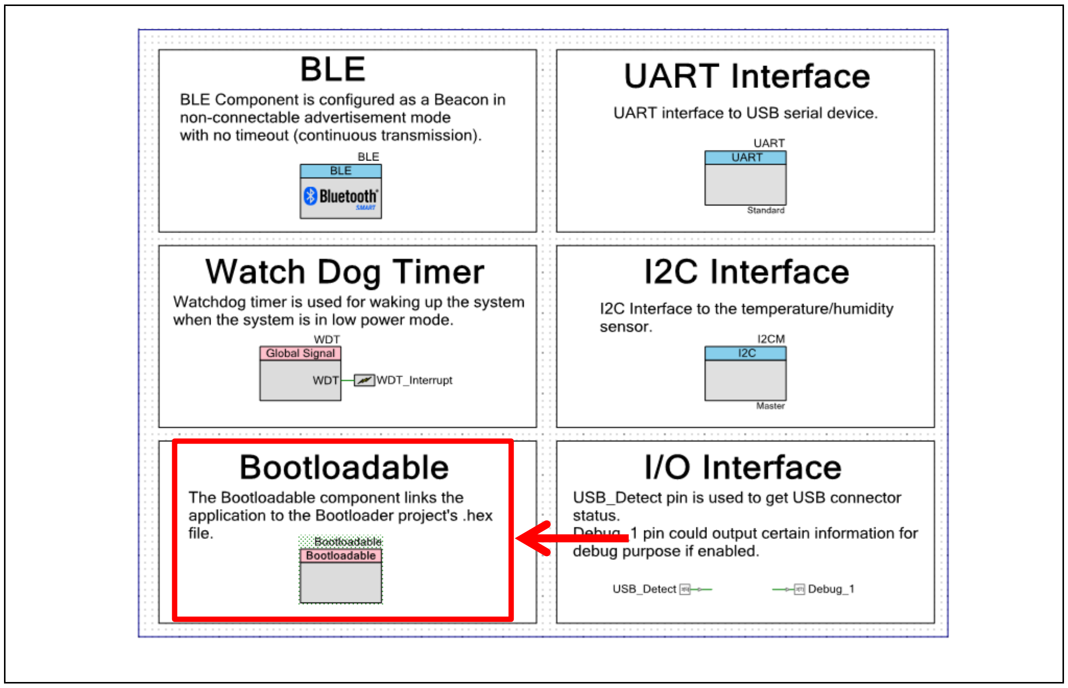
```
Log file for this session is located at: C:\Users\efukawa\AppData\Local\Temp\PSoC Creato
```

```
Programming -  505 of 1024 blocks
```

```
Ready
```

7.2.2 Debug

The sample project for this kit uses a bootloadable component for a UART Boot Loader. In the PSoC Creator bootloader system, the bootloader project executes first (at device reset) and then the bootloadable project. The jump from the bootloader to the bootloadable project is done through a software controlled device reset. This resets the debugger interface, which means that the bootloadable project cannot be run in debugger mode. To debug a bootloadable project, convert the Application Type to Normal, debug it, and then convert it back to Bootloadable after debugging is done. Another option is to program the Bootloadable project .hex file onto the device and then use the "Attach to running target" option for debugging, while the bootloadable project is running. In this case, you can debug the bootloadable project only from the point where debugger is attached to the device. Refer to AN68272 (http://www.cypress.com/?rID=50230) for detailed information.



BLE

BLE Component is configured as a Beacon in non-connectable advertisement mode with no timeout (continuous transmission).

UART Interface

UART interface to USB serial device.

Watch Dog Timer

Watchdog timer is used for waking up the system when the system is in low power mode.

I2C Interface

I2C Interface to the temperature/humidity sensor.

Bootloadable

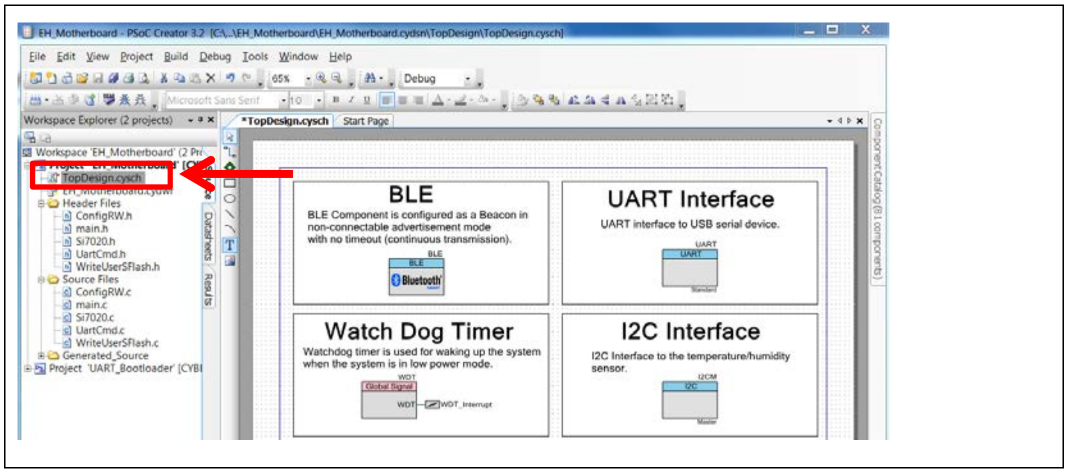
The Bootloadable component links the application to the Bootloader project's .hex file.

I/O Interface

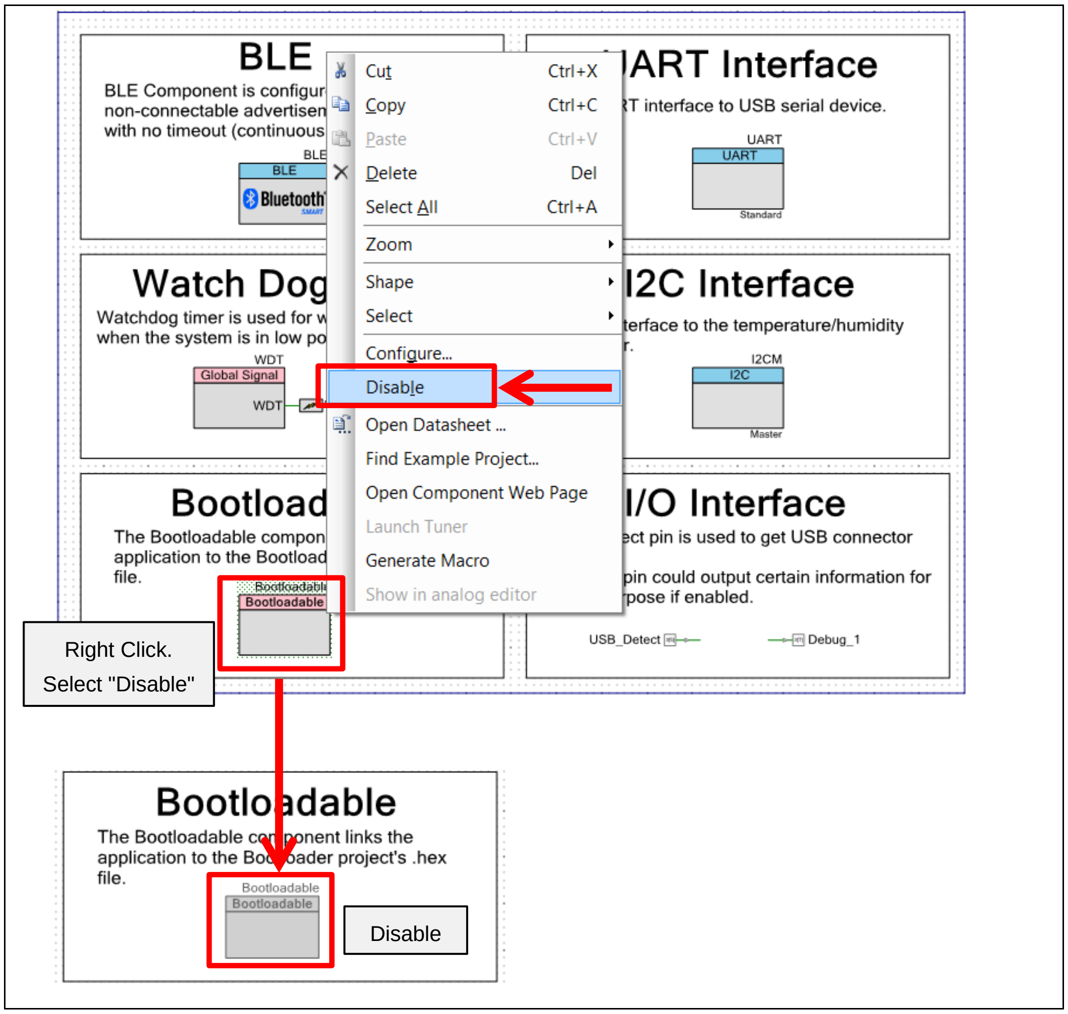
USB_Detect pin is used to get USB connector status.
Debug_1 pin could output certain information for debug purpose if enabled.

7.2.2.1 Convert the Application Type to Normal

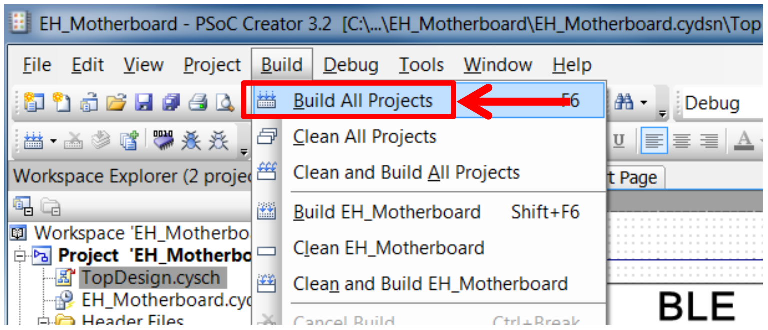
1. Open the "TopDesign.cysch".


EH_Motherboard - PSoC Creator 3.2 [C:\...\EH_Motherboard\EH_Motherboard.cydsn\TopDesign\TopDesign.cysch]
File Edit View Project Build Debug Tools Window Help
Workspace Explorer (2 projects)
TopDesign.cysch
Header Files
ConfigRW.h
main.h
Si7020.h
UartCmd.h
WriteUserSFlash.h
Source Files
ConfigRW.c
main.c
Si7020.c
UartCmd.c
WriteUserSFlash.c
Generated_Source
Project 'UART_Bootloader'
*TopDesign.cysch
Start Page
BLE
BLE Component is configured as a Beacon in non-connectable advertisement mode with no timeout (continuous transmission).
UART Interface
UART interface to USB serial device.
Watch Dog Timer
Watchdog timer is used for waking up the system when the system is in low power mode.
I2C Interface
I2C Interface to the temperature/humidity sensor.

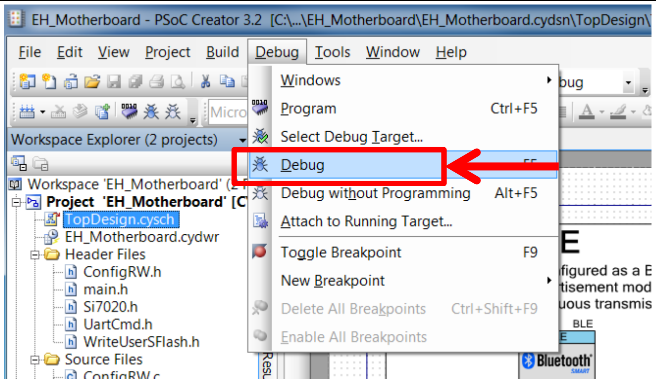
2. Right click on the Bootloadable component and disable it. The Bootloadable component will be disabled.



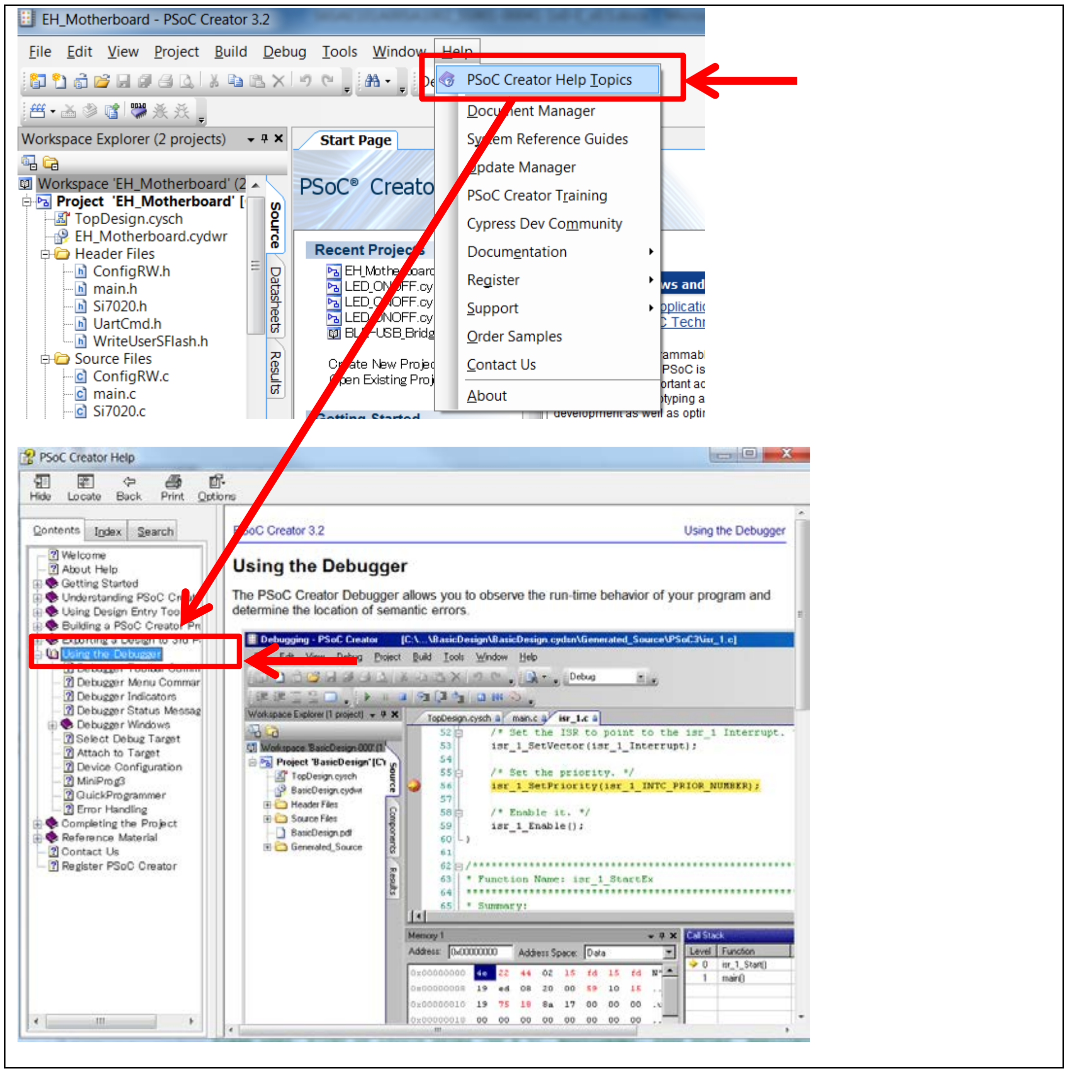
3. Select the "Menu>Build>Build All Projects" to rebulid the projects.


EH_Motherboard - PSoC Creator 3.2 [C:\...\EH_Motherboard\EH_Motherboard.cydsn\Top
File Edit View Project Build Debug Tools Window Help
Build All Projects F6
Clean All Projects
Clean and Build All Projects
Build EH_Motherboard Shift+F6
Clean EH_Motherboard
Clean and Build EH_Motherboard
Debug
Workspace Explorer (2 proje
Workspace 'EH_Motherbo
Project 'EH_Motherbo
TopDesign.cysch
EH_Motherboard.cy
BLE

4. Select the "Menu>Debug>Debug".


EH_Motherboard - PSoC Creator 3.2 [C:\...\EH_Motherboard\EH_Motherboard.cydsn\TopDesign\
File Edit View Project Build Debug Tools Window Help
Windows
Program Ctrl+F5
Select Debug Target...
Debug F5
Debug without Programming Alt+F5
Attach to Running Target...
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Enable All Breakpoints
Workspace Explorer (2 projects)
Workspace 'EH_Motherboard'
Project 'EH_Motherboard'
TopDesign.cysch
EH_Motherboard.cydwr
Header Files
ConfigRW.h
main.h
Si7020.h
UartCmd.h
WriteUserSFlash.h
Source Files
BLE
Bluetooth

5. Use "Help->PSoC Creator Help Topics->Using the Debugger" for information on using the debugger.



Using the Debugger

The PSoC Creator Debugger allows you to observe the run-time behavior of your program and determine the location of semantic errors.

6. Once done with debugging you need to:

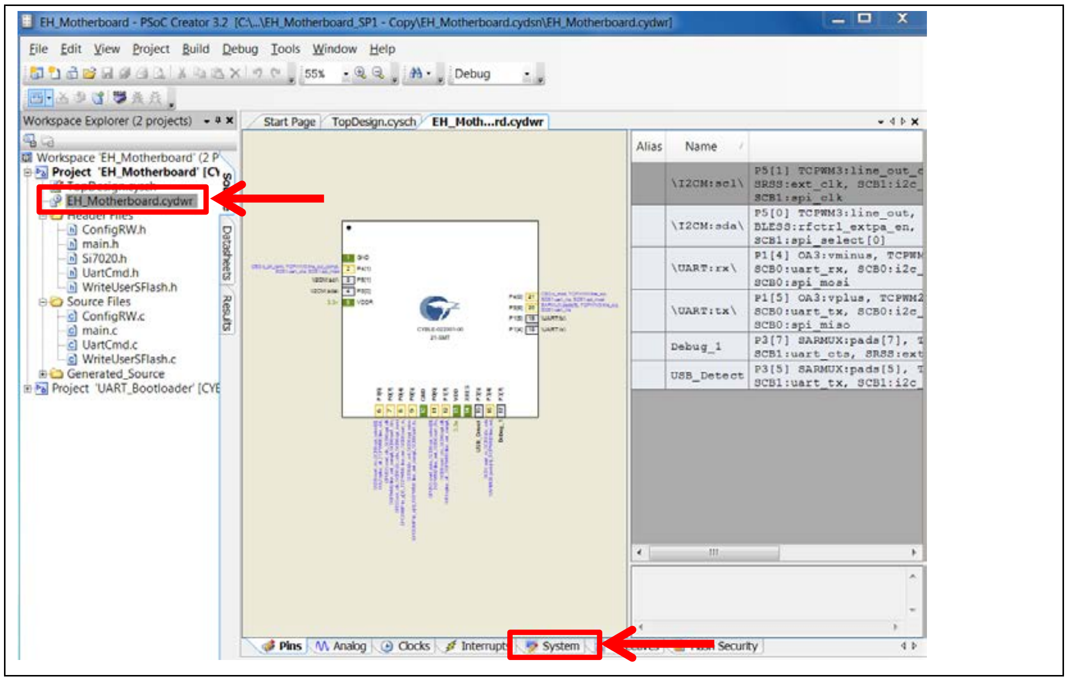
a. Right click on the bootloadable component and re-enable it.

b. Rebuild

c. Reprogram.

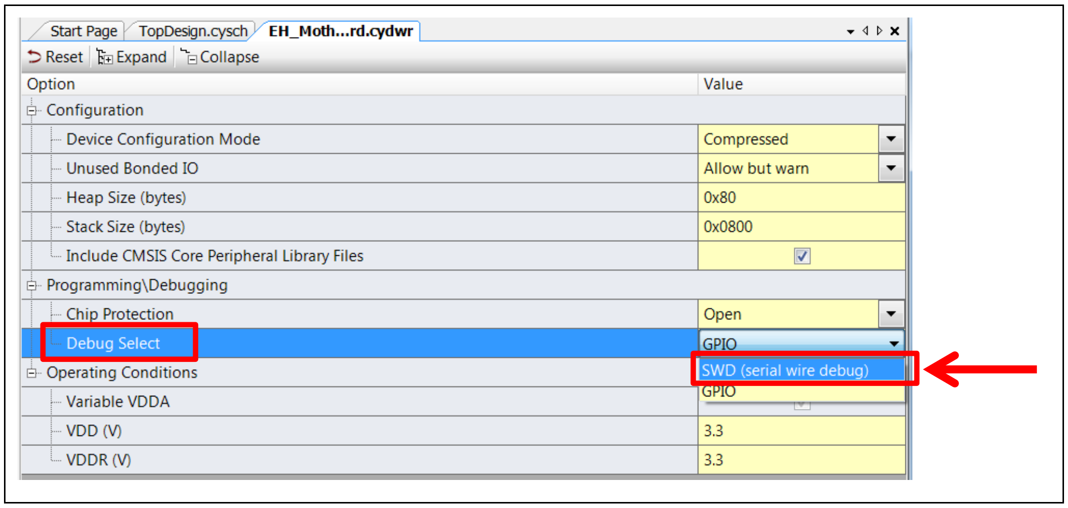
7.2.2.2 Attach to Running Target Option

1. Open the "EH_Motherboard.cydwr", then click the system tab.


EH_Motherboard - PSoC Creator 3.2 [C:\...\EH_Motherboard_SP1 - Copy\EH_Motherboard.cydsn\EH_Motherboard.cydwr]
File Edit View Project Build Debug Tools Window Help
55%
Debug
Workspace Explorer (2 projects)
Workspace 'EH_Motherboard' (2 P
Project 'EH_Motherboard' [C\
EH_Motherboard.cydwr
Header Files
ConfigRW.h
main.h
Si7020.h
UartCmd.h
WriteUserSFlash.h
Source Files
ConfigRW.c
main.c
UartCmd.c
WriteUserSFlash.c
Generated_Source
Project 'UART_Bootloader' [CYE
Start Page
TopDesign.cysch
EH_Moth...rd.cydwr
Datasheets
Results
Pins
Analog
Clocks
Interrupts
System
Security

Alias	Name	
	\I2CM:scl\	P5[1] TCPWM3:line_out_c SRSS:ext_clk, SCB1:i2c_ SCB1:spi_clk
	\I2CM:sda\	P5[0] TCPWM3:line_out, BLESS:rfctrl_extpa_en, SCB1:spi_select[0]
	\UART:rx\	P1[4] OA3:vminus, TCPWM SCB0:uart_rx, SCB0:i2c_ SCB0:spi_mosi
	\UART:tx\	P1[5] OA3:vplus, TCPWM2 SCB0:uart_tx, SCB0:i2c_ SCB0:spi_miso
	Debug_1	P3[7] SARMUX:pads[7], T SCB1:uart_cts, SRSS:ext
	USB_Detect	P3[5] SARMUX:pads[5], T SCB1:uart_tx, SCB1:i2c_

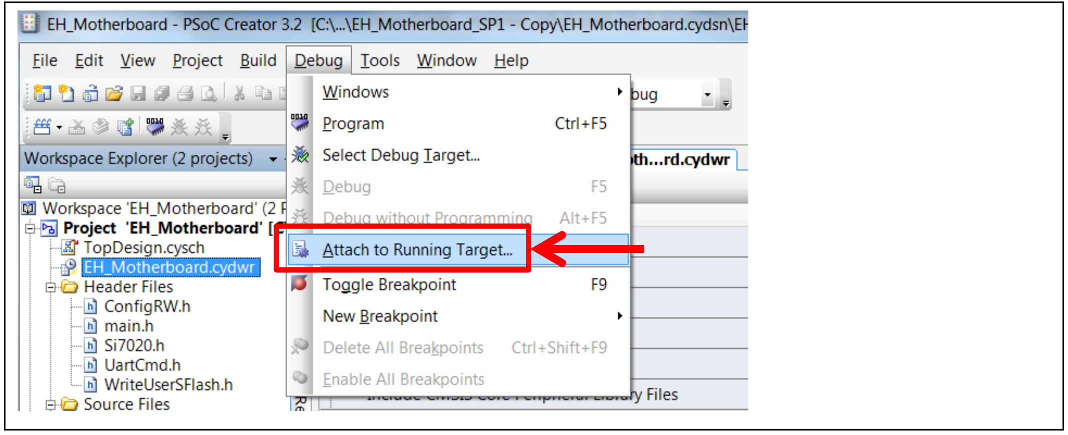
2. Change the Debug Select to the SWD (serial wire debug). The consumption current will be increased when set to the SWD. Therefore please re-select the GPIO to restore the Solar-Powered operation.



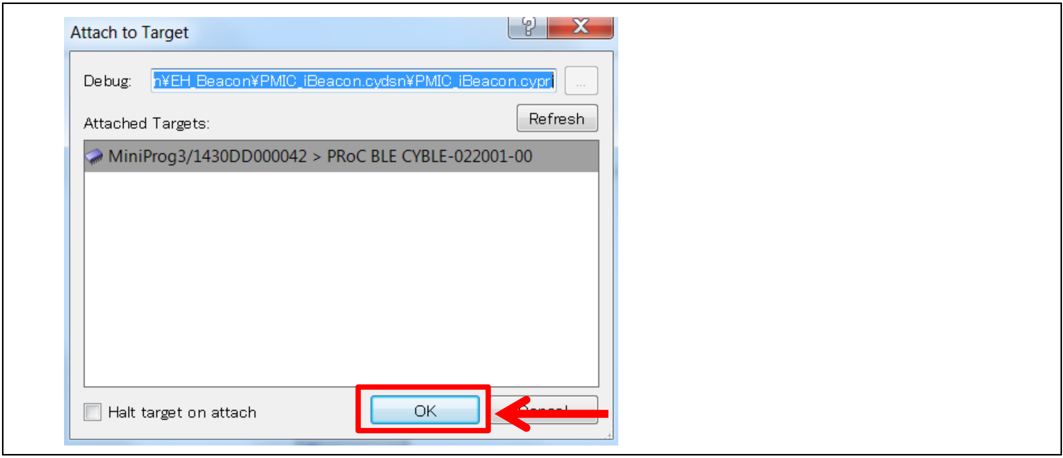
Option	Value
Configuration	
Device Configuration Mode	Compressed
Unused Bonded IO	Allow but warn
Heap Size (bytes)	0x80
Stack Size (bytes)	0x0800
Include CMSIS Core Peripheral Library Files	✓
Programming\Debugging	
Chip Protection	Open
Debug Select	GPIO
Operating Conditions	SWD (serial wire debug) / GPIO
Variable VDDA	
VDD (V)	3.3
VDDR (V)	3.3

3. Run the Program. Refer to "7.2.1 Program".

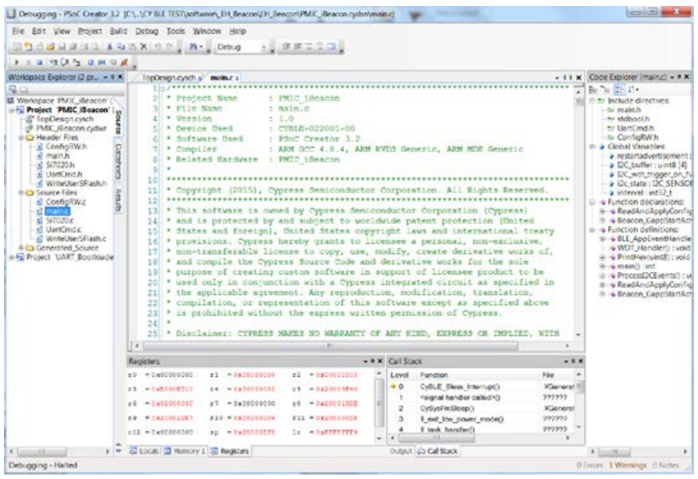
4. Select "Menu>Debug>Attach to Running Target....".


EH_Motherboard - PSoC Creator 3.2 [C:\...\EH_Motherboard_SP1 - Copy\EH_Motherboard.cydsn\EH
File Edit View Project Build Debug Tools Window Help
Workspace Explorer (2 projects)
Workspace 'EH_Motherboard' (2 P
Project 'EH_Motherboard' [
TopDesign.cysch
EH_Motherboard.cydwr
Header Files
ConfigRW.h
main.h
Si7020.h
UartCmd.h
WriteUserSFlash.h
Source Files
Windows
Program Ctrl+F5
Select Debug Target...
Debug F5
Debug without Programming Alt+F5
Attach to Running Target...
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Enable All Breakpoints

5. Attached to Target window will appear, and the click OK button.


Attach to Target
Debug: n¥EH_Beacon¥PMIC_iBeacon.cydsn¥PMIC_iBeacon.cypr
Attached Targets:
Refresh
MiniProg3/1430DD000042 > PRoC BLE CYBLE-022001-00
Halt target on attach
OK
Cancel

6. The window of PSoC Creator is changed to Debug mode automatically.



```
/*******************************************************************************
* Project Name      : PMIC_iBeacon
* File Name         : main.c
* Version           : 1.0
* Device Used       : CYBLE-022001-00
* Software Used     : PSoC Creator 3.2
* Compiler          : ARM GCC 4.8.4, ARM RVDS Generic, ARM MDK Generic
* Related Hardware  : PMIC_iBeacon
*
********************************************************************************
* Copyright (2015), Cypress Semiconductor Corporation. All Rights Reserved.
********************************************************************************
* This software is owned by Cypress Semiconductor Corporation (Cypress)
* and is protected by and subject to worldwide patent protection (United
* States and foreign), United States copyright laws and international treaty
* provisions. Cypress hereby grants to licensee a personal, non-exclusive,
* non-transferable license to copy, use, modify, create derivative works of,
* and compile the Cypress Source Code and derivative works for the sole
* purpose of creating custom software in support of licensee product to be
* used only in conjunction with a Cypress integrated circuit as specified in
* the applicable agreement. Any reproduction, modification, translation,
* compilation, or representation of this software except as specified above
* is prohibited without the express written permission of Cypress.
*
* Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH
```

7. Use "Help->PSoC Creator Help Topics->Using the Debugger" for information on using the debugger.

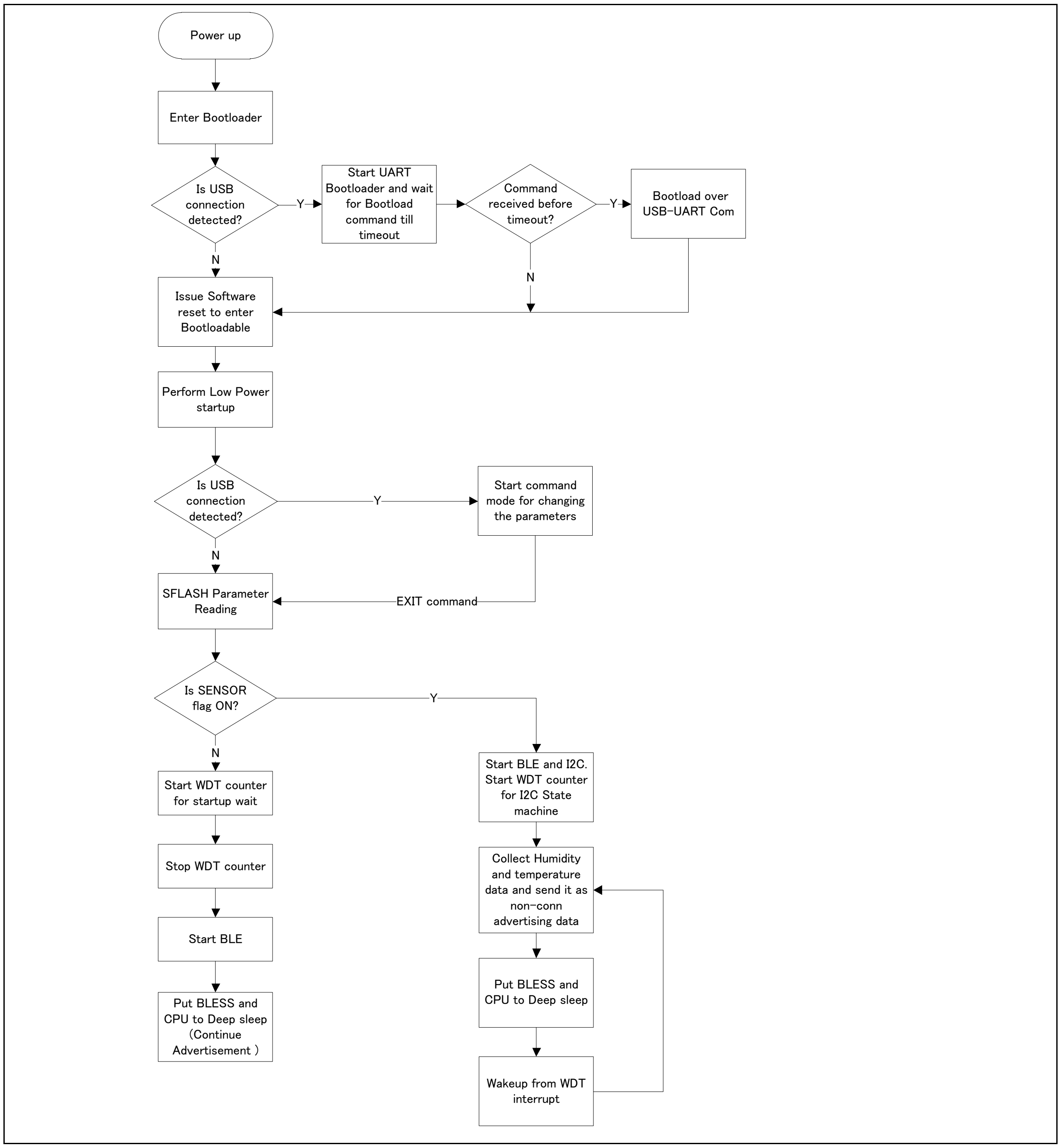
8. Once done with debugging you need to:
    a. Change the Debug Select to the GPIO.
    b. Rebuild
    c. Reprogram.

8. Example Project

This chapter introduces you to the initial provided firmware of the Solar-Powered IoT Device Kit. We will discuss features such as the BLE Beacon Process and WSN with BLE Beacon Process. Refer to "Getting Started with EZ-BLE PRoC Module" for standard reference design except energy harvesting.

8.1 Flow Diagram

Following is the flow diagram for the example project of the kit firmware.


Power up
Enter Bootloader
Is USB connection detected?
Y
Start UART Bootloader and wait for Bootload command till timeout
Command received before timeout?
Y
Bootload over USB-UART Com
N
N
Issue Software reset to enter Bootloadable
Perform Low Power startup
Is USB connection detected?
Y
Start command mode for changing the parameters
N
SFLASH Parameter Reading
EXIT command
Is SENSOR flag ON?
Y
N
Start WDT counter for startup wait
Stop WDT counter
Start BLE
Put BLESS and CPU to Deep sleep (Continue Advertisement)
Start BLE and I2C. Start WDT counter for I2C State machine
Collect Humidity and temperature data and send it as non-conn advertising data
Put BLESS and CPU to Deep sleep
Wakeup from WDT interrupt

8.2 Function List

Following is the function list for the example project of the kit firmware.

Function Name	Description
CyDelay	Blocks for milliseconds
CyDelayUs	Blocks for microseconds
USB_Detect_Read	Read the current value on the pins (P3.5) of the Digital Port in right justified
UART_Start	Invokes SCB_Init() and SCB_Enable()
UART_UartPutString	Places a NULL terminated string in the transmit buffer to be sent at the next available bus time
CyBtldr_Start	Entire bootload operation
Bootloader_SET_RUN_TYPE	Schedule Bootloadable to start after reset
CySoftwareReset	Forces a software reset of the device.
CySysClkWriteEcoDiv	Set the divider for ECO
CySysClkEcoStart	Starts the External Crystal Oscillator (ECO)
CySysClkEcoStop	Stops the megahertz crystal
WDT_Interrupt_StartEx	Sets up the interrupt and enables
CySysClkWcoStart	Enables Watch Crystal Oscillator (WCO)
CySysClkWcoSetPowerMode	Sets the power mode for the 32 KHz WCO
CySysWdtLocked	Reports the WDT lock state
CySysClkSetLfclkSource	Sets the clock source for the LFCLK clock
CySysWdtEnable	Enables the specified WDT counters
CySysWdtLock	Locks out configuration changes to the Watchdog timer registers and ILO configuration register
CySysWdtUnlock	Unlocks the Watchdog Timer configuration register
CySysPmDeepSleep	Puts the part into the Deep Sleep state
CySysWdtEnable	Enables the specified WDT counters
CySysWdtDisable	Disables the specified WDT counters
ConfigRW_CheckSFlash	Checks whether there is configuration data in User SFlash
cmd_uart_sub	Function for processing UART commands
CyBle_Start	Initializes the BLE Stack, which consists of the BLE Stack Manager, BLE Controller, and BLE Host modules
CyBle_ProcessEvents	Checks the internal task queue in the BLE Stack, and pending operation of the BLE Stack
CyBle_EnterLPM	requests the underlying BLE modules such as BLE Controller, BLE Host Stack and BLE Stack manger to enter into one of the supported low power modes
CyEnterCriticalSection	No interrupts allowed while entering system low power modes
CySysPmDeepSleep	Puts the part into the Deep Sleep state

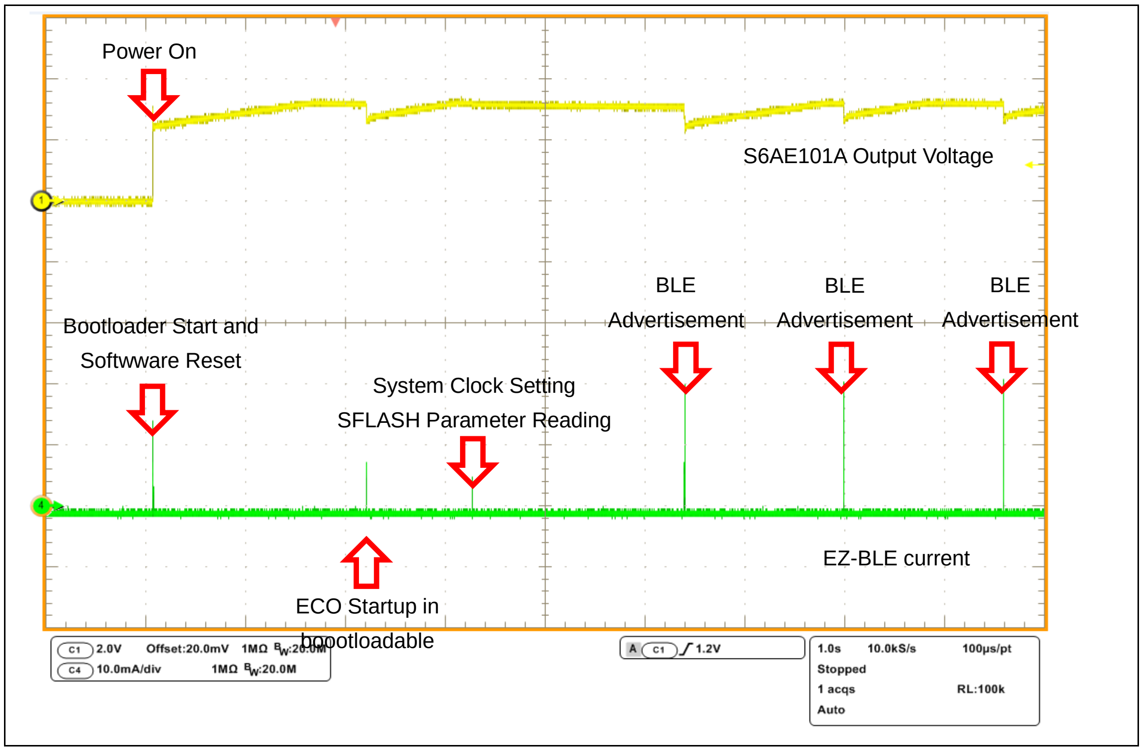
Function Name	Description
CySysClkWriteHfclkDirect	Selects the direct source for the HFCLK
CySysClkImoStart	Enables the IMO
CySysClkImoStop	Disables the IMO
CySysPmSleep	Puts the part into the Sleep state
ProcessI2CEvents	Handles I^2C events according to current state and update the state value
ProcessBeaconEvents	Handles Beacon events according to current state and update the state value
Si7020_Init	Init Temperature and Humidity sensor Si7020
Si7020_WriteRead	Send conversion command and read Temperature and Humidity data
Beacon_GappStartAdvertisement	Be used to start the advertisement using the specified interval
CyBle_GappStopAdvertisement	To exit from discovery mode

8.3 BLE Beacon Process

Following is the detail information of BLE Beacon Process.

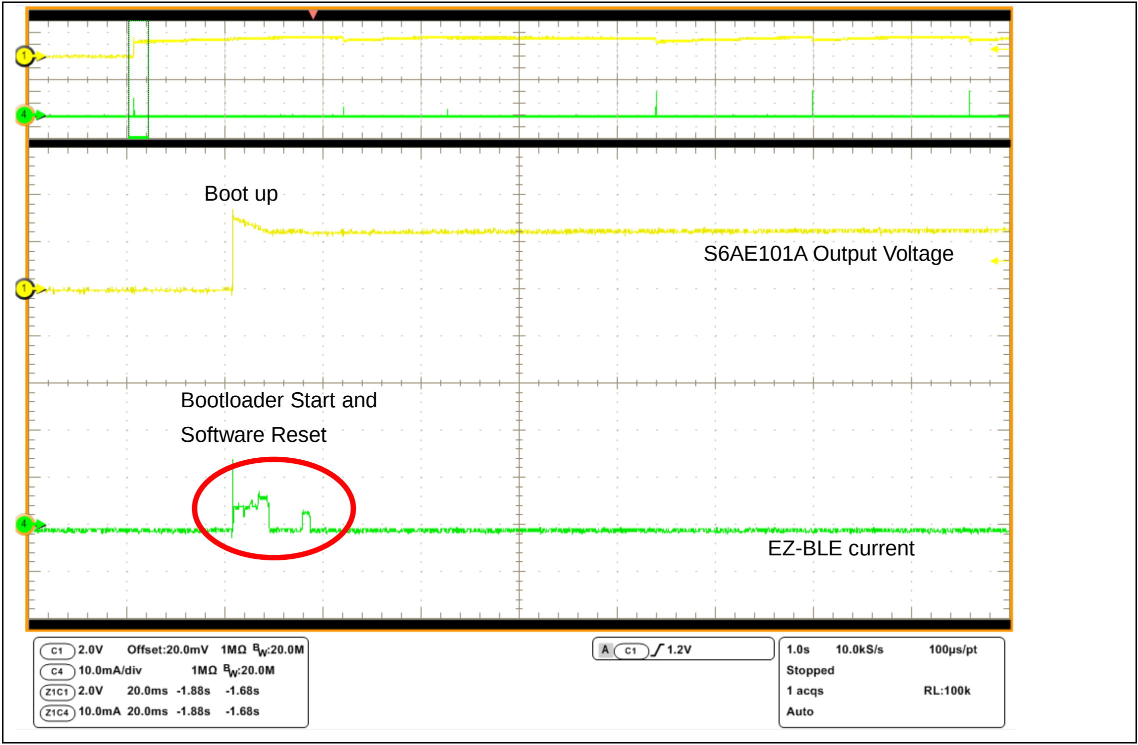
0. Complete BLE Beacon Process

The complete BLE Beacon process invokes running the Beacon_GappStartAdvertisement().The following waveform shows the BLE Advertisement states.



1. Power Up and Bootloader Start

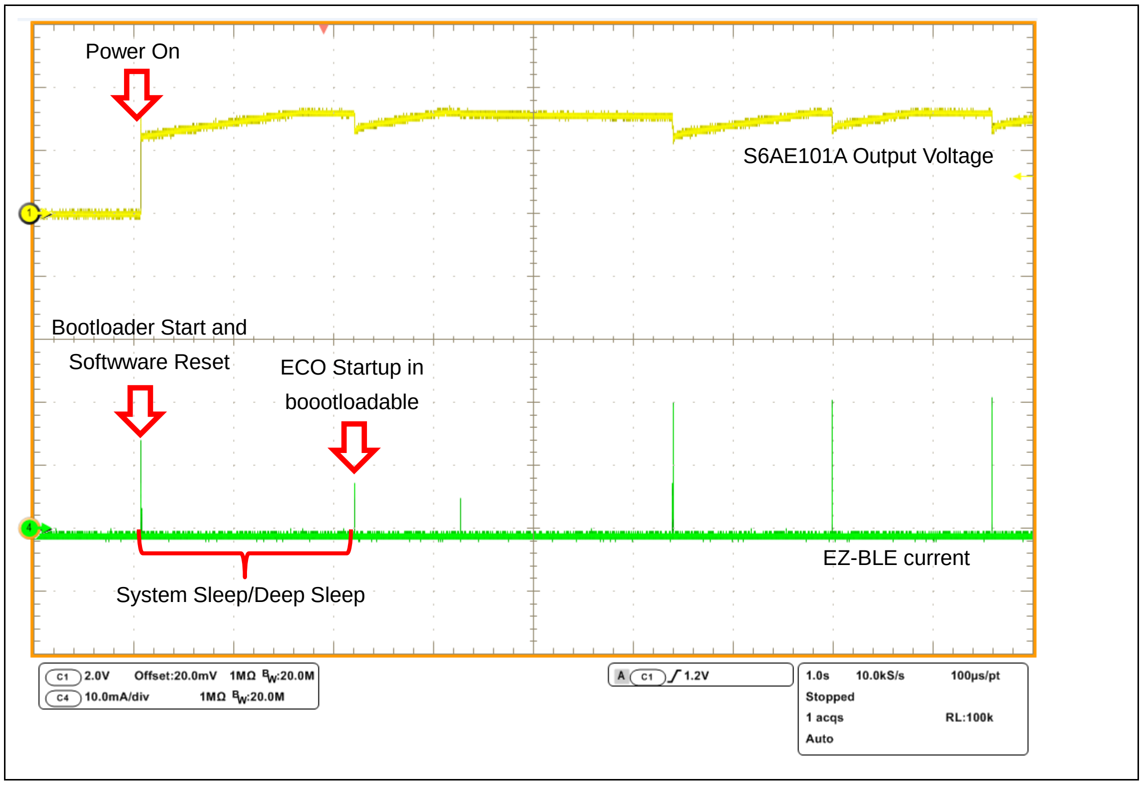
The code starts executing from the bootloader which can be found in the "UART_Bootloader" project. The code is optimized to read the USB detect line (P3.5 of EZ-BLE) and immediately switch to Bootloadable code if it was not detected.



```
CyGlobalIntEnable;

if(USB_DETECT_Read())
{
    UART_Start();
    CyDelay(100u);
    UART_UartPutString("UART Bootloader Starting. It will take 10s.\r\n");

 /* This API does the entire bootload operation. After a succesful
  * bootload operation, this API transfers program control to the
  * new application via a software reset */
    CyBtldr_Start();
}
else
{
  /* Schedule Bootloadable to start after reset */
    Bootloader_SET_RUN_TYPE(Bootloader_START_APP);

    CySoftwareReset();
}
```

2. System Initialization and Low Power Startup

Once the bootloader lanuches the bootloadable application, the code from the "EH_Motherboard" PSoC Creator project is started. This code begins with low power startup functions. This allows the system to conserve power during clock startup, especially the WCO which takes 2 seconds for startup. This is crucial to allow the project to survive on solar power.



```
CyGlobalIntEnable;
/* Set the divider for ECO, ECO will be used as source when IMO is switched
off to save power, to drive the HFCLK */
CySysClkWriteEcoDiv(CY_SYS_CLK_ECO_DIV8);

/* If USE_WCO_FOR_TIMING is set, then do the following:
 * 1. Shut down the ECO (to reduce power consumption while WCO is starting)
 * 2. Enable WDT to wakeup the system after 500ms (WCO startup time).
 * 3. Configure PRoC BLE device in DeepSleep mode for the 500ms WCO startup
time
 * 4. After WCO is enabled, restart the ECO so that BLESS interface can
function */
#if USE_WCO_FOR_TIMING
CySysClkEcoStop();
WDT_Interrupt_StartEx(WDT_Handler);
CySysClkWcoStart();
CySysWdtUnlock(); /* Unlock the WDT registers for modification */
CySysWdtWriteMode(SOURCE_COUNTER, CY_SYS_WDT_MODE_INT);
CySysWdtWriteClearOnMatch(SOURCE_COUNTER, COUNTER_ENABLE);
CySysWdtWriteMatch(SOURCE_COUNTER, COUNT_PERIOD_WCO);
CySysWdtEnable(CY_SYS_WDT_COUNTER0_MASK);
CySysWdtLock();
CySysPmDeepSleep(); /* Wait for the WDT interrupt to wake up the device
*/
(void)CySysClkEcoStart(2000u);
CyDelayUs(500u);
(void)CySysClkWcoSetPowerMode(CY_SYS_CLK_WCO_LPM);    /* Switch to the
low power mode */
CySysClkSetLfclkSource(CY_SYS_CLK_LFCLK_SRC_WCO);
CySysWdtUnlock();
CySysWdtDisable(CY_SYS_WDT_COUNTER0_MASK);
CySysWdtLock();
#endif
```

3. System Clock Setting

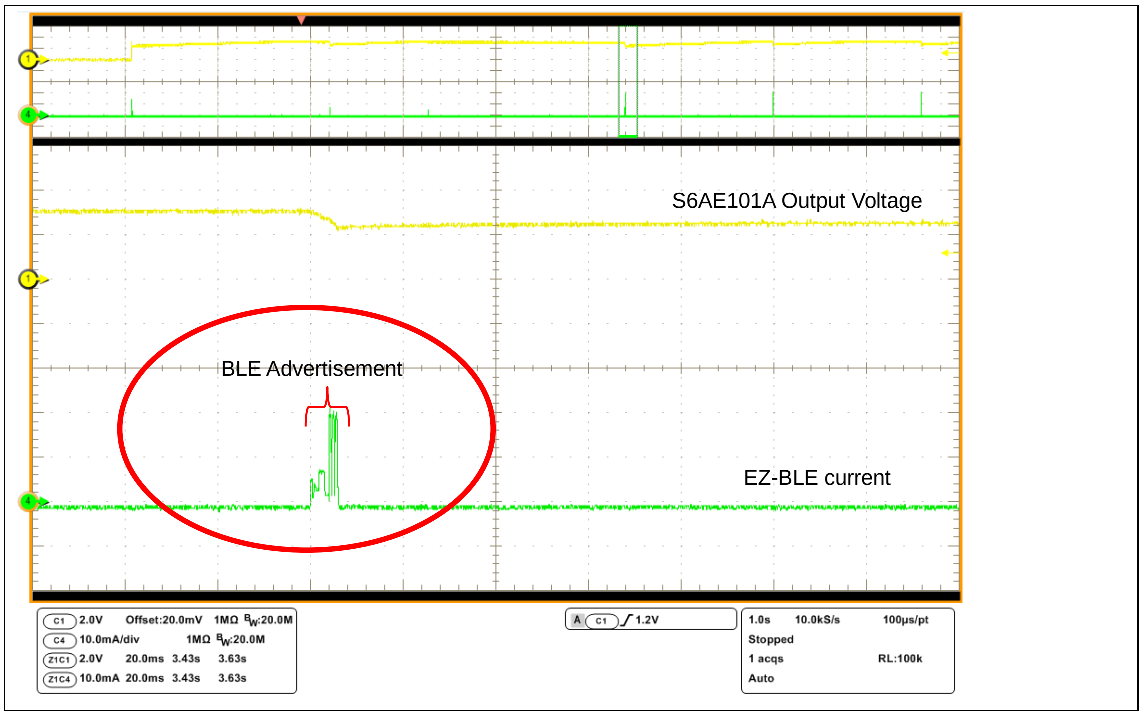
System is set with IMO at 12 MHz and ECO at 3 MHz.

4. SFLASH Parameter Reading

The user flash read is done at every start of the bootloadable to read the stored beacon data in SFLASH and then use it for broadcasting. The Flash read is done in CYBLE_EVT_STACK_ON to allow some breathing space between system on and flash read.

5. WDT Configuration and BLE Beacon Starting

WDT counter is configured for startup waiting. Then ProcessBeaconEvents() is called to process BLE Beacon events.



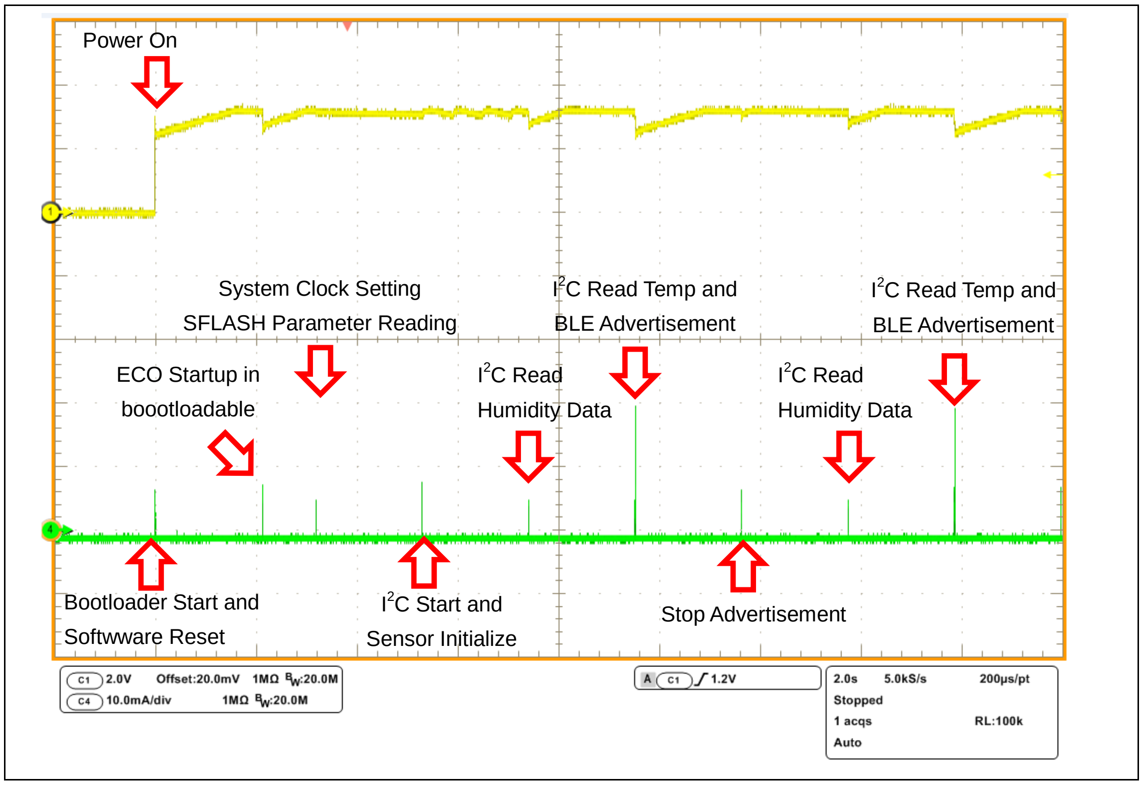
```
void ProcessBeaconEvents(void)
{/* If I2C_COUNTER triggers a new interrupt after another 3 seconds. */
    if(wdt_trigger_on_flag)
    {
        CYBLE_API_RESULT_T apiResult;
         switch(beacon_state)
        {
            case BEACON_WAIT:
       /* Wait two secounds to earn enough power to start advertising */
                beacon_state = BEACON_START;
            break;
            case BEACON_START:
                /* Stop I2C_COUNTER, and start advertisement */
                CySysWdtUnlock();
                /* Unlock the WDT registers for modification */
                CySysWdtDisable(CY_SYS_WDT_COUNTER1_MASK);
                CySysWdtLock();
                apiResult = Beacon_GappStartAdvertisement(interval);
                /* If fails to start advertisement, halt the processor. */
                if(apiResult != CYBLE_ERROR_OK)
                {
                    CYASSERT(0);
                }
                beacon_state = BEACON_RUN;
            break;
            default:
            break;
        }
        wdt_trigger_on_flag = false;
    }
}
```

8.4 WSN with BLE Beacon Process

Following is the detail information of WSN with BLE Beacon Process.

0. Complete WSN with BLE Beacon Process

The complete WSN with Beacon process involves running the I^{2}C state machine on each WDT interrupt and to send new data as part of ADV packet. The following waveform shows the various states in rotation:



1 to 4. Same Process as BLE Beacon

Refer to the process 1 to 4 of "8.3 BLE Beacon Process".

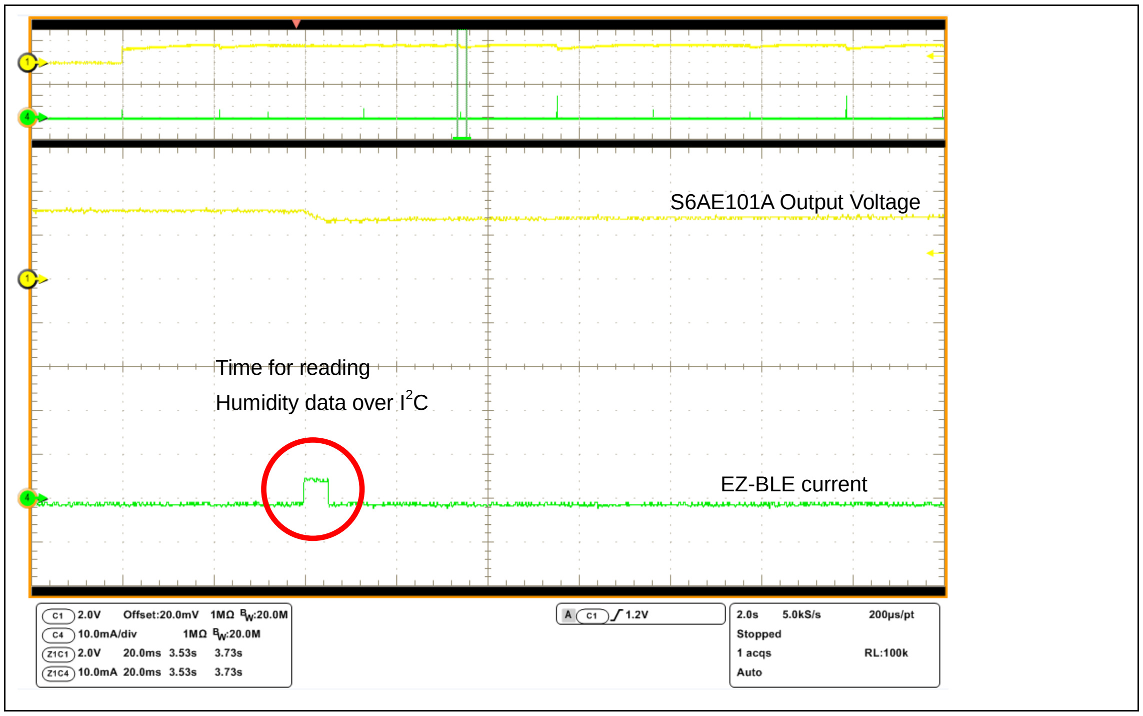
5. Components Start and WDT Configuration

Along with BLE start, WDT counter is configured for periodic interrupts to run the I^{2}C state machine. This state machine initiates I^{2}C as well as reads the data from the sensor to send it as part of the Advisement (ADV) packet. Also, CyBle_ProcessEvents() is called to process BLE events.

```
CySysWdtUnlock();
CySysWdtDisable(CY_SYS_WDT_COUNTER0_MASK);
CySysWdtWriteMode(I2C_COUNTER, CY_SYS_WDT_MODE_INT);
CySysWdtWriteClearOnMatch(I2C_COUNTER, COUNTER_ENABLE);
#if I2C_SENSOR_ENABLE
 /* If I2C sensor is enabled and sensor setting is on. */
 if(Sensor_Flag)
 {
   CySysWdtWriteMatch(I2C_COUNTER, I2C_COUNT_PERIOD);
 }
else
#endif
/* If I2C sensor is disabled or sensor setting is off. */
{
  CySysWdtWriteMatch(I2C_COUNTER, ((uint32)32767*3));
}
CySysWdtEnable(CY_SYS_WDT_COUNTER1_MASK);
CySysWdtLock();
```

6. I^{2}C Read Humidity Data

After I^{2}C is started, the next state reads the relative humidity value from I^{2}C sensor and updates the ADV packet with new humidity data. The system then goes back to deep sleep.



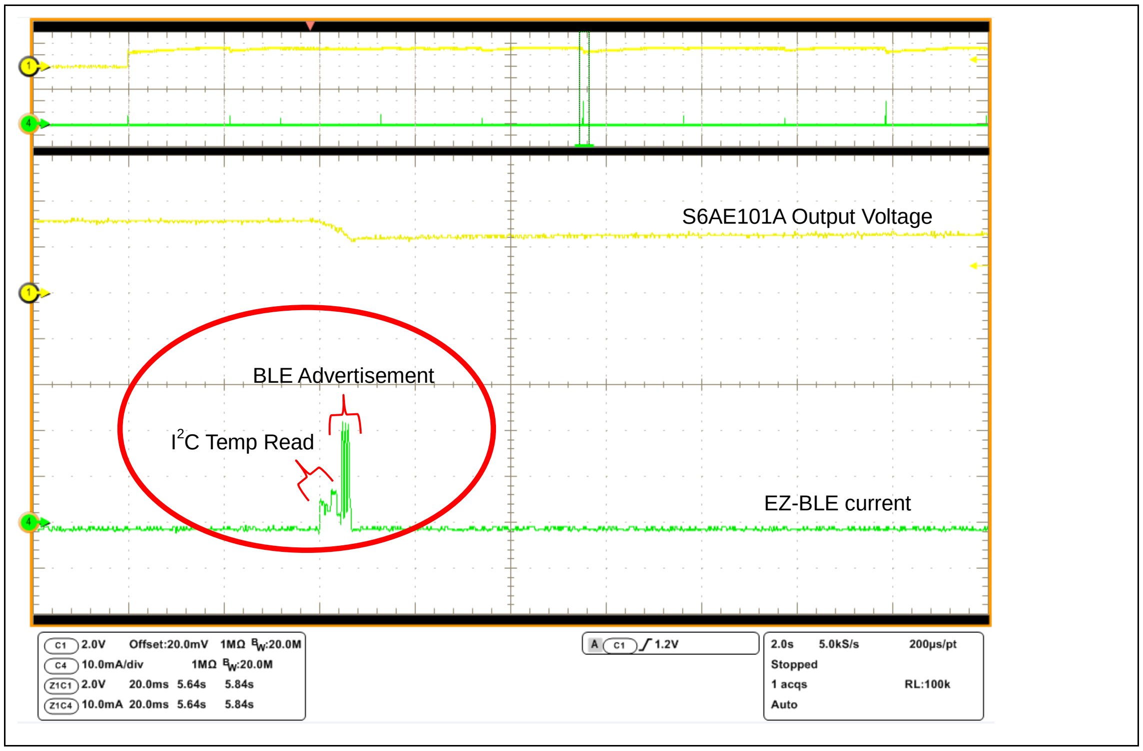
```
C1   2.0V     Offset:20.0mV  1MΩ  BW:20.0M
C4   10.0mA/div              1MΩ  BW:20.0M
Z1C1 2.0V     20.0ms  3.53s  3.73s
Z1C4 10.0mA   20.0ms  3.53s  3.73s

A C1 1.2V

2.0s    5.0kS/s    200μs/pt
Stopped
1 acqs             RL:100k
Auto
```

```
case I2C_READ_HUMIDITY:
      /* Read RH data from I2C Sensor */
      I2C_buffer[0] = SI7020_MEASURE_RH;
      Si7020_WriteRead(I2C_buffer, 1, 2);

      /* Update RH index of ADV packet with new value */
      cyBle_discoveryData.advData[ADDR_TEM_OFFSET]= I2C_buffer[0];

      /* Set next I2C state */
      i2c_state = I2C_READ_TEMP;
break;
```

Note: Si7020_WriteRead(I2C_buffer, 1, 2); 1 = number of write byte, 2= number of read byte

7. I^{2}C Read Temperature Data and BLE Advertisement

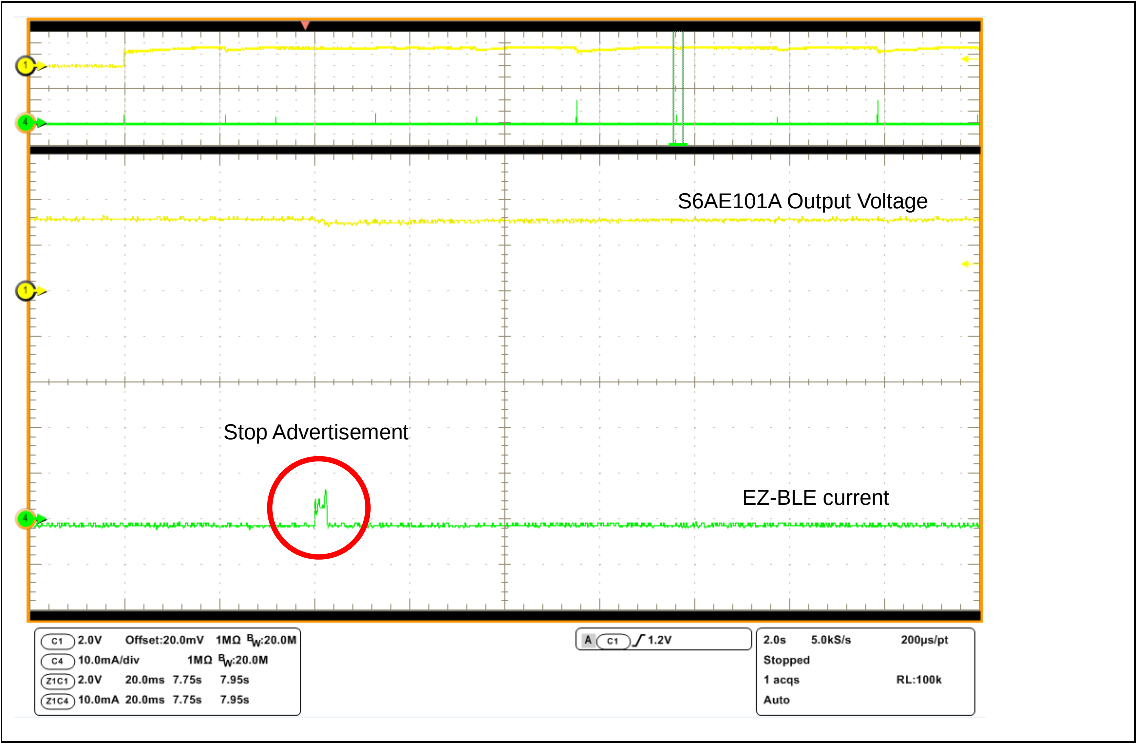
At next WDT interrupt, the system wakes up and enters the I^{2}C temperature state. Here the I^{2}C data for temperature is read and ADV packet is updated with this new data. Also, the GAP Advertisement is started so that the new ADV packet can be transmitted by BLE.



```
case I2C_READ_TEMP:
      /* Read Temperature data from I2C Sensor */
      I2C_buffer[2] = SI7020_READ_TEMP;
      Si7020_WriteRead(&I2C_buffer[2], 1, 2);

      /* Update Temperature index of ADV packet with new value */
      cyBle_discoveryData.advData[ADDR_TEM_OFFSET]= I2C_buffer[2];

       /* When sensor used, advertise interval is fixed to 10.24s. */
      apiResult = Beacon_GappStartAdvertisement(ITRVL_SENSOR_ON);
      /* If fails to start advertisement, halt the processor. */
       if(apiResult != CYBLE_ERROR_OK)
       {
          CYASSERT(0);
       }
       CyBle_ProcessEvents();

      /* Set next I2C state */
      i2c_state = I2C_STOP_ADV;
break;
```

8. Stop Current BLE Advertisement

At next WDT interrupt, the stop advertisement API is called to stop the current advertisement. This is required before the ADV packet can be updated with new data in next I^2C state.



8.5 BLE Beacon Format

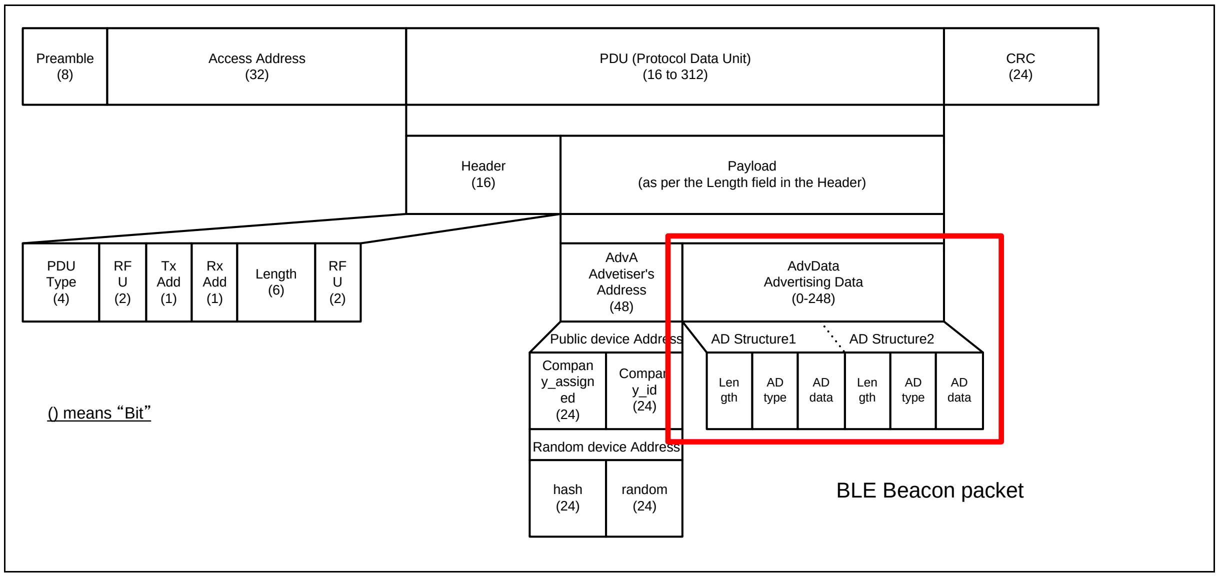
BLE Beacon is a one-way communication method that broadcast at a regular interval. It consists small packets of data (30byte) of "Advertisements".

Beacons that want to be "discovered" can be used for a variety of smartphone or computer applications to trigger things like push messages app actions, and prompts.

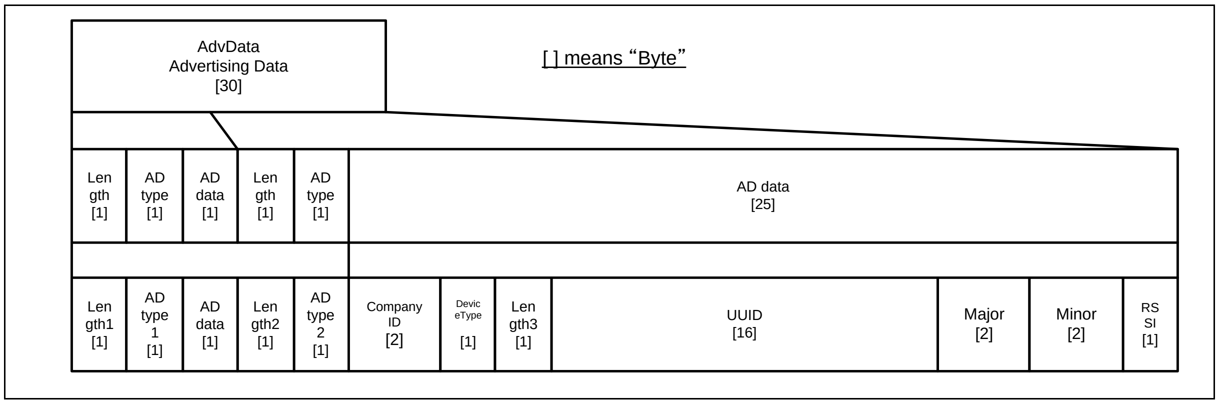
Following is a link layer format of Bluetooth Low Energy for Advertising channel packet format. The link layer of Bluetooth Low Energy has "Preamble", "Access Address" and "PDU (Protocol Data Unit)" and "CRC". Note that following information are for Advertising channel packets format, the information does not include "Data channel packets".

- "Preamble" must set "10101010b"
- "Access Address" must set "10001110100010011011111011010110b (0x8E89BED6)"
- "PDU" has "Header" and "Payload"

The packet structure of BLE Beacon belongs to "Advertising Data" in "Payload".



Following is the detail of BLE Beacon packet format for the kit.



Following are the initial value in the example project for "Solar-Powered BLE Beacon Operation".

<BLE Beacon Format>

–	Length1	0x02
–	AD type1	0x01
–	AD data	0x04
–	Length2	0x1A
–	AD type2	0xFF
–	Company ID	0x0131 [Cypress Semiconductor Corporation]
–	Device type	0x02
–	Length3	0x15
–	UUID[1]	00050001-0000-1000-8000-00805F9B0131 [hex]
–	Major[2]	0x0001
–	Minor[3]	0x0001
–	RSSI[4]	0xC3 [-61dBm]

<Others>

–	Transmitter power	3 dBm
–	Advertise interval	1500 ms

The kit uses "ReadAndApplyConfig" function in main.c to update the BLE Beacon packet. Note that the kit does not use "GAP Settings" of Configure 'BLE' on TopDesign.cysch, because the SFLASH Parameter is read and the BLE Beacon packet is updated when every turn the power on.

[1] This is a 16 byte string used to differentiate a large group of related beacons.

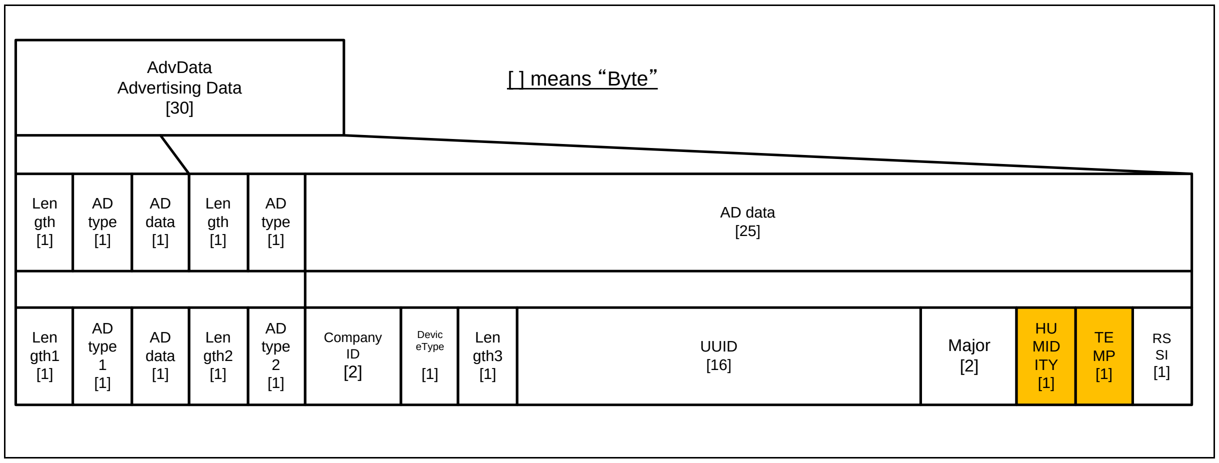
[2] This is a 2 byte string used to distinguish a smaller subset of beacons within the larger group.

[3] This is a 2 byte string meant to identify individual beacons.

[4] Received Signal Strength Indication. This is used to determine proximity (distance) from the beacon.

8.6 Sensor Transmitter Specification of WSN

When the “SENSOR” command is ON, the kit transmits the sensor data in the Beacon protocol packet where temperature and humidity sensor information are stored in the Minor region (2 bytes). Humidity data is stored in the upper byte and temperature data is stored in the lower byte.

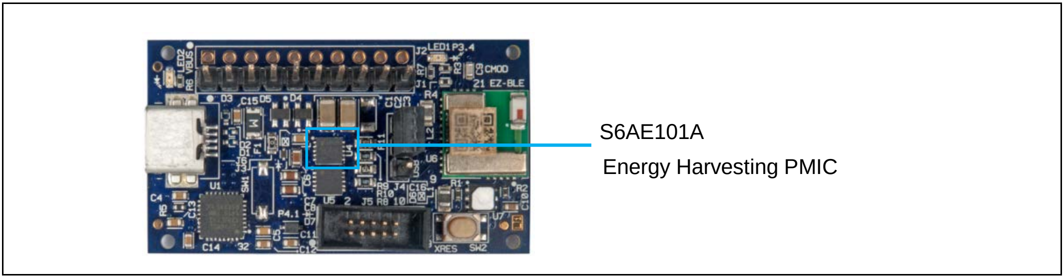


Stores temperature and humidity sensor information in Minor region (2 bytes) and sends packets

Meanwhile, whether temperature and humidity sensor information is written on Minor region is set by changing the ON/OFF status with the "SENSOR" command described. If user sets SENSOR OFF, the board will send data based on the Beacon protocol excluding sensor information.

9. Energy Harvesting PMIC (S6AE101A)

The following is the specification of the Energy Harvesting Power Management IC (S6AE101A) on the Energy Harvesting Motherboard. Refer to the datasheet of S6AE101A (DS405-00026) for the latest information.



9.1 Recommended Operating Conditions

Table 9-1 Recommended Operating Conditions

Parameter	Symbol	Condition	Value			Unit
			Min	Typ	Max	
Power supply voltage 1 (*1)	V_{VDD}	VDD pin	2.0	3.3	5.5	V
Power supply voltage 2 (*1)	V_{VBAT}	VBAT pin	2.0	3.0	5.5	V
VOUT1 setting resistance	R_{VOUT}	Sum of R1, R2, R3	10	−	−	MΩ
VDD capacitance	C1	VDD pin	10	−	−	µF
VINT capacitance	C2	VINT pin	1	−	−	µF
VOUT maximum setting voltage	V_{SYSH}	VSTORE1 pin	1.25	−	5.2	V
VOUT minimum setting voltage	V_{SYSL}	VSTORE1 pin	1.1	−	$V_{SYSH} \times 0.9$	V
Operating ambient temperature	Ta	−	−40	−	+85	°C

*1: When GND = 0V

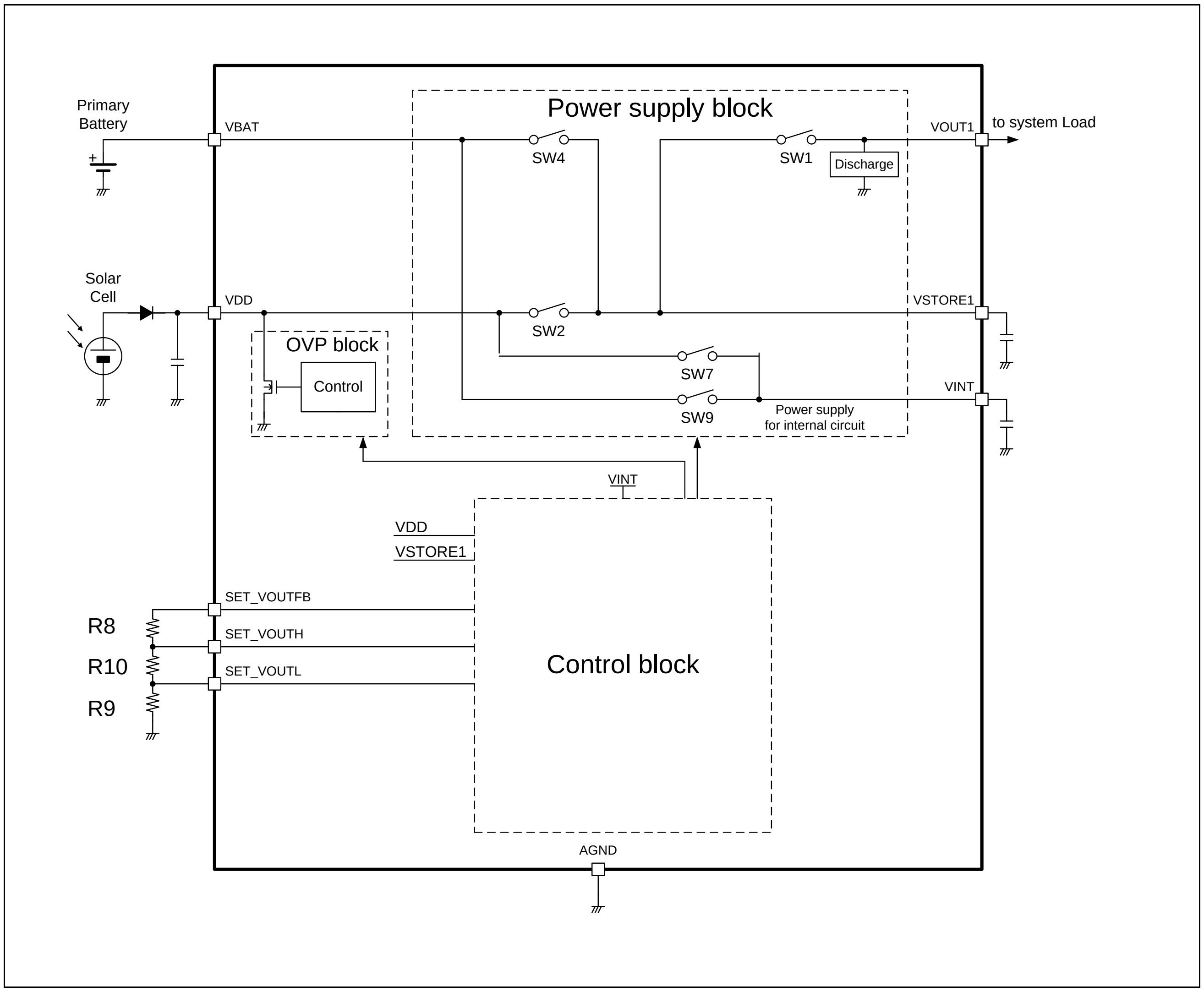
9.2 DC Characteristics

Table 9-2 DC Characteristics

Parameter	Symbol	Condition	Value			Unit
			Min	Typ	Max	
Minimum Input power in start-up	W_{START}	VDD pin, Ta = +25°C, V_{VOUTH} setting =3V, By applying 0.4µA to VDD, when VOUT1 reaches 3V×95% after the point when VDD reaches 3V.	−	−	1.2	µW
Consumption current	I_{QIN1}	VDD pin input current, VDD=3V, Open VBAT pin, V_{VOUTH}=1.25V setting, Ta=+25°C, SET_VOUTFB resistance>100MΩ, VOUT1 Load=0mA	−	250	375	nA
OVP detection voltage	V_{OVPH}	VDD pin	5.2	5.4	5.5	V
OVP release voltage	V_{OVPL}		5.1	5.3	5.4	V
OVP detection hysteresis	V_{OVPHYS}		−	0.1	−	V
OVP protection current	I_{OVP}	VDD pin input current	6	−	−	mA

9.3 Block Diagram

Figure 9-1 Block Diagram



The initial output voltage setting of Energy Harvesting Motherboard is following.

- VVOUTH

$$\mathrm{VVOUTH} = \frac{57.5 \times (R10 + R9)}{11.1 \times (R8 + R10 + R9)} = \frac{57.5 \times (6.8M + 4.7M)}{11.1 \times (6.8M + 6.8M + 4.7M)} = 3.27[\mathrm{V}]$$

- VVOUTL

$$\mathrm{VVOUTL} = \frac{57.5 \times R9}{11.1 \times (R8 + R10 + R9)} = \frac{57.5 \times 4.7M}{11.1 \times (6.8M + 6.8M + 4.7M)} = 1.93[\mathrm{V}]$$

10. Hardware

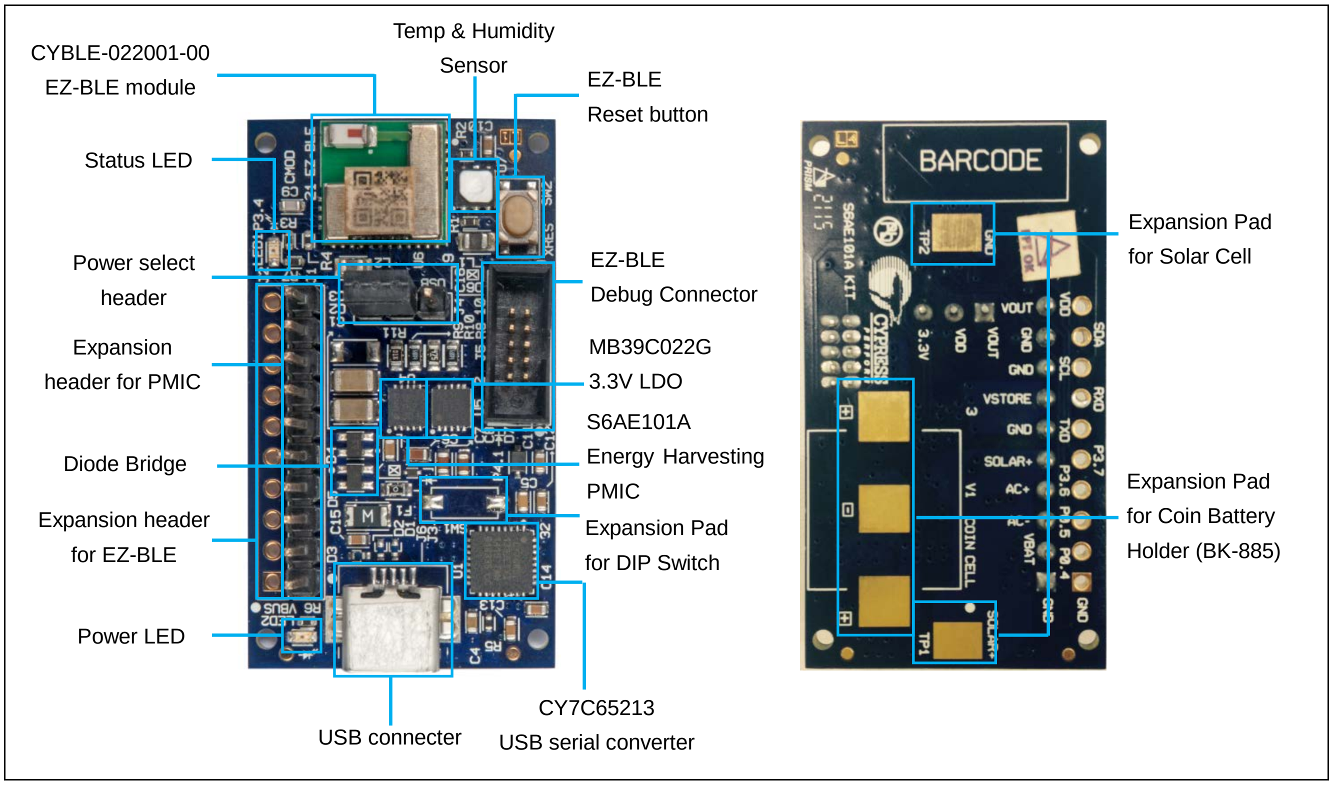
10.1 Energy Harvesting Motherboard

10.1.1 Board Detail

The Energy Harvesting Motherboard consists of the blocks shown in Figure 10-1 .

- Energy Harvesting Power Management IC (S6AE101A)
- EZ-BLE PRoC Module (CYBLE-022001-00)
- USB Serial Converter IC (CY7C65213)
- LDO for USB bus power (MB39C022G)
- Temp & Humidity Sensor (Si7020-A10)
- Diode Bridge for the vibration energy input (1SS383 x2pcs)
- USB mini-B connector
- Jumpers with socket to select power supply
- MiniProg3 programing header for EZ-BLE
- 10pin expansion header for Energy Harvesting Power Management IC
- 10pin expansion header for EZ-BLE
- USB power LED, Status LED
- Push switch for EZ-BLE reset
- Pads of DIP switch for future expansion (optional)
- Pads for BK-885 coin battery holder (optional)
- Pads for AM-1801 or BCS4630B9 solar cell (optional)

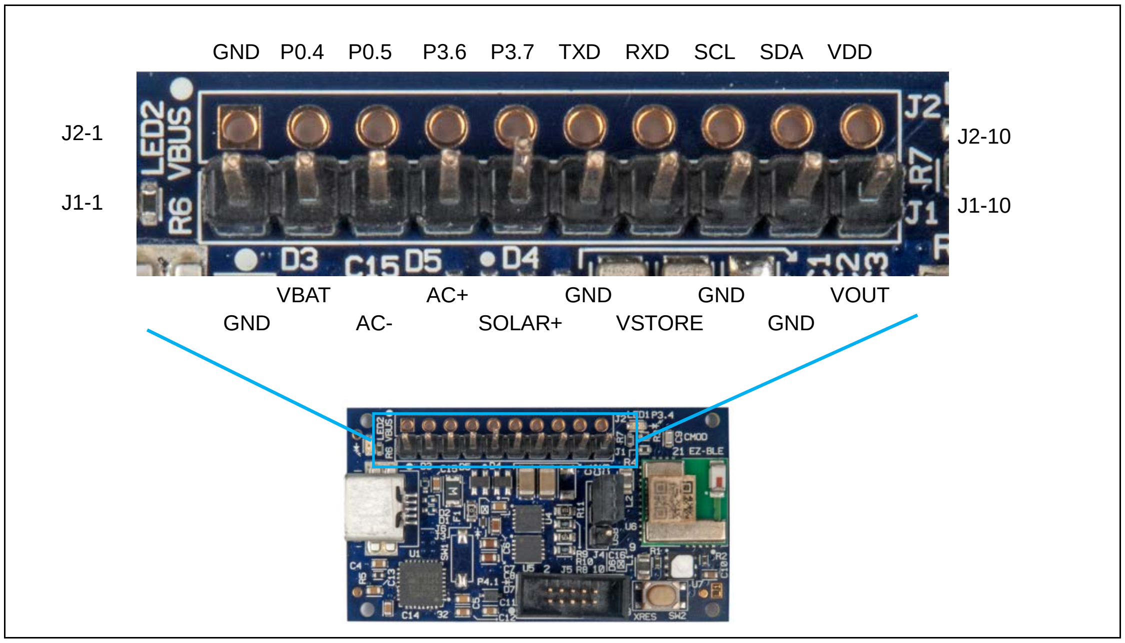
Figure 10-1 Energy Harvesting Motherboard



10.1.2 Input/Output Pin Description

Table 10-1 Input/Output Pin Description

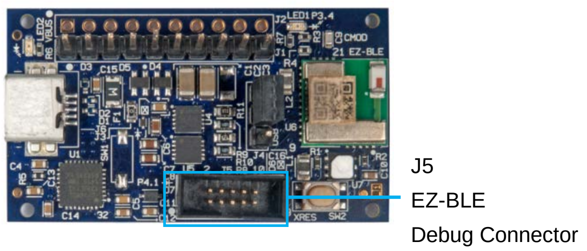
Circuit Pin No.	Silk-Printed Name	I/O	Description
Expansion pins for S6AE101A Energy Harvesting PMIC (J1)			
J1-1	GND	-	GND pin
J1-2	VBAT	I	Primary battery input pin for Hybrid operation
J1-3	AC-	I	AC- voltage input for vibration energy
J1-4	AC+	I	AC+ voltage input for vibration energy
J1-5	SOLAR+	I	Solar cell input
J1-6	GND	-	GND pin
J1-7	VSTORE	O	Storage output pin
J1-8	GND	-	GND pin
J1-9	GND	-	GND pin
J1-10	VOUT	O	Output voltage pin
Expansion pins for EZ-BLE PRoC Module (J2)			
J2-1	GND	-	GND pin
J2-2	P0.4	I/O	GPIO P0.4 of EZ-BLE
J2-3	P0.5	I/O	GPIO P0.5 of EZ-BLE
J2-4	P3.6	I/O	GPIO P3.6 of EZ-BLE
J2-5	P3.7	I/O	GPIO P3.7 of EZ-BLE
J2-6	TXD	O	UART TXD of USB serial (RXD of EZ-BLE)
J2-7	RXD	I	UART RXD of USB serial (TXD of EZ-BLE)
J2-8	SCL	O	I^{2}C clock pin of EZ-BLE
J2-9	SDA	I/O	I^{2}C data pin of EZ-BLE
J2-10	VDD	-	Power Supply Input



10.1.3 Debug Connector Description

Table 10-2 Debug Connector Description

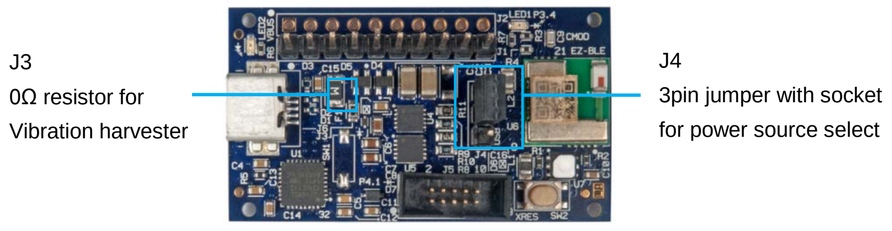
Circuit Pin No.	Silk-Printed Name	I/O	Description
J5-1	VDD	I	Power pin
J5-2	SWDIO	I/O	SWD data pin
J5-3	GND	-	GND pin
J5-4	SWDCLK	I/O	SWD clock pin
J5-5	GND	-	GND pin
J5-6	N.C	-	Non connection
J5-7	GND	-	GND pin
J5-8	N.C	-	Non connection
J5-9	GND	-	GND pin
J5-10	XRES	I	Reset pin of EZ-BLE



10.1.4 Jumper Description

Table 10-3 Jumper Description

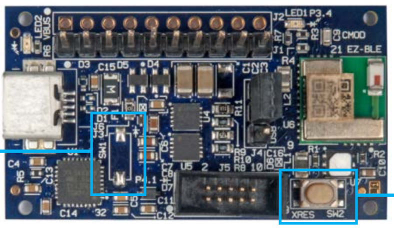
Circuit Pin No.	Description	Default Settings
J3	0Ω resistor for bridge rectifier of vibration harvester	Short
J4	3pin jumper with socket for power source select of EZ-BLE EH: 3.3V to 1.9V Energy Harvesting PMIC VOUT USB: 3.3V LDO output via USB bus power	EH



10.1.5 Switch Description

Table 10-4 Switch Description

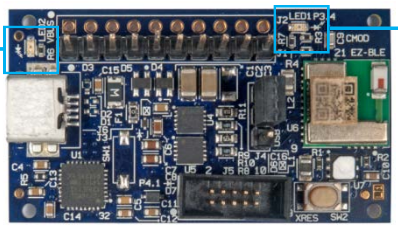
Circuit Pin No.	Silk-Printed Name	Description
SW1	SW1 (Not mounted)	DIP switch connection to GPIO P4.1 of EZ-BLE PRoC Module for future expansion
SW2	XRES SW2	Reset Button for BLE module



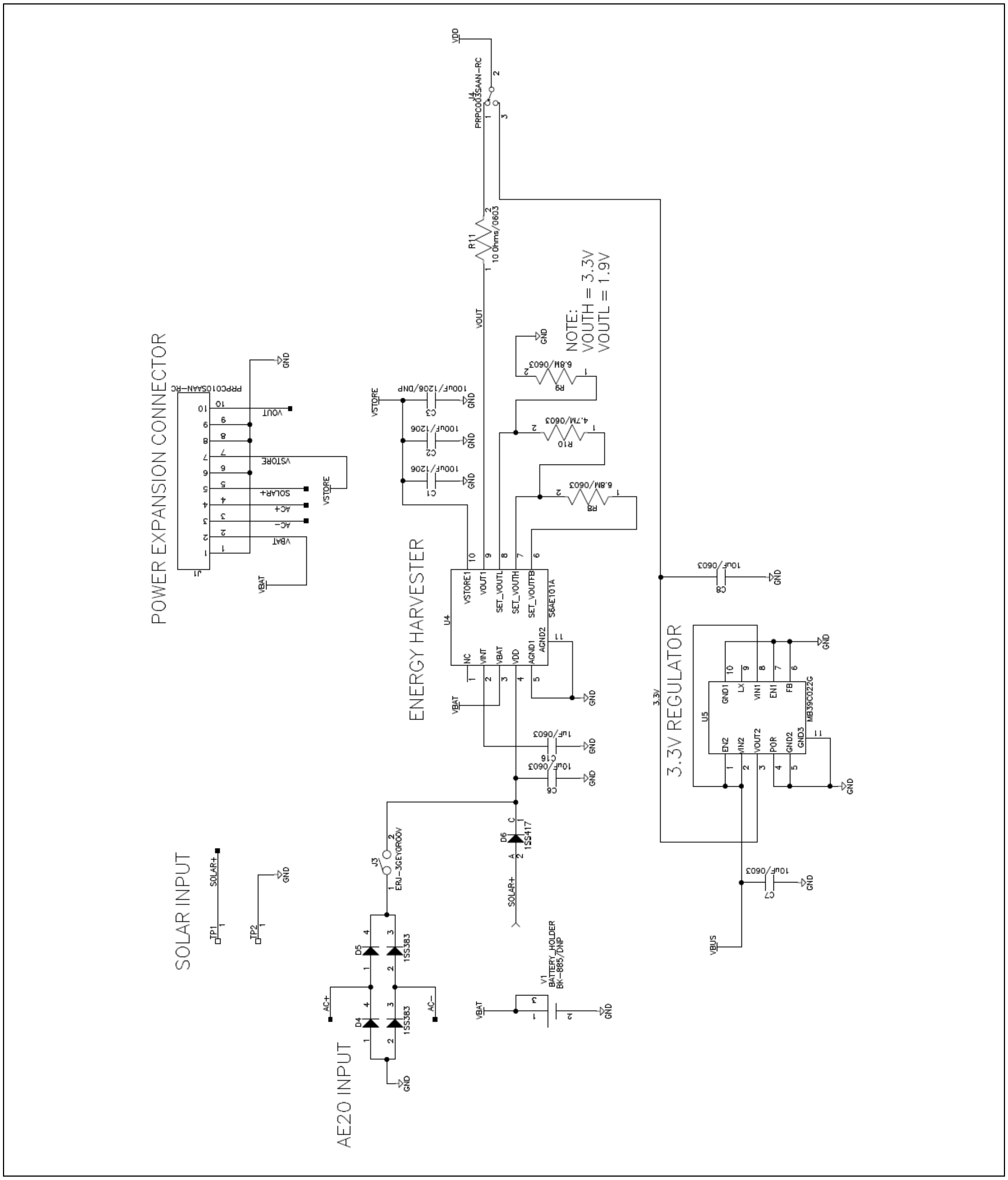
10.1.6 LED Description

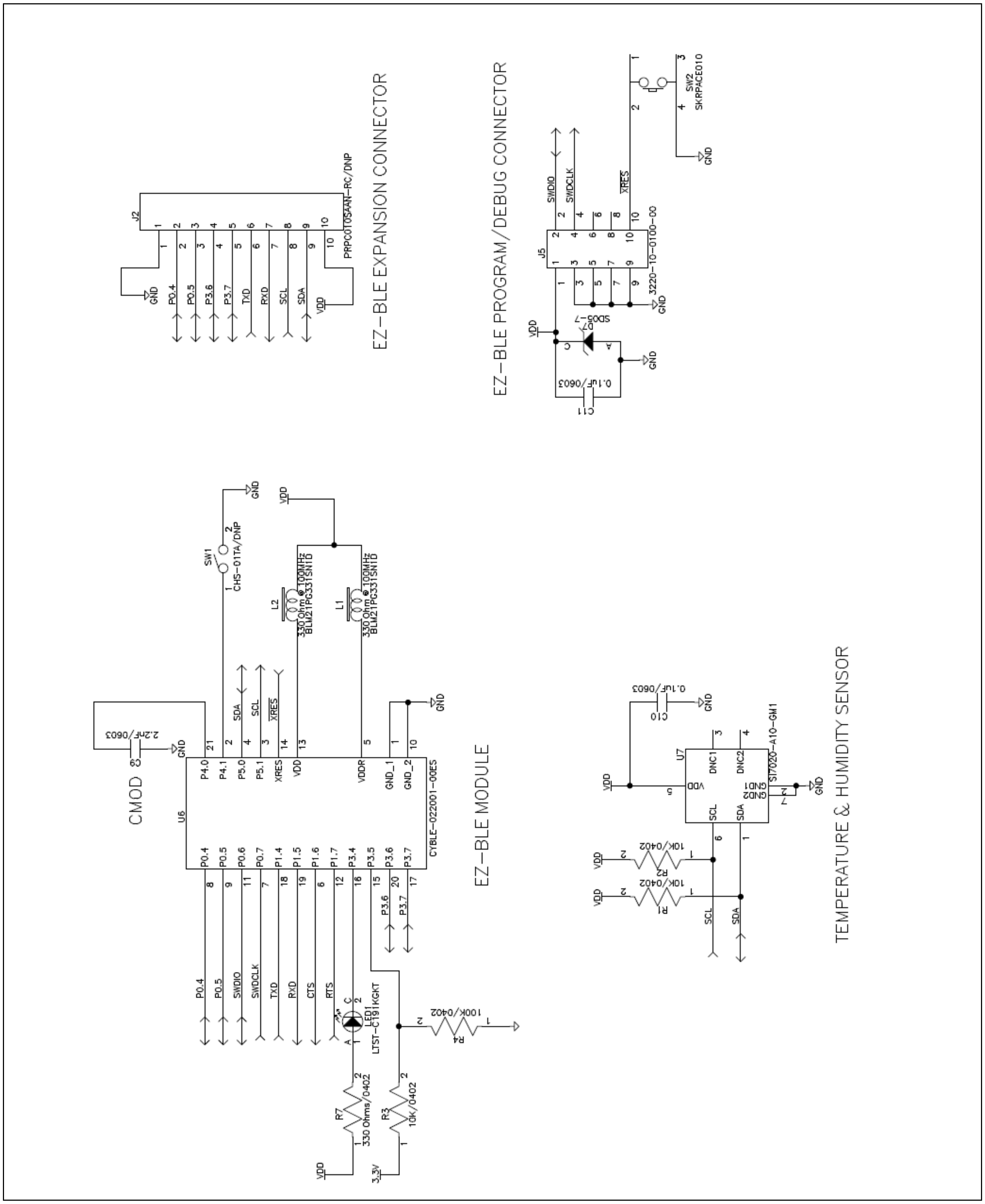
Table 10-5 LED Description

Circuit Pin No.	Silk-Printed Name	Description
LED1	LED1 P3.4	Status LED (GPIO P3.4 of EZ-BLE)
LED2	LED2 VBUS	USB power LED



10.1.7 Circuit




EZ-BLE EXPANSION CONNECTOR
J2
PRPC010SAAN-RC/DNP
GND
P0.4
P0.5
P3.6
P3.7
TXD
RXD
SCL
SDA
VDD
EZ-BLE PROGRAM/DEBUG CONNECTOR
J5
3220-10-0100-00
SWDIO
SWDCLK
XRES
SW2
SKRPACE010
D7
SD05-7
C11
0.1uF/0603
CMOD
C9
2.2nF/0603
U6
CYBLE-022001-00ES
EZ-BLE MODULE
SW1
CHS-01TA/DNP
L1
L2
330 Ohm @ 100MHz
BLM21PG331SN1D
VDDR
GND_1
GND_2
XRES
CTS
RTS
LED1
LTST-C191KGKT
R4
100K/0402
R7
330 Ohms/0402
R3
10K/0402
3.3V
TEMPERATURE & HUMIDITY SENSOR
U7
SI7020-A10-GM1
DNC1
DNC2
GND1
GND2
C10
0.1uF/0603
R1
R2
10K/0402

10.1.8 BOM List

No	Qty	Reference	Parts Number	Description	Manufacture	Note
1	2	C1, C2, (C3)	GRM31CR60J107ME39K	100 µF/1206	TDK Corporation	C3 Non mount
2	3	C4, C5, C16	GRM188R61C105KA93D	1 µF/0603	TDK Corporation	
3	3	C6, C7, C8	GRM188R61C106MA73D	10 µF/0603	TDK Corporation	
4	1	C9	GRM1885C1H222JA01D	2.2 nF/0603	TDK Corporation	
5	6	C10, C11, C12, C13, C14, C15	GRM188R71C104KA01D	0.1 µF/0603	TDK Corporation	
6	3	D1, D2, D3	CG0402MLC-05LG	TVS/0402	Bourns, Inc.	
7	2	D4, D5	1SS383	DUAL DIODE	TOSHIBA CORPORATION	
8	1	D6	1SS417	DIODE	TOSHIBA CORPORATION	
9	1	D7	SD05-7	DIODE	Diodes Incorporated	
10	1	F1	1206L050/15YR	FUSE 500mA	Littelfuse	
11	2	L1, L2	BLM21PG331SN1D	330 Ω @ 100MHz	Murata Manufacturing Co., Ltd.	
12	2	LED1, LED2	LTST-C191KGKT	GREEN	Lite-On Inc	
13	1	U1	CY7C65213-32LTXI	USB-UART LP Bridge Controller	Cypress Semiconductor Corporation	
14	1	U4	S6AE101A	Energy Harvesting Power Management IC	Cypress Semiconductor Corporation	
15	1	U5	MB39C022G	1ch DCDC, 1ch LDO	Cypress Semiconductor Corporation	
16	1	U6	CYBLE-022001-00ES	Bluetooth Low Energy Module	Cypress Semiconductor Corporation	
17	1	U7	SI7020-A10-GM1	Temp & Humidity Sensors	Silicon Labs	
18	3	R1, R2, R3	ERJ-2GEJ103X	10 kΩ/0402	Panasonic Corporation	
19	1	R4	ERJ-2GEJ104X	100 kΩ/0402	Panasonic Corporation	
20	1	R5	ERJ-2GEJ472X	4.7 kΩ/0402	Panasonic Corporation	
21	1	R6	ERJ-2GEJ471X	470Ω/0402	Panasonic Corporation	
22	1	R7	ERJ-2GEJ331X	330Ω/0402	Panasonic Corporation	
23	2	R8, R9	CRCW06036M80FKEA	6.8 MΩ/0603	Vishay Dale	
24	1	R10	RC0603FR-074M7L	4.7 MΩ/0603	Yageo	
25	1	R11	MCR03ERTF10R0	10Ω/0603	Rohm Semiconductor	
26	0	(SW1)	CHS-01TA	1pin DIP Switch	Copal Electronics Inc	Non mount
27	1	SW2	SKRPACE010	Push Switch	ALPS ELECTRIC CO., LTD	
28	0	(V1)	BK-885	Coin Battery Holder	-	Non mount
29	1	J1, (J2)	PRPC010SAAN-RC	2.54mm 10pin Header	Sullins Connector Solutions	J2 Non mount
30	1	J3	ERJ-3GEY0R00V	0Ω/0603	Panasonic Corporation	

No	Qty	Reference	Parts Number	Description	Manufacture	Note
31	1	J4	PRPC003SAAN-RC	2.54mm 3pin Header with socket	Sullins Connector Solutions	
32	1	J5	3220-10-0100-00	1.27mm 10pin Box Header	CNC Tech	
33	1	J6	UX60SC-MB-5ST	USB Mini Connector	Hirose Electric Co., Ltd	
34	2	TP1, TP2	TEST_PAD_3.0x4.0	-	-	

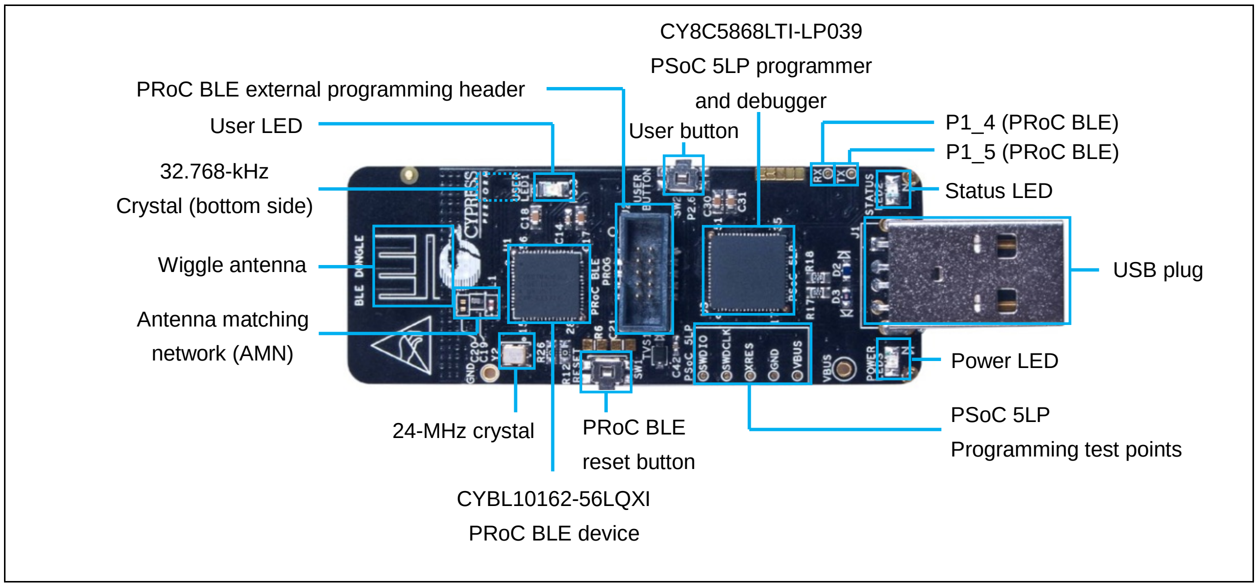
10.2 BLE-USB Bridge

10.2.1 Board Detail

The BLE-USB Bridge consists of the blocks shown in Figure 10-2. The initial firmware programmed into the BLE-USB Bridge does not support the CySmartTM Software Utility. Instead, the firmware is a custom version used for demonstrating the features of this kit.

- CYBL10162-56LQXI PRoC BLE device
- CY8C5868LTI-LP039 PSoC 5LP programmer and debugger
- Antenna matching network (AMN)
- Wiggle antenna
- 24-MHz crystal
- 32.768-kHz Crystal (bottom side)
- PRoC BLE external programming header
- PSoC 5LP Programming test points
- PRoC BLE reset button
- User button
- Power LED
- Status LED
- User LED
- Pads for P1_4 (PRoC BLE)
- Pads for P1_5 (PRoC BLE)
- USB plug

Figure 10-2 BLE-USB Bridge



10.2.2 Test Pin Description

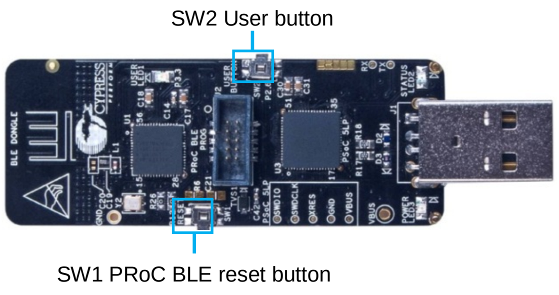
Table 10-6 Test Pin Description

Circuit Pin No.	Circuit Name	I/O	Description
TP1	BLE_TEST1	I/O	GPIO P3.0 of PRoC BLE
TP2	BLE_TEST2	I/O	GPIO P0.4 of PRoC BLE
TP3	VDDA	-	Power Pin of PRoC BLE
TP4	VDDR	-	Power Pin of PRoC BLE
TP5	VDDD	-	Power Pin of PRoC BLE
TP6	VBUS	-	Power Pin of USB Bus
TP7	GND	-	GND pin
TP8	P5LP_SWDIO	I/O	SWD data pin of PSoC 5LP
TP9	P5LP_SWDCLK	I/O	SWD clock pin of PSoC 5LP
TP10	GND	-	GND pin
TP11	P5LP_XRES	I	Reset pin of PSoC 5LP
TP12	P5LP15_4	I/O	GPIO P15_4 of PSoC 5LP
TP13	VBUS	-	Power pin of USB Bus
TP14	BLE_TX	O	UART TXD of USB serial (RXD of PSoC 5LP)
TP15	BLE_RX	I	UART RXD of USB serial (TXD of PSoC 5LP)

10.2.3 Switch Description

Table 10-7 Switch Description

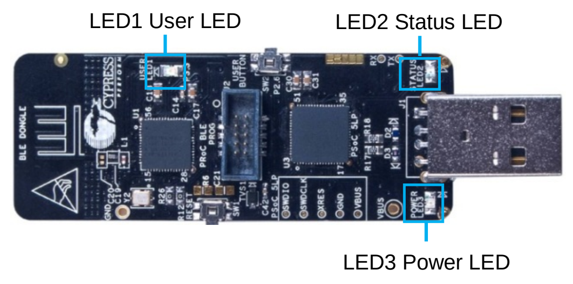
Circuit Pin No.	Silk-Printed Name	Description
SW1	RESET	Hardware Reset Button
SW2	USER BUTTON	User Button of PRoC BLE



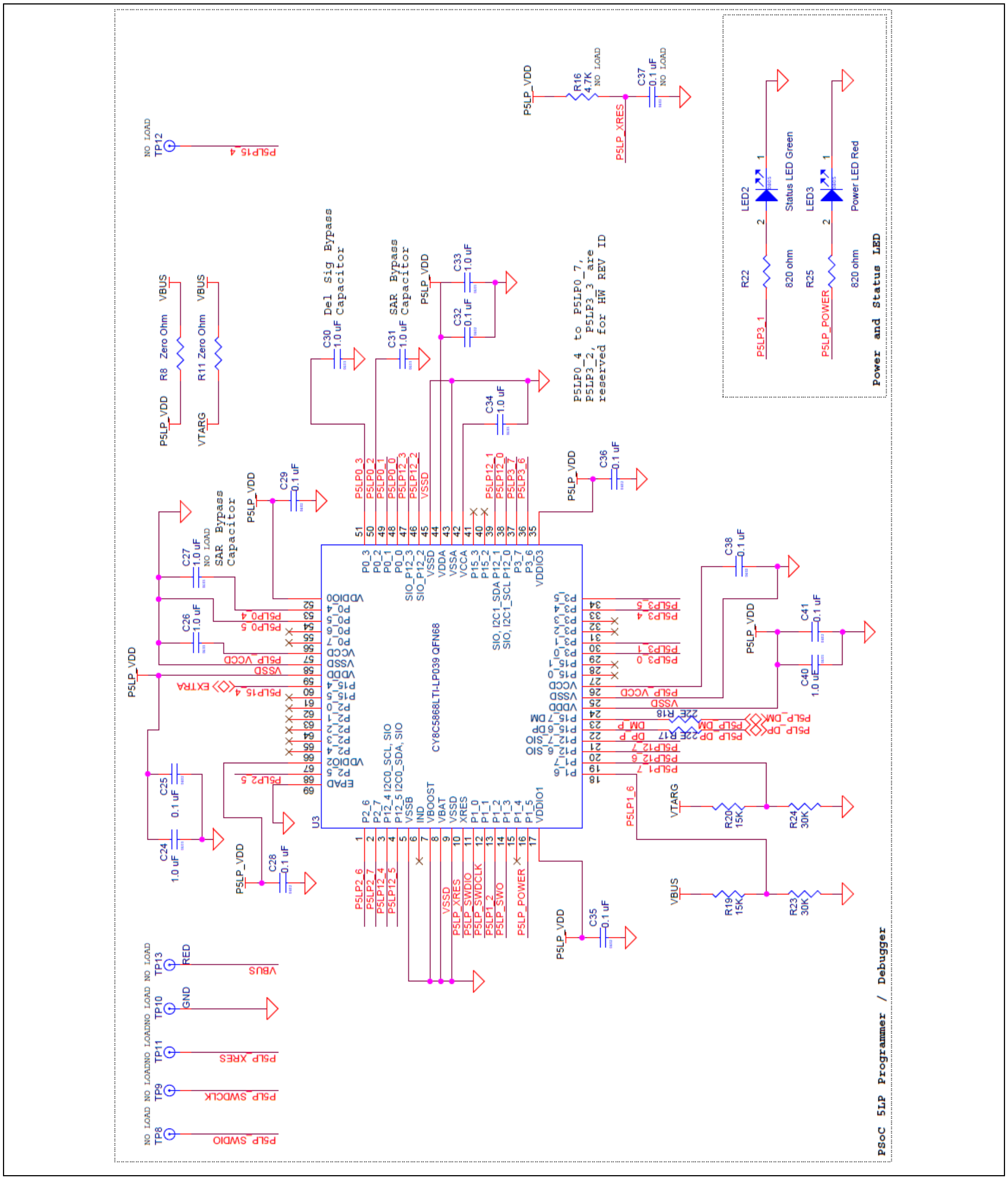
10.2.4 LED Description

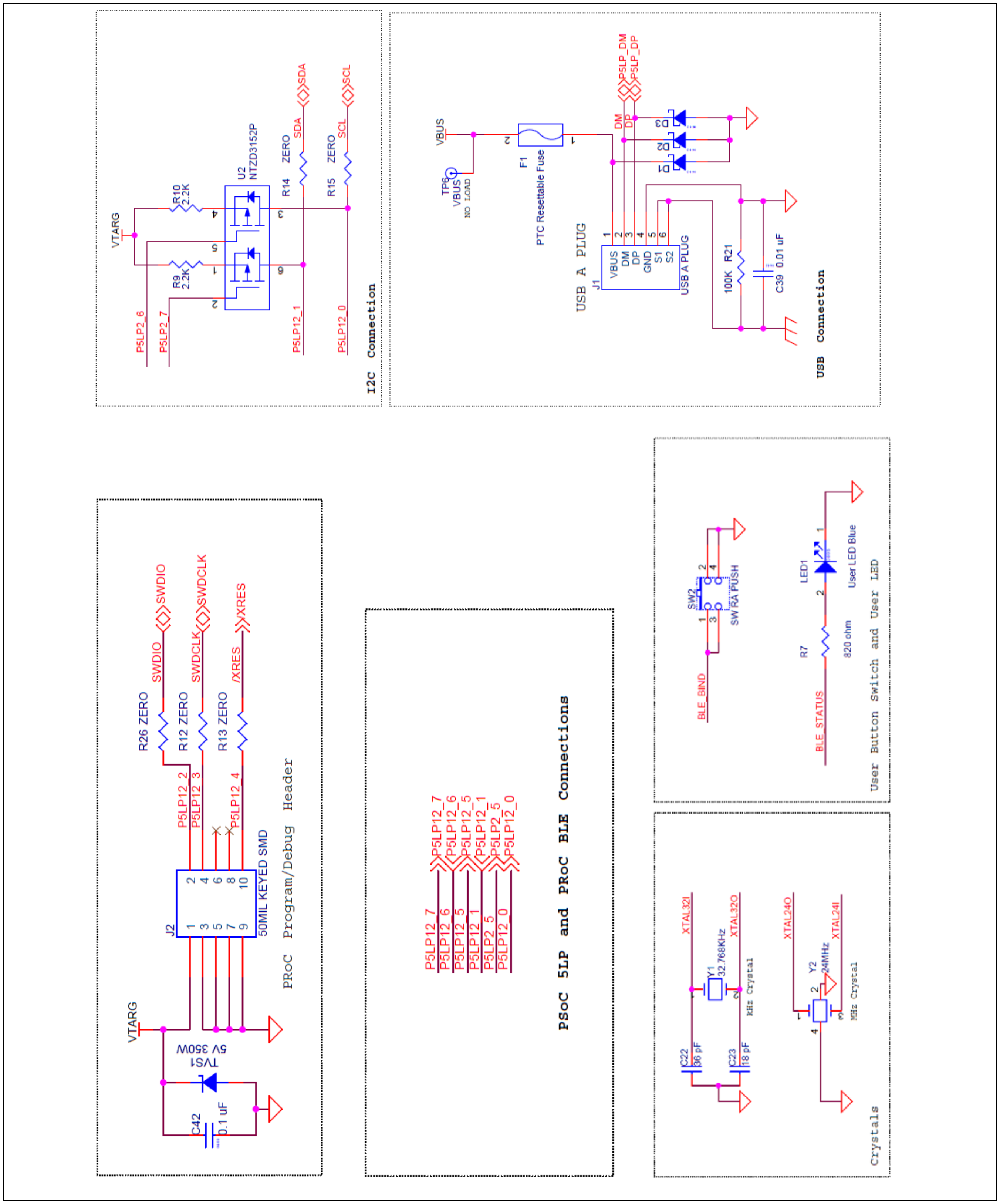
Table 10-8 LED Description

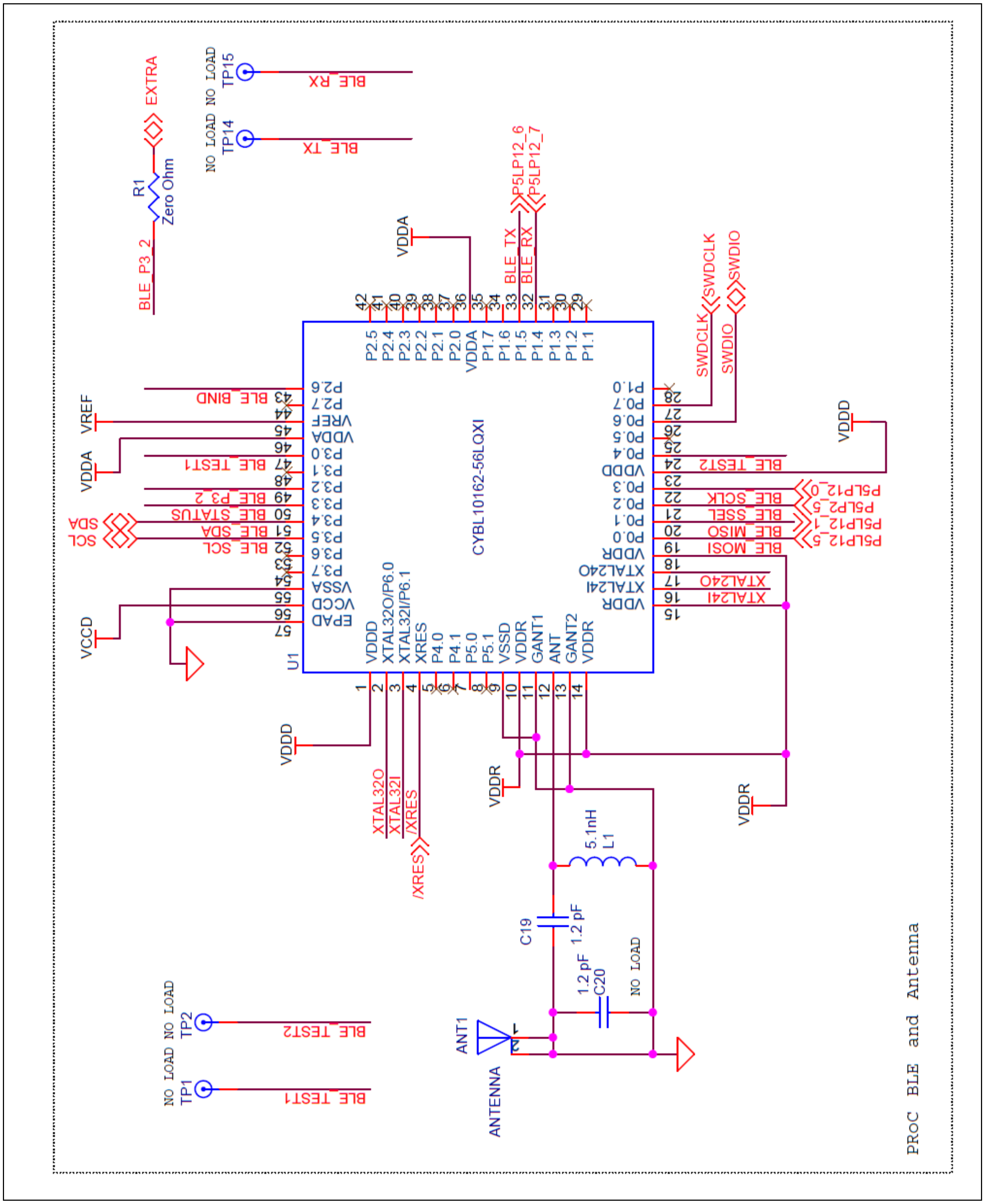
Circuit Pin No.	Silk-Printed Name	Description
LED1	USER LED1	GPIO P3.3 of PRoC BLE
LED2	STATUS LED2	GPIO P3_1 of PSoC 5LP
LED3	POWER LED3	Power LED of PSoC 5LP

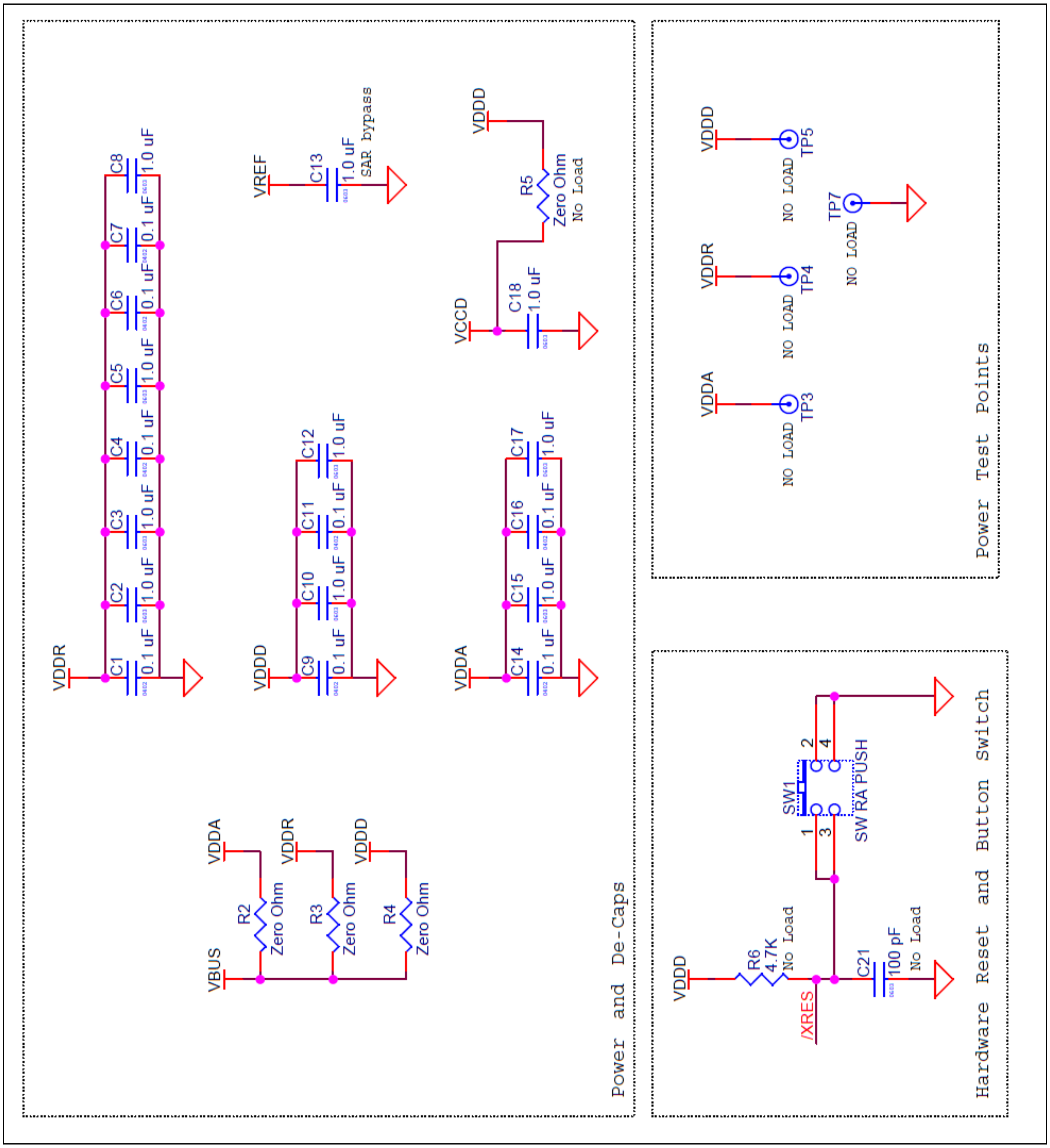


10.2.5 Circuit




I2C Connection
U2 NTZD3152P
R10 2.2K
R9 2.2K
R14 ZERO
R15 ZERO
VTARG
P5LP2_6
P5LP2_7
P5LP12_1
P5LP12_0
SDA
SCL
USB Connection
USB A PLUG
J1
VBUS
DM
DP
GND
S1
S2
F1 PTC Resettable Fuse
TP6 VBUS NO LOAD
D1
D2
D3
P5LP_DM
P5LP_DP
R21 100K
C39 0.01 uF
PROC Program/Debug Header
J2 50MIL KEYED SMD
TVS1 5V 350W
C42 0.1 uF
R26 ZERO
R12 ZERO
R13 ZERO
SWDIO
SWDCLK
/XRES
P5LP12_2
P5LP12_3
P5LP12_4
PSoC 5LP and PROC BLE Connections
P5LP12_7
P5LP12_6
P5LP12_5
P5LP12_1
P5LP2_5
P5LP12_0
User Button Switch and User LED
SW2 SW RA PUSH
BLE_BIND
BLE_STATUS
R7 820 ohm
LED1 User LED Blue
Crystals
Y1 32.768KHz KHz Crystal
Y2 24MHz MHz Crystal
C22 36 pF
C23 18 pF
XTAL32I
XTAL32O
XTAL24O
XTAL24I


PRoC BLE and Antenna
U1
CYBL10162-56LQXI
R1
Zero Ohm
BLE_P3_2
EXTRA
TP14
TP15
NO LOAD
BLE_TX
BLE_RX
VDDA
P5LP12_6
P5LP12_7
SWDCLK
SWDIO
VDDD
VREF
VCCD
SDA
SCL
BLE_BIND
BLE_TEST1
BLE_TEST2
BLE_STATUS
BLE_SDA
BLE_SCL
BLE_SCLK
BLE_SSEL
BLE_MISO
BLE_MOSI
P5LP12_0
P5LP2_5
P5LP12_1
P5LP12_5
XTAL24O
XTAL24I
XTAL32O
XTAL32I
/XRES
VDDR
5.1nH
L1
C19
1.2 pF
C20
ANT1
ANTENNA
TP1
TP2


VDDR
C1 0.1 uF
C2 1.0 uF
C3 1.0 uF
C4 0.1 uF
C5 1.0 uF
C6 0.1 uF
C7 0.1 uF
C8 1.0 uF
VDDD
C9 0.1 uF
C10 1.0 uF
C11 0.1 uF
C12 1.0 uF
VDDA
C14 0.1 uF
C15 1.0 uF
C16 0.1 uF
C17 1.0 uF
VREF
C13 1.0 uF SAR bypass
VDDD
R5 Zero Ohm No Load
VCCD
C18 1.0 uF
VBUS
VDDA
R2 Zero Ohm
VDDR
R3 Zero Ohm
VDDD
R4 Zero Ohm
Power and De-Caps
VDDA
TP3 NO LOAD
VDDR
TP4 NO LOAD
VDDD
TP5 NO LOAD
TP7 NO LOAD
Power Test Points
VDDD
R6 4.7K No Load
/XRES
C21 100 pF No Load
SW1 SW RA PUSH
1
2
3
4
Hardware Reset and Button Switch

10.2.6 BOM List

No	Qty	Reference	Parts Number	Description	Manufacture	Note
1	17	C1,C4,C6,C7,C9,C11, C14,C16,C25,C28,C29, C32,C35,C36,C38, C41,C42	C1005X5R1A104K050BA	0.1 µF/0402	TDK Corporation	
2	17	C2,C3,C5,C8,C10,C12, C13,C15,C17,C18,C24, C26,C30,C31,C33,C34, C40	TMK107BJ105KA-T	1 µF/0603	Taiyo Yuden	
3	1	C19	500R07S1R2BV4T	1.2 pF/0402	Johanson Technology Inc	
4	1	C22	GRM1555C1H360JA01D	36 pF/0402	Murata Manufacturing Co., Ltd.	
5	1	C23	GRM1555C1H180FA01D	18 pF/0402	Murata Manufacturing Co., Ltd.	
6	1	C39	C1005X7R1C103K050BA	0.01 µF/0402	TDK Corporation	
7	3	D1,D2,D3	CG0603MLC-05LE	ESD diode	Bourns, Inc.	
8	1	F1	MF-MSMF050-2	FUSE	Bourns, Inc.	
9	1	J1	480370001	USB Type-A PLUG	Molex Inc	
10	1	J2	20021521-00010T1LF	50MIL KEYED SMD	FCI	
11	1	LED1	LTST-C171TBKT	Status LED Blue	LiteOn Inc	
12	1	LED2	CMD17-21VGC/TR8	Status LED Green	Chicago Miniature	
13	1	LED3	LTST-C170KRKT	Power LED Red	LiteOn Inc	
14	1	L1	L-07C5N1SV6T	5.1 nH/0402	Johanson Technology Inc	
15	2	R8,R11	ERJ-6GEY0R00V	0Ω/0805	Panasonic Corporation	
16	1	R7	ERJ-3GEYJ821V	820Ω/0603	Panasonic Corporation	
17	2	R22,R25	ERJ-6GEYJ821V	820Ω/0805	Panasonic Corporation	
18	2	R9,R10	ERJ-3GEYJ222V	2.2 kΩ/0603	Panasonic Corporation	
19	9	R1,R2,R3,R4,R12,R13, R14,R15,R26	ERJ-3GEY0R00V	0Ω/0603	Panasonic Corporation	
20	2	R17,R18	ERJ-3EKF22R0V	22Ω/0603	Panasonic Corporation	
21	1	R21	ERJ-2GEJ104X	100 kΩ/0402	Panasonic Corporation	
22	2	R19,R20	ERJ-3GEYJ153V	15 kΩ/0603	Panasonic Corporation	
23	2	R23,R24	ERJ-3GEYJ303V	30 kΩ/0603	Panasonic Corporation	
24	2	SW1,SW2	EVQ-P3401P	Push Switch	Panasonic Corporation	
25	1	TVS1	SD05-7	5V 350W	Diodes Inc.	
26	1	U1	CYBL10162-56LQXI	PRoC BLE, Programmable Radio on Chip	Cypress Semiconductor Corporation	
27	1	U2	NTZD3152PT1G	DUAL PMOS	ON Semiconductor	

No	Qty	Reference	Parts Number	Description	Manufacture	Note
28	1	U3	CY8C5868LTI-LP039	PSoC 5LP Programmable System on Chip	Cypress Semiconductor Corporation	
29	1	Y1	ECS-.327-12.5-34B	32.768 kHz	ECS Inc	
30	1	Y2	ECS-240-8-36CKM	24 MHz	ECS Inc	
31	0	(C20)	500R07S1R2BV4T	1.2 pF/0402	Johanson Technology Inc	Non mount
32	0	(C21)	C0603C101K5RACTU	100 pF/0603	Kemet Corporation	Non mount
33	0	(C37)	C1005X5R1A104K050BA	0.1 μF/0402	TDK Corporation	Non mount
34	0	(C27)	TMK107BJ105KA-T	1 μF/0603	TAIYO YUDEN CO., LTD.	Non mount
35	0	(R5)	1623094-1	0Ω/0603	TE Connectivity Ltd.	Non mount
36	0	(R6),(R16)	ERJ-3GEYJ472V	4.7 kΩ/0603	Panasonic Corporation	Non mount
37	15	TP1,TP2,TP3,TP4,TP5, TP6,TP7,TP8,TP9,TP10, TP11,TP12,TP13,TP14, TP15	-	-	-	

11. Appendix

11.1 Other Sample Project

11.1.1 LED ONOFF Project

This project demonstrates the simple LED control. (<Install directory>/Solar-Powered IoT Device Kit/1.0/Firmware//Other/LED_ONOFF/LED_ONOFF.cydsn/ LED_ONOFF.cyprj)

This sample firmware only supports LED blinking control using USB bus power, therefore please reprogram the "EH_Motherboard" project in the Firmware folder to restore the Solar-Powered BLE Beacon operation.

This following introduces you to the LED_ONOFF project:

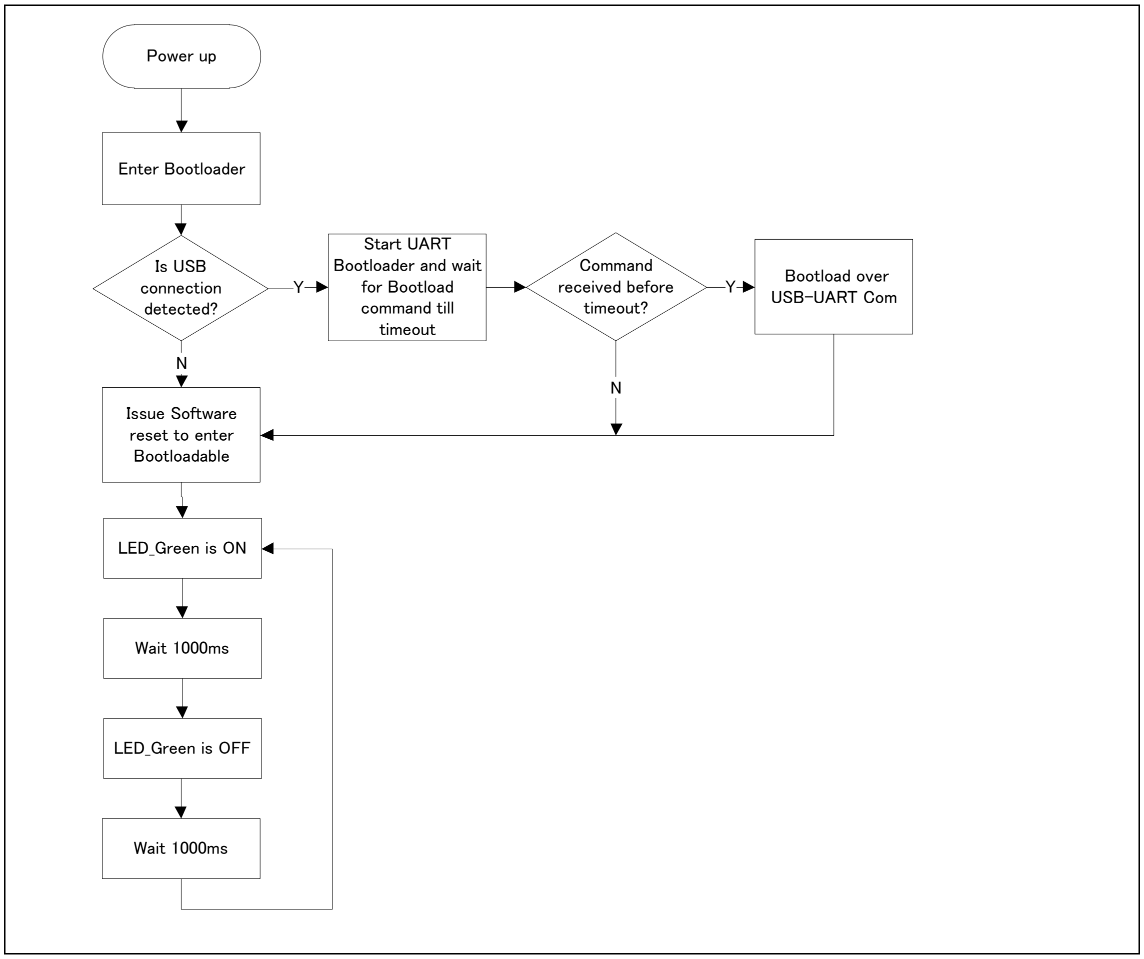
- main.c

This file contains the main function, which is only control the LED turning on and off. The P3.4 of EZ-BLE module is control a status LED on Energy Harvesting Motherboard. The blinking interval is set 1000 ms.

```
#include <project.h>

#define ON             (1u)        /* P3.4 is high      */
#define OFF            (0u)        /* P3.4 is low       */
#define BLINK_INTERVAL (1000)      /* 1000 ms           */

int main()
{
    while(1)
    {
        LED_Green_Write(ON);
        CyDelay(BLINK_INTERVAL);
        LED_Green_Write(OFF);
        CyDelay(BLINK_INTERVAL);
    }
    return 0;
}
```

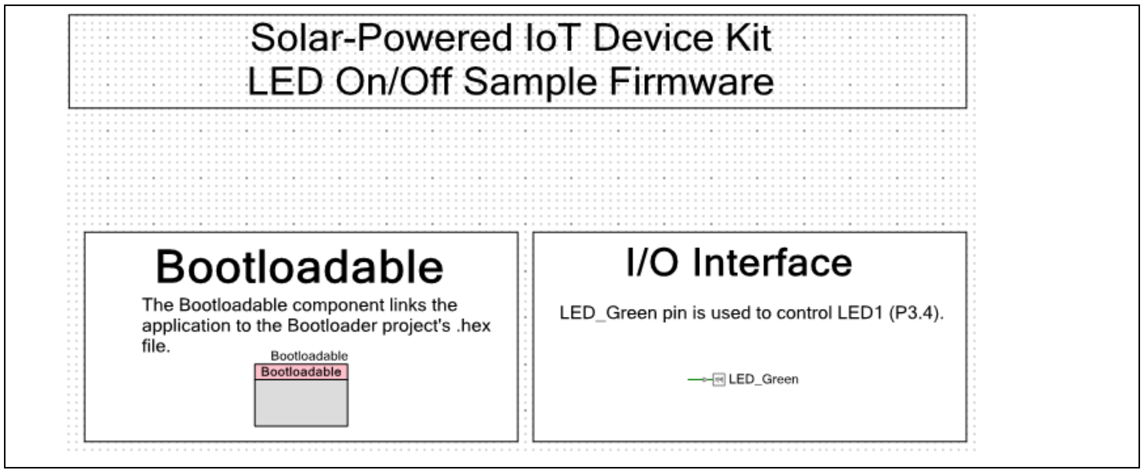
■ Flow Diagram

Following is the flow diagram for the LED ONOFF project.


Power up
Enter Bootloader
Is USB connection detected?
Y
N
Start UART Bootloader and wait for Bootload command till timeout
Command received before timeout?
Y
N
Bootload over USB-UART Com
Issue Software reset to enter Bootloadable
LED_Green is ON
Wait 1000ms
LED_Green is OFF
Wait 1000ms

■ TopDesign.cysch

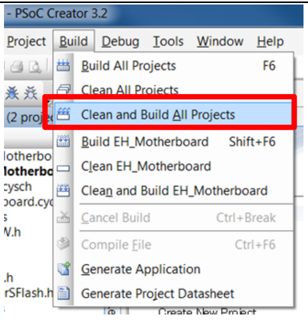
Following is the top design for LED_ONOFF project. The Bootloadable component links the application to the Bootloader project's hex file. A LED_Green of I/O Interface is used to control the status LED1 (P3.4).


Solar-Powered IoT Device Kit
LED On/Off Sample Firmware
Bootloadable
The Bootloadable component links the application to the Bootloader project's .hex file.
Bootloadable
Bootloadable
I/O Interface
LED_Green pin is used to control LED1 (P3.4).
LED_Green

- Demonstration steps

1. Open the sample project (LED_ONOFF.cypj) for this kit.
(<Install directory>/Solar-Powered IoT Device Kit/1.0/Firmware//Other/LED_ONOFF/LED_ONOFF.cydsn/ LED_ONOFF.cyprj)

2. Select the "Menu>Build>Clean and Build All Projects", and then the build status will appear on lower right side of PSoC Creator. The build process is finished when the "Rebuild succeeded" appears on massage window.

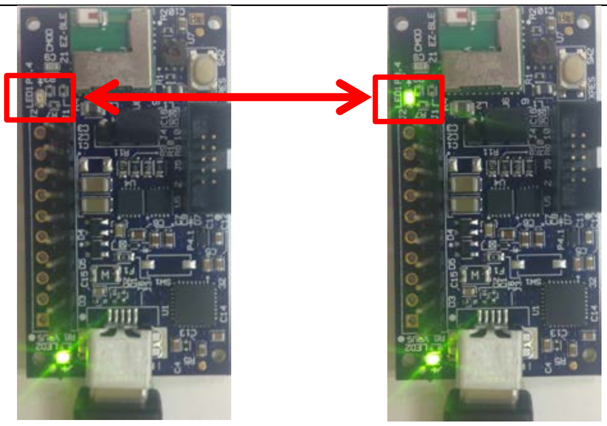


3. Program the LED_ONOFF project.

a. When you use the UART Bootloader, refer to "7.1 UART Bootloader (Program Only)".
The .cyacd file is generated in the project folder of PSoC Creater as shown below.
(<Install directory>/Solar-Powered IoT Device Kit/1.0/Firmware//Other/LED_ONOFF/LED_ONOFF.cydsn/CortexM0/ARM _GCC_484/Debug/ LED_ONOFF.cyacd)

b. When you use the MiniProg3, Refer to "7.2 PSoC Creator with MiniProg3 (Program and Debug)" of step 8 to step 12.

4. The status LED on Energy Harvesting Motherboard will continue the turning on and off for a second using USB bus power.



11.1.2 Simple BLE Project

This project demonstrates the simple BLE Beacon transmitter. (<Install directory>/Solar-Powered IoT Device Kit/1.0/Firmware//Other/Simple_BLE/Simple_BLE.cydsn/ Simple_BLE.cyprj)

This sample firmware only supports the BLE Beacon and Bootloader with no other frills (no I^{2}C, no flash read/write, no UART serial configuration). This project have just one source file (main.c).

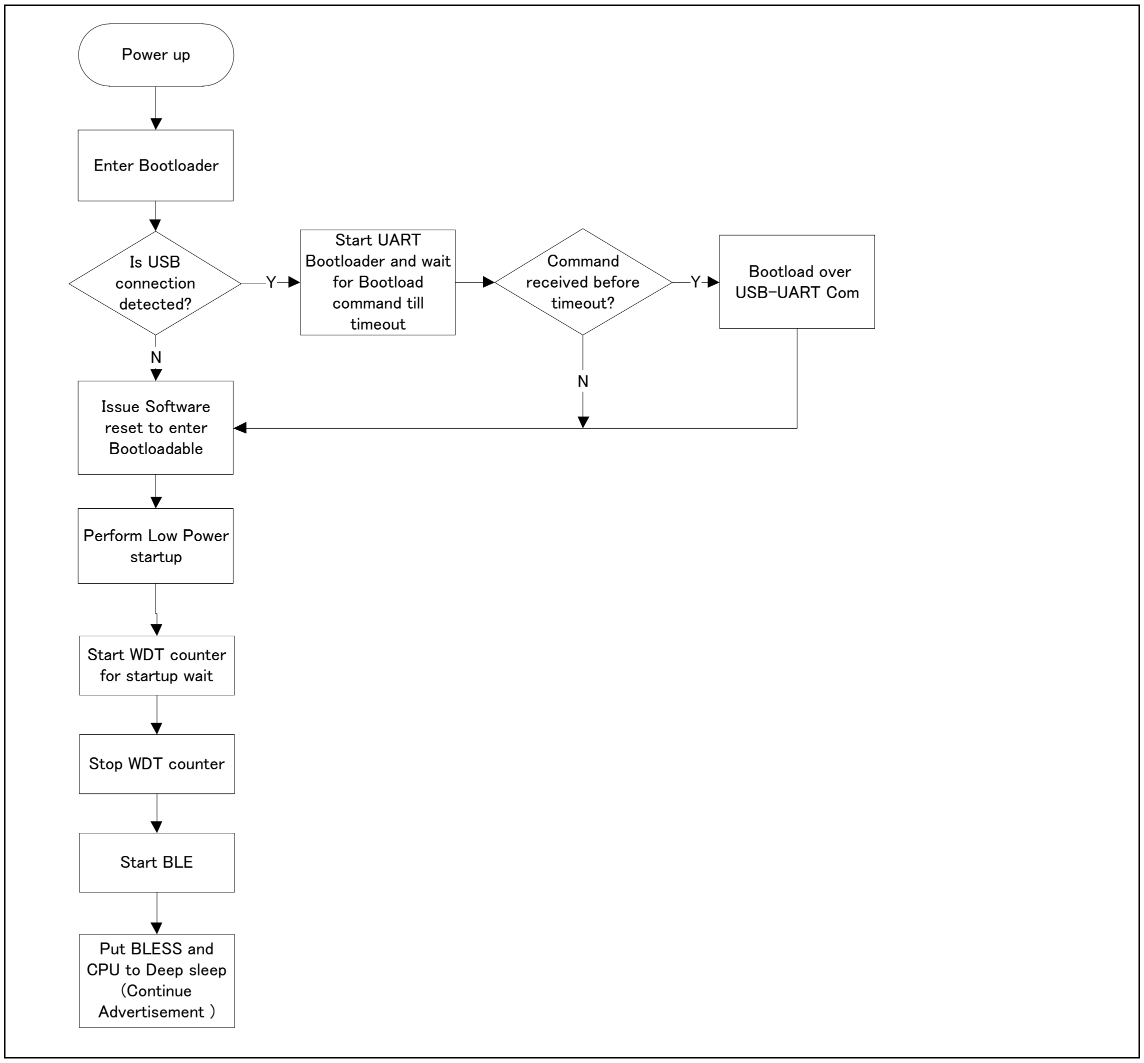
This following introduces you to the Simple_BLE project:

- main.c
  This file contains the main function, which is only transmitting the BLE Beacon as shown below. This firmware flow is same as "8.3 BLE Beacon Process" except the SFLASH parameter reading. All parameter such as UUID, MAJOR, MINOR is fixed by the BLE Component on TopDesign.cysch.

```
int main()
{
    CyGlobalIntEnable;
    /* Set the divider for ECO, ECO will be used as source when IMO is
     * switched off to save power, to drive the HFCLK */
    CySysClkWriteEcoDiv(CY_SYS_CLK_ECO_DIV8);

    /* Start WCO & ECO in low power mode */
    LowPower_WCO_ECO_Start();
    CyBle_Start(BLE_AppEventHandler);
    CyBle_ProcessEvents();

    for(;;)
    {
        CYBLE_LP_MODE_T pwrState;
        CYBLE_BLESS_STATE_T blessState;
        uint8 intStatus = 0;
        /* BLE stack processing state machine interface */
        CyBle_ProcessEvents();
        /* Configure BLESS in Deep-Sleep mode */
        pwrState  = CyBle_EnterLPM(CYBLE_BLESS_DEEPSLEEP);
        /* No interrupts allowed while entering system low power modes
         */
        intStatus = CyEnterCriticalSection();
        blessState = CyBle_GetBleSsState();
        /* Make sure BLESS is in Deep-Sleep before configuring system in
         * Deep-Sleep mode */
        if(pwrState == CYBLE_BLESS_DEEPSLEEP)
        {
            /* If BLESS is in Deep Sleep or is in the process of waking
             * up from Deep Sleep, put system in Deep Sleep mode */
            if(blessState == CYBLE_BLESS_STATE_ECO_ON || blessState ==
            CYBLE_BLESS_STATE_DEEPSLEEP)
            {
                CySysPmDeepSleep(); /* System Deep-Sleep. 1.3uA mode */
            }
        }
        /* If BLESS is in Active state,
         * and if BLESS Tx/Rx Event is not complete, stop IMO and put CPU
         * to Sleep */
        else if (blessState != CYBLE_BLESS_STATE_EVENT_CLOSE)
        {
            /* Change HF clock source from IMO to ECO, as IMO can be stopped
             * to save power */
            CySysClkWriteHfclkDirect(CY_SYS_CLK_HFCLK_ECO);
            /* Stop IMO for reducing power consumption */
            CySysClkImoStop();
            /* Put the CPU to Sleep. 1.1mA mode */
            CySysPmSleep();
            /* Starts execution after waking up, start IMO */
            CySysClkImoStart();
            /* Change HF clock source back to IMO */
            CySysClkWriteHfclkDirect(CY_SYS_CLK_HFCLK_IMO);
        }
        CyExitCriticalSection(intStatus);
        /* If restartadvertisement flag is set, which means it's the first
         * time cpu goes here. */
        if(restartadvertisement)
        {
            restartadvertisement = false;
            CySysWdtUnlock();
            CySysWdtDisable(CY_SYS_WDT_COUNTER0_MASK);
            CySysWdtWriteMode(SOURCE_COUNTER, CY_SYS_WDT_MODE_INT);
            CySysWdtWriteClearOnMatch(SOURCE_COUNTER, COUNTER_ENABLE);
            CySysWdtWriteMatch(SOURCE_COUNTER, COUNT_PERIOD_1S);
            CySysWdtEnable(CY_SYS_WDT_COUNTER0_MASK);
            CySysWdtLock();
            wdt_trigger_on_flag = true;
        }
        ProcessBeaconEvents();
    }
}
```

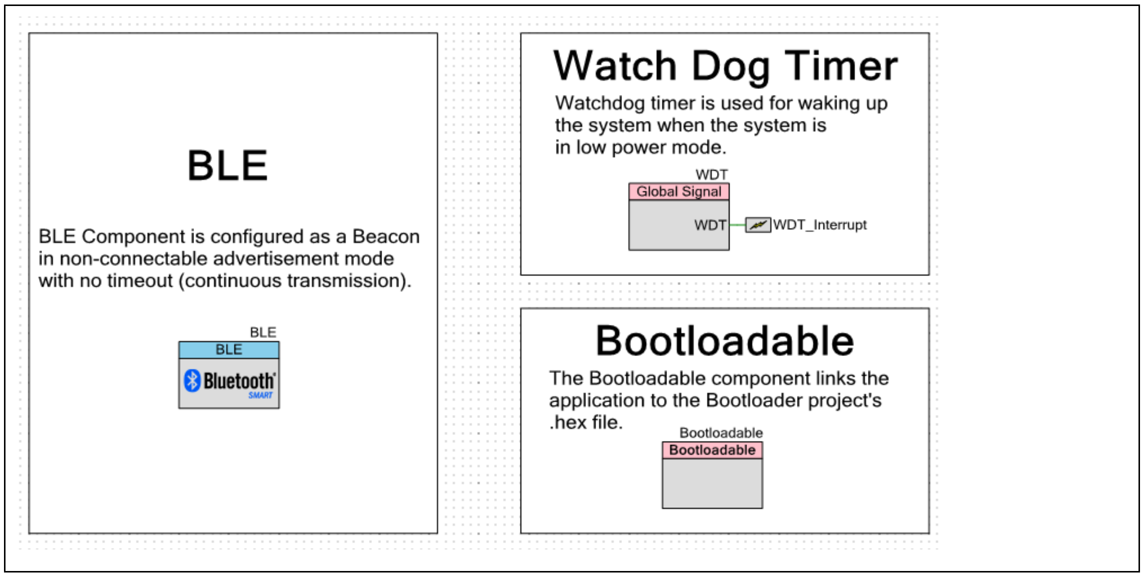
■ Flow Diagram

Following is the flow diagram for the Simple BLE project.


Power up
Enter Bootloader
Is USB connection detected?
Y
Start UART Bootloader and wait for Bootload command till timeout
Command received before timeout?
Y
Bootload over USB-UART Com
N
N
Issue Software reset to enter Bootloadable
Perform Low Power startup
Start WDT counter for startup wait
Stop WDT counter
Start BLE
Put BLESS and CPU to Deep sleep (Continue Advertisement)

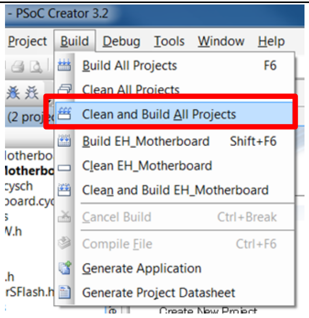
- TopDesign.cysch

Following is the top design for Simple_BLE project. The BLE Component is cofigured as a Beacon with fixed parameter. The Watch Dog Timer is used for waking up the system when the system is in low power mode. The Bootloadable component links the application to the Bootloader project's hex file.


BLE
BLE Component is configured as a Beacon in non-connectable advertisement mode with no timeout (continuous transmission).
BLE
BLE
Bluetooth SMART
Watch Dog Timer
Watchdog timer is used for waking up the system when the system is in low power mode.
WDT
Global Signal
WDT
WDT_Interrupt
Bootloadable
The Bootloadable component links the application to the Bootloader project's .hex file.
Bootloadable
Bootloadable

- Demonstration steps
    1. Open the sample project (Simple_BLE.cyprj) for this kit. (<Install directory>/Solar-Powered IoT Device Kit/1.0/Firmware//Other/Simple_BLE/Simple_BLE.cydsn/Simple_BLE.cyprj)

    2. Select the "Menu>Build>Clean and Build All Projects", and then the build status will appear on lower right side of PSoC Creator. The build process is finished when the "Rebuild succeeded" appears on massage window.

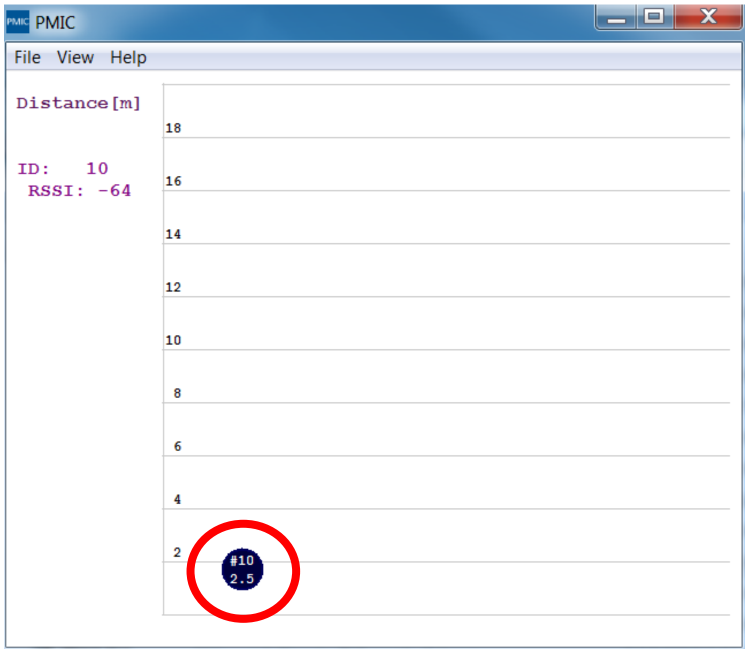


3. Program the Simple_BLE project.

   a. When you use the UART Bootloader, refer to "7.1 UART Bootloader (Program Only)".
   The .cyacd file is generated in the project folder of PSoC Creater as shown below.
   (<Install directory>/Solar-Powered IoT Device Kit/1.0/Firmware//Other/Simple_BLE/Simple_BLE.cydsn/CortexM0/ARM_GCC_484/Debug/ Simple_BLE.cyacd)

   b. When you use the MiniProg3, Refer to "7.2 PSoC Creator with MiniProg3 (Program and Debug)"of step 6 to step 12.

4. Run PMIC.exe, and then find ten MAJOR number (fixed value by firmware) of the Motherboard on the PMIC Software (Refer to "6.1.4 Establishing BLE Connection" for detail). Then move the Motherboard further away from your computer. The Received Signal Strength Indicator (RSSI) value will change and the graphic will be updated.

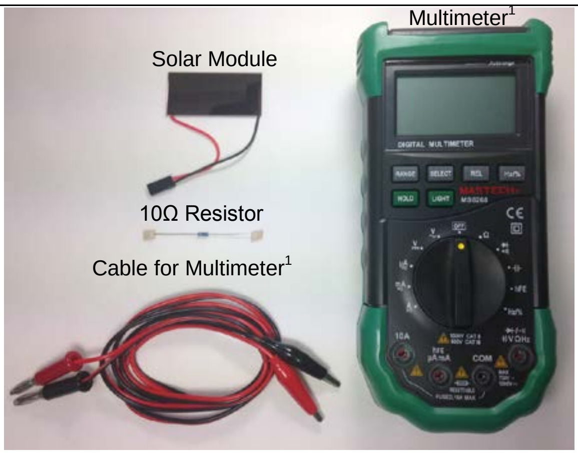


ID: 10
RSSI: -64

11.2 How to Use Extra Components

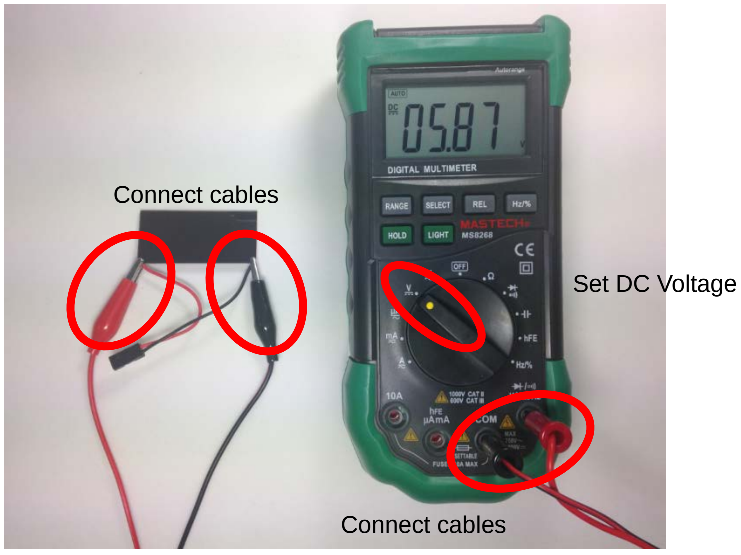
11.2.1 10Ω Resistor for Current Measurement

This chapter uses the Solar Module, 10Ω Resistor and a Multimeter (MASTECH: MS8268, Not included in this kit.) to measure the voltage and current for the solar module using 10Ω Resistor that is included in the Solar-Powered IoT Device Kit. First, you will take measurements under constant ambient light. Then, you will vary the available light and observe how the energy output of the solar module changes.



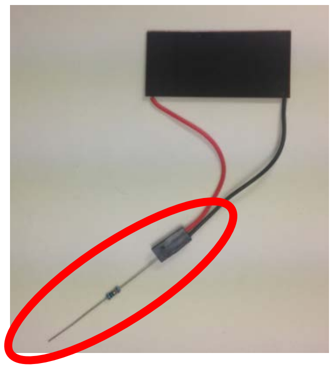
[1] Not included in the kit.

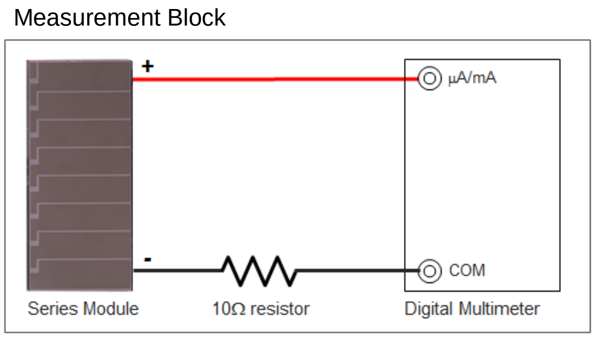
1. Set the Multimeter to measure DC Voltage. Connect the Multimeter to the Solar Module as shown in picture, and observe the DC voltage (V1).



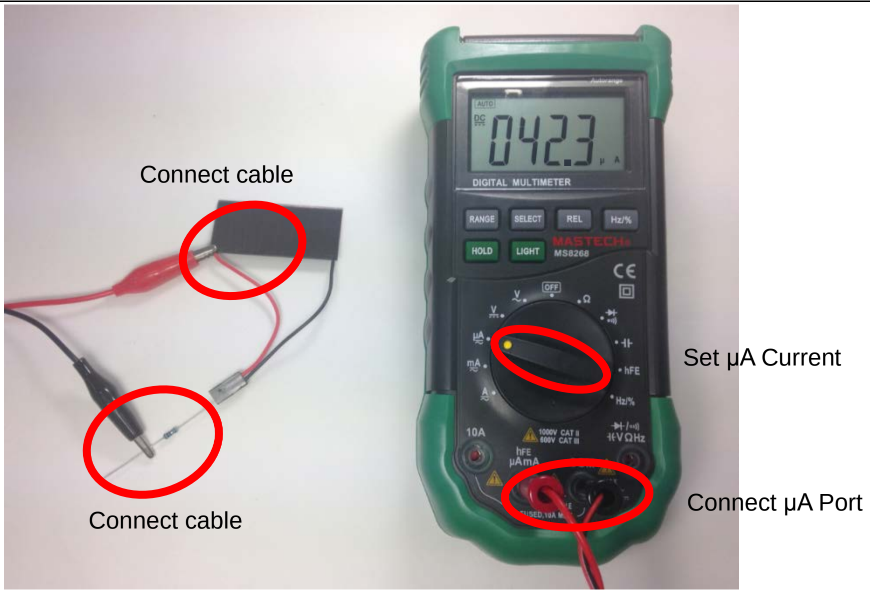
2. Put your hand about 2 cm above the Solar Module to block most of the light and observe the DC voltage (V2).

3. Now, you will measure the current produced by the Solar Module. Connect a 10Ω resistor to the black wire on the Solar Module as shown in the picture.

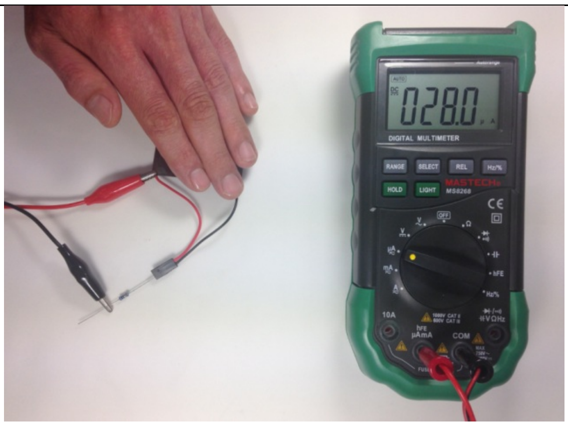




4. Set the Multimeter to measure Current in the microamp range. Connect the positive cable of the Multimeter to the positive lead of the Solar Module, and the negative cable to the 10Ω resistor. Observe the current (I1).



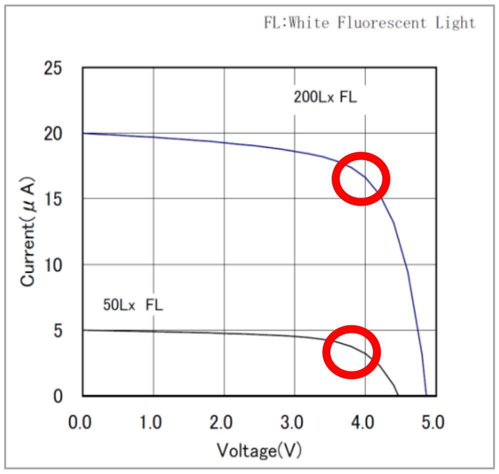
5. Put your hand about 2 cm above the Solar Module to block most of the light and observe the DC voltage (I2).



6. Calculate the energy produced by the Solar Module. Calculate the Maximum Power Point (MPP[1]) energy of the Solar Module as shown below.

- Under constant ambient light (Example 400 Lux)
  E1 = (0.8 x V1) x (0.8 x I1) = (0.8 x 5.87V) x (0.8 x 42.3uA) = 158.91 [uW]
- Covering Solar Module with hand (Example 250 Lux)
  E2 = (0.8 x V2) x (0.8 x I2) = (0.8 x 5.75V) x (0.8 x 28.0uA) = 103.04 [uW]

I-V Characteristics of Solar Module (AM-1801)

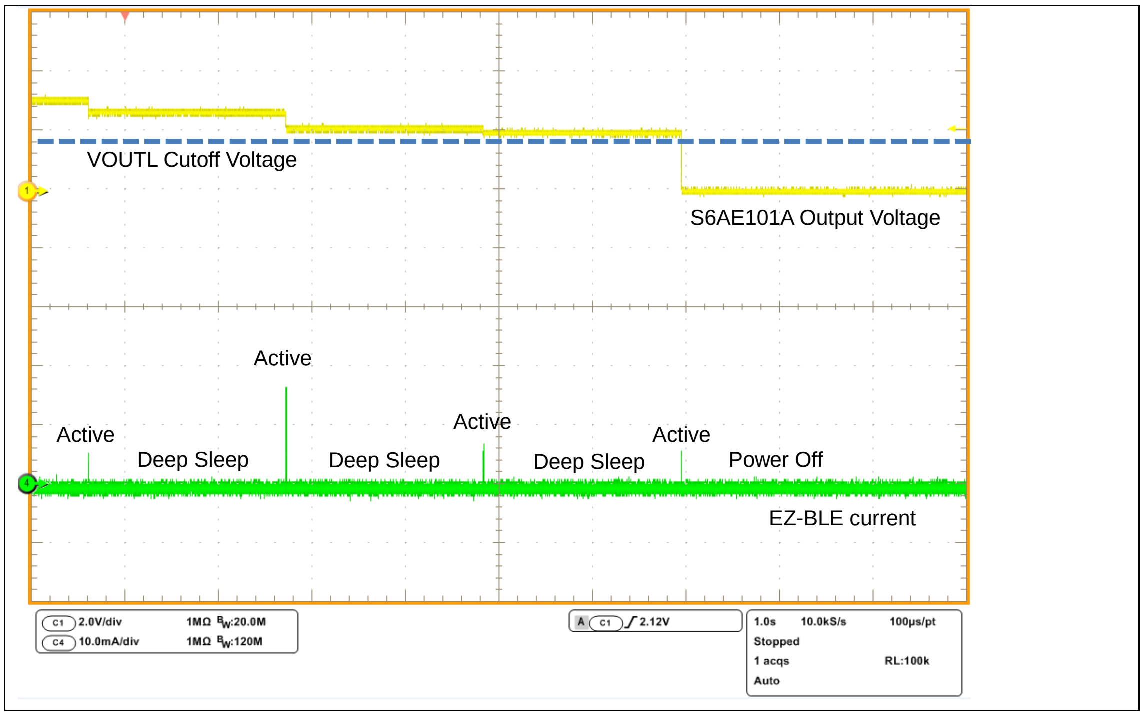


[1] The MPP of a typical solar cell is 80% of the maximum voltage and maximum current.

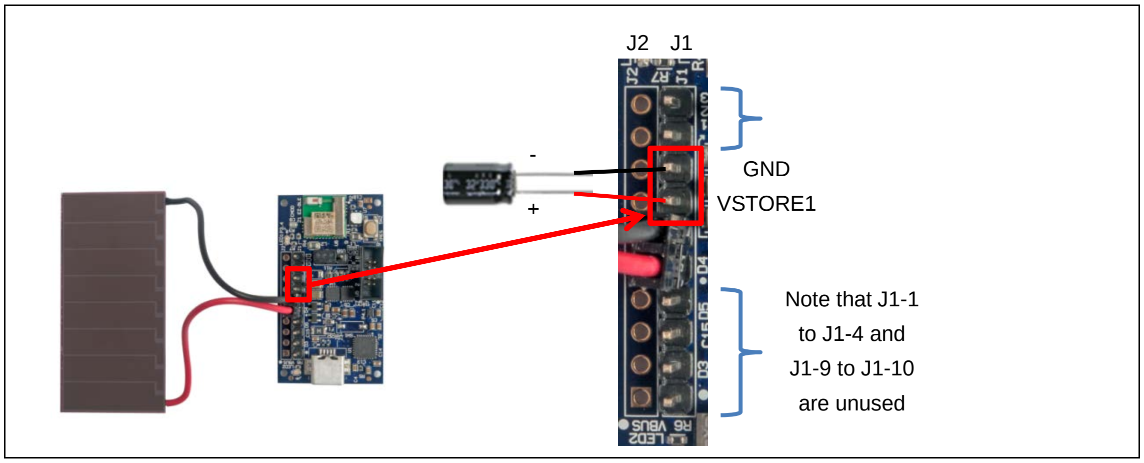
11.2.2 220 µF capacitor for additional

An Energy Harvesting System (EHS) is made of an Energy Harvesting Device (EHD) such as a solar module, an Energy Harvesting PMIC, and a Storage Device (such as a capacitor). Its purpose is to supply power to the EHS Load. When designing an EHS, you have to consider voltage, current, and time in 3 contexts: Power supplied by the EHD, power stored in the Storage Device, and power required to operate the system.

Following is example of failed waveform for EHS. The output voltage of the S6AE101A slowly drops due to consumption, eventually shutting down the output completely.



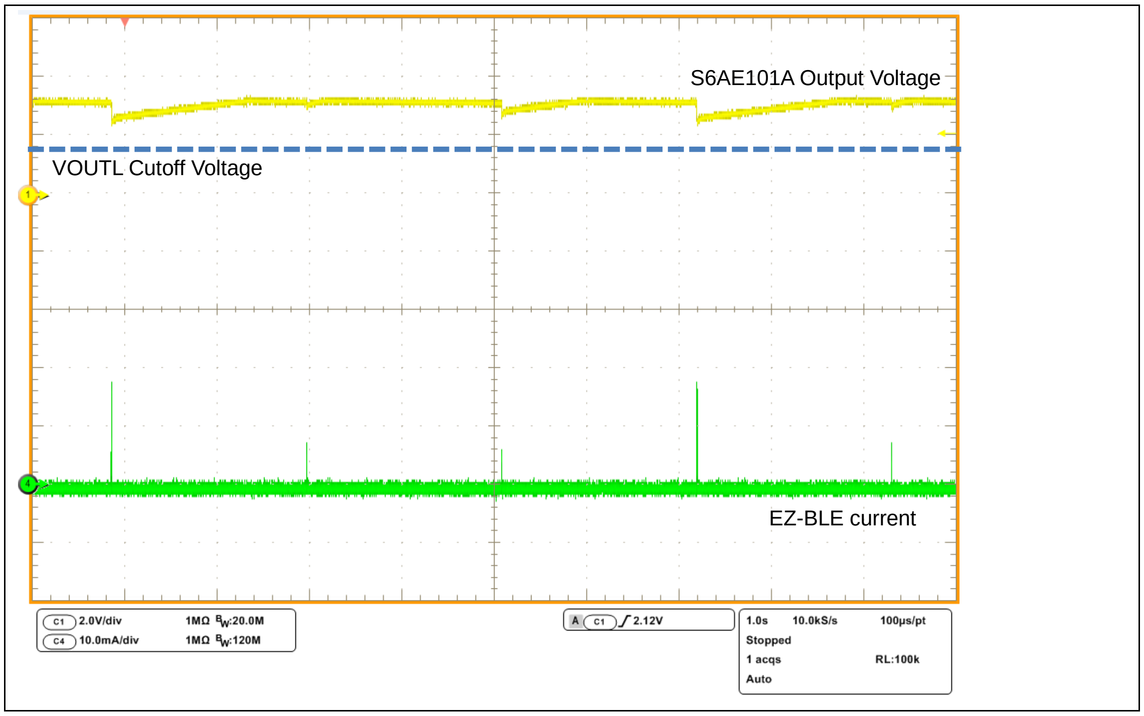
One way to resolve this fail, connect the extra capacitor to VSTORE1 of S6AE101A by using soldering or two jumper wires that is included in the kit.



The waveform for EHS is completed to add the extra capacitor to VSTORE1 pin. Note that value of capacitor depends on the consumption energy. You can calculate the energy of capacitor using following formula.

- Calculation formula for energy of capacitor

  $ECAP = 0.5 \times C \times (VOUTH^2 - VOUTL^2)$



12. Ordering Information

Table 12-1 Ordering Information

Part Number	Version	Note
S6SAE101A00SA1002	Rev 1.0	

Revision History

Document Revision History

Document Title: S6SAE101A00SA1002 - Solar-Powered Internet of Things (IoT) Device Kit Guide **Document Number: 002-00297**			
Revision	**Issue Date**	**Origin of Change**	**Description of Change**
**	08/25/2015	EIFU	New Kit Guide

Стандарт
Электрон
Связь

Мы молодая и активно развивающаяся компания в области поставок электронных компонентов. Мы поставляем электронные компоненты отечественного и импортного производства напрямую от производителей и с крупнейших складов мира.

Благодаря сотрудничеству с мировыми поставщиками мы осуществляем комплексные и плановые поставки широчайшего спектра электронных компонентов.

Собственная эффективная логистика и склад в обеспечивает надежную поставку продукции в точно указанные сроки по всей России.

Мы осуществляем техническую поддержку нашим клиентам и предпродажную проверку качества продукции. На все поставляемые продукты мы предоставляем гарантию .

Осуществляем поставки продукции под контролем ВП МО РФ на предприятия военно-промышленного комплекса России , а также работаем в рамках 275 ФЗ с открытием отдельных счетов в уполномоченном банке. Система менеджмента качества компании соответствует требованиям ГОСТ ISO 9001.

Минимальные сроки поставки, гибкие цены, неограниченный ассортимент и индивидуальный подход к клиентам являются основой для выстраивания долгосрочного и эффективного сотрудничества с предприятиями радиоэлектронной промышленности, предприятиями ВПК и научно-исследовательскими институтами России.

С нами вы становитесь еще успешнее!

Наши контакты:

Телефон: +7 812 627 14 35

Электронная почта: sales@st-electron.ru

Адрес: 198099, Санкт-Петербург, Промышленная ул, дом № 19, литера Н, помещение 100-Н Офис 331