

ncp-uart-hw: Configuring and building it in Simplicity Studio

i For detailed information about the special build and configuration of NCP-UART-HW for Yellow, please refer to its dedicated page [here](#).

Introduction

The **ncp-uart-hw** project by Silicon Labs is designed to enable communication with Zigbee Network Co-Processor (NCP) devices using UART hardware interfaces. Silicon Labs does not provide pre-compiled binaries, so users must compile and deploy the firmware themselves. This guide provides detailed instructions for configuring, compiling, and deploying the NCP firmware using UART hardware, tailored for integration with the Hemla project. By following these steps, you will set up a functional UART-based interface for your Zigbee NCP devices, ensuring reliable communication and control.

Prerequisites

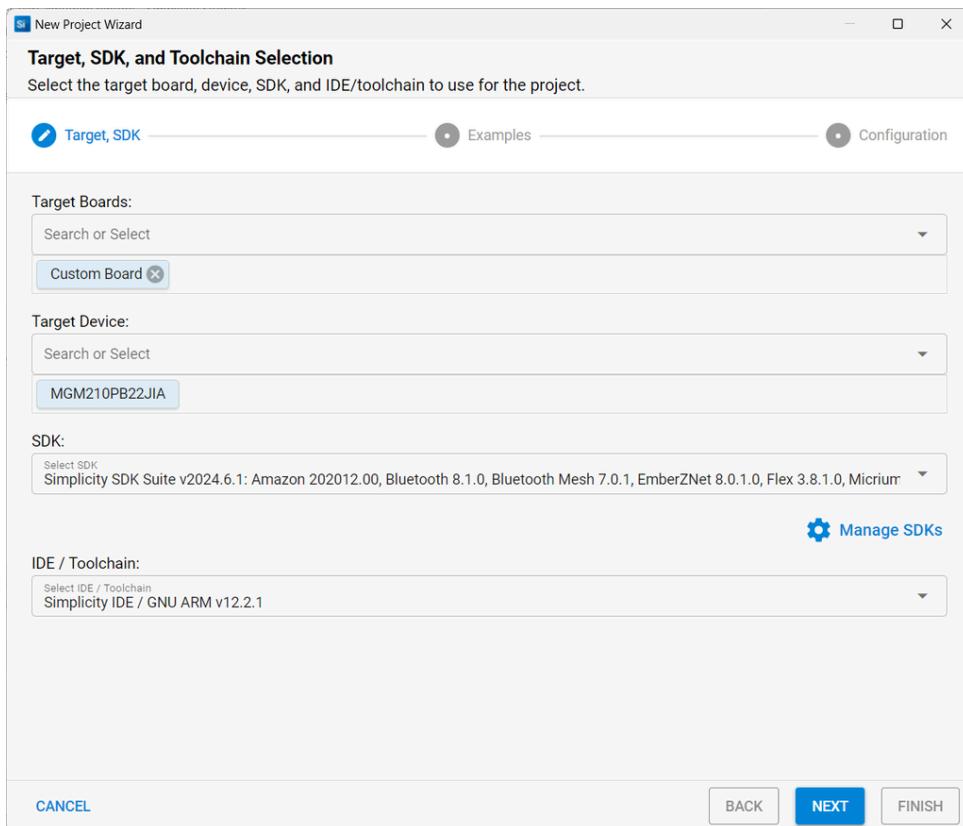
Before proceeding with the configuration, ensure that the following tools and hardware are available:

- **Simplicity Studio:** Version 5.9.2.1
- **Simplicity SDK Suite v2024.6.1:** Amazon 202012.00, Bluetooth 8.1.0, Bluetooth Mesh 7.0.1, EmberZNet 8.0.1.0, Flex 3.8.1.0, Micrium OS Kernel 5.16.00, OpenThread 2.5.1.0 (GitHub-1fce225b), Platform 5.0.1.0, Sidewalk 2.2.0, Silicon Labs Matter 2.3.1-1.3, USB 1.3.1.0, Wi-Fi SDK 3.3.2, Wi-SUN 2.1.0.0, Z-Wave SDK 7.22.1.0
- **UART Interface** (e.g., USB-to-UART converter)
- **Zigbee 3.0 device:** A Zigbee device you can connect on your coordinator
- **Silicon Labs Microcontroller:** MGM210PB221JIA

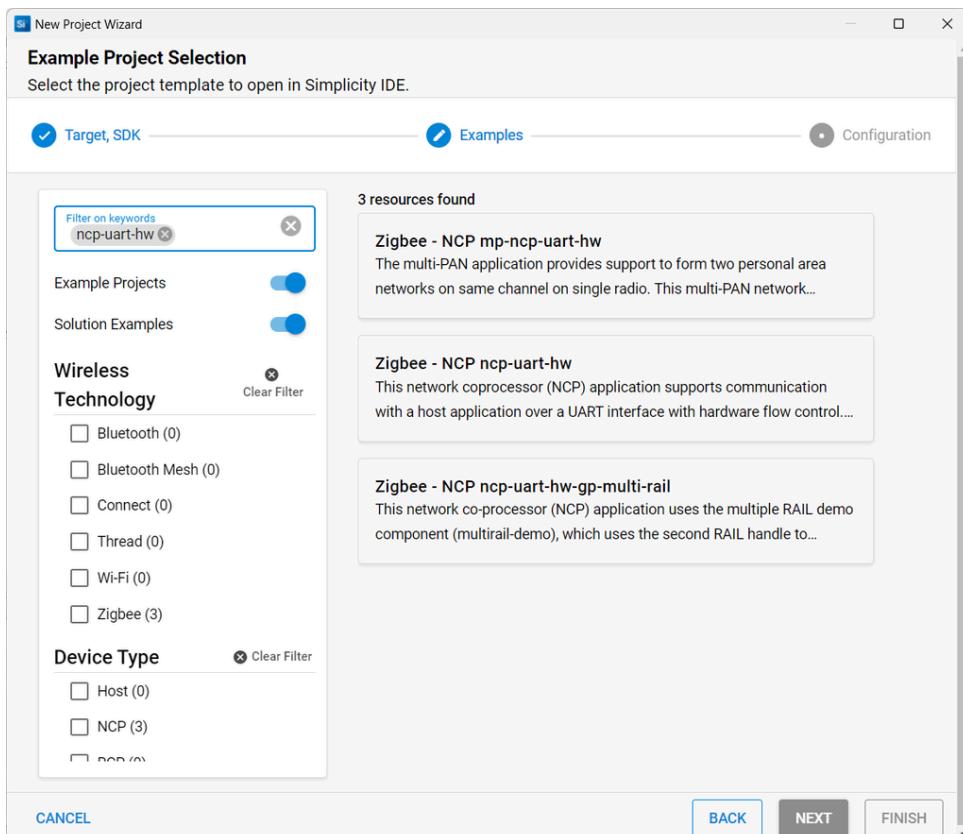
Create a ncp-uart-hw Project

1. Create a New Project:

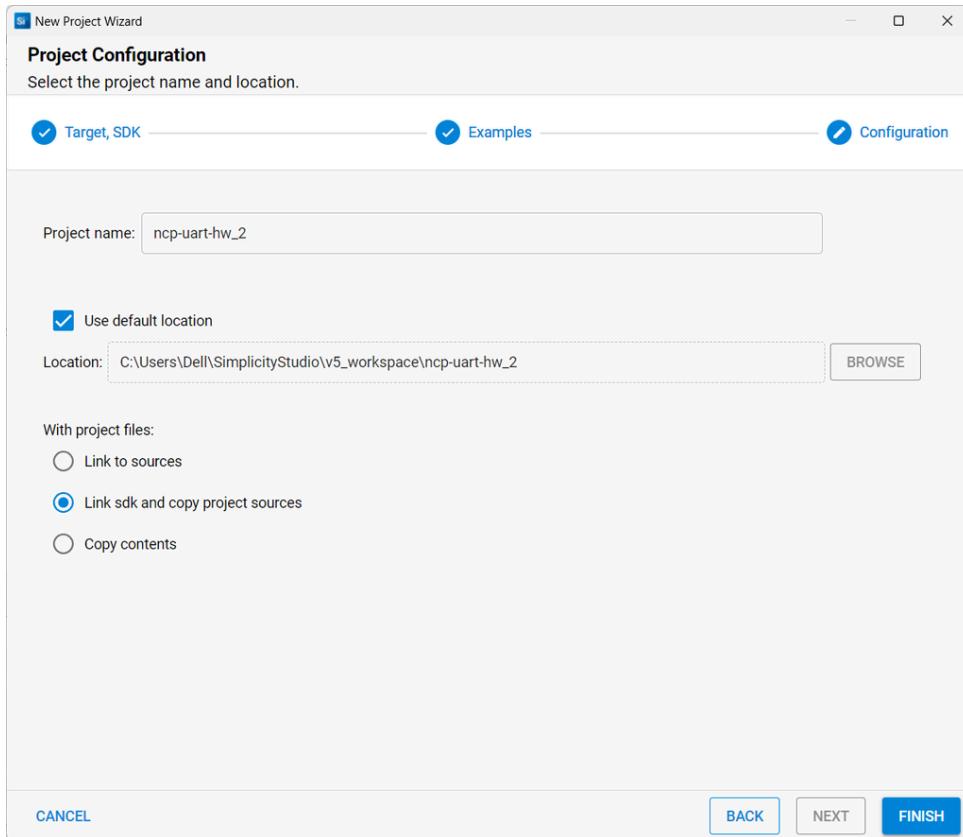
- Navigate to **File** → **New** → **Silicon Labs Project Wizard**.
- A screen like this will appear, be sure the fields are populated as in this image, then click **Next**:



- A new screen will appear, on the search box type “ncp-uart-hw”, and a project called “Zigbee - NCP ncp-uart-hw” should be visible, select it and click **Next**.



- A new screen will appear, leave the defaults, click **Finish**:



2. Configure the new project:

- On the Project Explorer you should see your recently created project:



- Click on the one that ends with slcp → Software Components → Search for “IO Stream” → Select it → Create new Component → vcom.
- Click on the one that ends with pintool, this will allow you to configure the pinout for the microcontroller. The pinout should be like this:

Pin #	Pin Name	Function	Software Component
5	PA01	GPIO_SWCLK	Debug : SL_DEBUG
6	PA02	GPIO_SWDIO	Debug : SL_DEBUG

7	PA03	GPIO_SWV	Debug : SL_DEBUG
9	PA05	USART0_RX	
10	PA06	USART0_TX	
22	PC01	USART0_CTS	
19	PD00	PTI_DOUT	RAIL Utility, PTI : SL_RAIL_UTIL_PTI
17	PD02	USART0_RTS	

Compile the project

Once the previous configuration is done, we can proceed to compile the project. Simply click the button with a hammer  and the process will start. Terminal should end with a blue message saying:

“Build finished. 0 errors, 0 warnings.”

The binaries generated can be found on the folder called “GNU ARM v12.2.1 - Default”.

Troubleshooting

Sometimes, it seems some packages are not available in Simplicity Studio, that is because your installation is corrupted (it failed to install everything due to unknown reasons), but it is not able to notify when this happened. The best solution is to uninstall and install it again, attached to a reliable internet connection. This will increase the probability of success to have a proper installation of Simplicity Studio.

There are two warnings that appear when compiling the project. These warnings can be [safely ignored](#).