MCM Standalone Project Guide

08/19/2024 | Document Version: 1.0

Table of Contents

Revision History

Date	Version	Author	Description
08-19-24	1.0	Oxit LLC	Initial Draft

1. Introduction

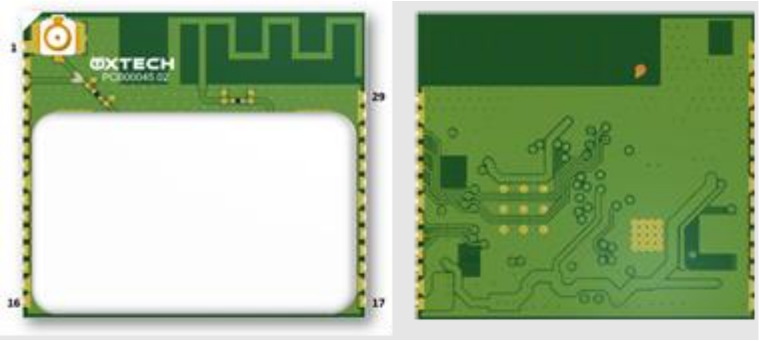
The OxTech MCM (Multi Connectivity Module) is a versatile hardware platform equipped with modern firmware for LoRaWAN and Sidewalk, IoT (Internet of Things) connectivity solutions.

The module embeds an EFR32MG24x series microcontroller, with advanced security characteristics and low power consumption and Semtech’s Sx126x LoRa transceiver for Sub-GHz conntivity.

2. OxTech MCM Module

The OxTech MCM is a 29-pin SMD module with a formfactor of 25.4 mm x 22.8 mm x 4.0 mm.

2.1.OxTech MCM (Multi Connectivity Module) PCB Appearance



2.2.MCM Pins

Pin No	Terminal Name	Type	Description
1	GND	Power	Ground
2	GND	Power	Ground
3	ANT	RF	LoRa RF path for application board antenna
4	GND	Power	Ground
5	JTAG-SWO	O	Debug Serial Wire Output
6	JTAG-SWIO	I/O	Debug Serial Wire I/O
7	JTAG-SWCLK	CLK	Debug Serial Wire Clock
8	RESET	I	Sidewalk Application MCU Reset

10	UART TX	O	UART TX, Module to host
11	UART RX	I	UART RX, Host to Module
12	COMMAND	I	Module Command input
13	EVENT	O	Module Event Out
14	BUSY	O	Module Busy Out
15	GND	Power	Ground
16	GND	Power	Ground
17	VDD	POWER	Module VDD Supply
18	GND	Power	Ground
19	EXT_GPIO4	I/O	General Purpose Input/Output
20	EXT_GPIO1	I/O	General Purpose Input/Output
21	EXT_GPIO2	I/O	General Purpose Input/Output
22	EXT_GPIO3	I/O	General Purpose Input/Output
23	GND	Power	Ground
24	FLASH_SCLK	CLK	Module/eFlash SPI Clock
25	FLASH_MOSI	I	Module/eFlash SPI Master Out Slave In
26	FLASH_MISO	O	Module/eFlash SPI Master In Slave Out
27	FLASH_CS	I	Module/eFlash SPI Chip Select
28	GND	Power	Ground
29	GND	Power	Ground

3. OxTech MCM EVK

The OxTech MCM EVK is an evaluation kit designed for easily interfacing with the MCM module and developing IoT (Amazon sidewalk and LoRaWAN) applications with an external application MCU (Host).

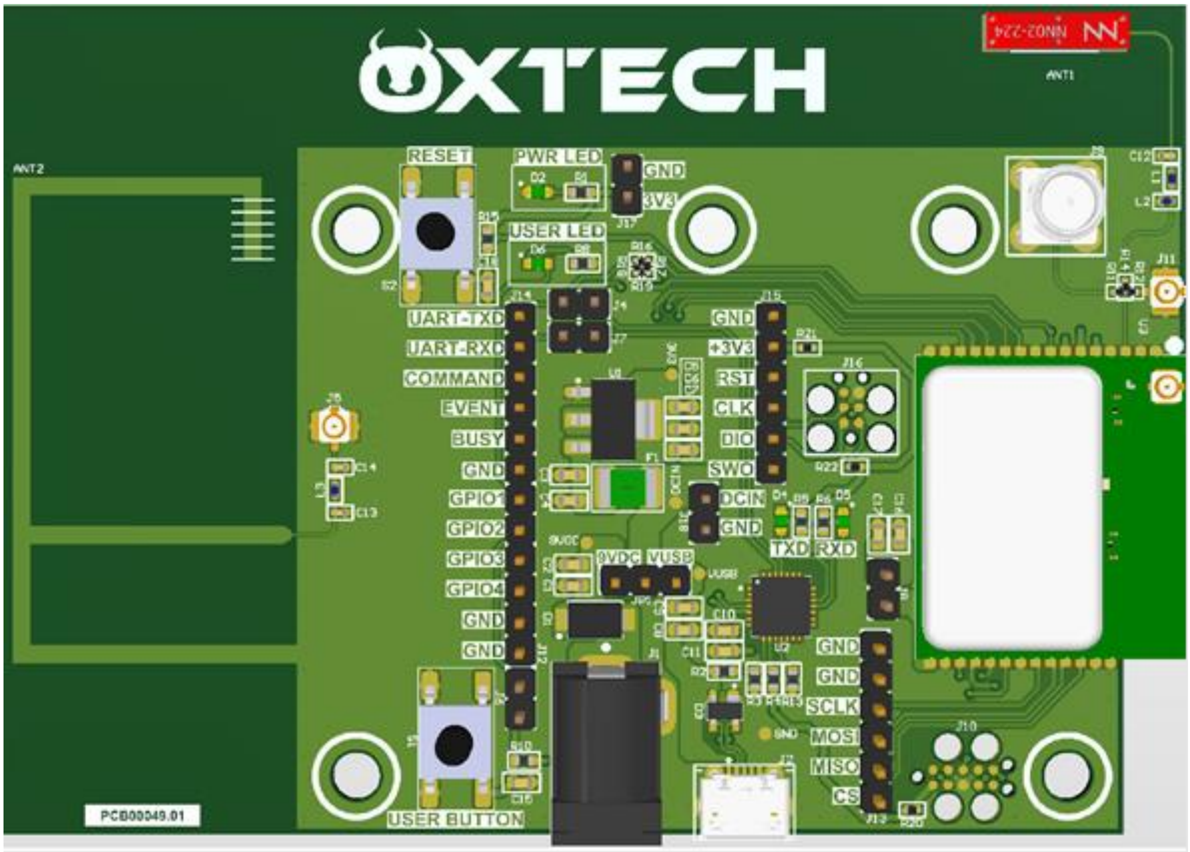


Figure 2: PCB Top View

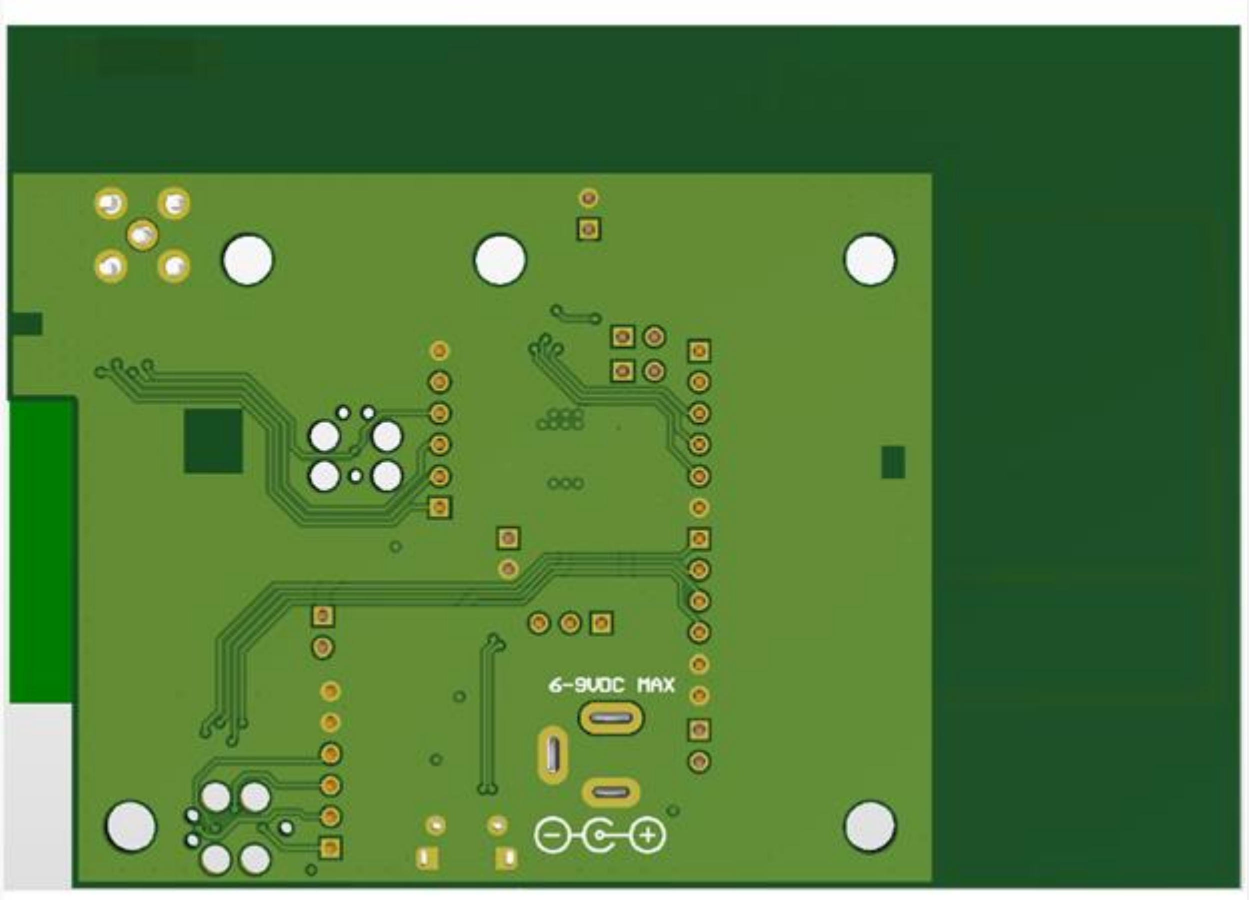
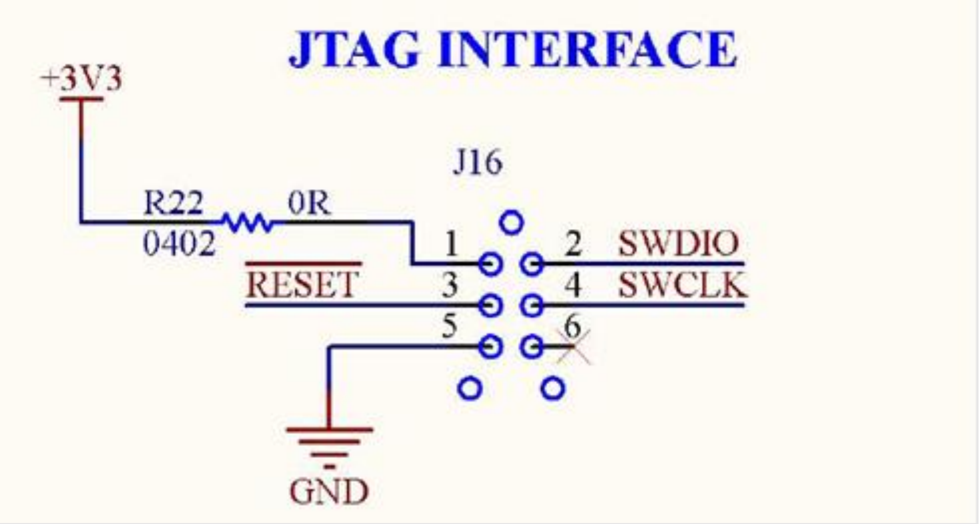


Figure 3 : PCB Bottom View

3.1.EVK's J-Link Connection

J-Link can be connected to the module through J16. This is only required when updating the MCM firmware via EVK. (Used in Standalone mode)



3.2.Module Power up via EVK

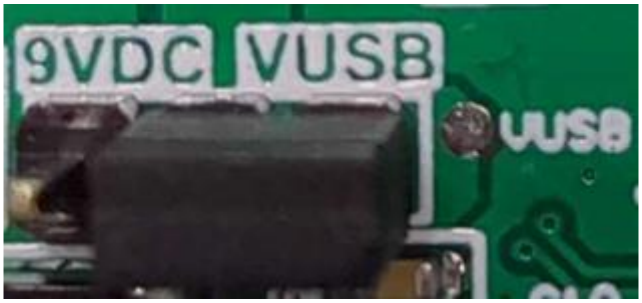
The EVK board has a USB Micro-B port and a barrel jack connector. The board can be powered up via either one of these.

Below are the necessary connections to power up the module:

- Connect a jumper to J8 (Enabling the power to MCM Module)
- Connect a jumper to J4 (For USB-UART connection)
- Connect a jumper to J7 (For USB-UART connection)

For powering the board using a USB Micro-B port:

- Short pins VUSB of JP1 (connect a jumper to pins 2 and 3 of JP1) (as shown in image below) -



- Connect USB from PC to J2

For powering the board using the barrel jack:

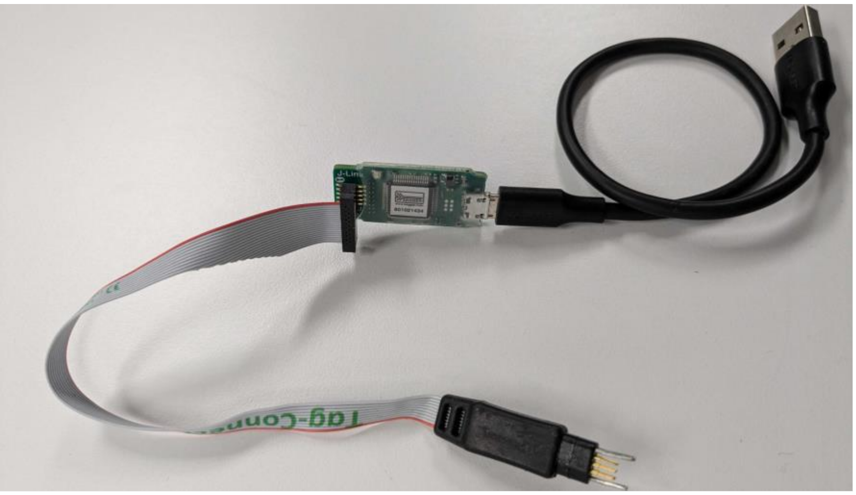
- Short pins 1 and 2 of JP1 (connect a jumper to pins 2 and 1 of JP1)
- Connect the 9V battery to the barrel jack connector

4. Hardware Setup

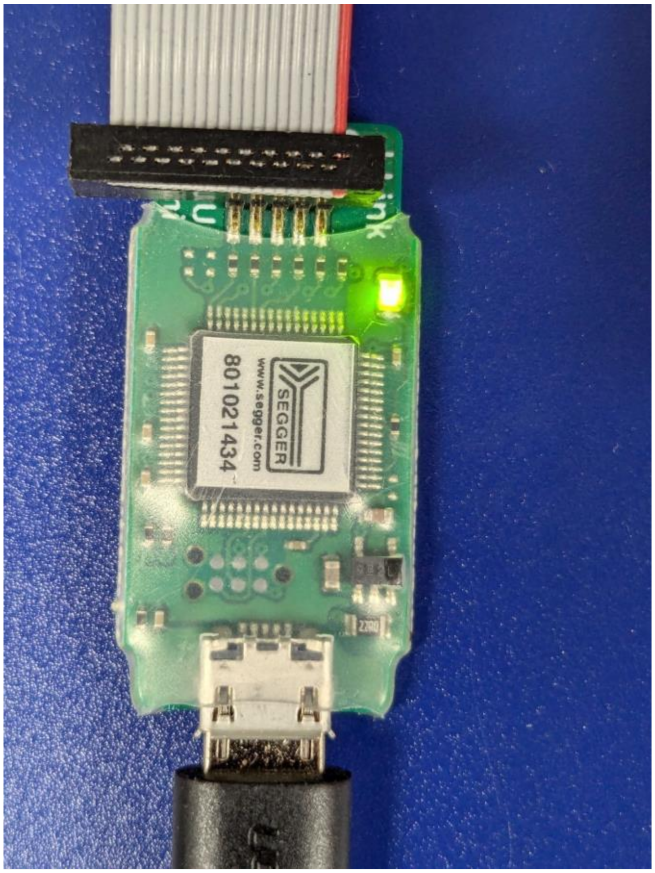
Users may have different versions of Jlink and its associate cable. Please use whatever is appropriate for your device.

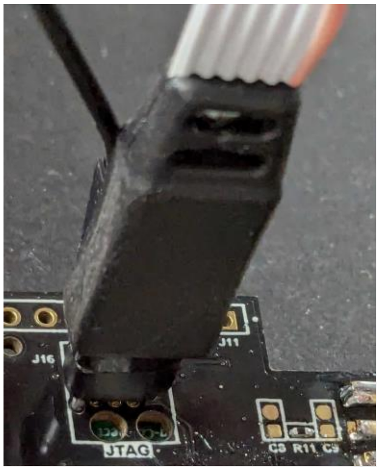
To perform this action, you may need administrative rights. Please login with the administrator account.

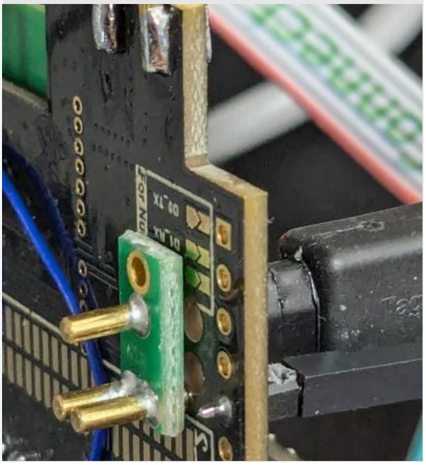
1. Connect the Jlink to the board using the tag connect cable.
2. Connect the Jlink to PC using the Micro-USB to USB cable.



3. Connect tag connector to the tag connector footprint on demo board.







4. Connect the demo board with pc using the USB C to USB cable.

5. Standalone mode

5.1.Prerequisite

- Simplicity Studio 5 (https://www.silabs.com/developers/simplicity-studio)
- J-Flash/J-Flash Lite (https://www.segger.com/downloads/jlink)

5.2.Configure AWS CLI

AWS CLI can be used to send downlinks to sidewalk devices.

5.3.Install and configure the AWS CLI on your system:

- Getting started with the AWS CLI
- Configure your AWS CLI

For configuration after installing AWS CLI and making sure it is in the environment path, go to you terminal and type AWS configure and input the following:

- Access Key ID and AWS Secret Access Key, you can create them after you create IAM User.
- For default region name choose **us-east-1**
- For default output format choose **JSON**

5.4.Create and Flash Manufacturing File

5.4.1.Create Manufacturing File

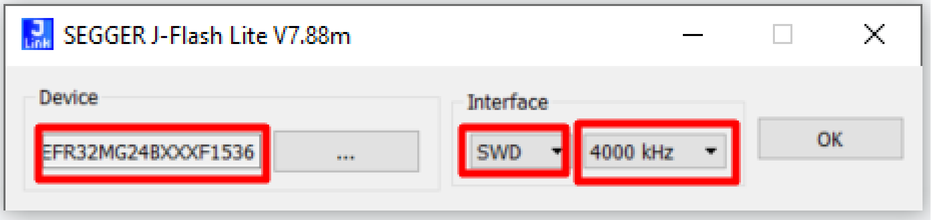
To create Manufacturing File for the Sidewalk devices in an automated way for prototyping follow this link: sidewalk-developers-guide
Make sure the Simplicity Commander is in system Path and python3 is installed along with AWS CLI.
You will basically have to clone the required scripts from the Github repository available in that link.
Make sure to modify *config.yaml* in the root directory after downloading the GitHub repository and change the DESTINATION_NAME to the corresponding destination used for uplink routing.

ⓘ INFO ⓘ: To create destination, role and rule for your sidewalk devices follow this link iot-sidewalk-destination-create

5.4.2.Flash Manufacturing File

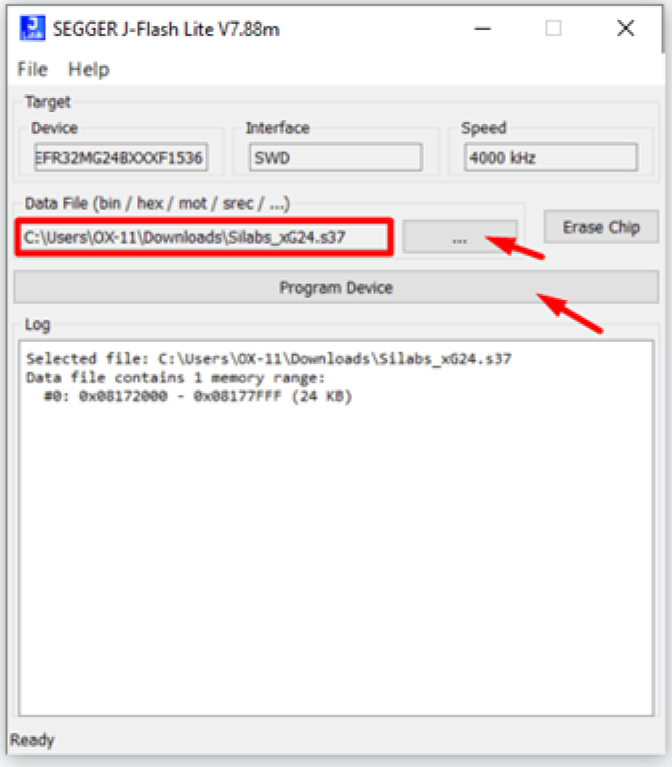
Connect J Link to Oxit Sidewalk Module Then Flash Manufacturing file *Silabs_xG24.s37* using J Flash lite.

To flash the manufacturing binary, install and open J-Flash/J-Flash Lite and select the parameters as shown then click *OK*



Device: EFR32MG24BXXXF1536
Interface: SWD
4000 kHz

Browse the manufacturing file by clicking the "…" and click *Program Device*



Device: EFR32MG24BXXXF1536
Interface: SWD
Speed: 4000 kHz
Data File (bin / hex / mot / srec / ...): C:\Users\OX-11\Downloads\Silabs_xG24.s37

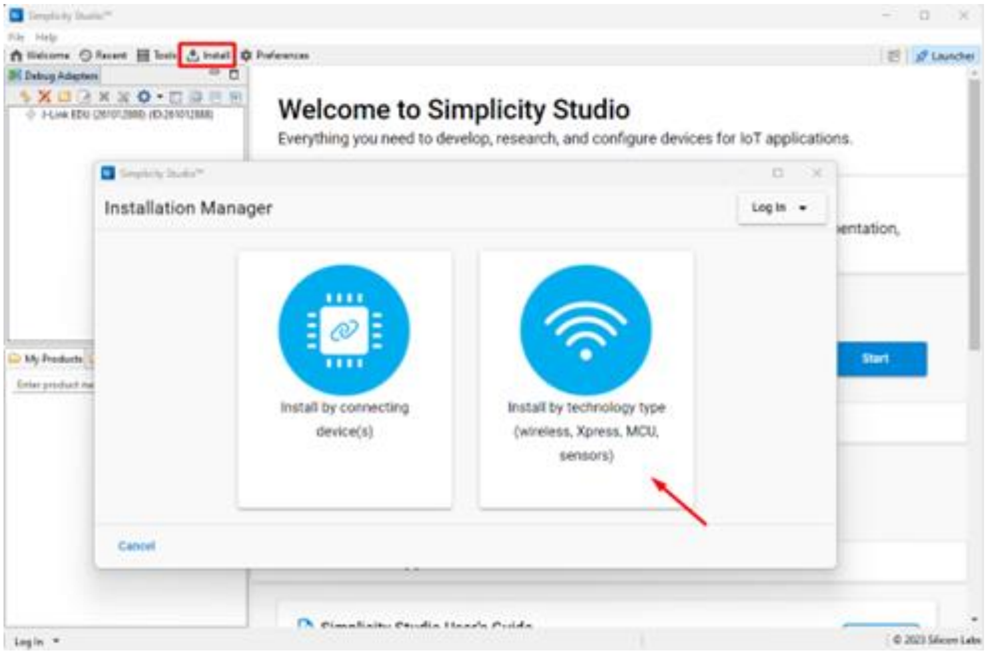
```
Selected file: C:\Users\OX-11\Downloads\Silabs_xG24.s37
Data file contains 1 memory range:
  #0: 0x08172000 - 0x08177FFF (24 KB)
```

6. Create an Amazon Sidewalk Project for Oxit Sidewalk Module

6.1.Installing SDK

Open Simplicity Studio 5 to download and install Gecko SDK 4.3.1 with its respective Sidewalk extension 1.2.0.

- Click on **Install** in the top navigation panel



Welcome to Simplicity Studio

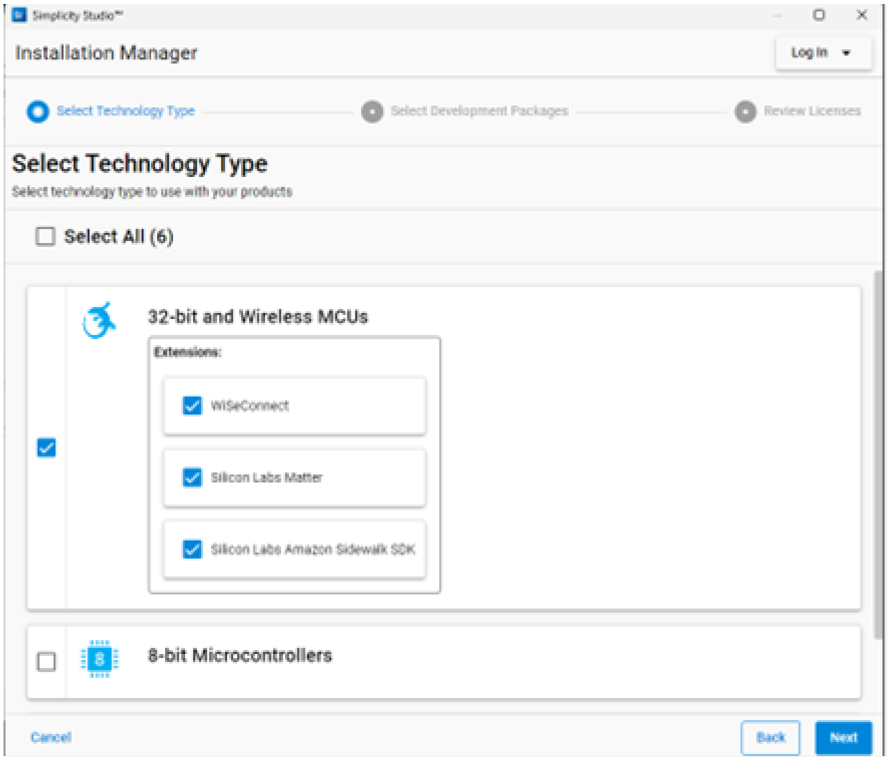
Everything you need to develop, research, and configure devices for IoT applications.

Installation Manager

Install by connecting device(s)

Install by technology type (wireless, Xpress, MCU, sensors)

- Select, **Gecko SDK - 32-bit and Wireless MCUs**



Select Technology Type

Select technology type to use with your products

□ Select All (6)

☑ 32-bit and Wireless MCUs

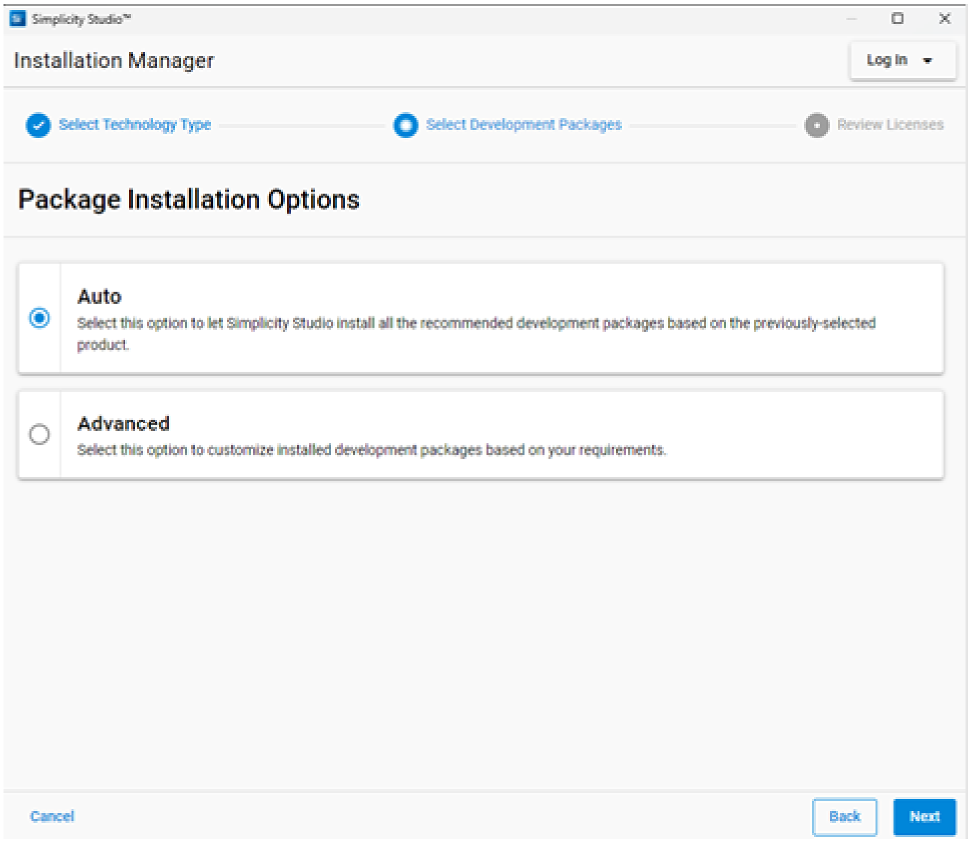
Extensions:

☑ WiSeConnect

☑ Silicon Labs Matter

☑ Silicon Labs Amazon Sidewalk SDK

□ 8-bit Microcontrollers

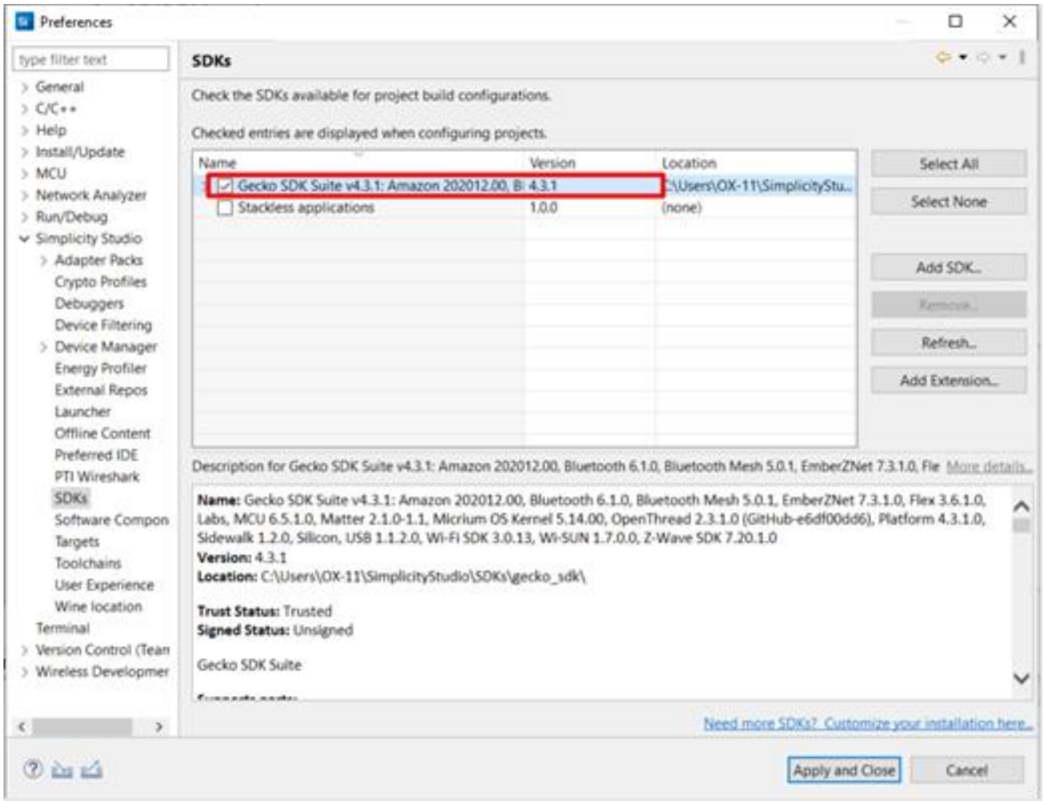


Package Installation Options

● Auto
Select this option to let Simplicity Studio install all the recommended development packages based on the previously-selected product.

○ Advanced
Select this option to customize installed development packages based on your requirements.

- From the top menu, go to **Window** > **Preferences**. In the preferences left pane menu, go to **Simplicity Studio** > **SDKs** and verify that the Gecko SDK 4.3.1 is installed.



SDKs

Check the SDKs available for project build configurations.

Checked entries are displayed when configuring projects.

Name	Version	Location
☑ Gecko SDK Suite v4.3.1: Amazon 202012.00, B	4.3.1	C:\Users\OX-11\SimplicityStu...
□ Stackless applications	1.0.0	(none)

Description for Gecko SDK Suite v4.3.1: Amazon 202012.00, Bluetooth 6.1.0, Bluetooth Mesh 5.0.1, EmberZNet 7.3.1.0, Fle More details...

Name: Gecko SDK Suite v4.3.1: Amazon 202012.00, Bluetooth 6.1.0, Bluetooth Mesh 5.0.1, EmberZNet 7.3.1.0, Flex 3.6.1.0, Labs, MCU 6.5.1.0, Matter 2.1.0-1.1, Micrium OS Kernel 5.14.00, OpenThread 2.3.1.0 (GitHub-e6df00dd6), Platform 4.3.1.0, Sidewalk 1.2.0, Silicon, USB 1.1.2.0, Wi-Fi SDK 3.0.13, Wi-SUN 1.7.0.0, Z-Wave SDK 7.20.1.0
Version: 4.3.1
Location: C:\Users\OX-11\SimplicityStudio\SDKs\gecko_sdk\

Trust Status: Trusted
Signed Status: Unsigned

Gecko SDK Suite

6.2.Create and Compile Sidewalk Application

Silicon Labs SDK for Amazon Sidewalk includes multiple example applications, including:

- Amazon Sidewalk - SoC Bluetooth Sub-GHz Hello Neighbor
- Amazon Sidewalk - SoC CLI

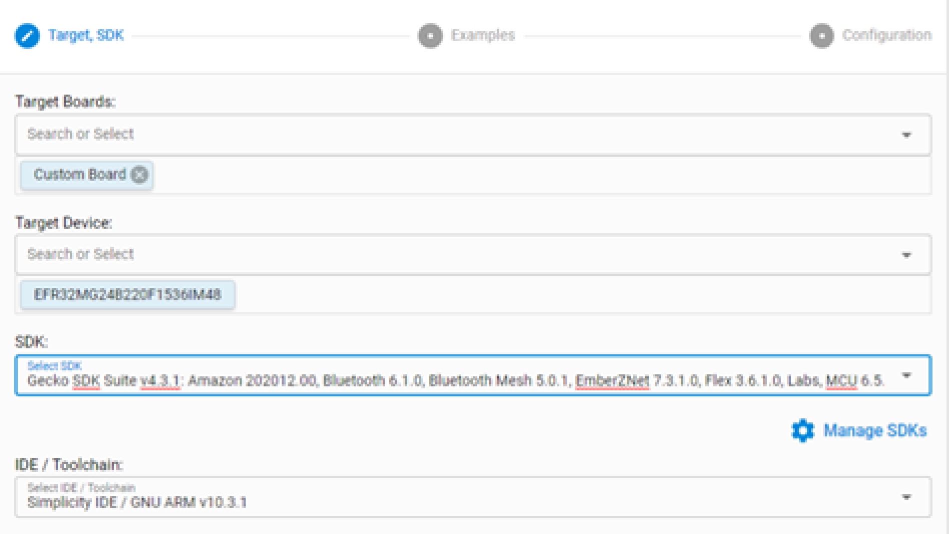
Create an Amazon Sidewalk Project for Oxit Sidewalk Module
With the Sidewalk resources added to your Gecko SDK, reopen Simplicity Studio 5.

∉ Click on **File** > **New** > **Silicon Labs Project Wizard**

∉ Choose appropriate target board, device, SDK and IDE/toolchain to use for the project.

  - Target Boards: Custom Board
  - Target Device: EFR32MG24B220F1536IM48
  - SDK: Gecko SDK Suite 4.3.1

○ IDE / Toolchain Simplicity IDE / GNU ARM v10.3.1



Target Boards:
Search or Select
Custom Board

Target Device:
Search or Select
EFR32MG24B220F1536IM48

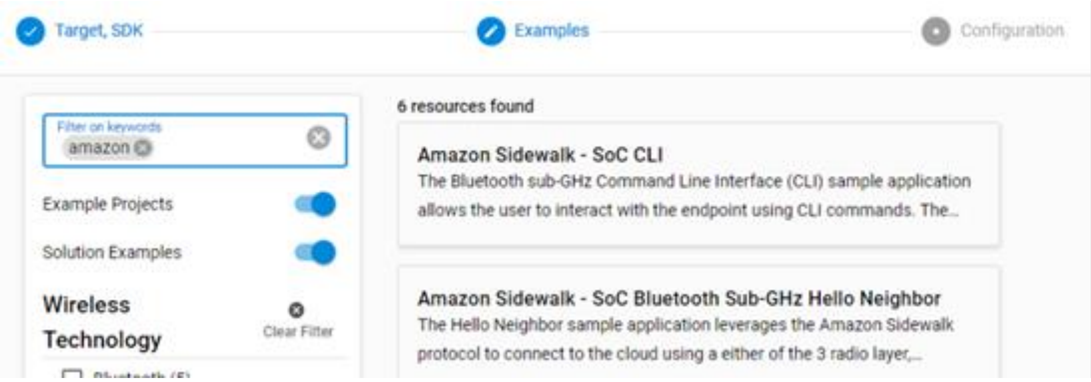
SDK:
Select SDK
Gecko SDK Suite v4.3.1: Amazon 202012.00, Bluetooth 6.1.0, Bluetooth Mesh 5.0.1, EmberZNet 7.3.1.0, Flex 3.6.1.0, Labs, MCU 6.5.

IDE / Toolchain:
Select IDE / Toolchain
Simplicity IDE / GNU ARM v10.3.1

∉ Click on Next

∉ In the Examples filter on keyword **Amazon**

∉ You will have two examples for **Amazon Sidewalk**



6 resources found

Amazon Sidewalk - SoC CLI
The Bluetooth sub-GHz Command Line Interface (CLI) sample application allows the user to interact with the endpoint using CLI commands. The...

Amazon Sidewalk - SoC Bluetooth Sub-GHz Hello Neighbor
The Hello Neighbor sample application leverages the Amazon Sidewalk protocol to connect to the cloud using a either of the 3 radio layer,...

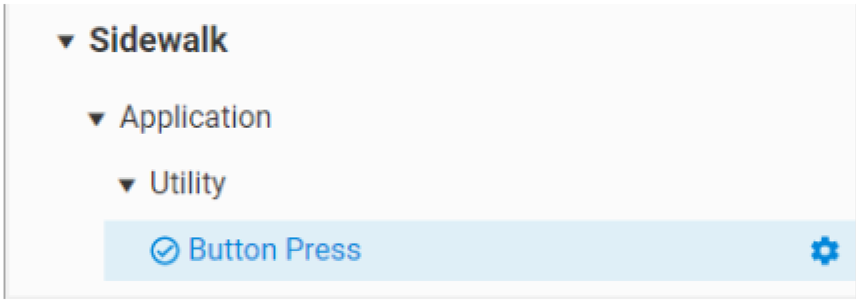
- You can choose one of the examples, for this demo choose **Amazon Sidewalk - SoC Bluetooth Sub-GHz Hello Neighbor**
- Click Next and Finish

7. Modify Amazon Sidewalk Sample Application

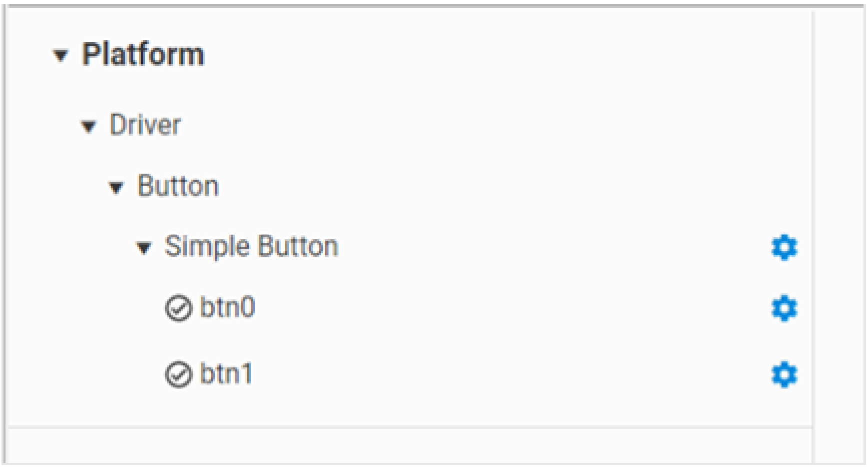
As Amazon sample applications are based on Silicon labs wireless starter kit, there are few modifications that need to be done:

1. Go to the project .slcp / Software components and follow the below instructions:

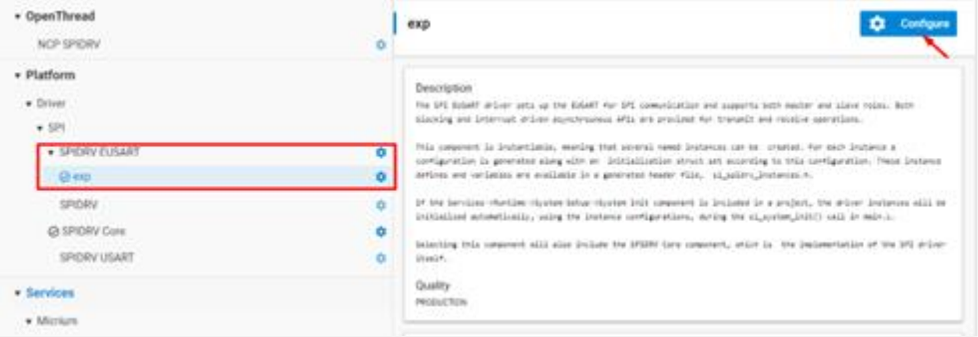
   a. Uninstall Button Press Utility



   b. Uninstall btn0 and btn1



   c. Modify SPIDRV EUSART exp pins to match Oxit Sidewalk module Sx1262 SPI pins as below -
   Module: EUSART1
   CS: PA05
   SCLK: PA06
   TX: PA07
   RX: PA08



Description

The SPI EUSART driver sets up the EUSART for SPI communication and supports both master and slave roles. Both blocking and interrupt driven asynchronous APIs are provided for transmit and receive operations.

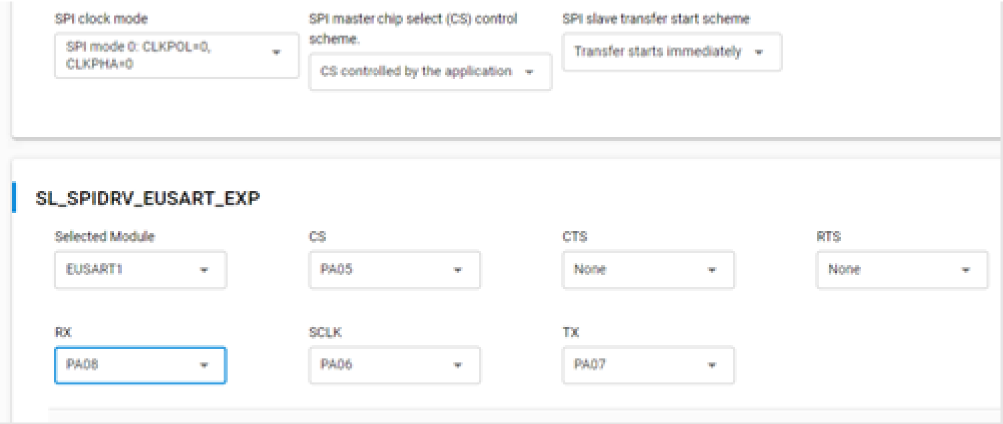
This component is instantiable, meaning that several named instances can be created. For each instance a configuration is generated along with an initialization struct set according to this configuration. These instance defines and variables are available in a generated header file, sl_spidrv_instances.h.

If the Services->Runtime->System Setup->System Init component is included in a project, the driver instances will be initialized automatically, using the instance configurations, during the sl_system_init() call in main.c.

Selecting this component will also include the SPIDRV Core component, which is the implementation of the SPI driver itself.

Quality

PRODUCTION



SPI clock mode: SPI mode 0: CLKPOL=0, CLKPHA=0

SPI master chip select (CS) control scheme: CS controlled by the application

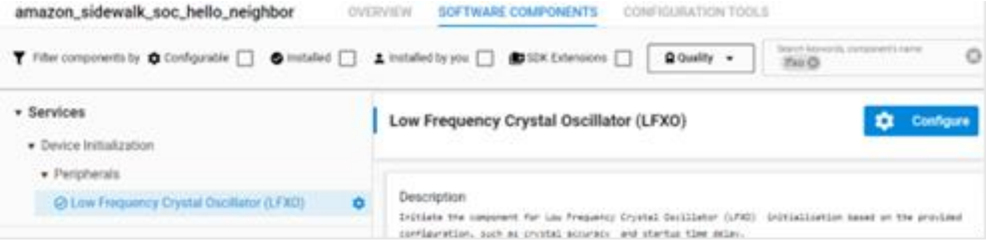
SPI slave transfer start scheme: Transfer starts immediately

SL_SPIDRV_EUSART_EXP

Selected Module	CS	CTS	RTS
EUSART1	PA05	None	None

RX	SCLK	TX
PA08	PA06	PA07

d. Install LFXO

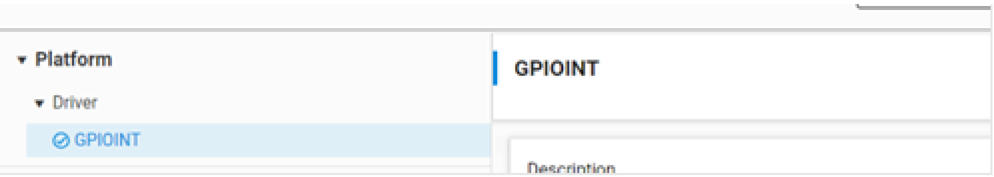


Low Frequency Crystal Oscillator (LFXO)

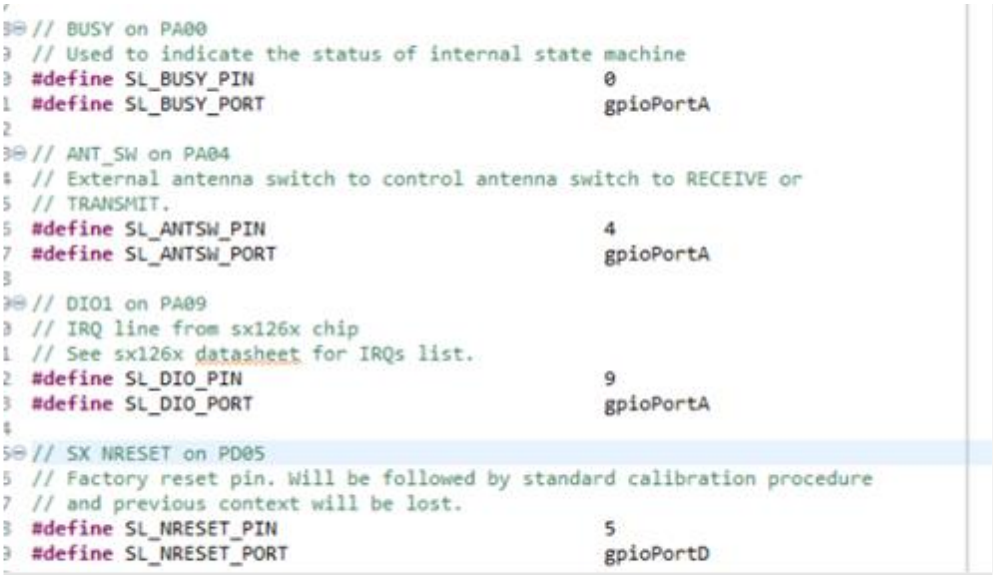
Description

Initiate the component for Low Frequency Crystal Oscillator (LFXO) initialization based on the provided configuration, such as crystal accuracy and startup time delay.

e. Install GPIOINT



2. Go to config folder in the project explorer and open app_gpio_config.h and modify SX1262 pins accordion to Oxit Sidewalk Module as follows:

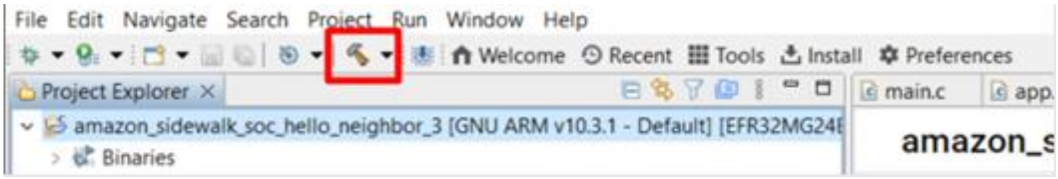


```
// BUSY on PA00
// Used to indicate the status of internal state machine
#define SL_BUSY_PIN                               0
#define SL_BUSY_PORT                              gpioPortA

// ANT_SW on PA04
// External antenna switch to control antenna switch to RECEIVE or
// TRANSMIT.
#define SL_ANTSW_PIN                              4
#define SL_ANTSW_PORT                             gpioPortA

// DIO1 on PA09
// IRQ line from sx126x chip
// See sx126x datasheet for IRQs list.
#define SL_DIO_PIN                                9
#define SL_DIO_PORT                               gpioPortA

// SX NRESET on PD05
// Factory reset pin. Will be followed by standard calibration procedure
// and previous context will be lost.
#define SL_NRESET_PIN                             5
#define SL_NRESET_PORT                            gpioPortD
```

3. Go to *app_init.c* and *app_process.c* files and remove ***#include “app_button_press.h”***
4. Go to *app_init.c*, *app_init(void)* function and remove ***app_button_press_enable();*** callback.

7.1.Build your Amazon Sidewalk Application

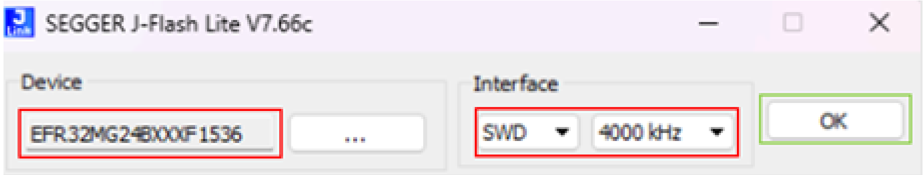
- After modifying the Amazon Sidewalk application, you can now successfully build the project by clicking the hammer button on the Simplicity studio.


File Edit Navigate Search Project Run Window Help
Welcome
Recent
Tools
Install
Preferences
Project Explorer
main.c
app.
amazon_sidewalk_soc_hello_neighbor_3 [GNU ARM v10.3.1 - Default] [EFR32MG24E
Binaries
amazon_s

7.2.Flash your Amazon Sidewalk Application

After installing the J-Link Software tools, in the Search bar search for J-Flash Lite and open it.

- For the device section choose EFR32MG24BXXXF536
- For the interface choose SWD and 400 kHz as speed
- Click ok
- Browse the path to .s37 file by clicking the “…” and click *Program Device*



SEGGER J-Flash Lite V7.66c

Device: EFR32MG24BXXXF1536

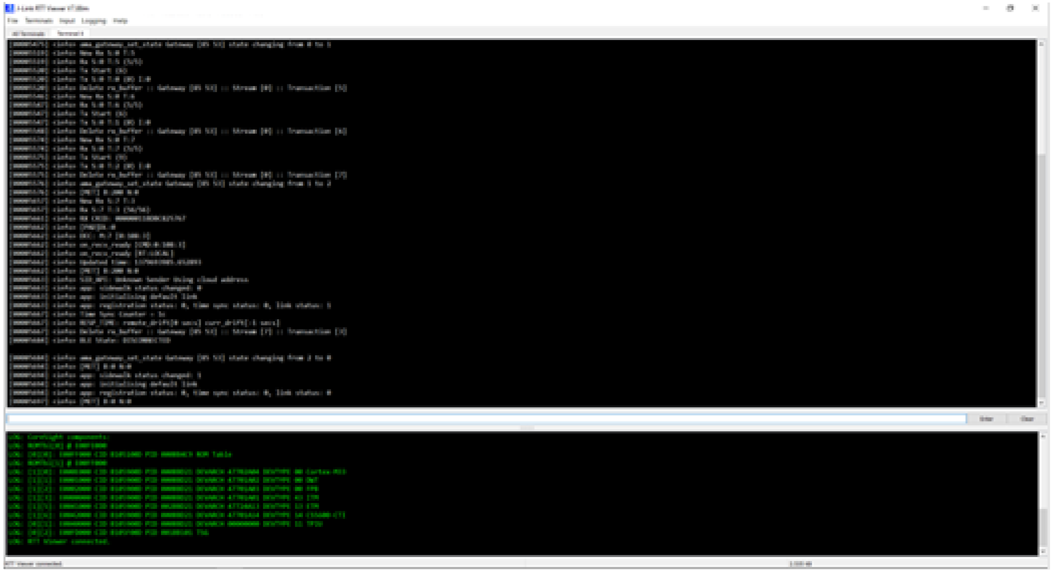
Interface: SWD, 4000 kHz

7.3.Check Amazon Sidewalk Application logs

To monitor standalone application logs, users can simply connect J Link to the board and open J Link RTT viewer.

- Install the J-Link RTT Viewer , if not already installed.
- Connect J-Link to MCM EVK board, using J16.
- Open the J-Link RTT Viewer.
- In the **Configuration** panel, **Connection to J-Link** RTT section, select **USB**.
- In the Specify Target Device list, select the connected part EFR32MG24BxxxF1536.
- In the **Target Interface & Speed panel**, select **SWD** and **4000 kHz**.
- In the **RTT Control Block** panel, select **Auto Detection**.
- Click **OK.**

A terminal opens and the Sidewalk application traces are output to it as shown in the image below.

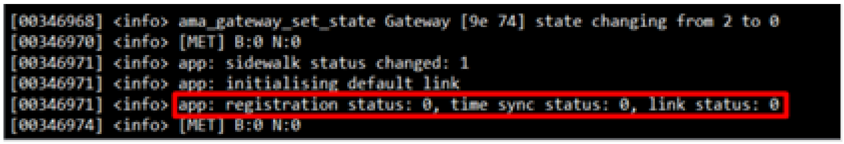


Note:

1. J Link RTT Viewer is used as default to view logs in Amazon Sidewalk example project.

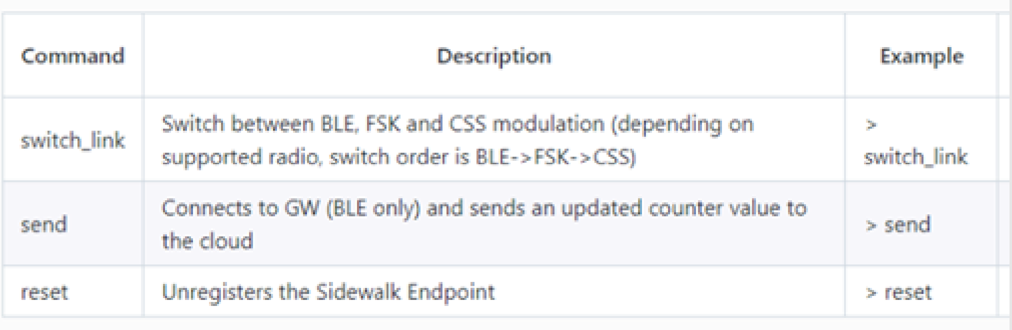
7.4.Test the Amazon Sidewalk Application

Successful registration is indicated by the following log -



```
[00346968] <info> ama_gateway_set_state Gateway [9e 74] state changing from 2 to 0
[00346970] <info> [MET] B:0 N:0
[00346971] <info> app: sidewalk status changed: 1
[00346971] <info> app: initialising default link
[00346971] <info> app: registration status: 0, time sync status: 0, link status: 0
[00346974] <info> [MET] B:0 N:0
```

By default, the Hello Neighbor application will start with the BLE default radio layer. User can use the following commands in the RTT Viewer to test the application by sending uplinks and switching the protocol between BLE, FSK and CSS modulation.

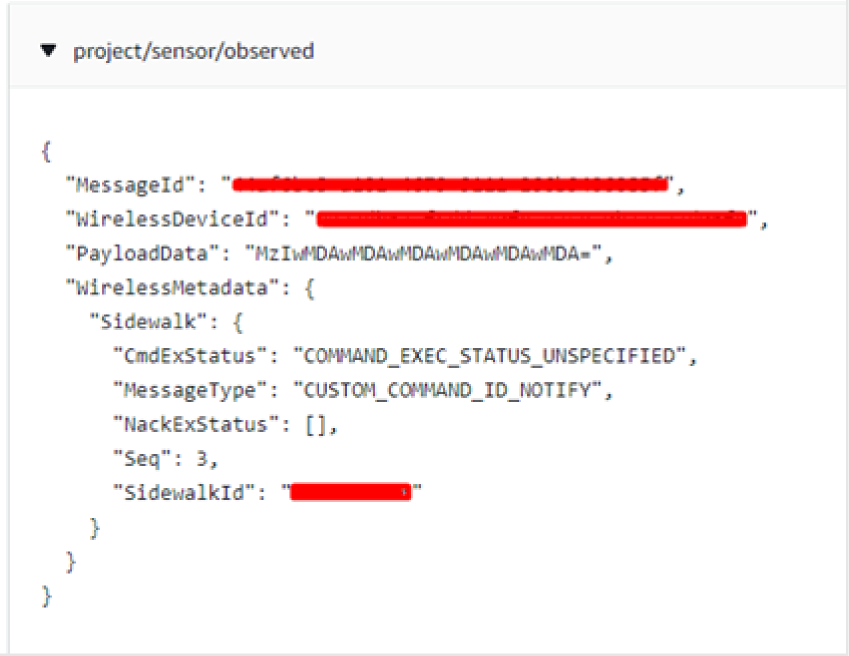


Command	Description	Example
switch_link	Switch between BLE, FSK and CSS modulation (depending on supported radio, switch order is BLE->FSK->CSS)	> switch_link
send	Connects to GW (BLE only) and sends an updated counter value to the cloud	> send
reset	Unregisters the Sidewalk Endpoint	> reset

8. Monitor Uplink Data

To monitor data sent from OxTech MCM Devkit to cloud follow below:

1. Login to your AWS account and click on IAM user and enter the user credentials.
2. Go to the MQTT test client in AWS.
3. Subscribe to the topic created when creating the destination for the device.
4. You should see messages come up on the MQTT broker.



```
{
  "MessageId": "[redacted]",
  "WirelessDeviceId": "[redacted]",
  "PayloadData": "MzIwMDAwMDAwMDAwMDAwMDAwMDA=",
  "WirelessMetadata": {
    "Sidewalk": {
      "CmdExStatus": "COMMAND_EXEC_STATUS_UNSPECIFIED",
      "MessageType": "CUSTOM_COMMAND_ID_NOTIFY",
      "NackExStatus": [],
      "Seq": 3,
      "SidewalkId": "[redacted]"
    }
  }
}
```

9. Send A Downlink

After installing and configuring AWS CLI on your system, the user should be able to run the AWS command to send payload data.

1. Run this command from the terminal:

```
aws iotwireless send-data-to-wireless-device --id=39c3d7af-6bf4-4dca-9858-465f7cc7e681 --transmit-mode 0 --payload-data="AQ==" --wireless-metadata "Sidewalk={Seq=1}"
```

   a. Seq=x, x should be unique any time you run this command
   b. –id should be the "Wireless Device ID" of the device in this case *39c3d7af-6bf4-4dca-9858-465f7cc7e681*
   c. –payload-data : base64 format of the payload to send in that case "AQ==", it represents 0x01 in hex.

10. General Notes

1. Boards do not come pre flashed with sidewalk manufacturing file.
2. Sidewalk manufacturing files need to be flashed separately to be tied to the desired test account.
3. Boards can be re-flashed at any time to host any custom firmware and manufacturing file.

11. Known Issues

1. Application will crash if CSS is started before FSK. This is a known issue with Sidewalk proprietary stack. The user must start FSK before CSS at first to prevent application crashes.
2. Application will crash if no MFG is flashed and the user sends BLE/FSK or CSS link request, this is also a known issue in Sidewalk stack.