UG392: Using Silicon Labs Green Power with EmberZNet PRO

This application note introduces the Silicon Labs Green Power components within the EmberZNet PRO stack and explains how to enable your network for Green Power. This document assumes readers are familiar with the basic Green Power concepts discussed in *UG103.15: Silicon Labs Green Power Fundamentals*.

KEY POINTS

- Introduces Green Power.
- Describes a basic Green Power network.
- Explains Green Power commissioning and operational model.
- Describes the Silicon Labs Green Power devices, their plugins, and callbacks.

1. Introduction to Green Power

Zigbee® Green Power (ZGP) is included in the Zigbee 3.0 specification (Z3) (Zigbee Alliance, Zigbee 3.0 specification). It is an end-to-end open standard that allows ultra-low power devices called Green Power Devices (GPDs) to operate on Zigbee networks.

Green Power is actually a number of technologies combined into one global standard that includes these features (Zigbee Green Power, Cam Williams):

- Energy harvesting technology.
  - Includes mechanical, heat, light, pressure (piezo), and other energy harvesting technology.
- Ultra-low power RF silicon that uses many orders of magnitude less power than required for a sleepy or fully networked wireless connection.
  - Uses ultra-low power, non-volatile memory such as Ferroelectric RAM (FRAM).
- An open, global standard network technology that saves even more energy by reducing packet length, round trips, connection rediscovery, and on-network time for devices that may be offline for extended periods of time.
  - Zigbee – 15.4 – 2.4Ghz modules
- An open, global, standard application layer protocol that supports compressed messages and limited transactions using:
  - Zigbee Application Framework (AF)
  - Zigbee Cluster Library (ZCL)

2. Basic Green Power Network

A basic Green Power (GP) network consists of three devices:

- Green Power Device (GPD)
- A Z3 Proxy or Green Power Proxy (GPP)
- A Green Power Sink (GPS)

GPD Frames (GPDF) are transmitted by the GPD devices and received by a Proxy or a Combination (Sink and Proxy) device. The GPP will then encapsulate the received GPDF within a standard Zigbee frame and forward the GPDF packets across the Zigbee PRO / Z3 network in the form of notifications to the Sink that that has been paired with the end device. In a Combination device, the Proxy side is responsible for forwarding the GPDF packets. The following figure illustrates the data flow from the GPD to the GPP and finally to the GPS.

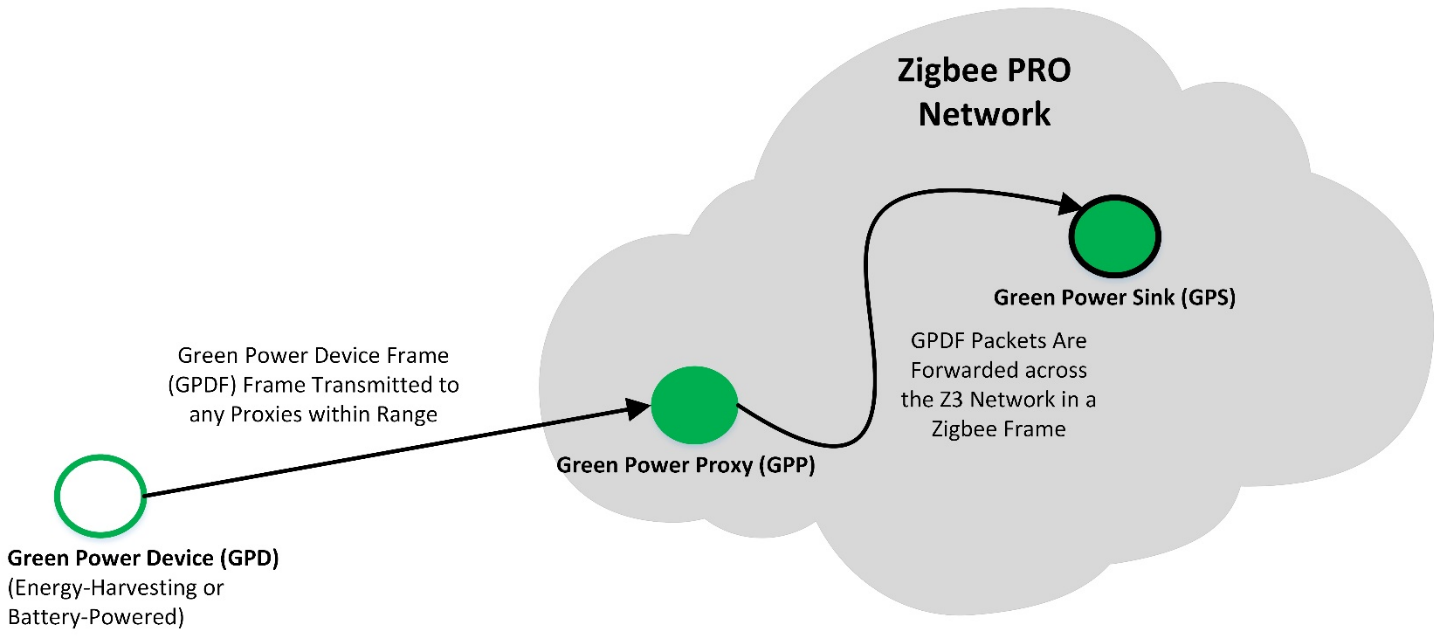

Zigbee PRO Network
Green Power Sink (GPS)
GPDF Packets Are Forwarded across the Z3 Network in a Zigbee Frame
Green Power Device Frame (GPDF) Frame Transmitted to any Proxies within Range
Green Power Proxy (GPP)
Green Power Device (GPD) (Energy-Harvesting or Battery-Powered)

Figure 2.1. Basic Green Power Message Transmission

As indicated in the following figure, the GPDF is shorter than a standard Zigbee frame (indicated by the dashed line). This allows a GPD to transmit a GPDF using less power than a standard Zigbee frame as the radio transmitter is active for less time.

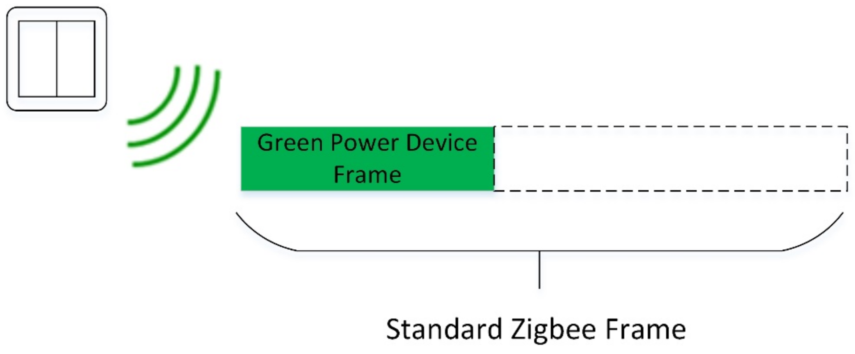


Figure 2.2. GPDF Size

GPDs are primarily one-way devices once in use, although they may optionally (if designed) support bidirectional data exchange during pairing. GPDs should not be considered end devices and Zigbee considers them as less than Zigbee End Devices (ZEDs). For more information on ZEDs, see *UG103.2: Zigbee Fundamentals*.

2.1 How Does Green Power Fit in a Zigbee Network?

The Zigbee stack architecture is illustrated in the following figure.

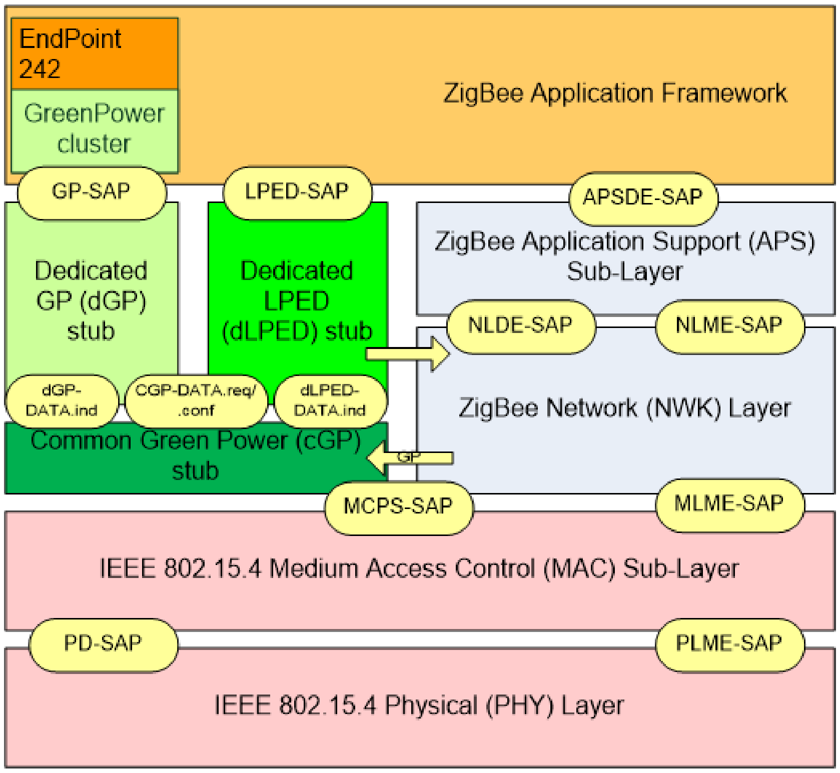

EndPoint 242
GreenPower cluster
ZigBee Application Framework
GP-SAP
LPED-SAP
APSDE-SAP
Dedicated GP (dGP) stub
Dedicated LPED (dLPED) stub
ZigBee Application Support (APS) Sub-Layer
NLDE-SAP
NLME-SAP
dGP-DATA.ind
CGP-DATA.req/.conf
dLPED-DATA.ind
ZigBee Network (NWK) Layer
Common Green Power (cGP) stub
GP
MCPS-SAP
MLME-SAP
IEEE 802.15.4 Medium Access Control (MAC) Sub-Layer
PD-SAP
PLME-SAP
IEEE 802.15.4 Physical (PHY) Layer

Figure 2.3. Zigbee Stack Architecture

Green Power data exchanges are handled by a dedicated block or "stub" for the Zigbee Network (NWK) Layer and the Application Support Sub-layer (APS). Green Power has three elements within the Zigbee stack architecture:

- Common GP (cGP) stub
- Dedicated GP (dGP) stub
- Dedicated LPED (dLPED) stub

The following table describes each of the stubs in more detail.

-

Table 2.1. Green Power Stubs

Stub name	Description
Common GP (cGP)	Performs the basic functions shared by Low Power EndPoint (LPED) and GP. It performs just enough processing to pass application data frames to the Medium Access Control (MAC) layer for transmission and to pass the GPDF payload from the MAC to the relevant dedicated stub on receipt. The cGP stub is accessible to the higher layers through two special Service Access Points (SAPs)—CGP-SAP and CZLPED-SAP.
Dedicated GP (dGP)	Performs just enough processing to pass application data frames to the cGP stub for transmission and to pass GPD commands from the cGP stub to the Green Power cluster on the Green Power EndPoint on receipt. The dGP stub is accessible to the higher layers through a special SAP—GP-SAP—which is parallel to the normal Application Support Sublayer Data Entity (APSDE)-SAP. The dGP communication architecture does not support simultaneous execution by multiple application entities. A ZigBee router is assumed to have only one proxy application entity (Green Power EndPoint) that will use the GP communication mechanism.
Dedicated LPED (dLPED)	This stub, as well as the corresponding LPED-SAPs, are out of scope of this User Guide and will be defined separately by the Low Power End Device Task

2.2 Green Power Devices

GPDs are ultra-low power and even batteryless devices that can utilize this communications method to send messages to Zigbee devices. GPDs can only send and receive GPDFs. GPDs can never be a part of the Zigbee network because they have a different frame format. (See Section 4.1.1 Frame Format for more information.) Some typical GPDs are shown in the following figure.

Figure 2.4. Typical Green Power Devices

2.3 Green Power Infrastructure Devices

ZGP devices that operate within a normal Zigbee network are called *infrastructure devices*. These devices can handle GPDFs in some way. There are two general types of infrastructure devices:

- Green Power Proxy (GPP)
- Green Power Sink (GPS)

The GP specification defines the entire set of features for each of these devices. Some of these features are optional and some are mandatory, called *Basic* by Zigbee. As a result, there is a set of sub-names: Green Power Proxy Basic (GPPB), Green Power Sink Basic (GPSB) and Green Power Combo Basic (GPCB + GPSB) that implements the functionality for both Proxy and Sink within a single device.

The following figure illustrates two GP infrastructure devices.

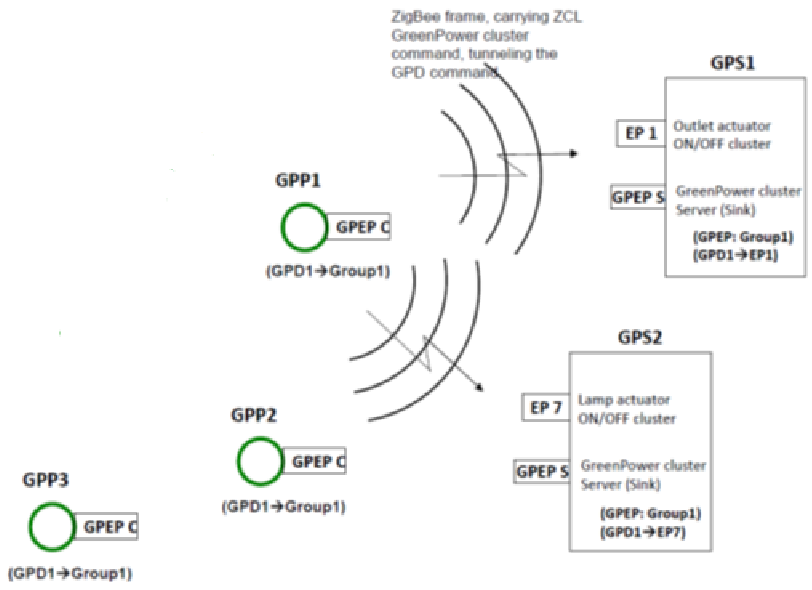

ZigBee frame, carrying ZCL GreenPower cluster command, tunneling the GPD command
GPS1
EP 1
Outlet actuator ON/OFF cluster
GPEP S
GreenPower cluster Server (Sink)
(GPEP: Group1)
(GPD1→EP1)
GPP1
GPEP C
(GPD1→Group1)
GPS2
EP 7
Lamp actuator ON/OFF cluster
GPEP S
GreenPower cluster Server (Sink)
(GPEP: Group1)
(GPD1→EP7)
GPP2
GPEP C
(GPD1→Group1)
GPP3
GPEP C
(GPD1→Group1)

Figure 2.5. Green Power Device Types

- Green Power Proxy (GPP)
  - Receives and forwards the GPDF wrapped in a ZCL command over the Zigbee network. This is sometimes called *tunneling* the GPD command.
  - Receives the GPDF, hence a Server for GPDF frames.
  - Sends the ZCL tunnel commands to the GPS. This is a client role of GP Cluster in ZCL.
- Green Power Sink (GPS)
  - Receives the tunneled commands and executes them.
  - Receives the ZCL tunnel commands from the GPP. This is a server role of GP Cluster in ZCL.

  Note: Even though it is possible to implement a standalone GPS device with EmberZNet PRO with the limitation of in-range direct communication with a GPD, a Green Power Combo (GPC) is preferred over a standalone GPS. The reason is that standalone sinks are rare because the Z3 specification requires all the routers to be Proxy Basic at minimum. The following sections of this User Guide discuss how to implement a GP Combo (GPC) rather than a GPS.

3. Green Power Commissioning

Commissioning is the process of adding a GPD to the Zigbee network so it can perform application functionality. This process establishes the GPD in the Proxy Table and in the Sink Table. If there is an infrastructure device (Sink or Combo) that has matching functionality with the GPD, they can be commissioned as a pair and work together.

The GPD sends a series of GP commands to the listening GP infrastructure device (Sink or Combo) which adds it as a commissioned node for further data transactions. In this command exchange, the GPD and the GP infrastructure device agree on the functionality that matches and the security requirements for additional command/data transactions.

There are two types of commissioning process based on the capabilities and design of the GPD:

- Unidirectional Commissioning: The GPD only sends the commissioning commands and does not receive any data back from the Zigbee network. In this type of commissioning, the application must predefine the Zigbee network channel and security requirements.
- Bidirectional Commissioning: The GPD sends and receives data through a series of commands, called *commissioning commands*. With bidirectional commissioning, the GPD can find the Zigbee network channel and negotiate the security level key. This process consumes more energy because it includes both transmission and receive steps.

3.1 Unidirectional Commissioning

The following figure illustrates how unidirectional commissioning works.

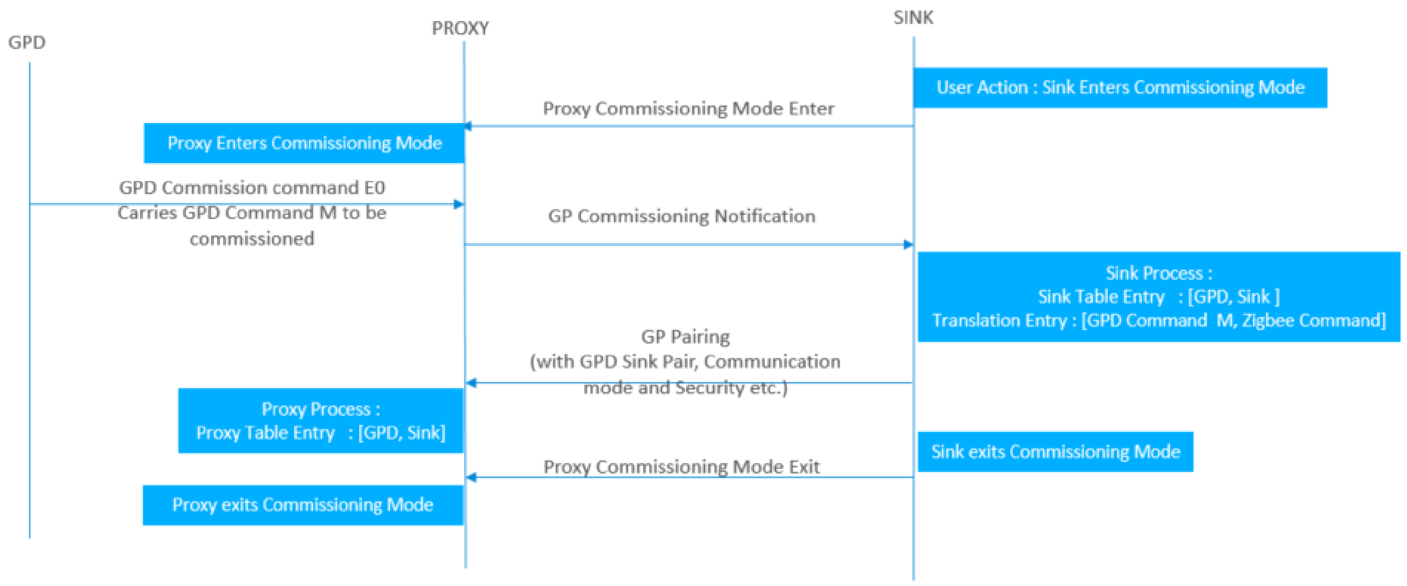

GPD
PROXY
SINK
User Action : Sink Enters Commissioning Mode
Proxy Commissioning Mode Enter
Proxy Enters Commissioning Mode
GPD Commission command E0
Carries GPD Command M to be commissioned
GP Commissioning Notification
Sink Process :
Sink Table Entry : [GPD, Sink]
Translation Entry : [GPD Command M, Zigbee Command]
GP Pairing
(with GPD Sink Pair, Communication mode and Security etc.)
Proxy Process :
Proxy Table Entry : [GPD, Sink]
Sink exits Commissioning Mode
Proxy Commissioning Mode Exit
Proxy exits Commissioning Mode

Figure 3.1. Unidirectional Commissioning

1. The Sink tells the Proxy (or Proxies) to be ready to commission a GPD and enters commissioning mode.
2. The GPD sends a commissioning packet to join the Proxy.
3. The Proxy sends this notification to the Sink, which establishes the Sink Table entries.
4. The Sink send a pairing packet to the Proxy (or Proxies), which populate the Proxy Table entries.
5. Both the Sink and the Proxy exit commissioning mode.

After completion of these steps, the GPD is ready to operate on the network.

3.2 Bidirectional Commissioning

The following figures illustrate how bidirectional commissioning works. It has two phases.

1. Find the Zigbee network channel.

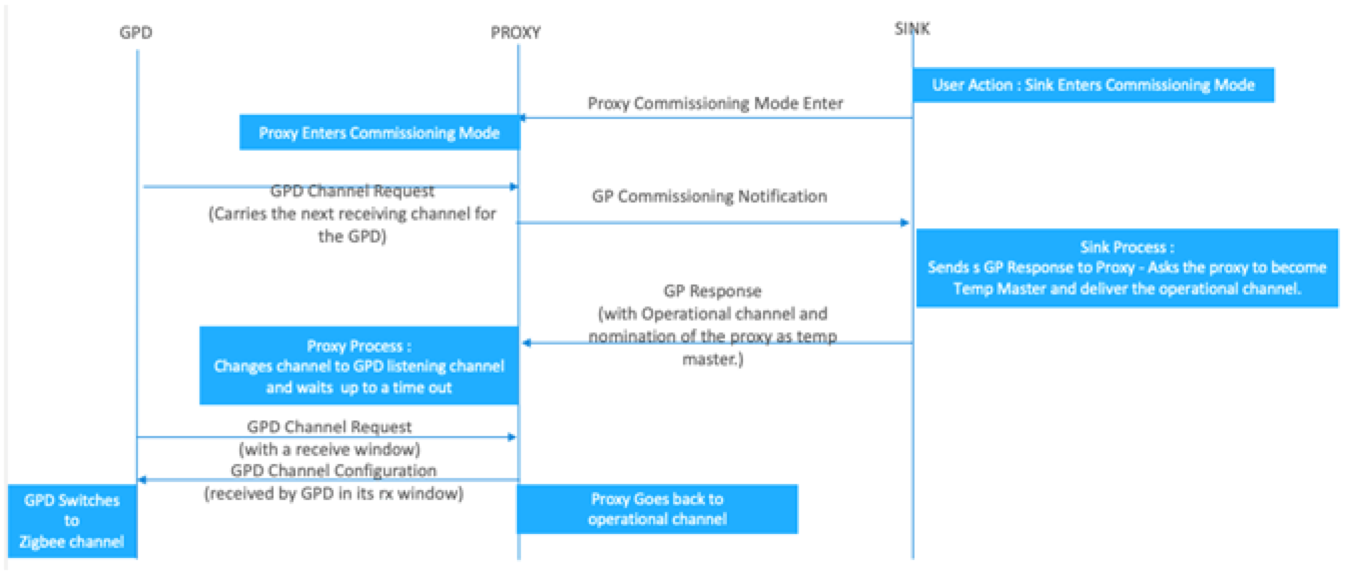

GPD
PROXY
SINK
User Action : Sink Enters Commissioning Mode
Proxy Commissioning Mode Enter
Proxy Enters Commissioning Mode
GPD Channel Request (Carries the next receiving channel for the GPD)
GP Commissioning Notification
Sink Process : Sends s GP Response to Proxy - Asks the proxy to become Temp Master and deliver the operational channel.
GP Response (with Operational channel and nomination of the proxy as temp master.)
Proxy Process : Changes channel to GPD listening channel and waits up to a time out
GPD Channel Request (with a receive window)
GPD Channel Configuration (received by GPD in its rx window)
GPD Switches to Zigbee channel
Proxy Goes back to operational channel

Figure 3.2. Finding the Zigbee Network Channel

2. Execute bidirectional commissioning.

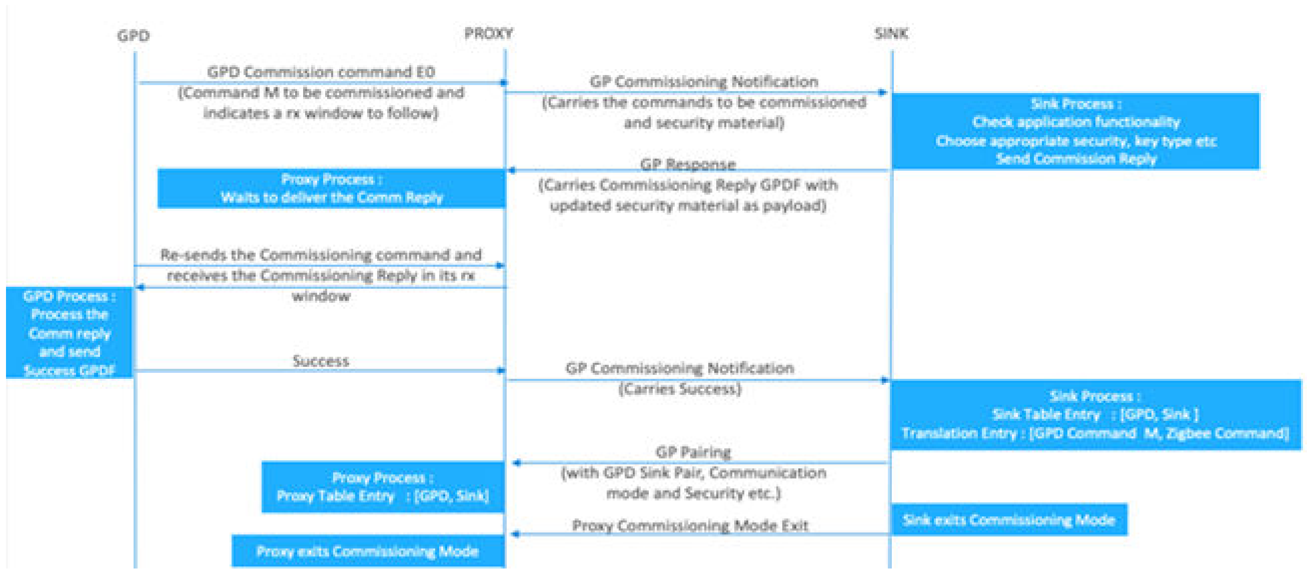

GPD
PROXY
SINK
GPD Commission command E0 (Command M to be commissioned and indicates a rx window to follow)
GP Commissioning Notification (Carries the commands to be commissioned and security material)
Sink Process : Check application functionality Choose appropriate security, key type etc Send Commission Reply
GP Response (Carries Commissioning Reply GPDF with updated security material as payload)
Proxy Process : Waits to deliver the Comm Reply
Re-sends the Commissioning command and receives the Commissioning Reply in its rx window
GPD Process : Process the Comm reply and send Success GPDF
Success
GP Commissioning Notification (Carries Success)
Sink Process : Sink Table Entry : [GPD, Sink] Translation Entry : [GPD Command M, Zigbee Command]
GP Pairing (with GPD Sink Pair, Communication mode and Security etc.)
Proxy Process : Proxy Table Entry : [GPD, Sink]
Sink exits Commissioning Mode
Proxy Commissioning Mode Exit
Proxy exits Commissioning Mode

Figure 3.3. Execute Bidirectional Commissioning

4. Green Power Operational Model

A GPD is said to be *operational* when it sends application control commands such as on/off. Typically, such operational commands are useful after the GPD is commissioned to a Zigbee network after the commissioning process. Like commissioning, a GPD can have two operational modes.

- Unidirectional Operation: Sends commands to the Zigbee network and requires a minimum energy budget.
- Bidirectional Operation: Sends commands receives a response back—for example, reading an attribute from the network. Because this mode sending data to the Zigbee network and receiving data back from the network, it uses more energy.

4.1 Green Power Command Transport

The following figure illustrates a high-level typical GP operational command transport with four GP devices.

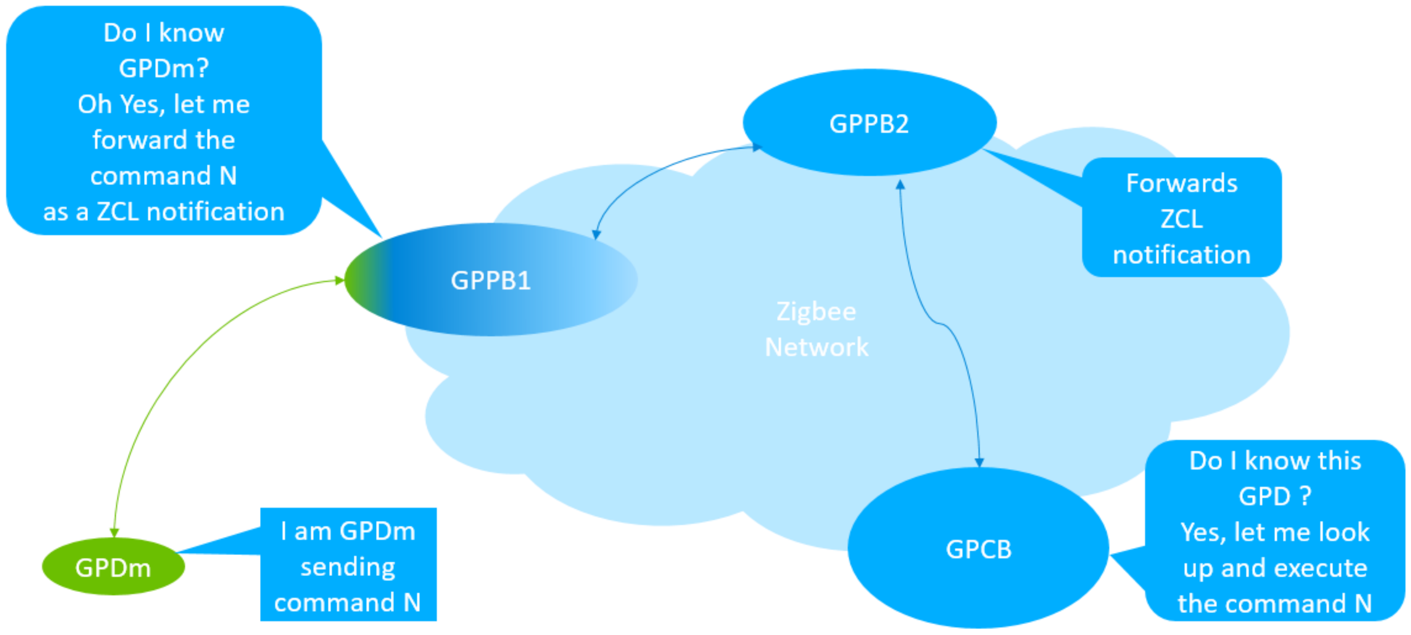

Do I know GPDm? Oh Yes, let me forward the command N as a ZCL notification
GPPB2
Forwards ZCL notification
GPPB1
Zigbee Network
Do I know this GPD ? Yes, let me look up and execute the command N
GPCB
GPDm
I am GPDm sending command N

Figure 4.1. Green Power Command Transport

These are the general steps in the Green Power command transport in the above figure.

1. GPDm (a GPD with Source Id = m) sends a command N (a GPDF frame) to GPPB1.
2. GPPB1 receives the command N.
3. GPPB1 looks up GPDm in its proxy table to determine whether GPDm has an entry paired with a Sink. If it does, GPPB1 forwards command N as a Zigbee Cluster Library (ZCL) notification to GPPB2. If it does not recognize GPDm after the proxy table lookup, GPPB1 drops the packet.
4. Once GPPB2 receives the notification, it forwards the command further as a ZCL command, as it would do for any other ZCL commands in the Zigbee network. (Any router in the Zigbee network would do the same because in Z3, all routers are GPPBs.)
5. The GPCB receives the ZCL notification and processes it by looking up its Sink table to obtain the required information. If all the information is correct and the GPDm is paired to this GPCB, the GPCB processes the command as follows:

   1. Looks up in its translation table to attempt to translate command N.

   2. Translates command N to an equivalent Zigbee command (which includes interpreting the command payload) and finds one or more paired application endpoints (which are in the translation table).

   3. Pushes the Zigbee packet to the identified application endpoint on the node.

4.1.1 Frame Format

The following figures illustrate the standard format of a Zigbee frame and the format of a generic GP frame. The size difference is because the entire GP frame must fit into the ZCL payload (or addressing).

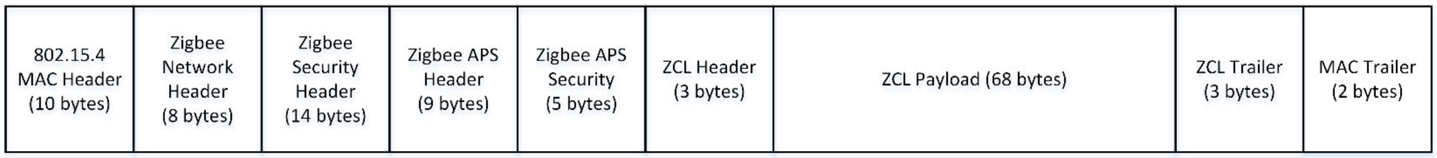

802.15.4 MAC Header (10 bytes)
Zigbee Network Header (8 bytes)
Zigbee Security Header (14 bytes)
Zigbee APS Header (9 bytes)
Zigbee APS Security (5 bytes)
ZCL Header (3 bytes)
ZCL Payload (68 bytes)
ZCL Trailer (3 bytes)
MAC Trailer (2 bytes)

Figure 4.2. Zigbee Frame Format

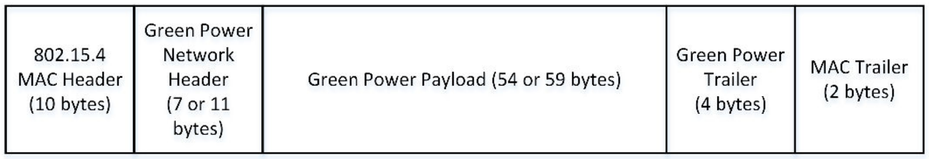

802.15.4 MAC Header (10 bytes)
Green Power Network Header (7 or 11 bytes)
Green Power Payload (54 or 59 bytes)
Green Power Trailer (4 bytes)
MAC Trailer (2 bytes)

Figure 4.3. Green Power Frame Format

Most of the information contained in the Zigbee NWK header and all the information in the APS headers is not relevant for GP operation. As a result, the GP frame contains a modified NWK header and no APS header, followed by a dedicated application payload. GP frames are compact and shorter, but they contain all the necessary information required for addressing and security.

5. Silicon Labs Green Power within EmberZNet PRO

There are three Silicon Labs Green Power devices within EmberZNet PRO:

- Green Power Device
- Green Power Proxy Basic
- Green Power Combo Basic

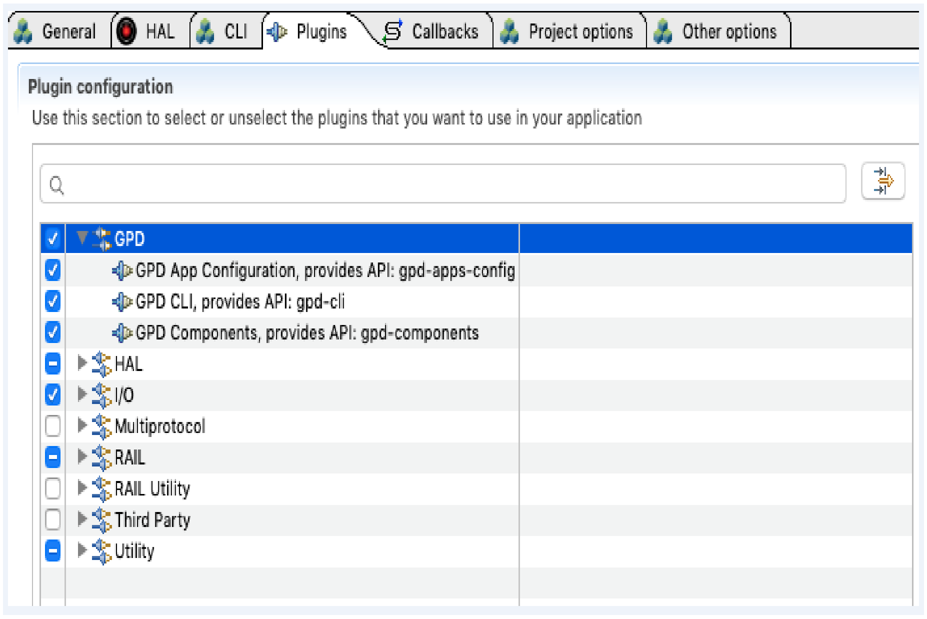
The following sections describe some of the main features, plugins, and callbacks for each device type.

5.1 Green Power Device

A device that is outside a Zigbee network and can send commands to the Zigbee network via a GPP.

5.1.1 Plugins

These are the plugins related to a GPD application and its components available in the GPD framework.



Plugin configuration

Use this section to select or unselect the plugins that you want to use in your application

- [✓] GPD
  - [✓] GPD App Configuration, provides API: gpd-apps-config
  - [✓] GPD CLI, provides API: gpd-cli
  - [✓] GPD Components, provides API: gpd-components
- [–] HAL
- [✓] I/O
- [ ] Multiprotocol
- [–] RAIL
- [ ] RAIL Utility
- [ ] Third Party
- [–] Utility

Figure 5.1. Green Power Device Plugins

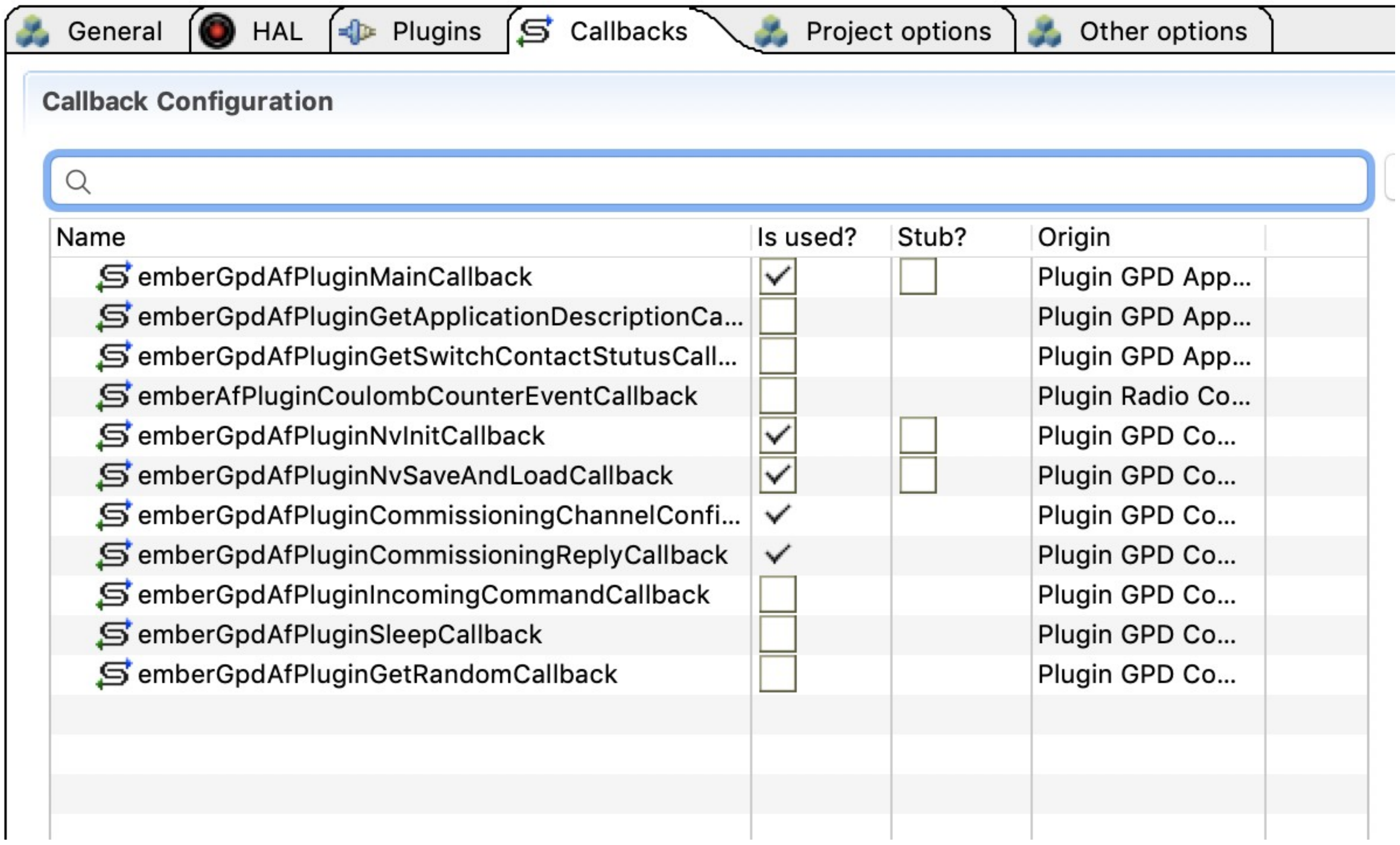
The GPD framework provides functional modules in the form of plugins described in the following table.

Table 5.1. Green Power Device Plugin Descriptions

Plugin Name	Description
GPD	GPD App Configuration: Configures application capability and device parameters. It also consists of a set of modules that provide application-level functions such as main, node configuration, and commissioning. GPD CLI: Provides a basic set of Command Line Interface (CLI) commands for development and testing. GPD Components: Includes a set of modules to provide the following capabilities: • Wrapper interface to RAIL layer • Sending and receiving bidirectional GPDFs • GPDF packet formation and parsing • GPDF security tagging and validation wrappers for the mbed TLS plugin • Nonvolatile memory wrapper functions
HAL	HAL Library: Includes a set of low-layer peripheral and board support modules.
RAIL	Includes the RAIL library.
Utility	Provides mbedTLS functionality.

5.1.2 Callbacks

The following figure illustrates the set of callbacks available for user interaction that provide information and accept inputs.



Name	Is used?	Stub?	Origin
emberGpdAfPluginMainCallback	✓	☐	Plugin GPD App...
emberGpdAfPluginGetApplicationDescriptionCa...	☐		Plugin GPD App...
emberGpdAfPluginGetSwitchContactStutusCall...	☐		Plugin GPD App...
emberAfPluginCoulombCounterEventCallback	☐		Plugin Radio Co...
emberGpdAfPluginNvInitCallback	✓	☐	Plugin GPD Co...
emberGpdAfPluginNvSaveAndLoadCallback	✓	☐	Plugin GPD Co...
emberGpdAfPluginCommissioningChannelConfi...	✓		Plugin GPD Co...
emberGpdAfPluginCommissioningReplyCallback	✓		Plugin GPD Co...
emberGpdAfPluginIncomingCommandCallback	☐		Plugin GPD Co...
emberGpdAfPluginSleepCallback	☐		Plugin GPD Co...
emberGpdAfPluginGetRandomCallback	☐		Plugin GPD Co...

Figure 5.2. Green Power Device Callbacks

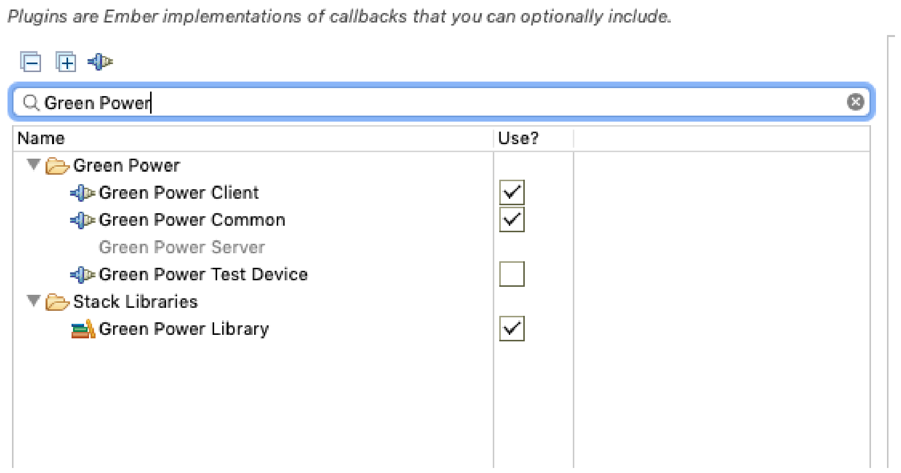
5.2 Green Power Proxy Basic

Green Power Proxy Basic has the following key features:

- Receives Green Power Frames.
- Converts Green Power Frames to ZCL commands.
- Is involved in commissioning new devices.

5.2.1 Plugins

Plugins implement the functionality for the mandatory incoming commands. On the Plugin tab, ensure the following plugins are enabled.



Plugins are Ember implementations of callbacks that you can optionally include.

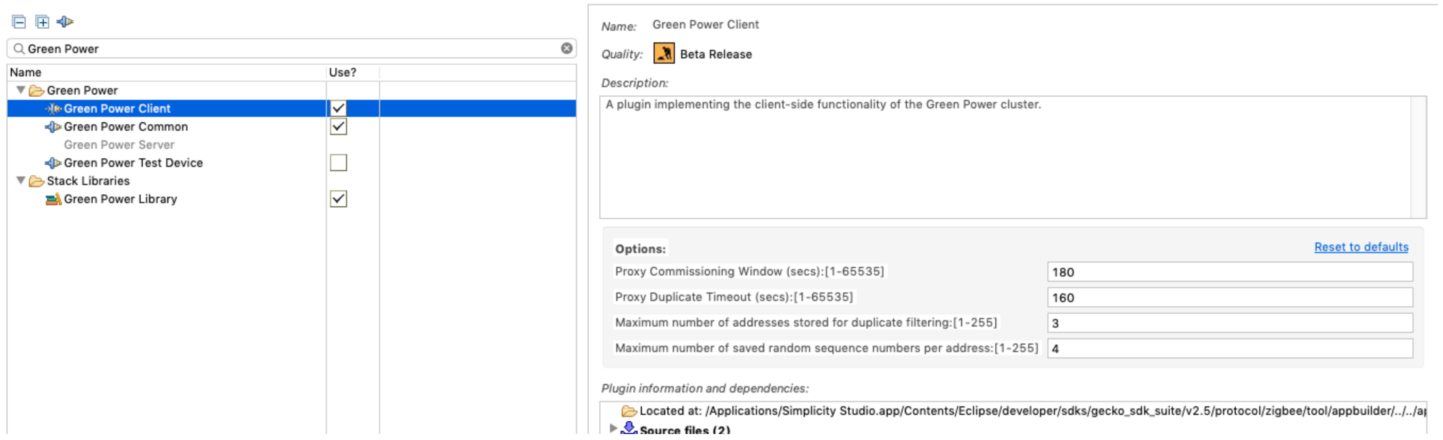
Green Power

Name	Use?
Green Power	
Green Power Client	✓
Green Power Common	✓
Green Power Server	
Green Power Test Device	☐
Stack Libraries	
Green Power Library	✓

Figure 5.3. Green Power Proxy Basic Plugins

The Green Power Client plugin implements most of the Proxy functionality while the Green Power Common plugin provides some of the common functions and utility that is shared by both the Green Power Server and the Green Power Client.

When you enable a Green Power Client plugin, make sure to set the correct desired options as shown in the following figure.



Green Power

Name	Use?
Green Power	
Green Power Client	✓
Green Power Common	✓
Green Power Server	
Green Power Test Device	☐
Stack Libraries	
Green Power Library	✓

Name: Green Power Client

Quality: Beta Release

Description:

A plugin implementing the client-side functionality of the Green Power cluster.

Options: Reset to defaults

Proxy Commissioning Window (secs):[1-65535]	180
Proxy Duplicate Timeout (secs):[1-65535]	160
Maximum number of addresses stored for duplicate filtering:[1-255]	3
Maximum number of saved random sequence numbers per address:[1-255]	4

Plugin information and dependencies:

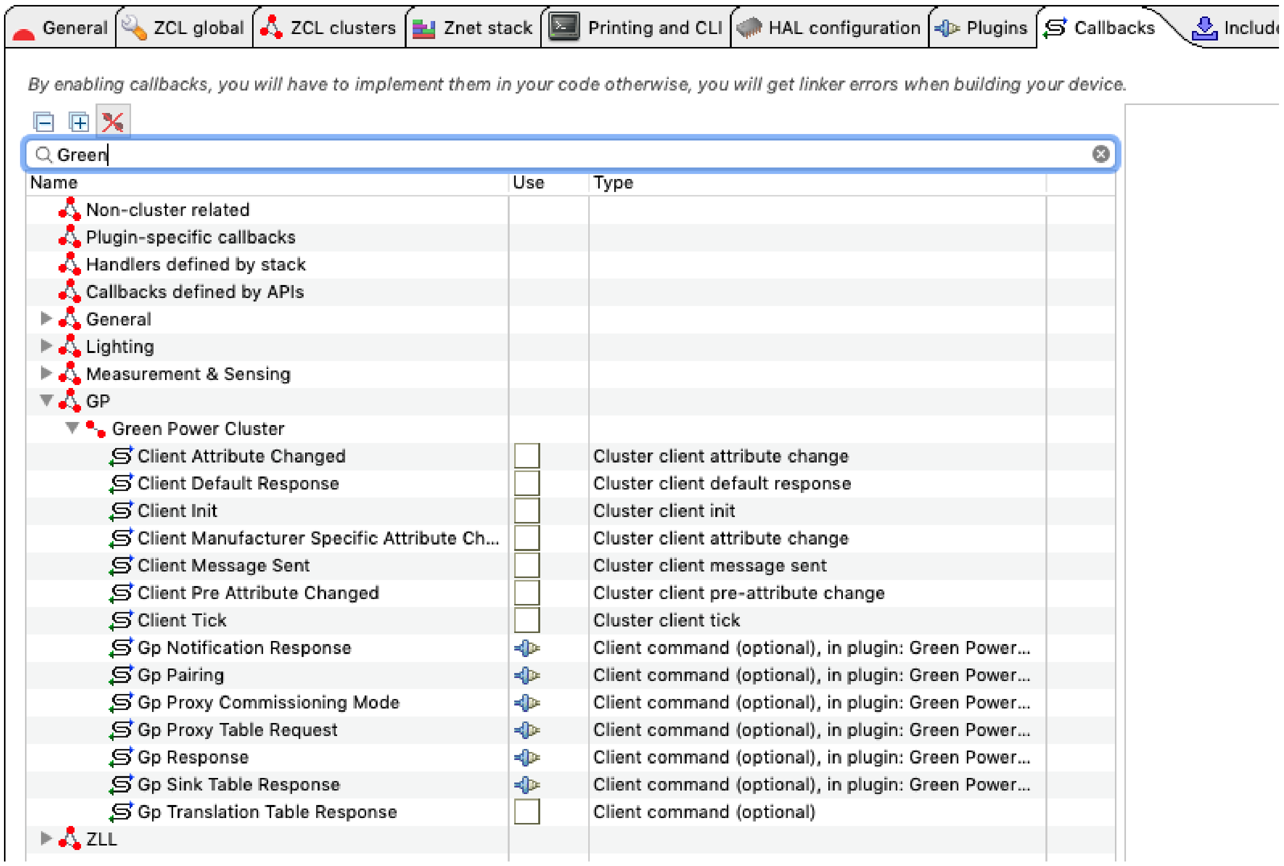
Located at: /Applications/Simplicity Studio.app/Contents/Eclipse/developer/sdks/gecko_sdk_suite/v2.5/protocol/zigbee/tool/appbuilder/../../ap

Source files (2)

Figure 5.4. Green Power Client Plugin Options

5.2.2 Callbacks

The following figure illustrates the Green Power cluster incoming commands callbacks that are implemented by the Green Power Client as part of the Green Power Proxy Basic.



By enabling callbacks, you will have to implement them in your code otherwise, you will get linker errors when building your device.

Green

Name	Use	Type
Non-cluster related		
Plugin-specific callbacks		
Handlers defined by stack		
Callbacks defined by APIs		
General		
Lighting		
Measurement & Sensing		
GP		
Green Power Cluster		
Client Attribute Changed	☐	Cluster client attribute change
Client Default Response	☐	Cluster client default response
Client Init	☐	Cluster client init
Client Manufacturer Specific Attribute Ch...	☐	Cluster client attribute change
Client Message Sent	☐	Cluster client message sent
Client Pre Attribute Changed	☐	Cluster client pre-attribute change
Client Tick	☐	Cluster client tick
Gp Notification Response		Client command (optional), in plugin: Green Power...
Gp Pairing		Client command (optional), in plugin: Green Power...
Gp Proxy Commissioning Mode		Client command (optional), in plugin: Green Power...
Gp Proxy Table Request		Client command (optional), in plugin: Green Power...
Gp Response		Client command (optional), in plugin: Green Power...
Gp Sink Table Response		Client command (optional), in plugin: Green Power...
Gp Translation Table Response	☐	Client command (optional)
ZLL		

Figure 5.5. Green Power Proxy Client Callbacks

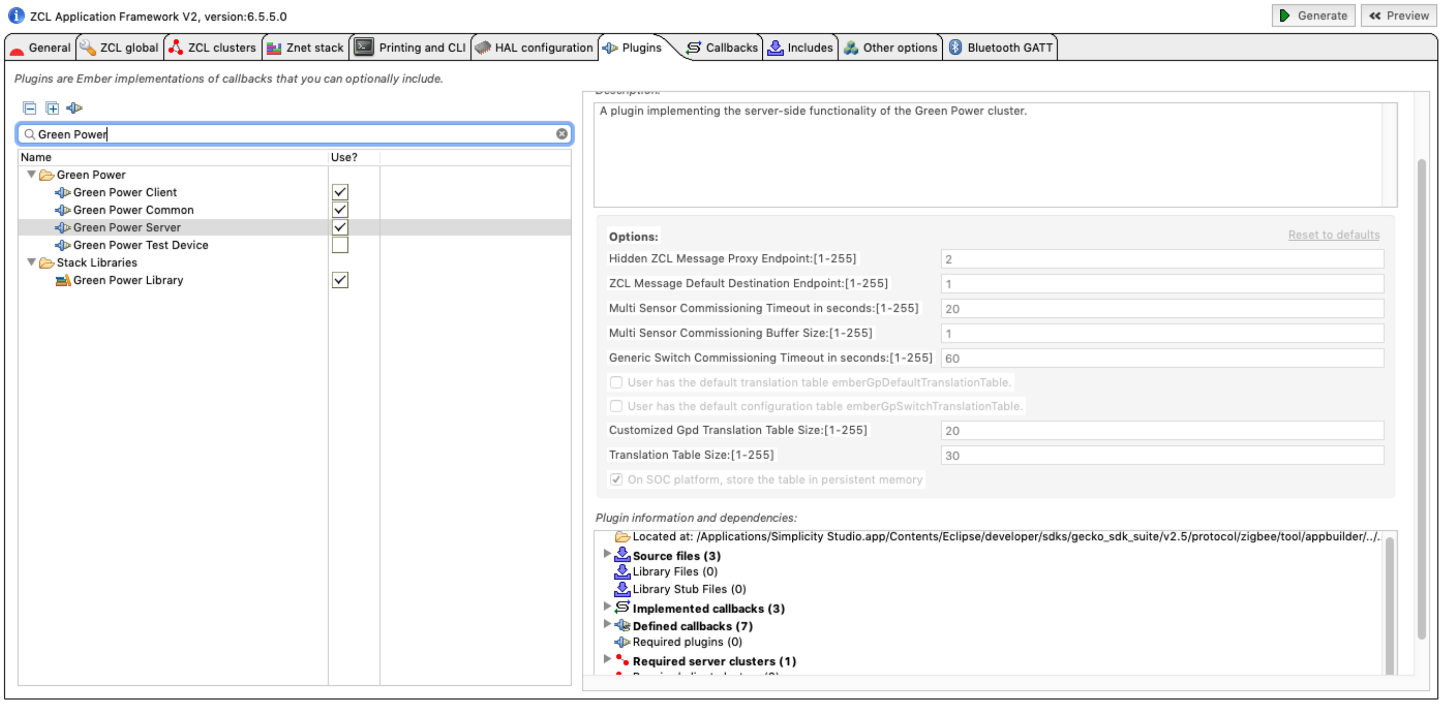
5.3 Green Power Combo Basic

Green Power Combo Basic has the following key features:

- Receives Green Power Frames.
- Converts Green Power Frames to ZCL commands.
- Commissions new devices.
- Is commanded by Green Power Devices.

5.3.1 Plugins

Plugins implement the functionality for the mandatory incoming commands for both the client and the server. On the Plugin tab, ensure the following plugins are enabled.



Plugins are Ember implementations of callbacks that you can optionally include.

Green Power

Name	Use?
Green Power	
Green Power Client	✓
Green Power Common	✓
Green Power Server	✓
Green Power Test Device	
Stack Libraries	
Green Power Library	✓

Description:
A plugin implementing the server-side functionality of the Green Power cluster.

Options: Reset to defaults

Hidden ZCL Message Proxy Endpoint:[1-255]	2
ZCL Message Default Destination Endpoint:[1-255]	1
Multi Sensor Commissioning Timeout in seconds:[1-255]	20
Multi Sensor Commissioning Buffer Size:[1-255]	1
Generic Switch Commissioning Timeout in seconds:[1-255]	60
☐ User has the default translation table emberGpDefaultTranslationTable.	
☐ User has the default configuration table emberGpSwitchTranslationTable.	
Customized Gpd Translation Table Size:[1-255]	20
Translation Table Size:[1-255]	30
✓ On SOC platform, store the table in persistent memory	

Plugin information and dependencies:

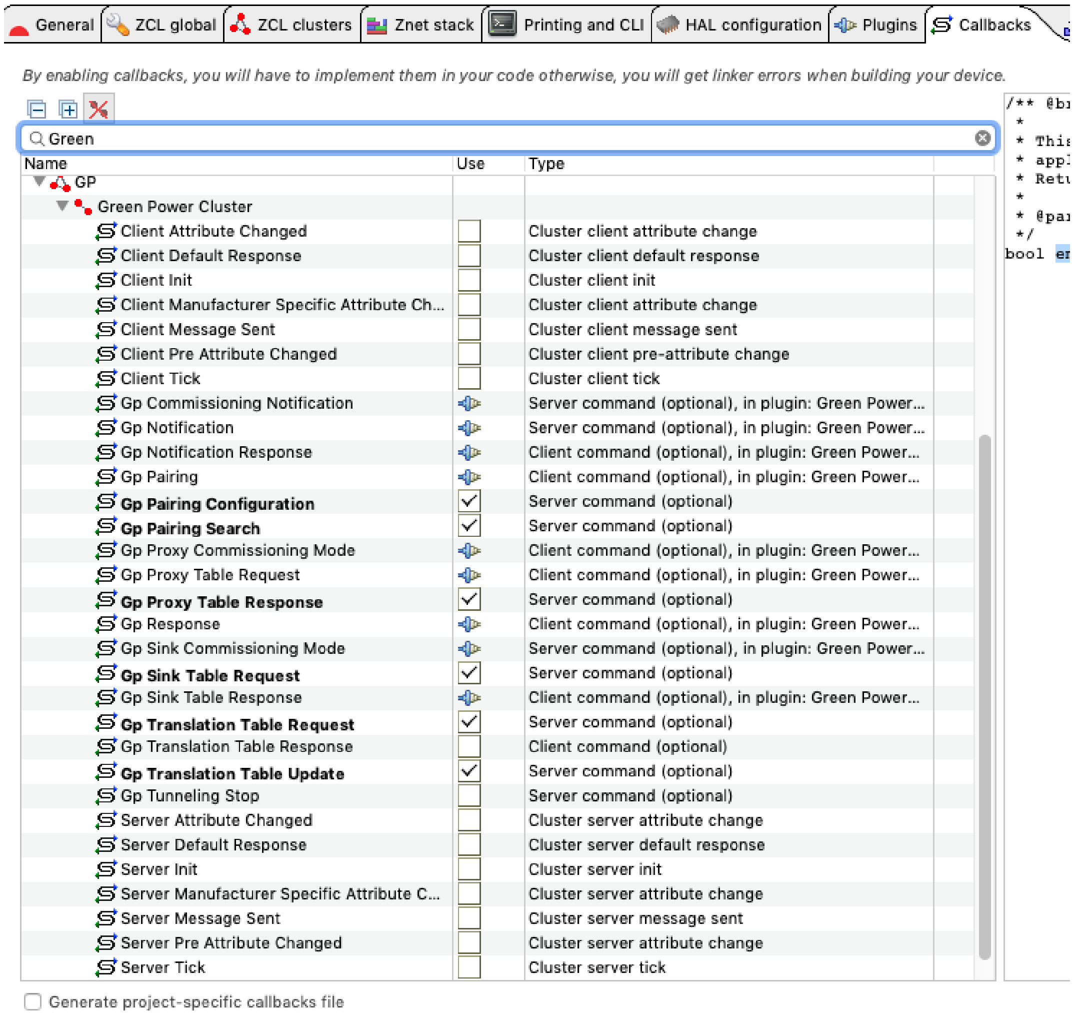
Located at: /Applications/Simplicity Studio.app/Contents/Eclipse/developer/sdks/gecko_sdk_suite/v2.5/protocol/zigbee/tool/appbuilder/../.
Source files (3)
Library Files (0)
Library Stub Files (0)
Implemented callbacks (3)
Defined callbacks (7)
Required plugins (0)
Required server clusters (1)

Figure 5.6. Green Power Sink Combo Plugins

Set **Hidden ZCL Message Proxy Endpoint** to one of the application endpoints implemented by the Green Power Combo Basic to allow the AF to send the operational command forwarding. Set **ZCL Message Default Destination Endpoint** to one of the application endpoints.

5.3.2 Callbacks

The following figure illustrates the Green Power Cluster incoming command callbacks that are implemented by the Green Power Client and the Green Power Server.



General | ZCL global | ZCL clusters | Znet stack | Printing and CLI | HAL configuration | Plugins | Callbacks

By enabling callbacks, you will have to implement them in your code otherwise, you will get linker errors when building your device.

Green

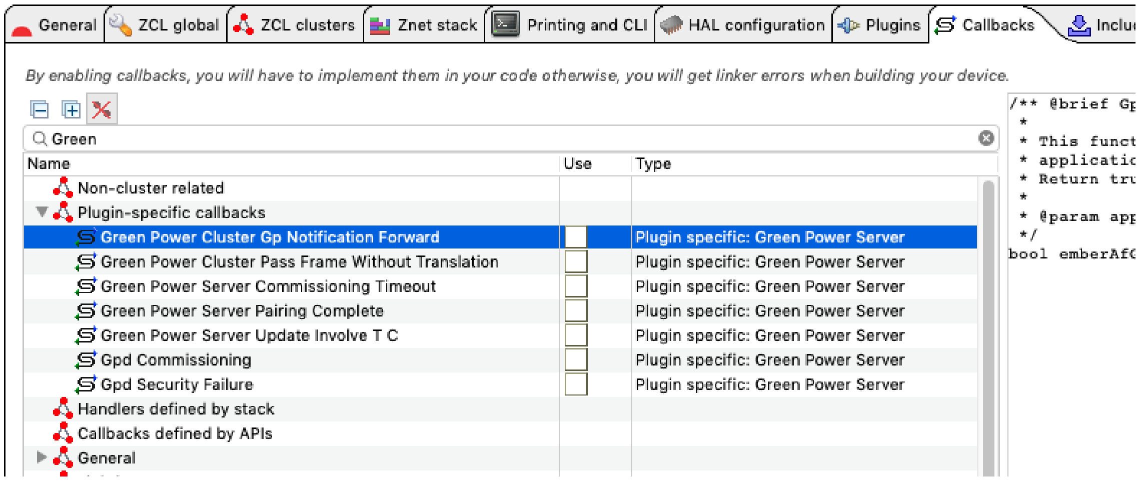
Name	Use	Type
GP		
Green Power Cluster		
Client Attribute Changed	☐	Cluster client attribute change
Client Default Response	☐	Cluster client default response
Client Init	☐	Cluster client init
Client Manufacturer Specific Attribute Ch...	☐	Cluster client attribute change
Client Message Sent	☐	Cluster client message sent
Client Pre Attribute Changed	☐	Cluster client pre-attribute change
Client Tick	☐	Cluster client tick
Gp Commissioning Notification	🔌	Server command (optional), in plugin: Green Power...
Gp Notification	🔌	Server command (optional), in plugin: Green Power...
Gp Notification Response	🔌	Client command (optional), in plugin: Green Power...
Gp Pairing	🔌	Client command (optional), in plugin: Green Power...
Gp Pairing Configuration	✓	Server command (optional)
Gp Pairing Search	✓	Server command (optional)
Gp Proxy Commissioning Mode	🔌	Client command (optional), in plugin: Green Power...
Gp Proxy Table Request	🔌	Client command (optional), in plugin: Green Power...
Gp Proxy Table Response	✓	Server command (optional)
Gp Response	🔌	Client command (optional), in plugin: Green Power...
Gp Sink Commissioning Mode	🔌	Server command (optional), in plugin: Green Power...
Gp Sink Table Request	✓	Server command (optional)
Gp Sink Table Response	🔌	Client command (optional), in plugin: Green Power...
Gp Translation Table Request	✓	Server command (optional)
Gp Translation Table Response	☐	Client command (optional)
Gp Translation Table Update	✓	Server command (optional)
Gp Tunneling Stop	☐	Server command (optional)
Server Attribute Changed	☐	Cluster server attribute change
Server Default Response	☐	Cluster server default response
Server Init	☐	Cluster server init
Server Manufacturer Specific Attribute C...	☐	Cluster server attribute change
Server Message Sent	☐	Cluster server message sent
Server Pre Attribute Changed	☐	Cluster server attribute change
Server Tick	☐	Cluster server tick

```
/** @b
 *
 * This
 * appl
 * Retu
 *
 * @par
 */
bool er
```

☐ Generate project-specific callbacks file

Figure 5.7. Green Power Cluster Command Callbacks

The following figure illustrates the set of callbacks that the Green Power Server uses as application callbacks.



By enabling callbacks, you will have to implement them in your code otherwise, you will get linker errors when building your device.

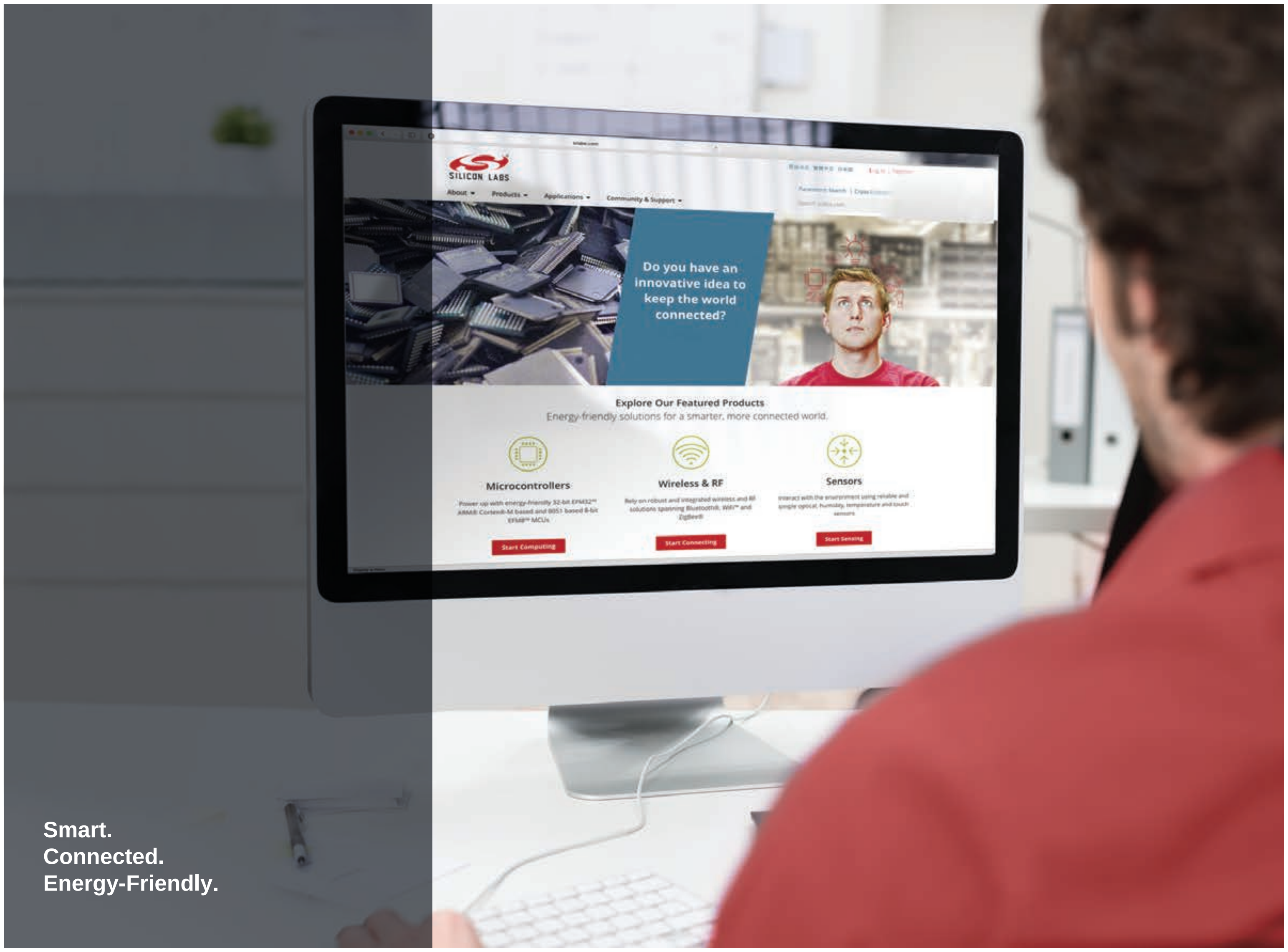
Green

Name	Use	Type
Non-cluster related		
Plugin-specific callbacks		
Green Power Cluster Gp Notification Forward	☐	Plugin specific: Green Power Server
Green Power Cluster Pass Frame Without Translation	☐	Plugin specific: Green Power Server
Green Power Server Commissioning Timeout	☐	Plugin specific: Green Power Server
Green Power Server Pairing Complete	☐	Plugin specific: Green Power Server
Green Power Server Update Involve T C	☐	Plugin specific: Green Power Server
Gpd Commissioning	☐	Plugin specific: Green Power Server
Gpd Security Failure	☐	Plugin specific: Green Power Server
Handlers defined by stack		
Callbacks defined by APIs		
General		

```
/** @brief Gp
 *
 * This funct
 * applicatio
 * Return tru
 *
 * @param app
 */
bool emberAfG
```

Figure 5.8. Green Power Server Application Callbacks

Note: Section 5.2 Green Power Proxy Basic and section 5.3 Green Power Combo Basic explained the implementation for a System-on-Chip (SoC) architecture. The Silicon Labs Zigbee application framework provides a Host-xNCP architecture for implementing a GPP or GPC. The plugins and callbacks remain the same and present on the host framework and the Green Power Library has the same settings as the above and remains on the xNCP.



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com