AN1321: Configuring Peripherals for 32 Bit Devices with Zigbee EmberZNet 7.0 and Higher

Peripherals for Zigbee devices, running applications built with EmberZNet SDK 7.0 and higher, are configured using the Pin Tool in Simplicity Studio® 5. The Pin Tool simplifies peripheral configuration by presenting peripherals and peripheral properties in a graphical user interface. For some SDKs, many peripherals can also be configured in the Simplicity IDE as component options.

If you are developing with the EmberZNet SDK 6.10.x and lower, see *AN1115: Configuring Peripherals for 32-Bit Devices using Hardware Configurator*.

KEY POINTS

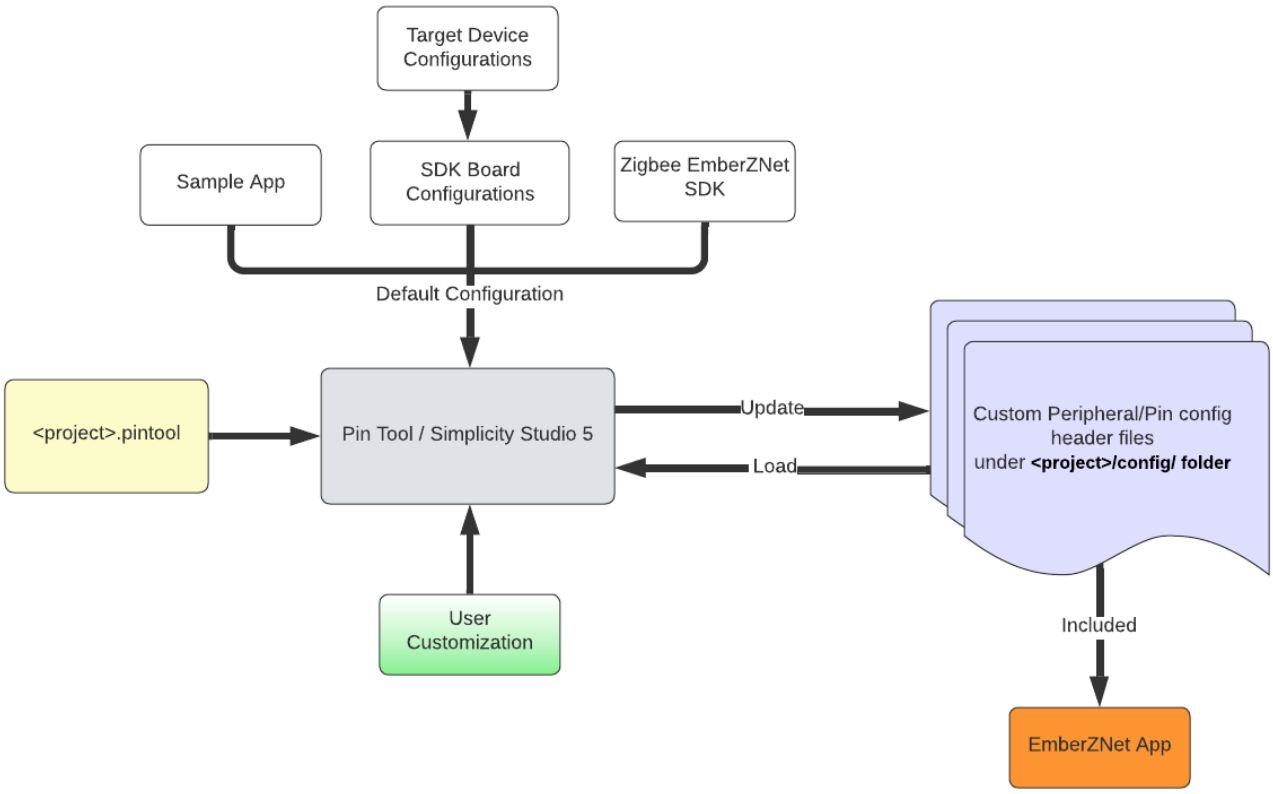
- Introducing peripheral configuration
- Using the Pin Tool in Simplicity Studio
- Pin Tool functions

1 Introduction

Pin Tool is an advanced graphical editor that allows developers to easily configure the peripherals on their Zigbee system. It provides three configuration perspectives to let the developers intuitively map physical pins and peripheral instances to software components on the target device.

Pin Tool editor is also flexible enough to be used in different development flows. The bottom-up approach lets developers start configuration with pins and connect them to functions/peripherals and then software components. However, an opposite but equally effective top-down approach lets developers starts with software component selections for peripherals and work down to peripheral functions and pins when required.

When a Zigbee application project is first created, an initial set of header files are provided to the new project based on the target board's configurations, EmberZNet SDK version, and so on, as shown in the following figure. Any subsequent customization of the peripherals can be made through the Pin Tool. Developers using EmberZet can also modify hardware options through the Component Editor. All customizations and changes through Pin Tool are updated to the configuration header files which are included by the application.



The C header files with hardware-specific configurations are used and monitored by the Pin Tool. These files can be found in the following project directory. The hardware-specific configurations are stored in the **Pin Config** section of the generated C header files.

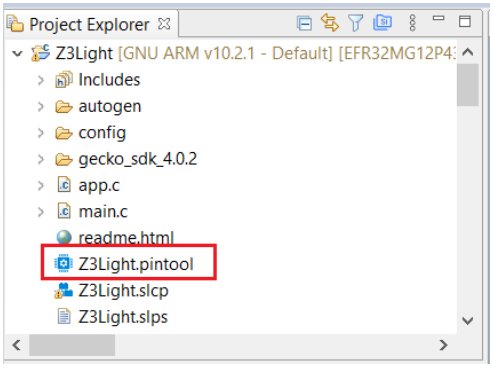
```
<workspace>/<project>/config/
```

2 Using the Pin Tool

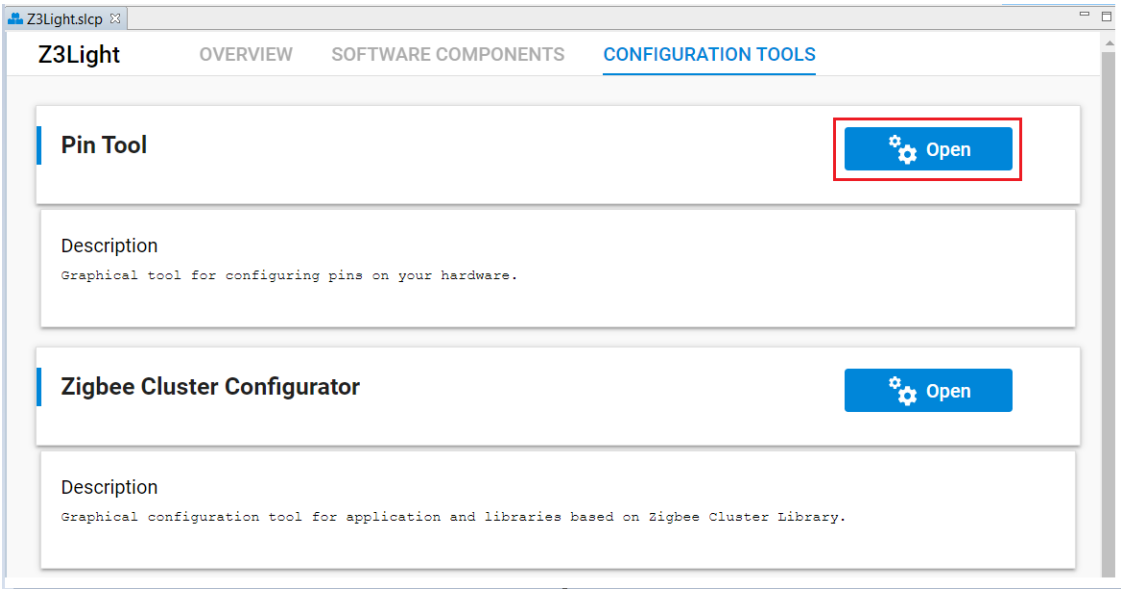
This chapter discusses the basic operation and functions of the Pin Tool. Before proceeding to the next section, it may be helpful to understand the GPIO functionality and peripheral signal routing controls of the target devices by reviewing *AN0012: General Purpose Input Output*, device datasheets, and reference manuals.

2.1 Opening the Pin Tool in Simplicity Studio

Open Pin Tool directly by double-clicking on the *<project>.pintool* file in the Project Explorer, as shown in the following figure.

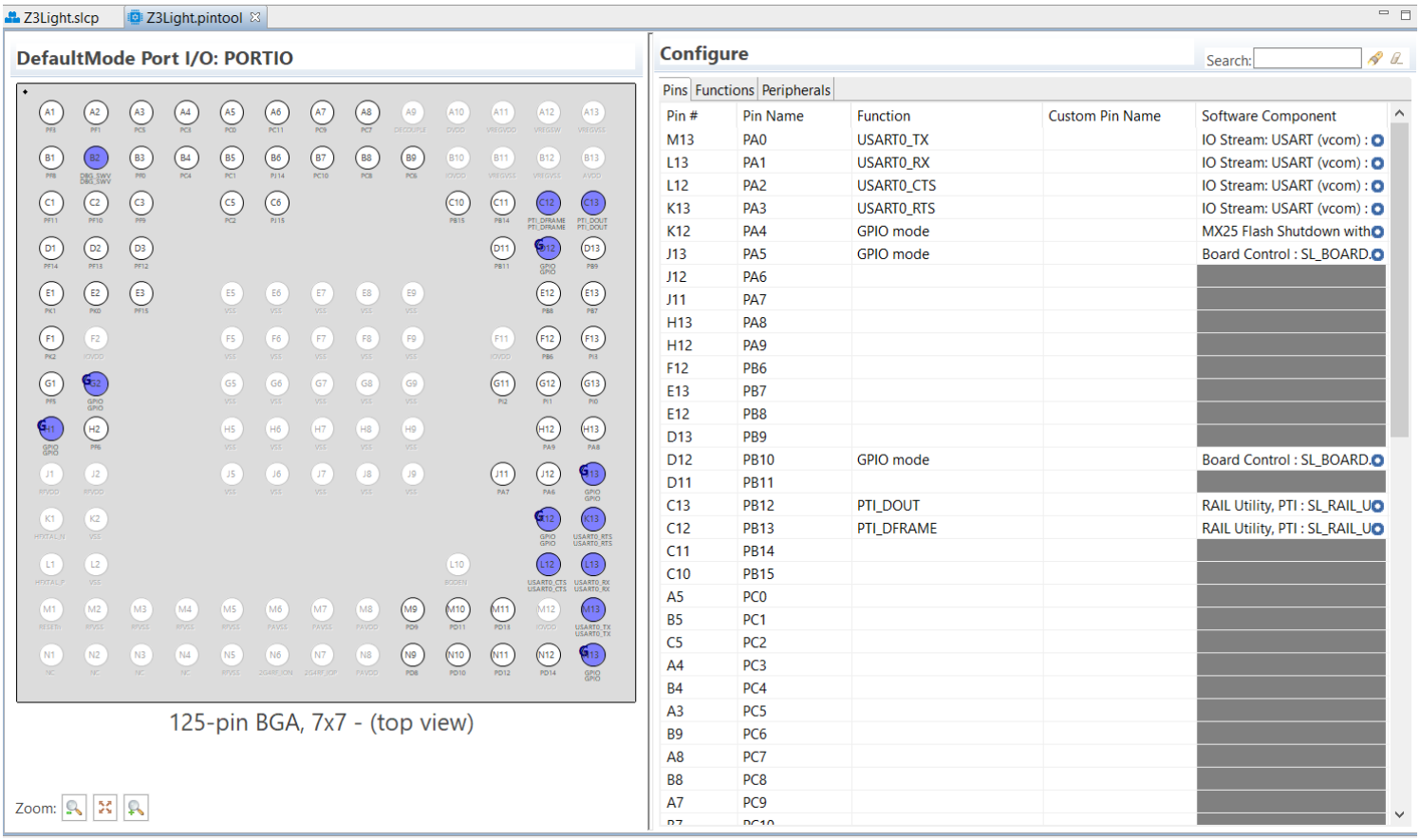


Pin Tool may also be started from the Project Configurator's CONFIGURATION TOOLS tab.



2.2 Pin Tool Functions

The following figure shows the Pin Tool editor window once it is open. The left “Port I/O” pane shows the device package’s Port I/O view. The right “Configure” has three tabs –Pin, Functions, and Peripherals. Each of these tabs gives a different detailed perspective with which to configure the hardware.

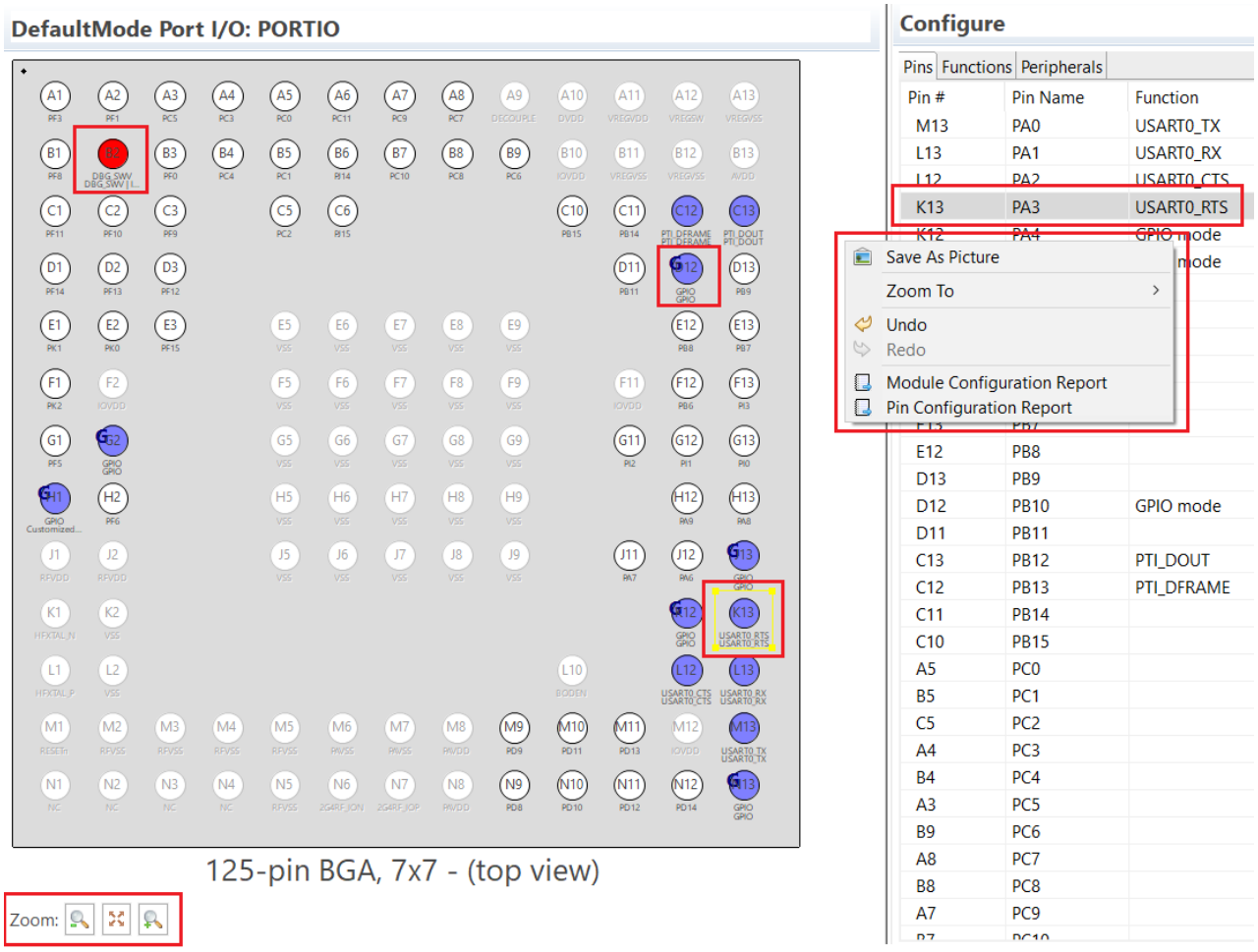


2.2.1 Port I/O Pane

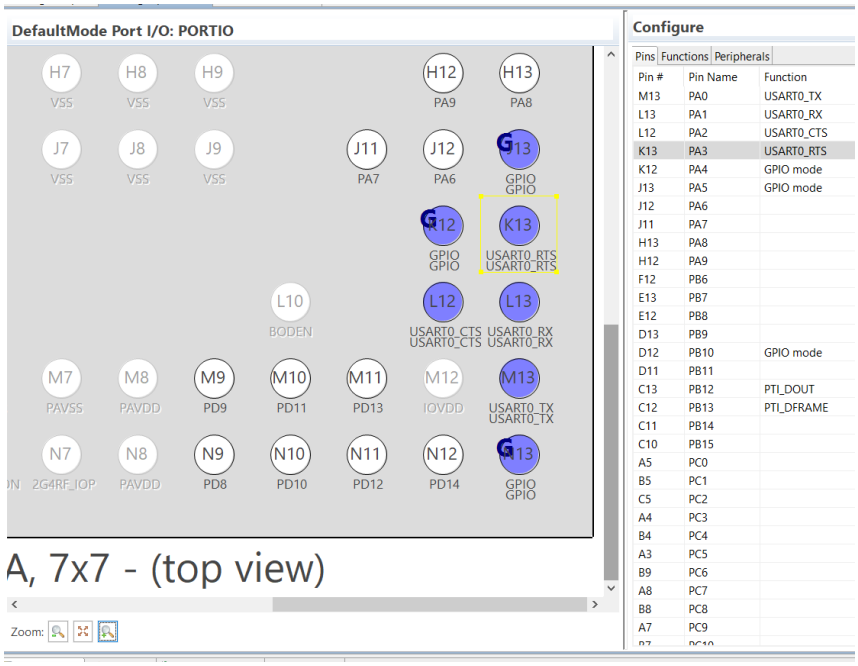
The Port I/O Pane is essentially a Pinout diagram that displays the physical pin locations on the target device package.

The Pinout diagram has the following color coding:

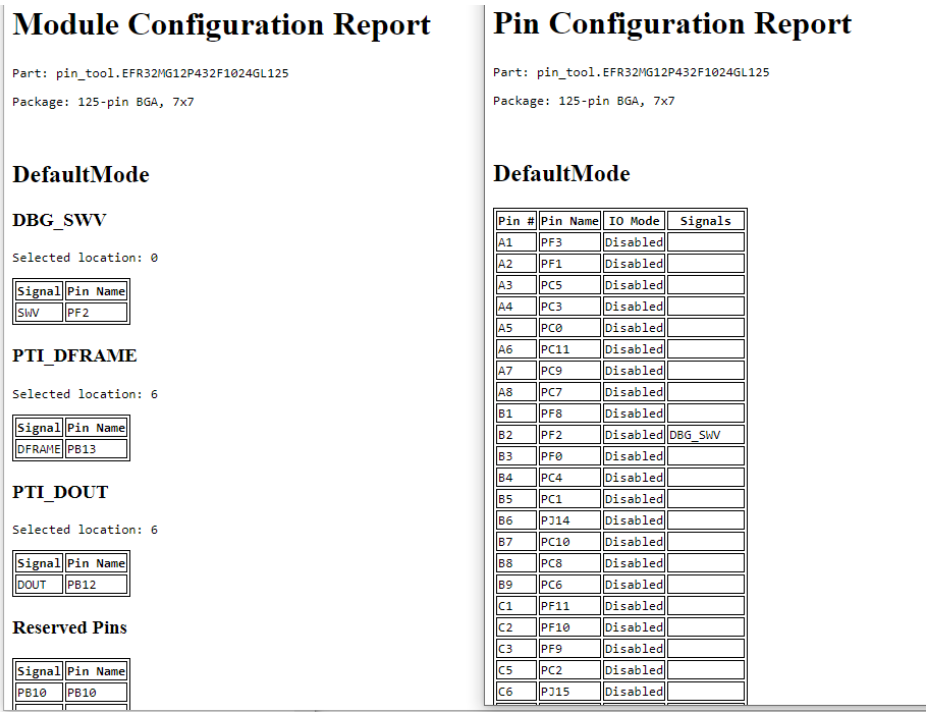
- Pins in blue are in use
- Pins in white are unused.
- Pins in red show **unallowed** conflicts with two or signals going into the same pin.
- Pins in orange show **allowed** conflicts with two or more signals going into the same pin (not shown in the diagram).
- Pins, such as E5/Vss, are greyed out because they are unavailable for configuration
- All pins configured for GPIO modes are marked by the boldfaced letter **G**.
- When one or more pins are selected in the Configure panel (e.g., K13), the corresponding pins are highlighted in yellow.



The Zoom controls at the lower left corner of the Port I/O pane provide a convenient way to zoom in to a specific location on the Pinout diagram to see more detailed information of a given pin.



A printable report can be generated by right-clicking the pinout diagram and selecting **Pin Configuration Report**. This opens a report as a webpage in a browser that can be saved, printed, or archived. The **Module Configuration Report** option generates a similar set of tables organized by module rather than by pin order.



Module Configuration Report

Part: pin_tool.EFR32MG12P432F1024GL125

Package: 125-pin BGA, 7x7

DefaultMode

DBG_SWV

Selected location: 0

Signal	Pin Name
SWV	PF2

PTI_DFRAME

Selected location: 6

Signal	Pin Name
DFRAME	PB13

PTI_DOUT

Selected location: 6

Signal	Pin Name
DOUT	PB12

Reserved Pins

Signal	Pin Name
PB10	PB10

Pin Configuration Report

Part: pin_tool.EFR32MG12P432F1024GL125

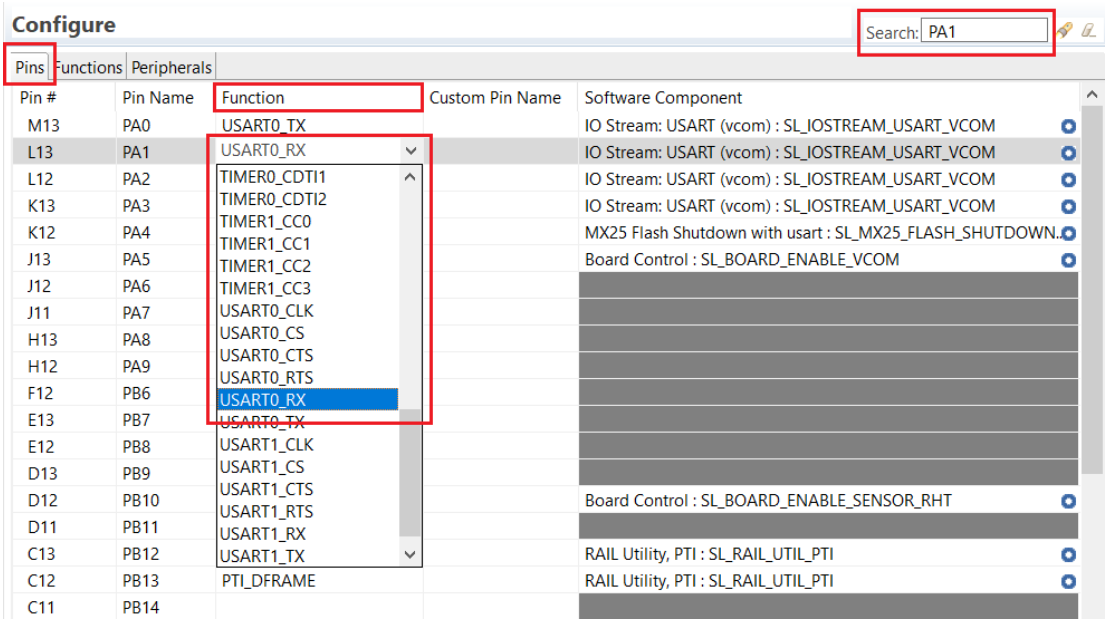
Package: 125-pin BGA, 7x7

DefaultMode

Pin #	Pin Name	IO Mode	Signals
A1	PF3	Disabled	
A2	PF1	Disabled	
A3	PC5	Disabled	
A4	PC3	Disabled	
A5	PC0	Disabled	
A6	PC11	Disabled	
A7	PC9	Disabled	
A8	PC7	Disabled	
B1	PF8	Disabled	
B2	PF2	Disabled	DBG_SWV
B3	PF0	Disabled	
B4	PC4	Disabled	
B5	PC1	Disabled	
B6	PJ14	Disabled	
B7	PC10	Disabled	
B8	PC8	Disabled	
B9	PC6	Disabled	
C1	PF11	Disabled	
C2	PF10	Disabled	
C3	PF9	Disabled	
C5	PC2	Disabled	
C6	PJ15	Disabled	

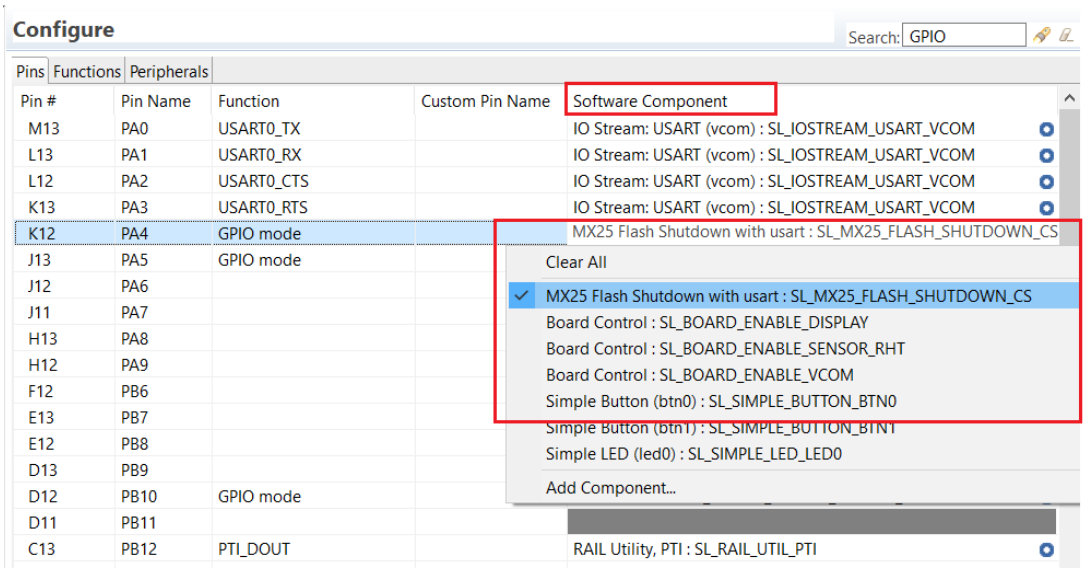
2.2.2 Pins Tab

The Pins tab gives a **pin-centric** table view of the device, similar to the datasheet's GPIO Functionality Table. The Pins table lets the user assign any valid alternate function to a pin, as shown in the following drop-down menu under the Function column.

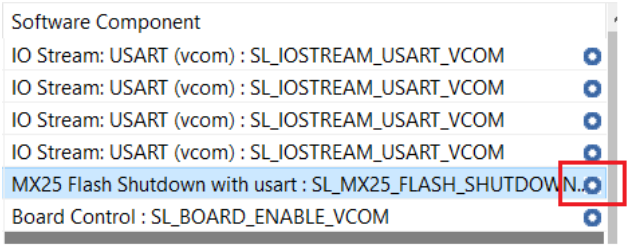


The Search box shown in the above figure allows user to quickly locate a Pin in a table.

Once a pin and the function have been chosen, the software component can be selected from the Software Component drop-down menu for the pin. The following figure shows the pin PA4 has been configured for GPIO mode and assigned to the software component **MX25 Flash Shutdown with usart**. Alternatively, the user can assign the pin through the Component Editor.



As a convenience, the user can open the Component Editor for a given component by double-clicking the blue circle in the "Software Component" cell as shown below.

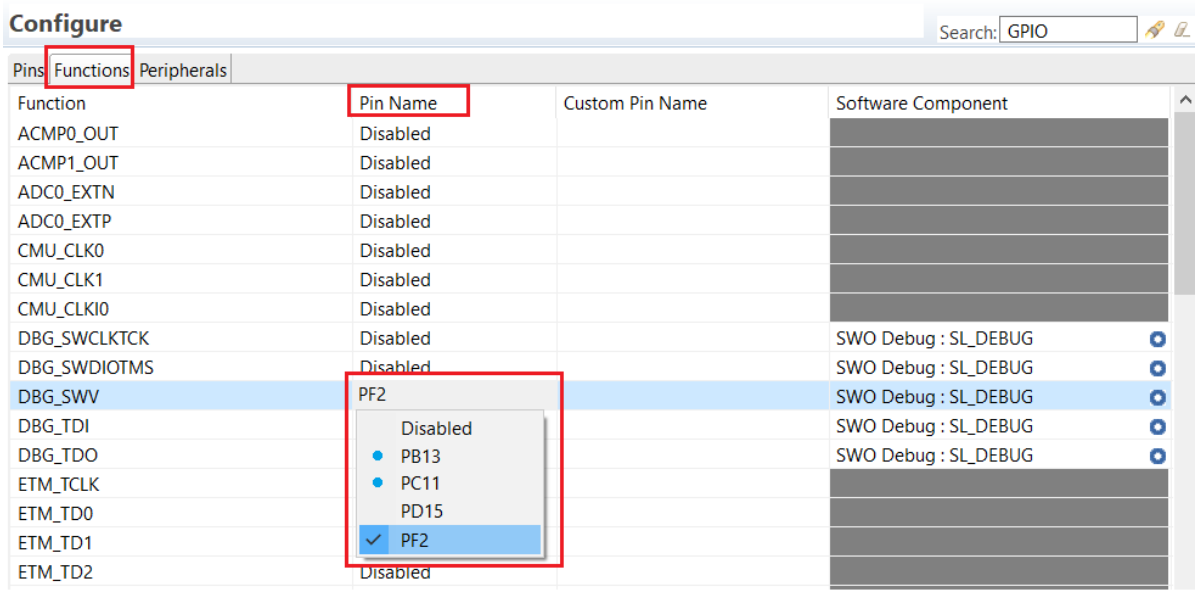


The "Custom Pin Name" column allows users to enter the custom pin name for a given pin.

2.2.3 Functions Tab

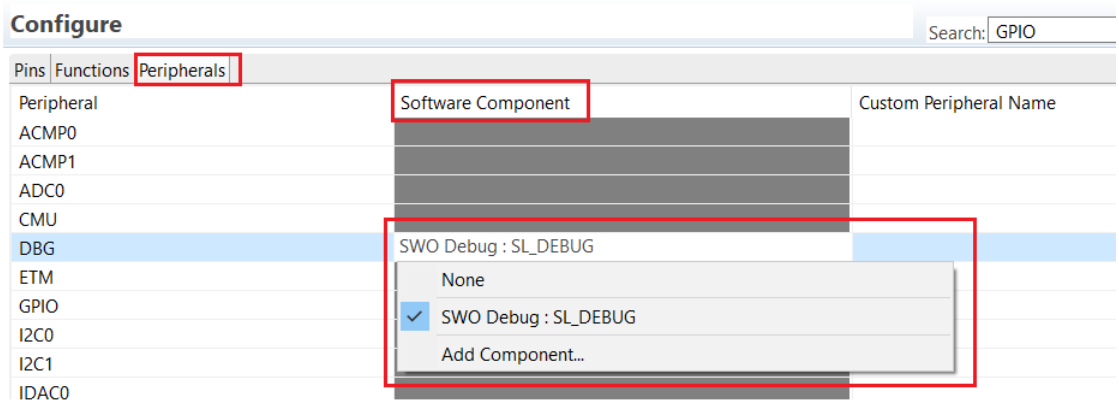
The Functions tab provides an alternate **function-centric** view of the device, similar to the datasheet's Alternate Functionality Table. The Functions tab lets the user assign available pins to an alternate function.

The valid pin for a specific alternate function can be selected from the drop-down menu in the "Pin Name" column. The blue dot for a pin in the same drop-down menu indicates the pin is already in use. The Component Editor can be opened for the entries in the "Software Component" column.

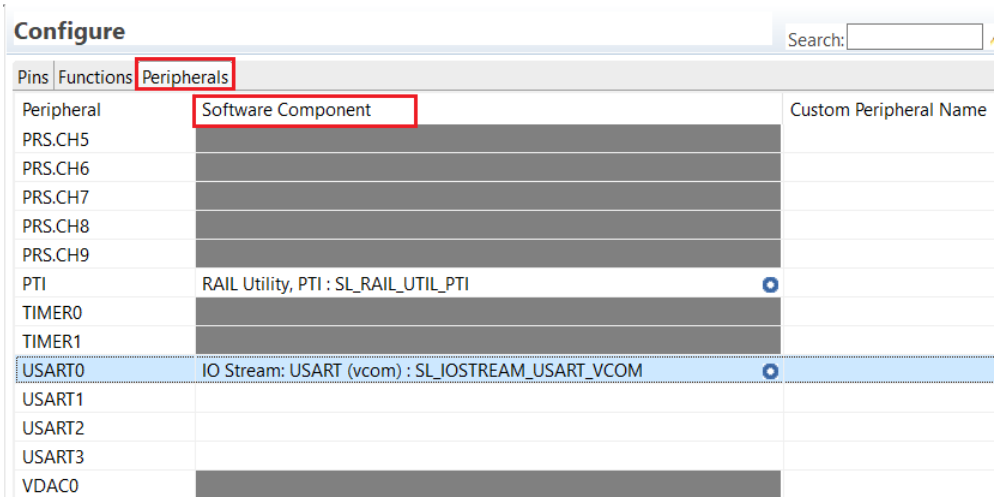


2.2.4 Peripherals Tab

The Peripherals tab shows a list of the peripherals on the device and their mapping to software components. The drop-down menu allows the user to select an available software component for a specific peripheral, as shown in the following figure.



The Software Component cell for a peripheral appears grey when no software component that uses the peripheral exists, and white when one exists but has not been assigned. The user can also provide a custom name for a given peripheral in the “Custom Peripheral Name” column.



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 & Community
www.silabs.com/community

SILICON LABS

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

www.silabs.com