



AN1181: Configuring Antenna Diversity for EmberZNet and Silicon Labs Thread

This document describes how to use Application Builder plugins and the Hardware Configurator to configure Antenna Diversity in EmberZNet and Silicon Labs Thread applications. Both receive and transmit antenna diversity configuration are discussed.

This information applies to EmberZNet version 6.5.0 and higher, and Silicon Labs Thread version 2.9.0 and higher.

KEY POINTS

- Describes antenna diversity and the transmit and receive implementations
- Provides instructions and examples for configuring transmit and receive antenna diversity.
- Includes additional configuration information if antenna diversity is implemented with a FEM.

1 About Antenna Diversity

Antenna diversity is a technique used to improve radio performance by using two different antennas to transmit and/or receive messages. For a more detailed explanation of the problems with signal transmission and reception in indoor environments and how antenna diversity can mitigate those problems, see <https://www.silabs.com/documents/public/white-papers/using-antenna-diversity-to-create-highly-robust-radio-links.pdf>.

Antenna diversity may be applied to transmission (Tx) and/or reception (Rx). Diversity is achieved using an external RF switch, either standalone or as part of a FEM (front end module)/LNA (low noise amplifier).

The Tx algorithm uses reception of the packet acknowledgement (ACK) to determine if it should change antennas. If the device does not receive an ACK after packet transmission it toggles the RF switch to the other antenna and tries again. It retries two more times, for a total of four attempts, before the MAC (media access layer) fails the transmit up to the network layer. Specifically, the worst-case scenario is:

- New MAC packet transmitted on antenna 1.
- No ACK received so antenna is switched to antenna 2.
- MAC retransmit #1 sent on antenna 2.
- No ACK received, so antenna is switched to antenna 1.
- MAC retransmit #2 sent on antenna 1.
- No ACK received, so antenna is switched to antenna 2.
- MAC retransmit #3 sent on antenna 2.
- No ACK received, so antenna is switched to antenna 1.
- (MAC retries have exhausted, so MAC fails transmit to network layer).
- The next transmit will start on antenna 1.

If transmission is successful, at the beginning of the next transmission the device starts on the last successfully-used antenna.

In antenna diversity with RSSI, the receiver alternates between antenna 1 and antenna 2 during the timing search looking for a valid timing pattern on the Rx signal. When a valid timing pattern is found, antenna diversity tries to select the best antenna for receiving the rest of the frame. To achieve this, the signal quality for the currently active antenna is saved/updated at every subsequent antenna switch. Therefore, at the first timing detect event the algorithm already has a fresh quality metric for one antenna. To perform a valid comparison between antenna 1 and antenna 2, the radio switches simultaneously with the timing detect event to the other antenna to perform a signal quality evaluation/update there. Finally, antenna quality results get compared, and the algorithm selects the better antenna for packet reception. If the better antenna is the current antenna, then the Rx operation carries on with packet reception without further antenna switching. If the better antenna is the other antenna then the radio switches to that one, reacquires timing and carries on with packet reception on that antenna.

In antenna diversity, longer preambles are often used to provide the antenna diversity algorithm time to detect and evaluate the signal on each antenna to ensure that a true preamble is found. However, shorter preambles are preferred as they reduce MCU on-time and in turn reduce MCU current consumption. The RSSI measurement technique for evaluating signal quality requires less preamble time than other methods such as timing correlation.

Note that antenna Rx diversity is available for testing and evaluation purposes on the Gecko SDK suite. Due to the short preambles on the 802.15.4 packets, customers will need to make their own assessment on the performance and production readiness of this feature.

Rx and Tx antenna diversity are independent operations. In practice this means that, for example, Tx antenna diversity will begin the next transmission on the last successfully used antenna for Tx (for example antenna 1), even though in the intervening receive Rx antenna diversity found better signal quality on antenna 2.

2 Configuring Antenna Diversity

The antenna diversity configuration options available for EmberZNet and Silicon Labs Thread consist of selecting Rx and/or Tx antenna diversity and configuring the underlying peripherals correctly. To configure antenna diversity, you must be familiar with your device's overall antenna configuration, that is, if it uses a FEM/LNA for either Tx or Rx, and also be familiar with the device's pin layout. Check the datasheet for your device for these settings or contact Silicon Labs support if you have questions.

These instructions assume you have installed Simplicity Studio and either the EmberZNet or Silicon Labs Thread SDK (software development kit), and that you have a project open in the Simplicity IDE (integrated development environment).

The steps to set up antenna diversity are described in detail below. In summary:

1. On the plugins tab, disable the antenna stub plugin and enable the antenna plugin
2. Configure the antenna plugin.
3. If you are configuring Rx diversity either alone or with Tx diversity, finish the configuration in the Hardware Configurator.

2.1 Antenna Plugin Settings

2.1.1 Enable the Antenna Plugin (required)

1. In the Plugins tab, search for Antenna plugin, located in the HAL plugins group. The interface is essentially the same in EmberZNet and Silicon Labs Thread. By default, the Antenna Stub plugin is enabled.
2. Disable (uncheck) the Antenna stub plugin and enable (check) the Antenna plugin.

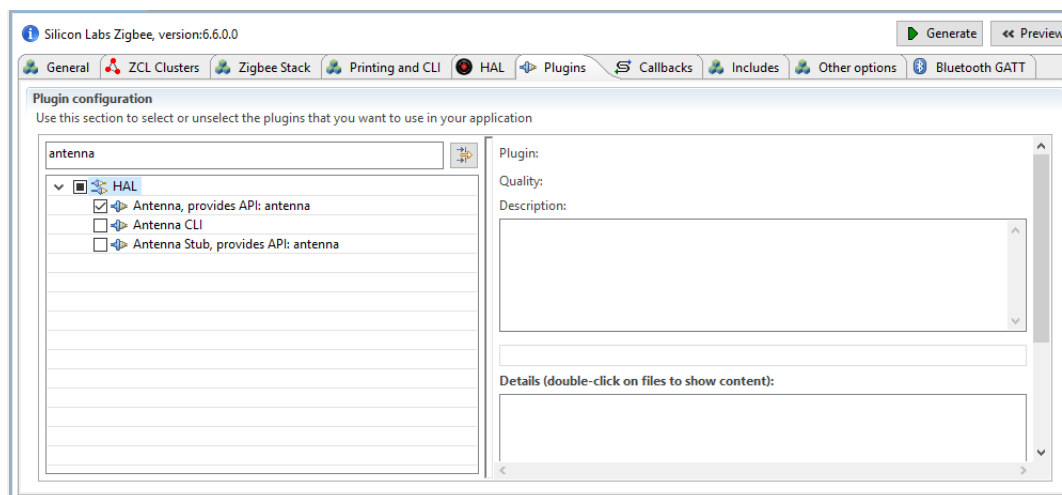


Figure 2-1. Antenna Plugin Enabled

When you enable the Antenna plugin, a warning is displayed. This is expected behavior. If you will be modifying configurations frequently, check **Do not show this anymore**. Click **OK**.

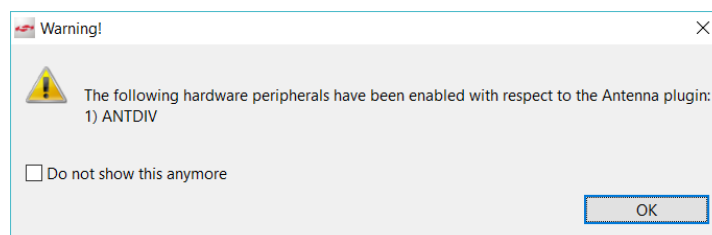


Figure 2-2. Antenna Diversity Peripherals Warning

3. Configure the Antenna Diversity Options for Tx and Rx using plugins, as described in the next sections.

2.1.2 Configure Tx Diversity in the Antenna Plugin

The default configuration for the antenna plugin is Tx Diversity enabled. An EmberZNet project is shown as an example. Note that some of these options are only available in Gecko SDK 2.6.0 and above.

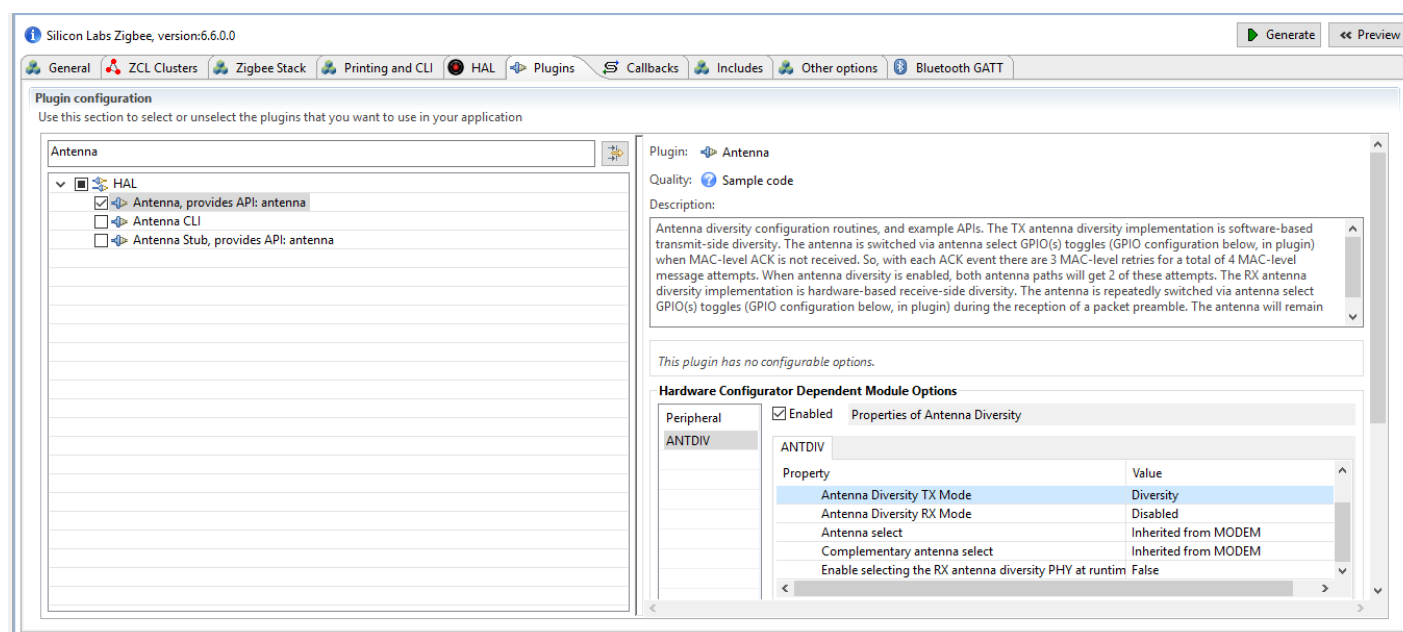


Figure 2-3. Antenna Diversity TX Mode

The options for **Antenna Diversity TX Mode** are:

- Disabled: Tx antenna diversity plugin does not control ANTENNA_SELECT_GPIO.
- Enable1: ANTENNA_SELECT_GPIO is set to high during Tx.
- Enable2: ANTENNA_SELECT_GPIO is set to low during Tx.
- Diversity: Tx antenna selection is dynamic and Tx diversity is enabled.

Antenna select: Pin used to control the external antenna switch if only Tx diversity is enabled. A pin can be set directly through this option, or it can inherit the setting from MODEM in Hardware Configurator (see section [2.2 Hardware Configurator](#)).

Complementary antenna select: Pin for the inverted external antenna signal. A pin can be set directly through this option, or it can inherit the setting from MODEM in Hardware Configurator.

2.1.3 Configure Rx Diversity in the Antenna Plugin

A Silicon Labs Thread project is shown as an example.

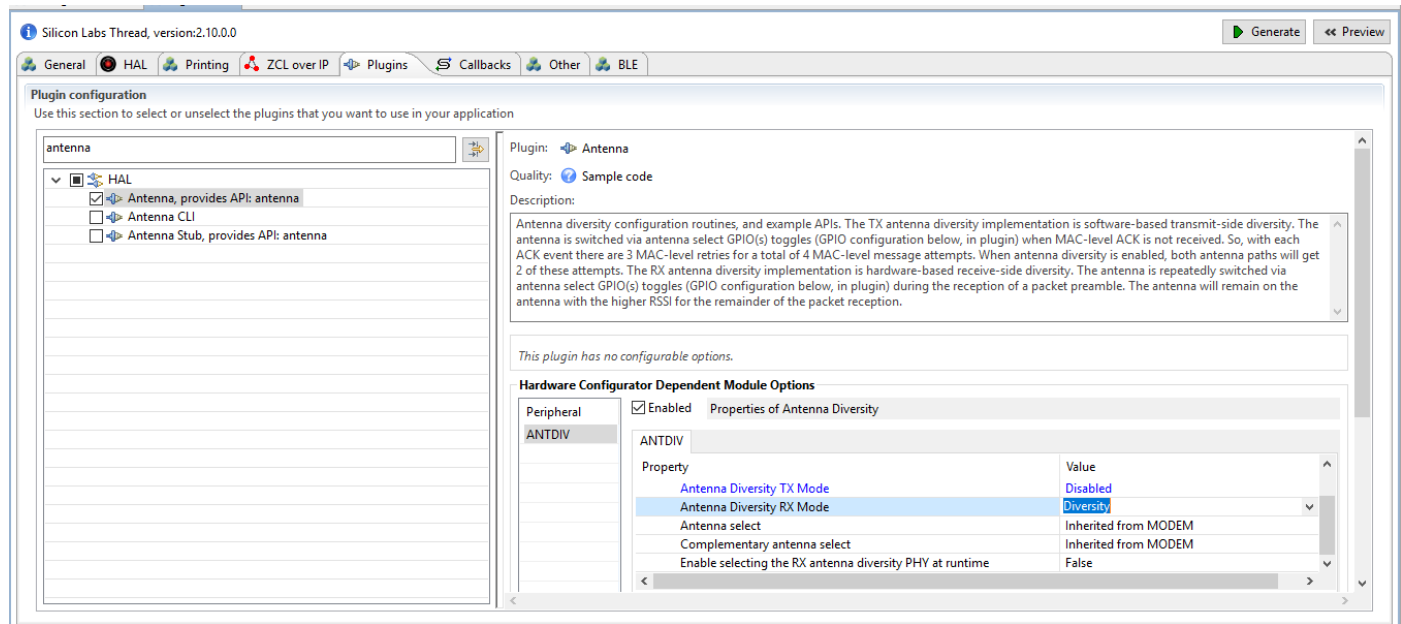


Figure 2-4. Antenna Diversity RX Mode

The options for **Antenna Diversity RX Mode** are:

- Disabled: Rx antenna diversity plugin does not control ANTENNA_SELECT_GPIO.
- Enable1: ANTENNA_SELECT_GPIO is set to high during Rx
- Enable2: ANTENNA_SELECT_GPIO is set to low during Rx
- Diversity: Rx antenna selection is dynamic and diversity is enabled.

Antenna Port pin selection: Selection of the antenna port pin must be done through the Modem peripheral in Hardware Configurator (see section [2.2 Hardware Configurator](#)). Note that this option changes after you pick an option for **Rx Diversity**. Select the **Diversity** option first before attempting to change the antenna select option.

Enable selecting the Rx antenna diversity PHY at runtime: On Gecko SDK 2.6.0 and above, there is an additional option to change diversity settings at runtime. When this option is set to True, both the standard and the diversity-enabled PHY are compiled into the image, thereby allowing the application to use API calls to change PHY (from standard to diversity or vice-versa) at runtime. Note that the code size of the image will increase significantly when this option is set to True.

2.1.4 Configure Rx and Tx Diversity in the Antenna Plugin

To configure both Rx and Tx antenna diversity, set them both in the Antenna plugin.

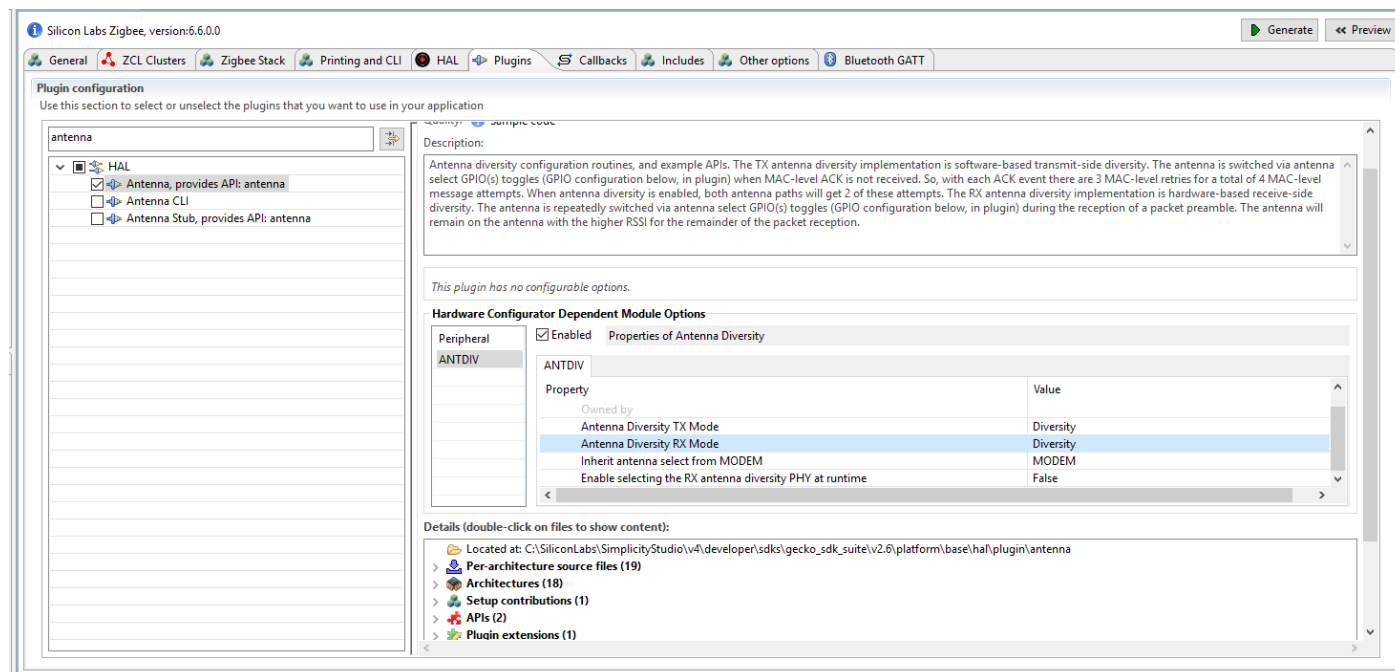


Figure 2-5. The Antenna Plugin Configured for both Tx and Rx Diversity

Because Rx diversity is enabled, pin configuration must be done in the Hardware Configurator, as described in the following section.

2.2 Hardware Configurator Settings

2.2.1 Configure the Antenna Pins

The **Antenna select** and **Complementary antenna select** pins can also be configured through Hardware Configurator for Tx diversity, and must be configured in Hardware Configurator for Rx diversity. To open the Hardware Configurator:

1. Select the **HAL Configuration (EmberZNet)/HAL** (Silicon Labs Thread) tab.
2. In the Hardware Configurator group, click **Open Hardware Configurator**.
3. Select the Default Mode Peripherals tab.
4. Select MODEM and the two parameters are displayed, configured as Disabled (or whatever pin was selected in the plugin).
5. Select the pins for **Antenna Select** and, if applicable, **Complementary antenna select**. The Antenna select signal goes high to select antenna 1 and low to select antenna 2.

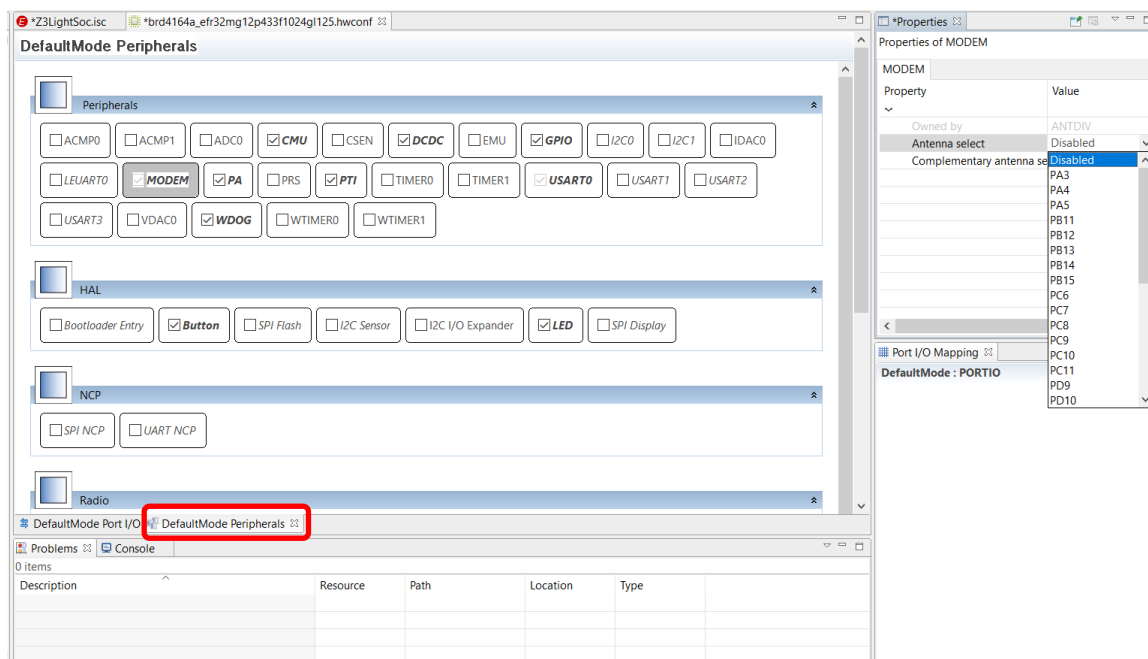


Figure 2-6. TX Antenna Diversity Pin Configuration in Hardware Configurator

2.2.2 Configure the FEM (optional)

If antenna diversity is implemented using a FEM, you must

1. Enable the FEM driver plugin on the Plugins tab in the HAL plugins group.
2. Configure the driver.

The driver can be configured in the Plugins tab, or in Hardware Configurator. The options are the same. These instructions are for the Hardware Configurator.

In the Radio group, select the **External FEM** peripheral.

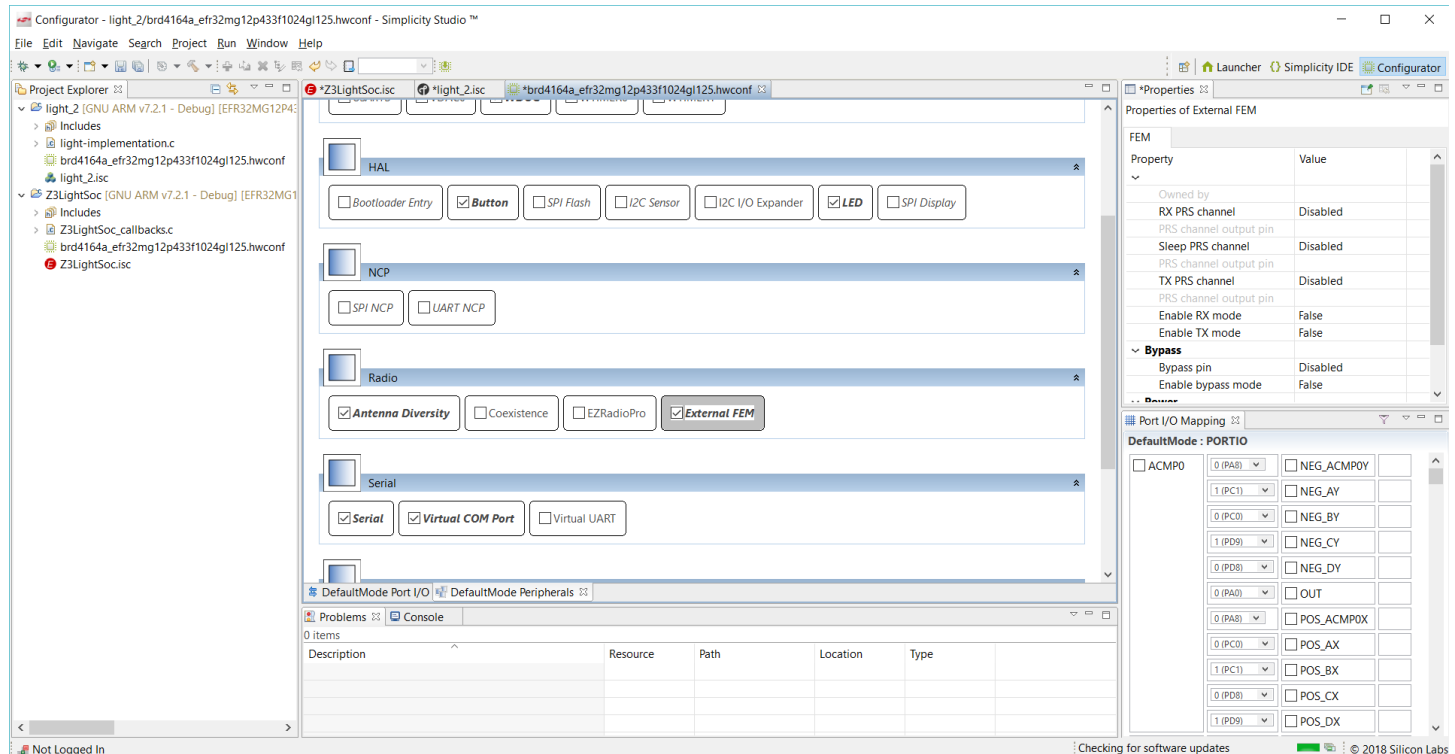


Figure 2-7. Hardware Configurator External FEM Peripheral

Not all FEM configuration options apply to all FEMs. Specifically **Bypass** and **Tx Power** only apply to FEMs with the pins to support the features, such as the SKY66112. Check your FEM datasheet for the settings required.

NOTE: Since there are a limited number of PRS channels, care must be taken so they do not conflict with channels that might be selected in other plugins.

The FEM configuration options are:

RX PRS channel: PRS Channel for Rx control (FEM pin CRX). If no Tx is defined, it is a dual-use (Rx/Tx) pin. The options are Disabled or a channel number. Once a channel is selected, the **PRS channel output pin** can be configured.

Sleep PRS channel: PRS channel for sleep control (FEM pin CSD). The options are Disabled or a channel number. If the setting is enabled, it must be configured to be the channel immediately following the RX PRS channel. If set incorrectly, it will result in a compile error. Once a channel is enabled, the **PRS channel output pin** can be configured.

TX PRS channel: PRS channel for Tx control (FEM pin CTX). The options are Disabled or a channel number. If the setting is disabled, then the software assumes that the FEM pin CRX is a dual use pin. Once a channel is enabled, the **PRS channel output pin** can be configured.

Enable RX mode: Configures RX mode on the FEM. The options are True or False. This option must be set to True and the Antenna Diversity Plugin enabled in order for the underlying antenna diversity radio configuration settings to be used. If it is set to false, then the standard radio configuration settings are used.

Enable TX mode: Configures TX mode on the FEM. The options are True or False.

Bypass (optional)

- **Bypass Pin:** If the FEM has a pin for the bypass signal (CPS) select it.
- **Enable bypass Mode:** Set to True to enable communication that bypasses the LNA (low noise amplifier).

Power (optional)

- **TX power pin:** If the FEM has a pin for Tx power mode (CHL) select it.
- **Enable high power tx:** Set to True to enable high power Tx, set to False to enable low power Tx.

For example, a FEM such as the Skyworks 66112 configured for Rx antenna diversity, with receive bypass pin configured but bypass mode not enabled, would look like the following figure:

Properties of External FEM		
FEM		
Property	Value	
Owned by		
RX PRS channel	CH3	
PRS channel 3 output	PD10	
Sleep PRS channel	CH4	
PRS channel 4 output	PD12	
TX PRS channel	Disabled	
PRS channel output p		
Enable RX mode	True	
Enable TX mode	False	
▼ Bypass		
Bypass pin	PB11	
Enable bypass mode	False	
▼ Power		
TX power pin	Disabled	
Enable high power tx	False	

Figure 2-8. FEM Configured for Rx Antenna Diversity

Note that options in blue have not yet been saved to the board header file.

2.2.3 Update the FEM Driver (applicable to older stacks only)

Project (ISC) files that use the legacy FEM control plugin (based on EmberZNet SDK version 6.1 or earlier or Silicon Labs Thread version 2.1 or earlier) do not update automatically to the new FEM plugin, which has merged the functionality of FEM control and LNA. The ISC file previously used plugin options to configure elements of its software platform, and this configuration is now done through Hardware Configurator. AppBuilder will attempt to migrate the relevant plugin options to Hardware Configurator. Verify that the property `ABPeripheral.included` on module FEM is set to `True`.

You must add the following to the `.hwconf` file based on your old plugin options. Note that if only TX location and channel are defined, it is changed to be RX.

- If only RX or TX channel and location are defined:
 - Set the `RX PRS channel` to the used channel.
 - Set the `PRS channel <rx or tx channel> output pin` to the used pin.
- If both RX and TX channel and location are defined:
 - Set RX properties as above, and do the same for TX.
- If `rxActive` or `txActive` is defined:
 - Set corresponding `Enable RX mode` or `Enable TX mode` to `True`.
- Select channel and output pin for SLEEP pin:
 - Set `Sleep PRS channel` and `PRS channel <sleep channel> output pin`.

2.3 Command Line Interface

Gecko SDK.2.6.0 and above provide a means to query and set Rx and Tx diversity modes using the Antenna CLI plugin.

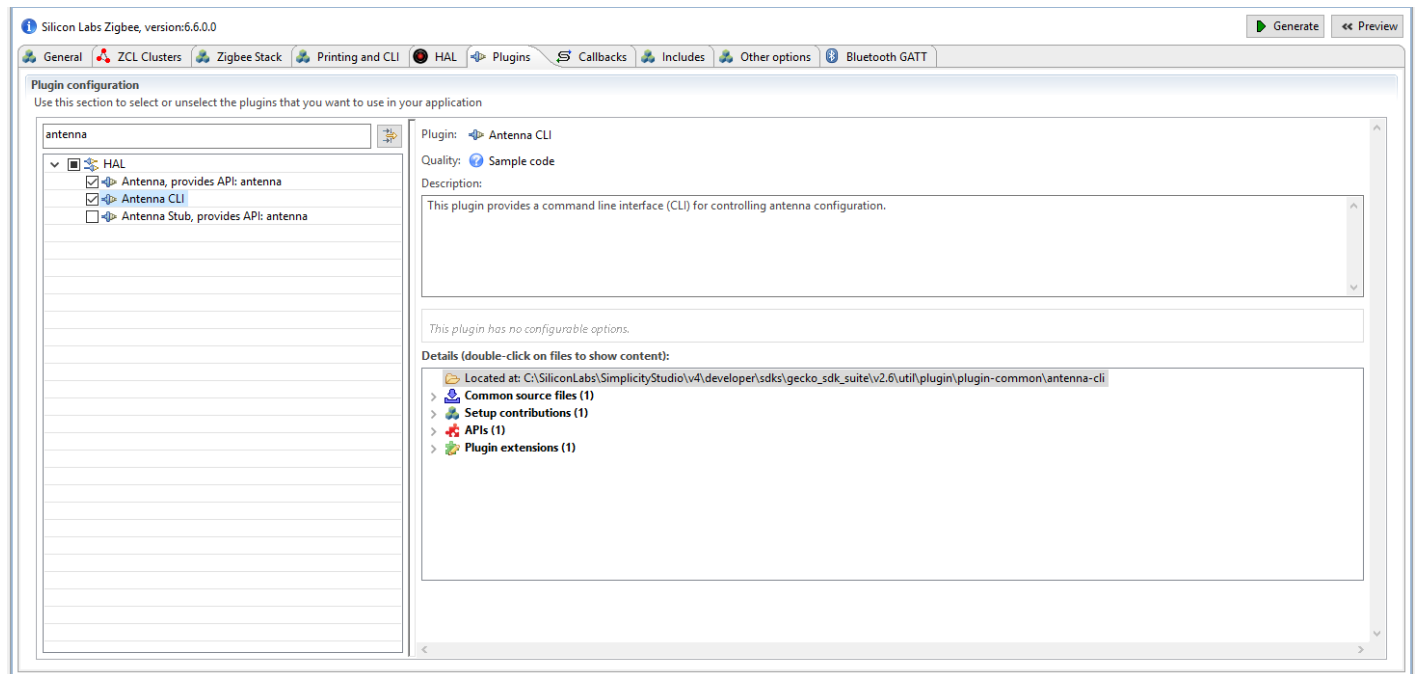


Figure 2-10. Antenna CLI Plugin

Command	Command Description	API Function	Arguments		
			Name	Type	Description
plugin antenna get-antenna-tx-mode	Returns the current setting for the antenna tx diversity mode	emberAfPluginAntennaGetAntennaTxMode	n/a	n/a	Interpretation of returned results for get operation and permissible values for set operation are as below: HAL_ANTENNA_MODE_DISABLED 0 HAL_ANTENNA_MODE_ENABLE1 1 HAL_ANTENNA_MODE_ENABLE2 2 HAL_ANTENNA_MODE_DIVERSITY 3
plugin antenna set-antenna-tx-mode	Sets tx diversity mode to argument	emberAfPluginAntennaSetAntennaTxMode	Tx Antenna Mode	uint8_t	
plugin antenna get-antenna-rx-mode	Returns the current setting for antenna rx diversity mode	emberAfPluginAntennaGetAntennaRxMode	n/a	n/a	
plugin antenna set-antenna-rx-mode	Sets rx diversity mode to argument	emberAfPluginAntennaSetAntennaRxMode	Rx Antenna Mode	uint8_t	

Table 2-1. Antenna Diversity CLI description

Tx diversity settings can be changed using the CLI without any restriction. However, Rx Diversity options that require switching from standard PHY to diversity PHY or vice-versa may only be permitted when the plugin option **Enable selecting RX antenna diversity PHY at runtime** is set to True.

Silicon Labs

Simplicity Studio™4



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



IoT Portfolio

www.silabs.com/IoT



SW/HW

www.silabs.com/simplicity



Quality

www.silabs.com/quality



Support and Community

community.silabs.com

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, ClockBuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



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

<http://www.silabs.com>