**Energy Harvesting Reference Design – Firmware QuickStart Guide**

The firmware for the Zigbee Green Power Energy Harvesting reference design was based off of the example, [Getting Started with Silicon Labs Green Power Device using EFR32MG22](https://www.silabs.com/community/wireless/zigbee-and-thread/knowledge-base.entry.html/2020/03/13/green_power_devicefromscratchusingefr32mg22-s6dh). You will use this example to create Green Power Devices from scratch.

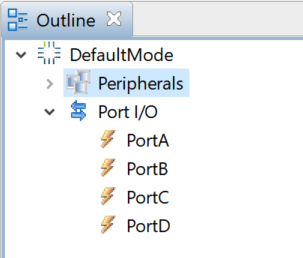
You can find the necessary files for this implementation at the [MG22-EnergyHarvesting GitHub](https://github.com/ArrowElectronicsESC/MG22-EnergyHarvesting) repo. Additionally you can download the sls project file to import into Simplicity Studio.

**Note**: You will need an EFR32xG22 radio board acting as the Green Power Device (GPD) and an EFR32xG21 or similar variant to act as the Green Power Combo (GPC). This example was written for and tested on a **BRD4182A** using **EmberZNet 6.7.5.0** in Simplicity Studio v4.

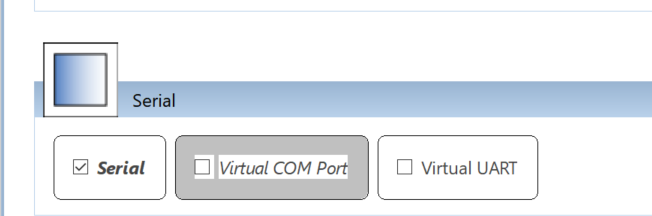
**Getting Started**

If you are starting from scratch, begin by going to the “**Getting Started with.. EFR32MG22**” mentioned above. Start by following Part 1, steps 1-13 in the **Usage** section. Make sure you validate that the project successfully generated and Build before moving forward.

1. Open the **.hwconf** file. You will need to disable VCOM.
   1. On the outline tab on the right panel of Studio, double click on **Peripherals**.

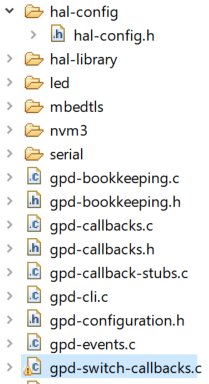


* 1. Uncheck the box for **Virtual COM Port**.



* 1. **Save** the file. This will re-generate the project automatically.
  2. **Build** the project again to ensure the changes took effect.

1. Next you will want to replace the following files with those found on the **MG22-EnergyHarvesting** GitHub repo.
   1. gpd-switch-callbacks.c, gpd-configuration.h, hal-config.h
   2. See the file structure below:



* 1. **Build** the project to ensure the changes have taken effect.
  2. **Flash** the **.s37** binary file to you EFR32xG22 BRD4182A radio board.
     1. You can flash the board by removing the board from the reference design PCB and inserting it into a WSTK board.
     2. Insert the radio board back into the Energy Harvesting Reference Design PCB.

1. Follow Part 2 in the **Usage** section of the “**Getting Started with.. EFR32MG22**” to create a Z3LightGPCComboSoc project on a WSTK board with a capable Mighty Gecko radio board.
   1. The guide calls for an EFR32MG12 or other board. You can use an EFR32xG21 as an option.
2. In the **Commissioning and Testing** section of the guide, you see the commands to issue to the GPC to initialize the commissioning process.
   1. Enter the following commands to the GPC:
      1. plugin network-creator start 0
      2. plugin green-power-server commission 9 0 0 1
3. Press the **small secondary button** on the Energy Harvesting Reference Design **once** to begin the single event commissioning process.
4. On the terminal log of the **GPC** you should see the commissioning commands joining the GPD to the GPC.
5. Pressing the **large primary button** of the reference design will toggle the LED on the **GPC** on/off.
6. Pressing the **small secondary button** of the reference design will issue the Green Power Step-Down command to the GPC.

The next section will outline the firmware created for the Energy Harvesting Reference Design and highlight functions and sections that you can change for your own implementations.

**Understanding the Firmware**

There are 3 main files to familiarize yourself with in the firmware; gpd-configuration.h, hal-config.h, gpd-switch-callbacks.c. If you make any changes to each file, you must **Build** the project again.

**gpd-configuration.h:**

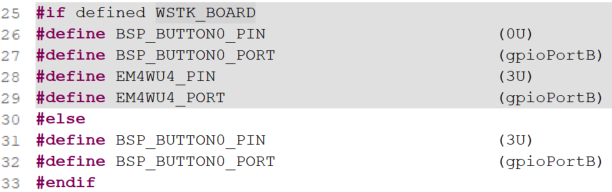
* This file has the associated build macros for enabling and changing configurations for the GPD.
* On line 17 of the file, you see a macro for using a WSTK board.



* The example can be run on a Silicon Labs WSTK board with a BRD4182A radio board attached. The changes associate with this macro are explained in the hal-config.h section of this document.
* There are other macros commented out to maximize power efficiency of the GPD. These can be enabled to add the associated functionality back to the design.
  + EMBER\_AF\_PLUGIN\_CLI on line 63
  + EMBER\_AF\_PLUGIN\_COMMAND\_INTERPRETER2 on line 69
  + EMBER\_AF\_PLUGIN\_EMBER\_MINIMAL\_PRINTF on line 72
  + EMBER\_AF\_PLUGIN\_LED on line 81

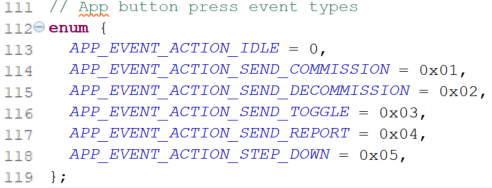
**hal-config.h:**

* Opening the hal-config.h file you will see the pertinent information for running this example on a WSTK board.
* It is important to note that the use of **BTN0** on the WSTK must be routed through the **Expansion Header** on the WSTK to enable BTN0 to wake the radio from EM4 low-power mode. See the associated comment on line 17.
* This will change the associated port and pin associated with BTN0 as well as enable the EM4 wake-up capabilities as seen below:



**gpd-switch-callbacks.c:**

* This file has all the core functionality for the GPD.
* On line 112 of the file, you see the event action types for the GPD; COMMISSION, DECOMMISSION, TOGGLE, REPORT, and STEP\_DOWN.



* These commands are passed into the processAppEvent function on line 429 to send GPD commands.



* The STEP\_DOWN command is a custom Zigbee Green power command used for dimming. This is defined on line 138. To see other custom commands you can implement, see Table 49 of the Zigbee Alliance document outlining GPDF commands:

<https://zigbeealliance.org/wp-content/uploads/2019/11/docs-09-5499-26-batt-zigbee-green-power-specification.pdf>

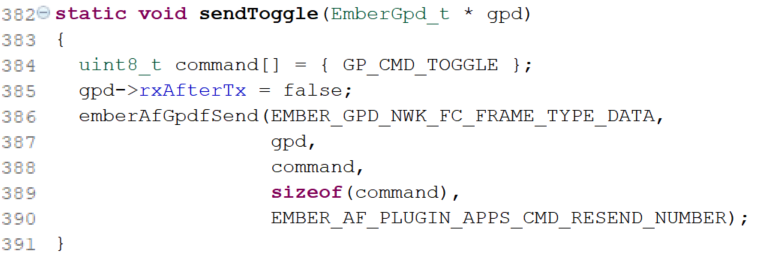
* The GPD is setup for single event commissioning. This enables a periodic timer to allow the user to press the secondary button once to commission to the GPC. This is seen on line 141. 
* The commissioning process has 4 steps:

1. Channel request: To find the channel of the Zigbee network and use that for commissioning requests
2. Commissioning command: Commissioning request
3. Commissioning command: To collect the commissioning reply
4. Commissioning command: To indicate the successful commissioning

* The GPD can be set up for single event commissioning or you can press the button 4 times to go through each step. There must be at least 1 second between each button press to go through each step properly. To enable manual commissioning, comment out the PERIODIC\_COMMISSION\_TIMER on line 141.
* The halButtonIsr function, seen on line 525, handles all button processing events for the large primary button and the small secondary button.



* The commissioning process using either the single event commissioning timer or processing 4 button presses can be seen on line 538 in the **halButtionIsr** function.
* Once the device is commissioned, the secondary button is re-purposed for the custom STEP\_DOWN command as well as decommissioning. Pressing the button once, will send the step-down command. Pressing and holding the button for 3 seconds will issue a decommission command from the GPD to the GPC. This function is seen on line 554.
* The primary button sends a toggle command to turn the LED on/off on the GPC seen on line 567.
* The format for sending the toggle and step-down command can be seen on lines 382 and 393, respectively. The toggle framework is seen below:



* The GPD is set to stay in EM4 low-power mode unless handling button presses and processing Zigbee commands. This is set on line 425.

