MCM EVK Guide - Amazon Sidewalk

09/18/2023 | Document Status: Preliminary

Revision History

Date	Version	Description
9/18/2023	0.9	Initial version

Table of Contents

1. Introduction

The MCM (Multi Connectivity Module) EVK is outlined in this document. This covers its air interfaces, hardware interfaces, serial commands, and provisioning details. It provides a convenient way to comprehend the MCM's interface specifications, electrical and mechanical aspects, and related information, facilitating the design and setup of Amazon Sidewalk applications using the module.

2. Product Overview

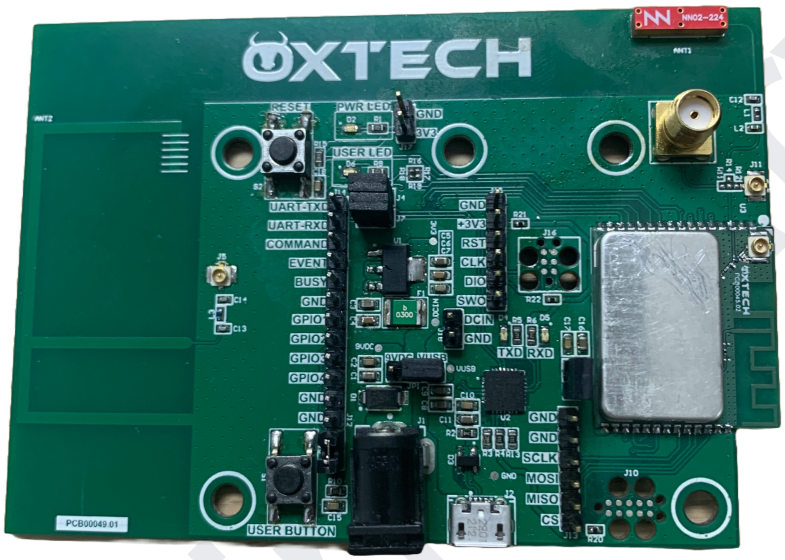


Figure: OxTech MCM Devkit

2.1 General Description

The MCM EVK is an evaluation kit based on the MCM module designed to aid in the prototyping and testing of your IoT applications. The EVK enables developers to connect peripherals using jumper wires and evaluate its functionality.

2.2 Key Features

Features	**Details**
Power Supply	• USB Power • 9V Barrel Jack Connector
Program/Debug Interface	JTAG
Operating Frequency	• LoRa/FSK: 863-928 MHz • BLE: 2.402-2.480 GHz

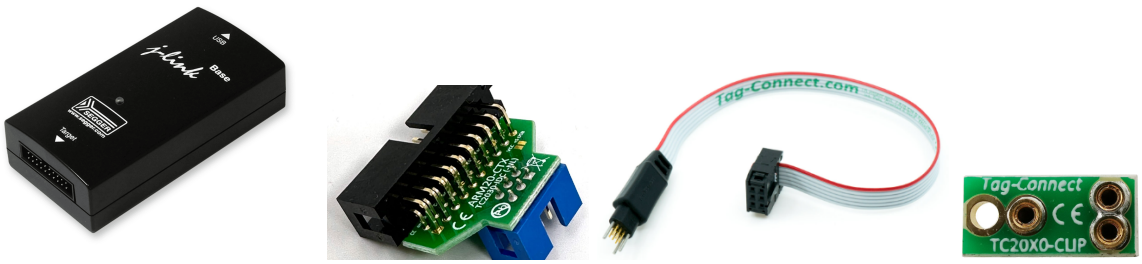
LoRa/FSK Features	• +22dBm or +15dBm high efficiency PA • Integrated DC-DC converter and LDO • High sensitivity: down to -148dBm • Built-in bit synchronizer for clock recovery • FSK, GFSK, MSK, GMSK, LoRa and Long Range FHSS modulations
Bluetooth Protocol	BLE 5.1
Bluetooth Operation Mode	BLE
Bluetooth Modulation	GFSK
Antenna Interfaces	• U.FL antenna port for LoRa/FSK • Trace antenna for BLE • Antenna interface for LoRa/FSK (Pin 3) • 50 Ω impedance
Temperature Range	• Operating temperature range: -30 °C to +85 °C • Storage temperature range: -40 °C to +95 °C
RoHS	All hardware components are fully compliant with RoHS directive

2.3 MCM Module

This EVK board uses OxTech MCM module, refer to the MCM summary datasheet.

2.4 Programming and Debugging

A J-Link segger with a 6-pin TAG connect cable is required to program and debug the MCM module on the EVK board. The programming and debugging hardware is **not included** in the EVK kit.



(a) **(b)** **(c)** **(d)**

(a) J-Link: https://shop-us.segger.com/product/j-link-base-8-08-00/
(b) Adapter: https://www.tag-connect.com/product/arm20-ctx-m
(c) 6-pin tag connect cable: https://www.tag-connect.com/product/tc2030-idc-nl
(d) Retaining clip: https://www.tag-connect.com/product/tc2030-retaining-clip-board-3-pack

3. EVK Interfaces

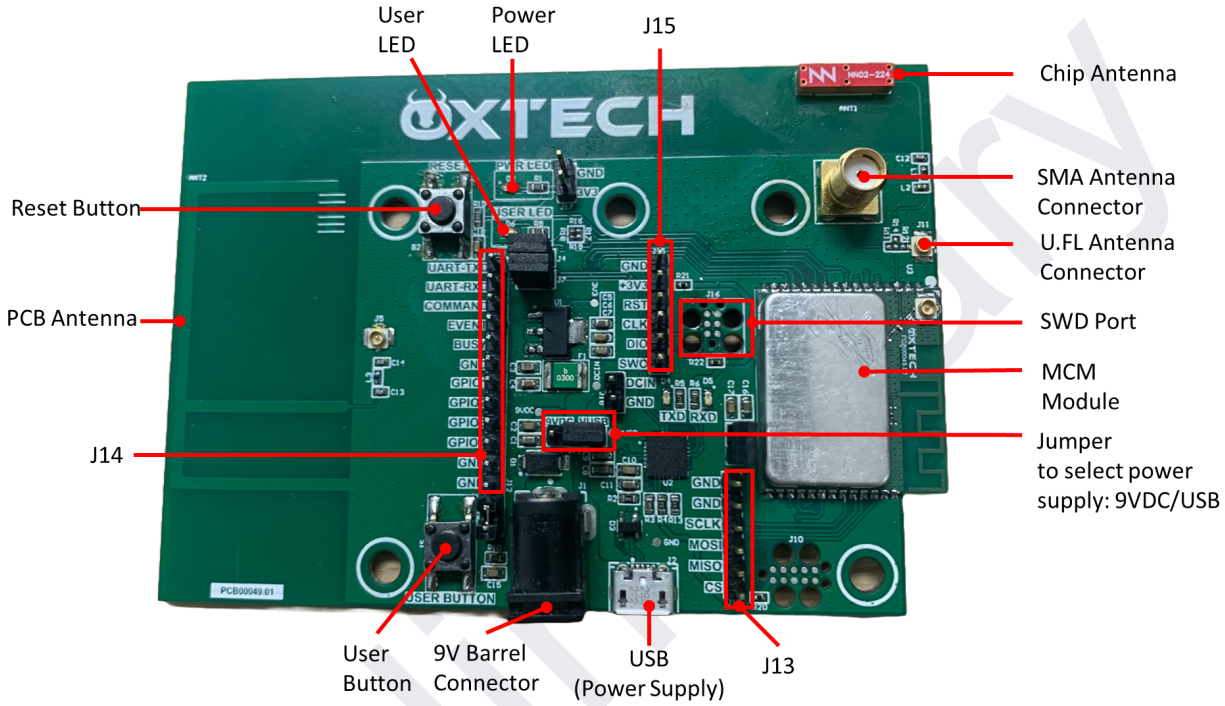


Figure: MCM EVK board breakout

Identifier	Terminal Name	Type	Description
J14	UART-TXD	O	UART TX, module to host
	UART-RXD	I	UART RX, host to module
	COMMAND	I	Module command input
	EVENT	O	Module event out
	BUSY	O	Module busy out
	GND	POWER	Module ground pin
	GPIO1	I/O	General purpose input/output

	GPIO2	I/O	General purpose input/output
	GPIO3	I/O	General purpose input/output
	GPIO4	I/O	General purpose input/output
	GND	POWER	Module ground pin
	GND	POWER	Module ground pin
J15	GND	POWER	Module ground pin
	3V3	POWER	Module 3.3V Ssupply
	RST	I	MCU reset
	CLK	CLOCK	Programing interface pin
	DIO	I/O	Programing interface pin
	SWO	O	Debug serial wire output
J13	GND	POWER	Module ground pin
	GND	POWER	Module ground pin
	SCLK	CLOCK	Module SPI clock
	MOSI	I	Module SPI master out slave in
	MISO	O	Module SPI master in slave out
	CS	I	Module SPI chip select
S1	USER BUTTON	I	General purpose user button
S2	RESET	I	MCU Reset
D2	PWR LED	O	Indicate board power
D6	USER LED	O	General purpose user LED

3.1 Debug Interface

Parameter	Value
Baud Rate	115,200 bps
Data Bits	8
Stop Bits	1
Parity	None

3.2 RF Antenna Interfaces

3.2.1 RF Antenna Pin Description

The MCM EVK has U.FL connector, SMA connector, PCB and chip antenna options for LoRa/FSK. The BLE antenna is built-in on the MCM module.

Operating frequency

Mode	Frequency	Unit
LoRa/FSK	863-928	MHz
BLE	2.402-2.480	GHz

4. Mode of Operation:

The MCM has the capability to operate in Host mode as well as in network coprocessor (NCP) mode. The EVK also complements both of these capabilities. It allows the developers to either interface the peripherals with the MCM running in host mode or interface another host MCU to the MCM running in NCP mode using UART.

In both scenarios, MCM requires the following binaries to operate with Amazon Sidewalk:

- **Manufacturing Binary:** This file contains information that is required to provision the device on Amazon Sidewalk. The process to generate these binaries is explained later in the “Prototyping API” section.
- **Application Binary (Host mode):** The application program that you want to run on the MCM. The user needs to develop these binaries.
- **Modem Binary (NCP mode):** The program that sends and receives data as well as commands to and from the host MCU.

4.1 Application Binary (Standalone Mode)

Build your own application or run the example applications available in the Amazon Sidewalk SDK available in the Simplicity Studio 5.

4.1.1 Prerequisites

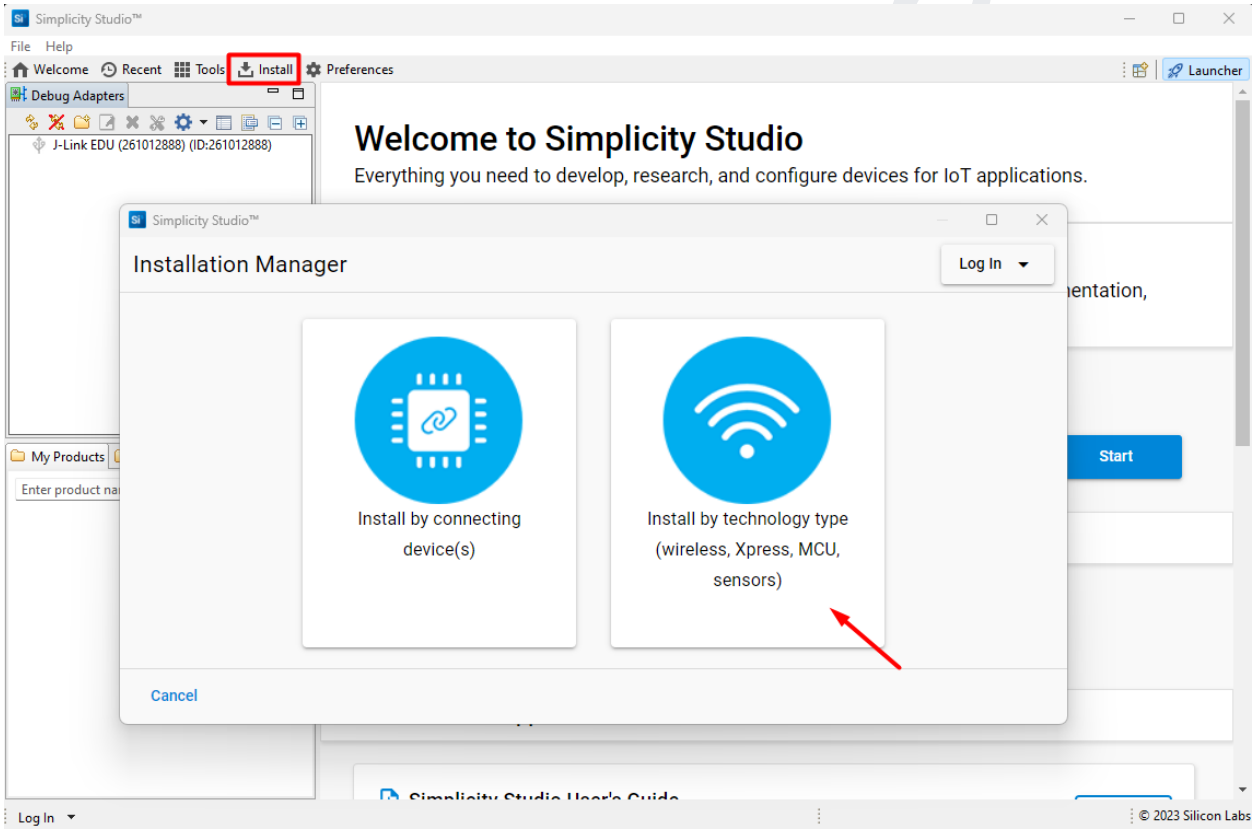
- Simplicity Studio 5 (https://www.silabs.com/developers/simplicity-studio)
- J-Flash/J-Flash Lite (https://www.segger.com/downloads/jlink)

Here we demonstrate how to build and flash an example application available in the SDK for the MCM EVK board.

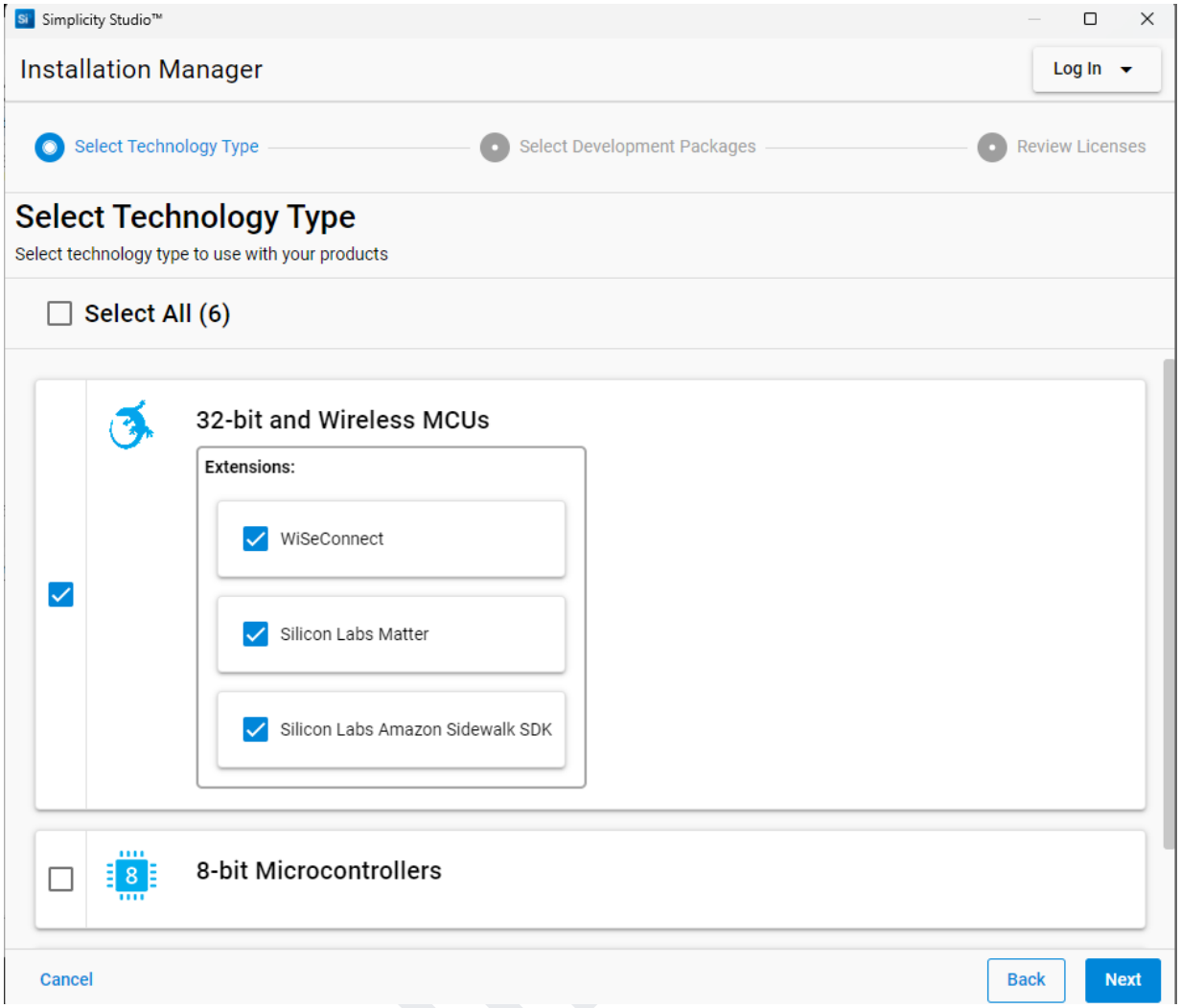
4.1.2 Installing SDK

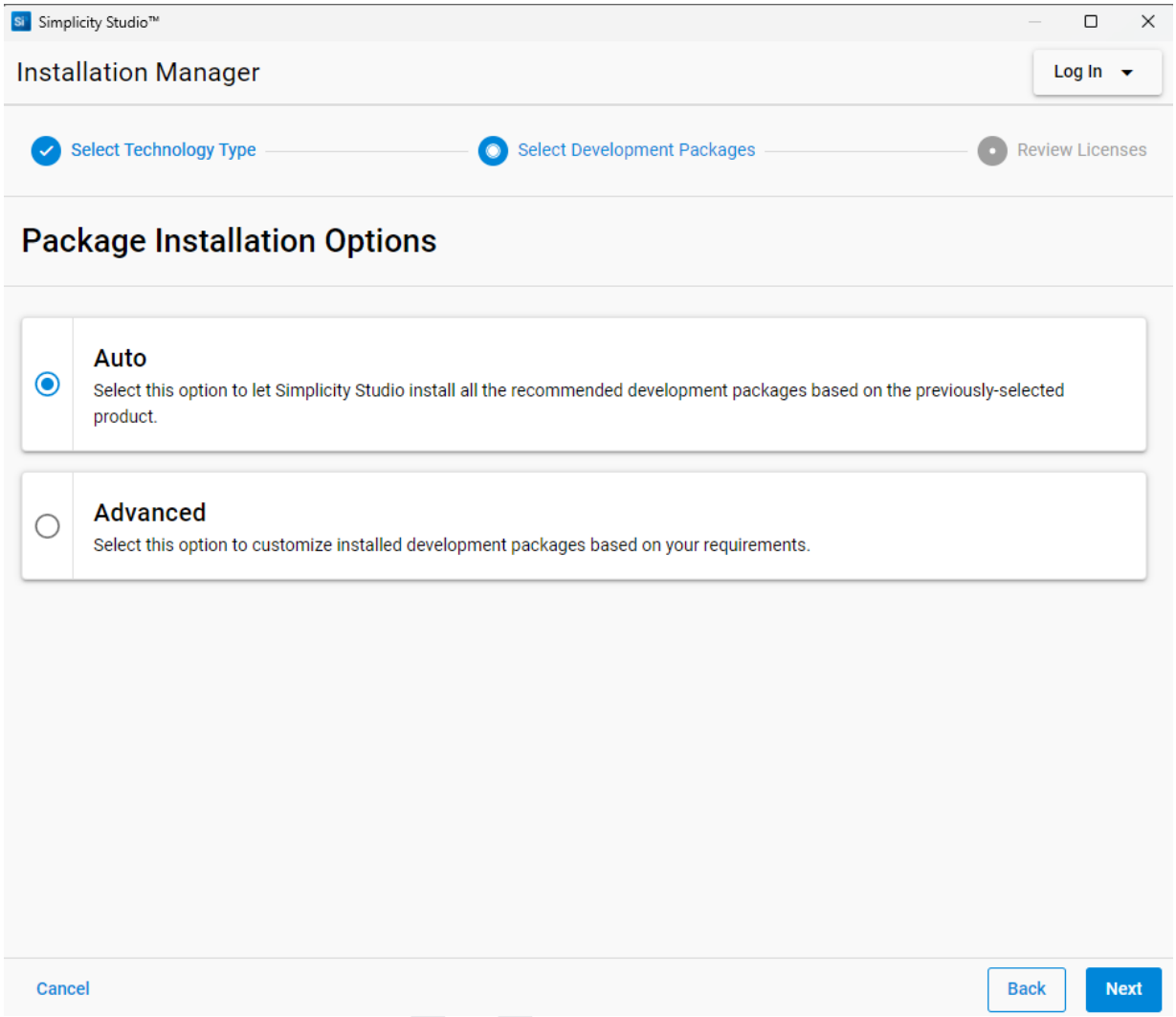
Open Simplicity studio to download the latest SDK:

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

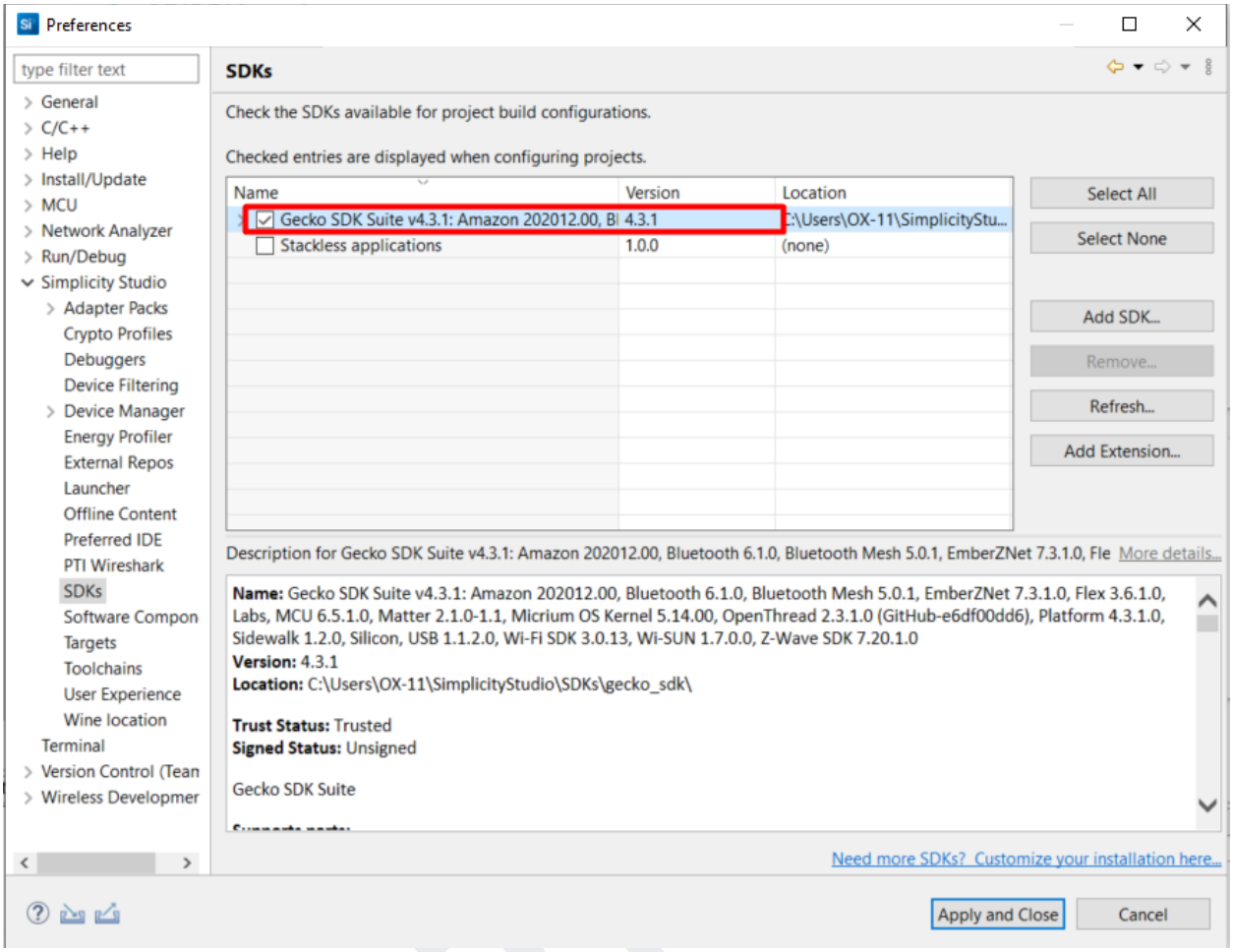


- Select, **Gecko SDK - 32-bit and Wireless MCUs**


Simplicity Studio™
Installation Manager
Log In
Select Technology Type
Select Development Packages
Review Licenses
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
Cancel
Back
Next



- From the top menu, go to **Window** > **Preferences**. In the preferences left pane menu, go to **Simplicity Studio** > **SDKs** and verify that the GSDK installed is latest (4.3.1 at the time of preparing this document)



4.1.3 Create and Compile Sidewalk Application

Amazon Sidewalk Sample Applications

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

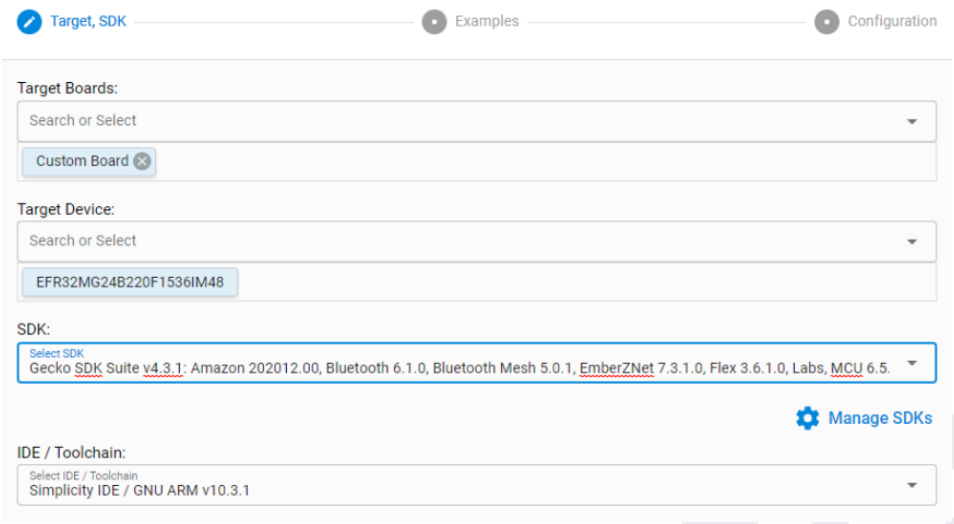
- Amazon Sidewalk - SoC Bluetooth Sub-GHz Hello Neighbor
- Amazon Sidewalk - SoC CLI

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 the 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 / ToolchainL Simplicity IDE / GNU ARM v10.3.1



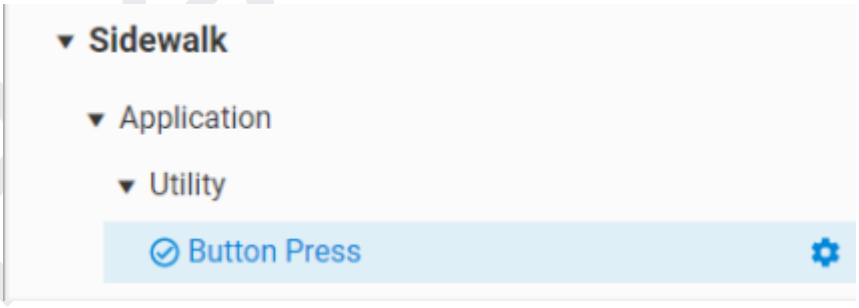
- Click on Next
- In the Examples, filter on keyword **Amazon**
- You can choose one of the examples, for this demo choose **Amazon Sidewalk - SoC Bluetooth Sub-GHz Hello Neighbor**
- Click Next and Finish

4.1.4 Modify Amazon Sidewalk Sample Application for EVK

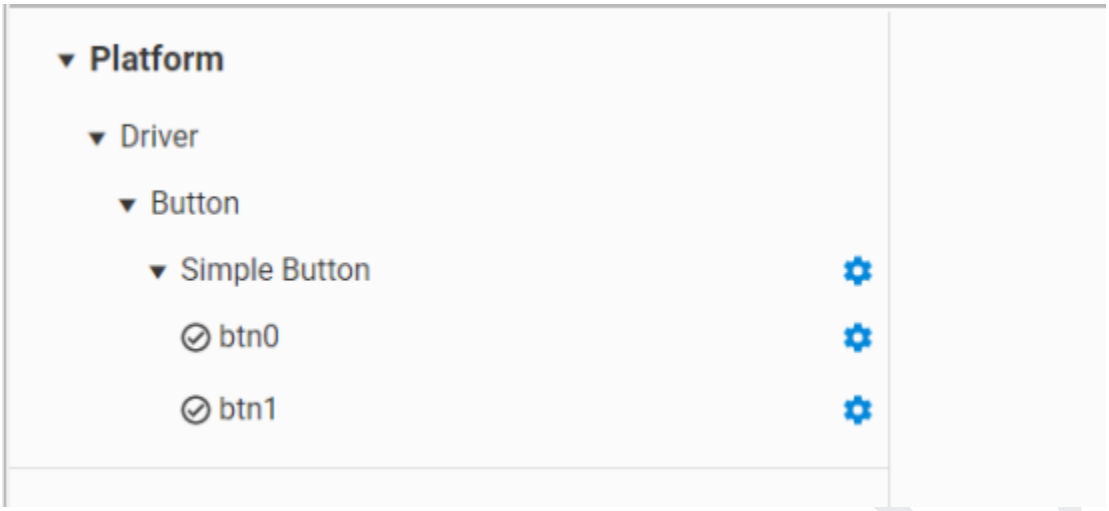
As Amazon sample applications are based on Silicon lab's wireless starter kit, there are few modifications that need to be made for the application to work with OxTech MCM EVK board:

Go to the **<project name>.slcp** > **Software components** and follow the below instructions:

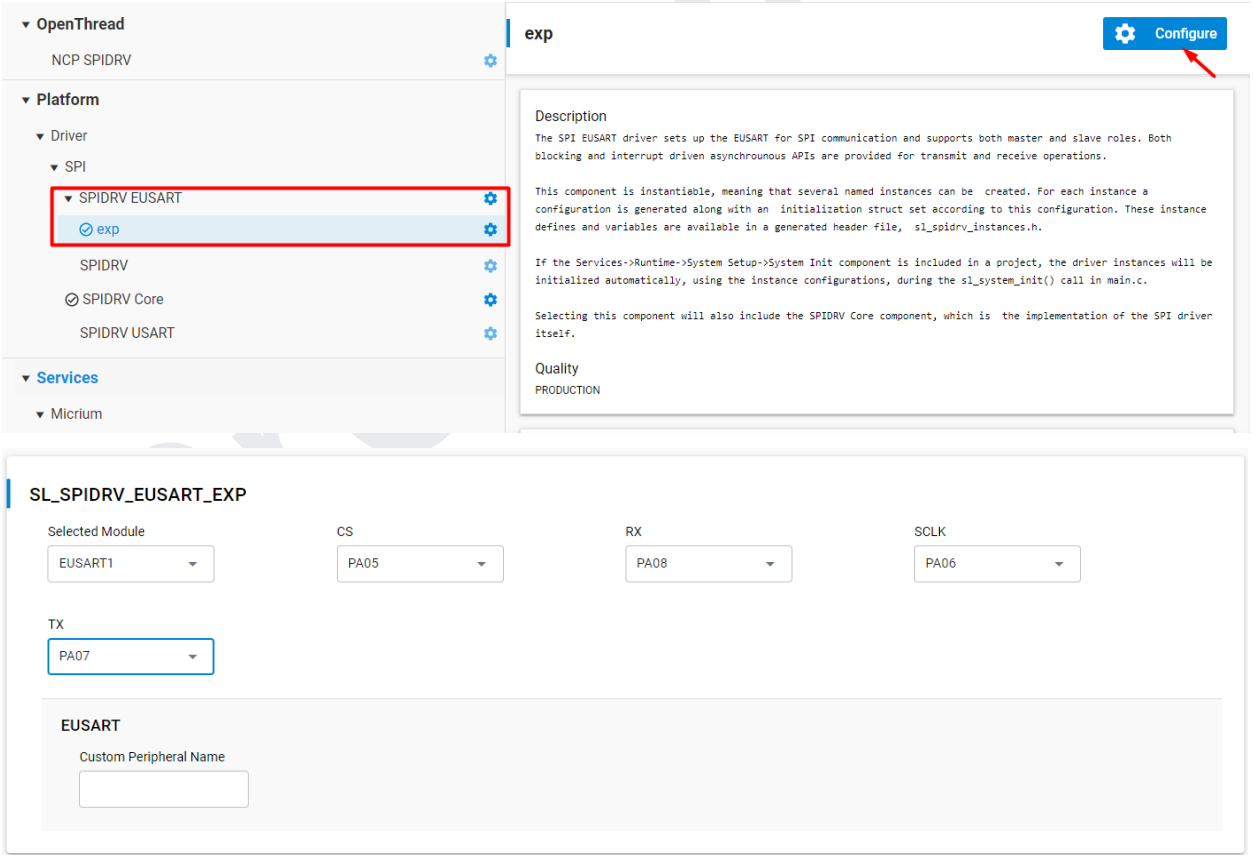
- Uninstall the **Button Press** Utility.
  Note: If it is already uninstalled then install it and then uninstall it



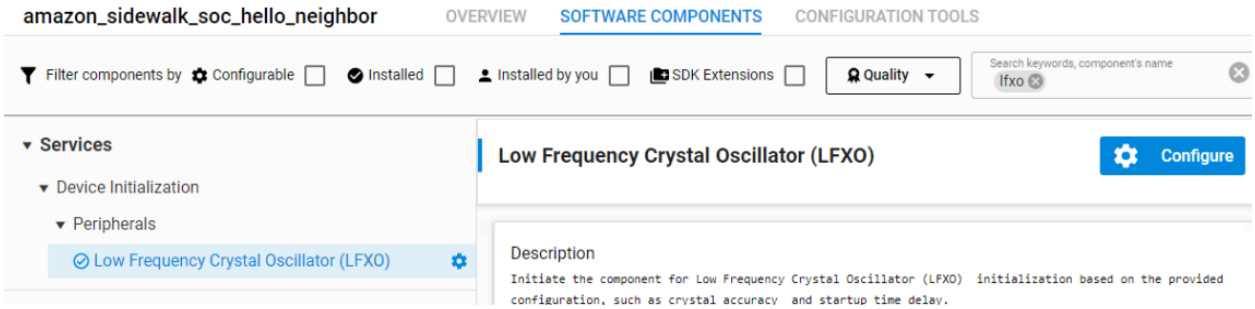
- Make sure that **btn0** and **btn1** are not installed



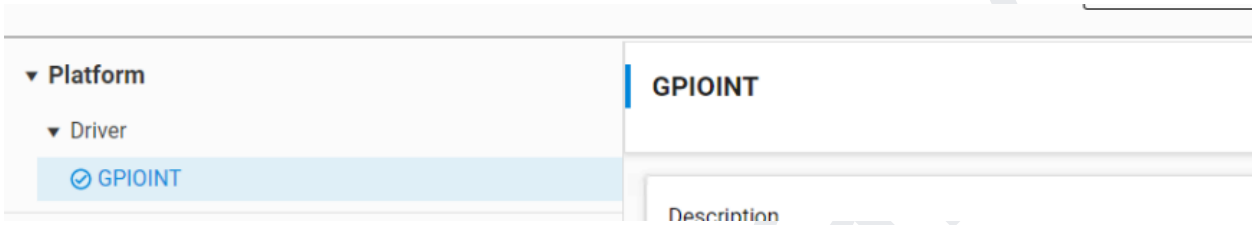
- Modify **SPIDRV EUSART** exp pins to match Oxit Sidewalk module sx1262 SPI pins
  - **Module:** EUSART1
  - **CS:** PA05
  - **SCLK:** PA06
  - **TX:** PA07
  - **RX:** PA08



- **Install LFXO**



- **Install GPIOINT**



- Go to ***config*** folder in the project and open ***app_gpio_config.h*** and modify sx1262 pins according to MCM Module as shown below:

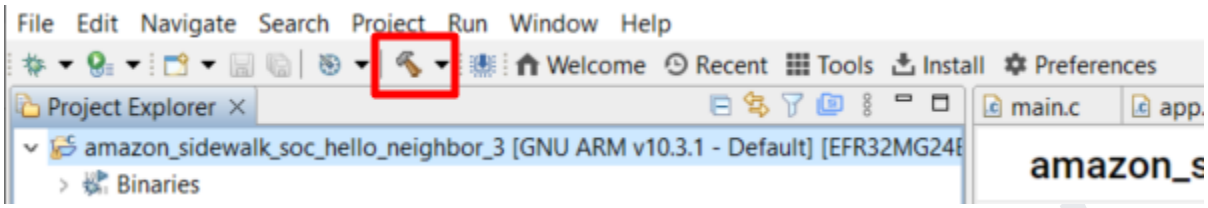
```
// BUSY on PA00
// Used to indicate the status of internal state machine
#define SL_BUSY_PIN                                     0
#define SL_BUSY_PORT                                    gpioPortA

// 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
```

- Go to ***app_init.c*** and ***app_process.c*** and remove ***#include "app_button_press.h"*** from both files

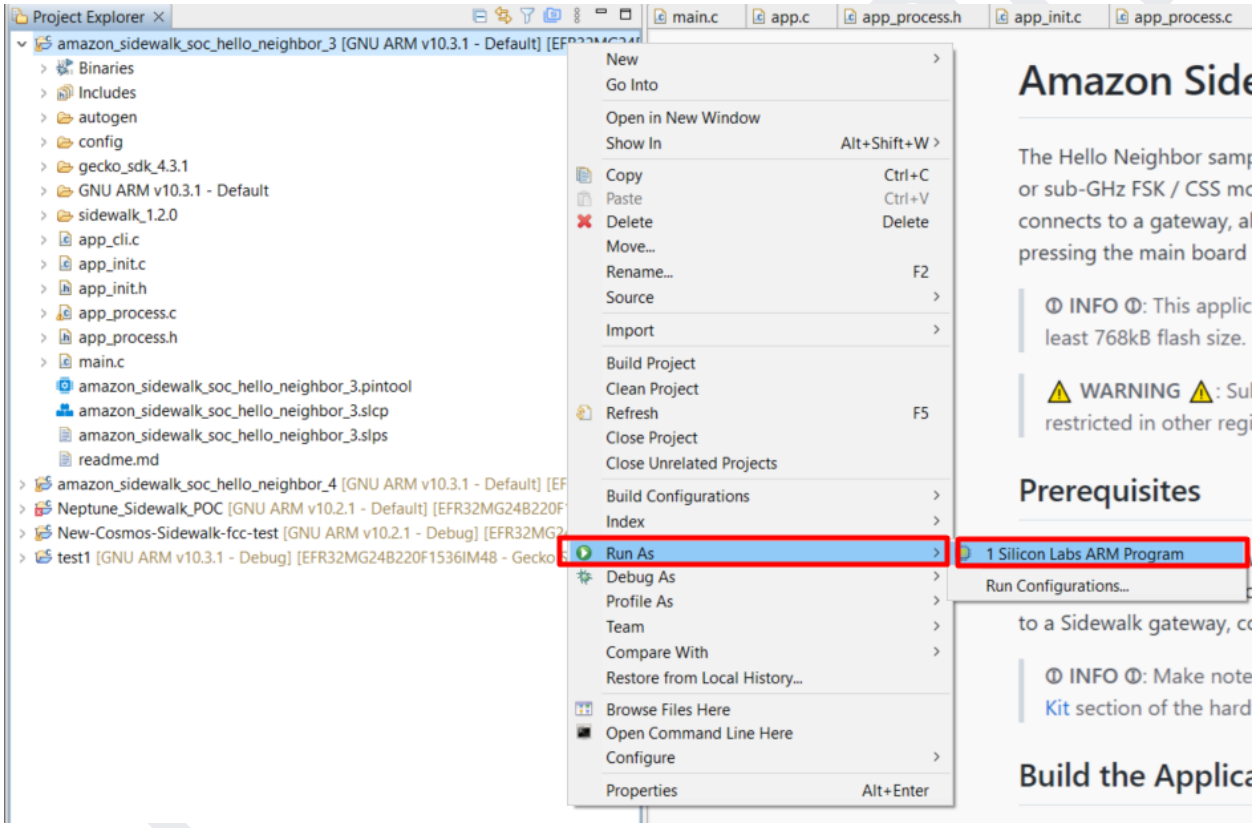
- Go to ***app_init.c***, in ***app_init(void)*** function and remove ***app_button_press_enable();*** callback.

4.1.5 Build and Flash your Amazon Sidewalk Application

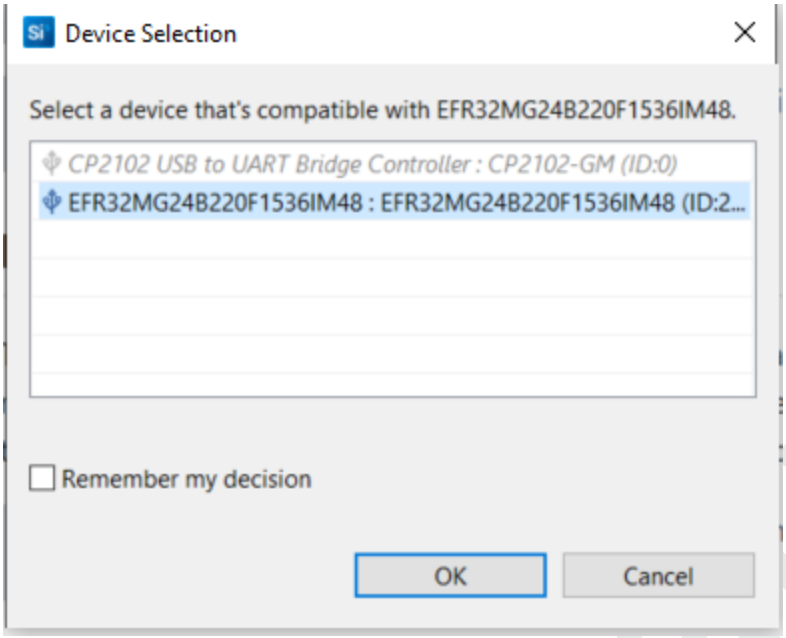
- After modifying the Amazon Sidewalk application you can now successfully build the project by clicking the hammer button on the Simplicity studio.



- After the build is successful, the generated binaries can be flashed by right-clicking on the project and then going to ***Run As*** > ***Silicon Labs ARM Program***.
  Note: Make sure that the EVK board is powered up and J-Link is connected to it.



- Select the device and click *OK*, the binaries will be flashed on the EVK.



4.2 Serial commands - NCP Mode

In NCP mode, these commands are called by the host MCU on the firmware level to communicate with the Amazon Sidewalk network. Refer to the API reference manual for details.

- Join
- Get Event
- Transmit

The application MCU sends join commands to the MCM module and waits for the module to synchronize with GW, after the module joins the network application MCU can now send uplinks and receive downlinks.

4.2.1 Join Command

Application MCU sends the join command to initialize sidewalk libraries and begin synchronization with GW.

4.2.2 Get Event Command

This command is used by the application MCU to retrieve pending events related to:

- Join failure
- Join success (device synced with GW and is registered)
- TX Event (failure or success)
- Down Data
- Reset

4.2.3 Transmit Command

Transmit command is used by the application MCU to send an uplink request to the module.

5. Prototyping API

To automate device creation for prototyping, scripts are provided that create the necessary objects in the cloud and the manufacturing page. The required scripts are available in the Amazon Getting Started Github repository.

5.1 Prerequisites for API

- **Set up an AWS account**
- **Python 3.6 and above**
- **Install and Configure AWS CLI (version 1)**
  AWS CLI is used to manage sidewalk devices using Amazon APIs for example send a downlink to the device.
  Install and configure the AWS CLI on your system:
    - Getting started with the AWS CLI
    - Configure your AWS CLI

  For configuration:
    - You can find the AWS Access Key ID and AWS Secret Access Key in the *.csv* file that has the AWS account user credentials.
    - For default region name choose ***us-east-1***
    - For default output format choose ***json***

5.2 Provisioning

Before proceeding with provisioning you will have to fill out ***config.yaml*** file in the cloned repository with your details, you will need to modify below:

- **AWS_PROFILE**: Profile to be used during the stack creation. If you have a custom-named profile in your AWS CLI configuration files, replace 'default' with the name of your profile. Usually, you'd have just one profile named 'default'.
- **DESTINATION_NAME**: The Sidewalk destination used for uplink traffic routing. Can be any string.
- **HARDWARE_PLATFORM**: Use SILABS
- **USERNAME**: User's AWS login Username
- **PASSWORD**: User's AWS login Password.

To create manufacturing files and wireless IDs for your devices, navigate to the root directory of the cloned repository using your preferred terminal application and run the following commands.

1. python -m pip install --user virtualenv
2. python -m venv sample-app-env
3. cd sample-app-env\Scripts\
4. activate.bat
5. Navigate back to the root directory
6. python -m pip install --upgrade pip
7. python -m pip install -r requirements.txt
8. python -m pip install pyjwt -t .\ApplicationServerDeployment\lambda\authLibs
9. python env_check.py

You should run step 1 to 8 only one time, step 9 below can be performed each time you will need to create a new manufacturing file for a new device.

10. python EdgeDeviceProvisioning/provision_sidewalk_end_device.py --instances <number_of_instances>
    Note: replace <number of instances> with the number of manufacturing files you want to create, each manufacturing file corresponds to one wireless device ID.

The script will create a directory structure as follows in the *EdgeDeviceProvisioning* folder:

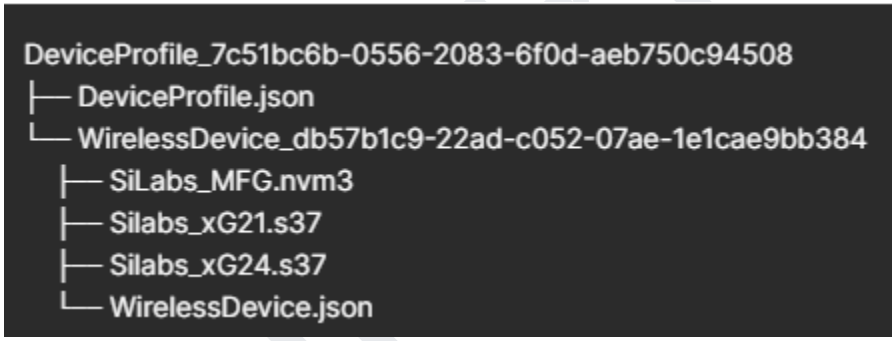
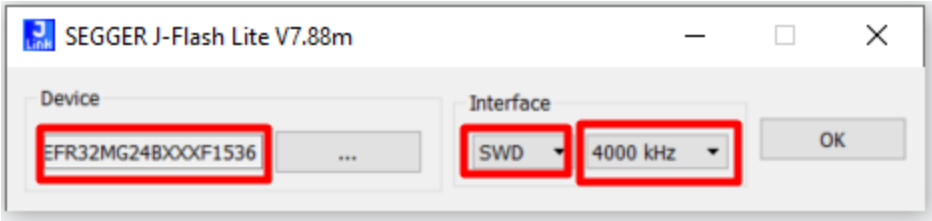


Figure: Directory structure

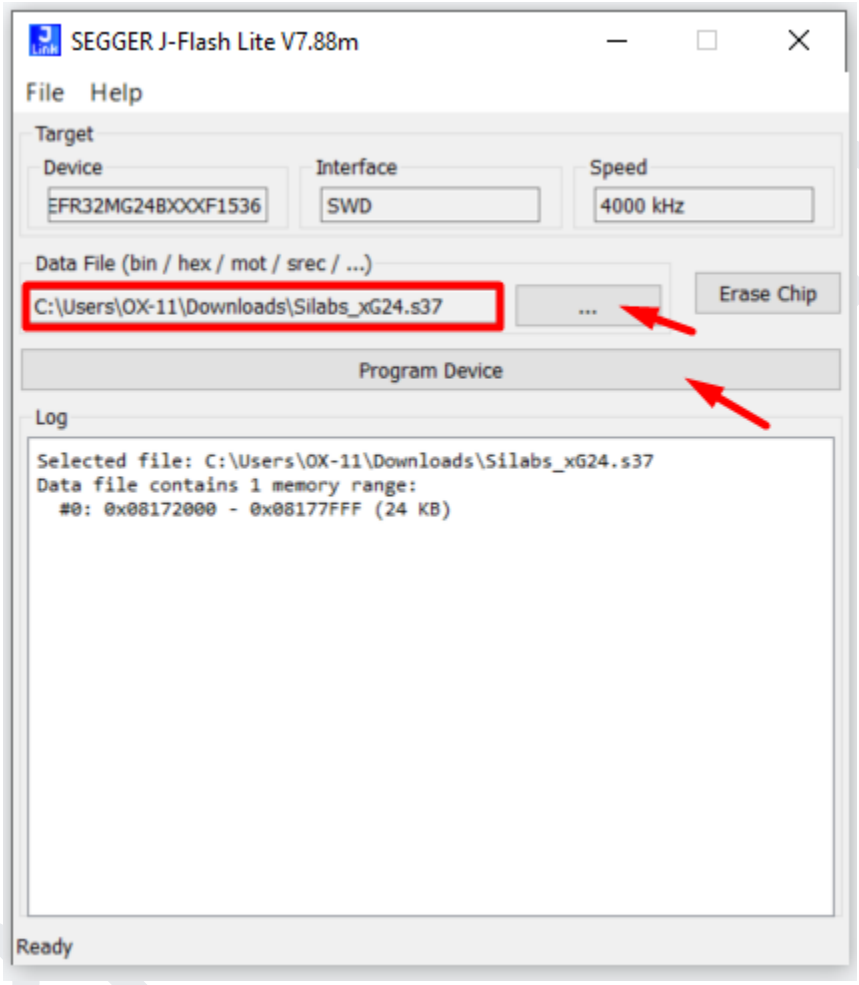
The first directory is named with the Device ID of your device profile and contains *DeviceProfile.json*, which contains your device profile information. Then a directory is created for every wireless device you created. One wireless device contains manufacturing pages for all supported platforms and a WirelessDevice.json file containing your device information (including private keys).

For the MCM EVK board, you will need to flash the *SiLabs_xG24.s37* file.

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



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



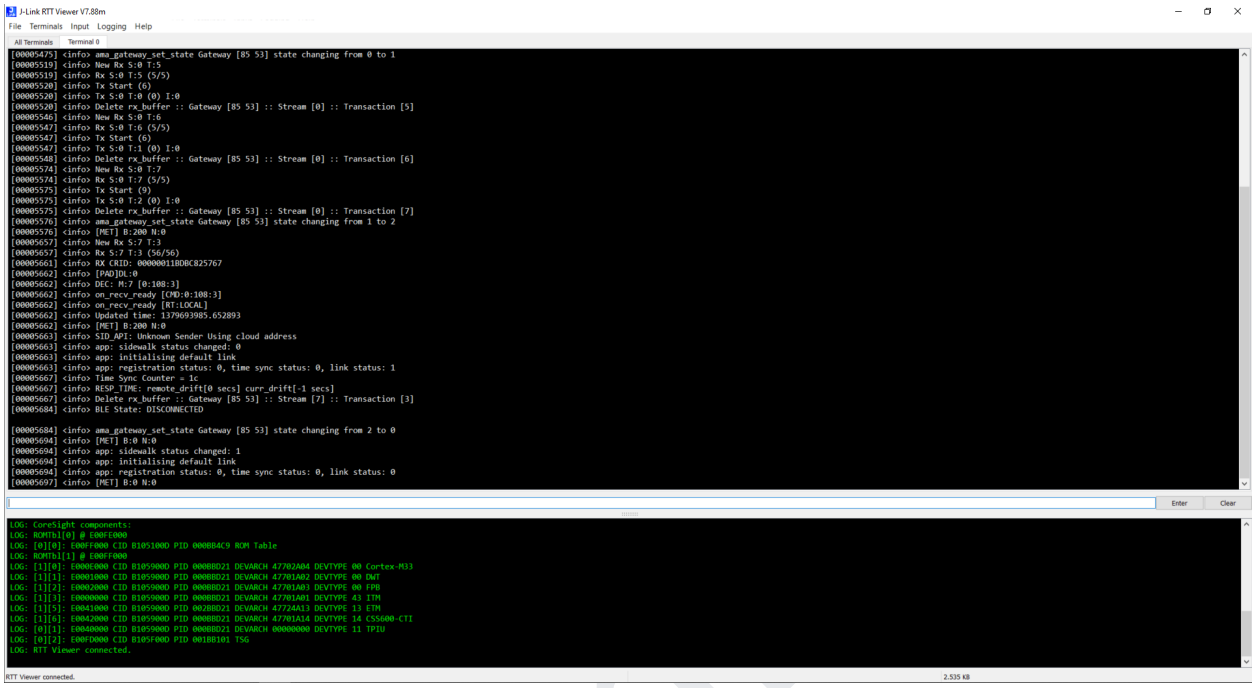
6. View the Application Logs through J-Link RTT

J-Link RTT interface is used to view Sidewalk application logs.
To set up the communication between your PC and the EVK, follow these instructions.

- Install the J-Link RTT Viewer.
- 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.



7. Test the Amazon Sidewalk Application

Successful registration is indicated by the following log

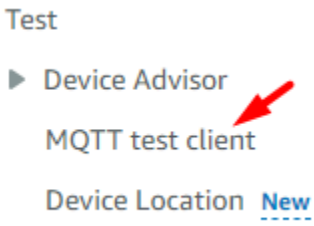
```
[00346968] <info> ama_gateway_set_state Gateway [9e 74] state changing from 2 to 0
[00346970] <info> [MET] B:0 N:0
[00346971] <info> app: sidewalk status changed: 1
[00346971] <info> app: initialising default link
[00346971] <info> app: registration status: 0, time sync status: 0, link status: 0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

By default, the Hello Neighbor application will start with the BLE default radio layer. You 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.

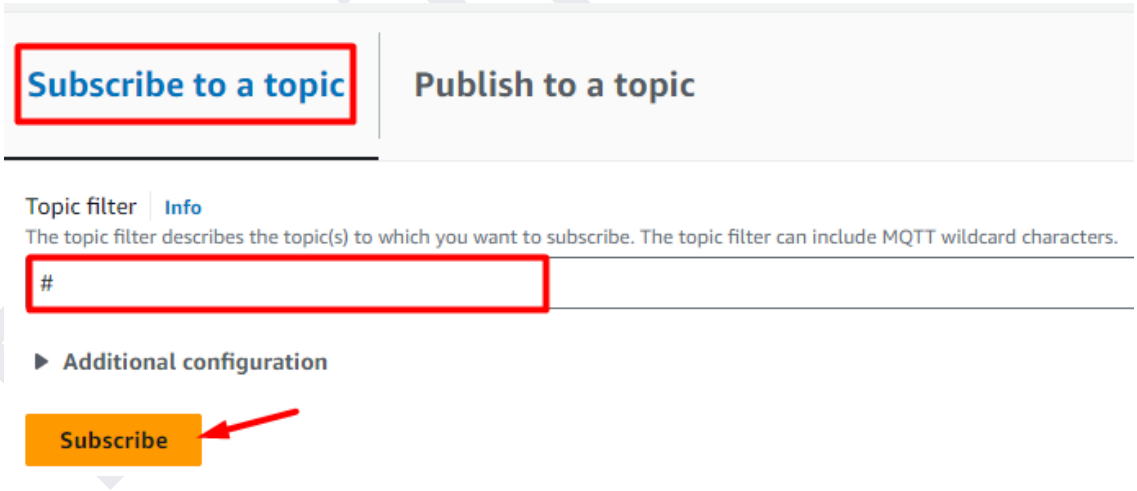
Command	Description	Example
switch_link	Switch between BLE, FSK and CSS modulation (depending on supported radio, switch order is BLE->FSK->CSS)	> switch_link
send	Connects to GW (BLE only) and sends an updated counter value to the cloud	> send
reset	Unregisters the Sidewalk Endpoint	> reset

The uplinks sent by the EVK board can be viewed on AWS using the MQTT test client

- Sign in to your AWS account and go to IoT Core
- Click on the "*MQTT test client*" from the left side menu



- In the topic filter, write # and click on subscribe



- The uplinks will appear on the AWS as shown

▼ project/sensor/observed


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

8. CONFIDENTIAL & PROPRIETARY

This document contains confidential and proprietary information belonging to Oxit LLC. It is intended solely for the use of the individual or entity to whom it is addressed and may contain privileged and confidential information. Any unauthorized review, use, disclosure, distribution, or copying of this document, in whole or in part, is strictly prohibited. If you have received this document in error, please notify the sender immediately and delete the original message and any attachments.

By accepting and reviewing this document, you agree to be bound by the terms and conditions stated herein. Failure to comply with these terms may result in legal action and remedies under applicable laws.

This document does not constitute an offer, agreement, or commitment of any kind, unless expressly stated otherwise. The views and opinions expressed in this document are those of the individual author and do not necessarily reflect the official policy or position of Oxit LLC.

Oxit LLC,

3131 Westinghouse Blvd, Charlotte, NC 28273

https://oxit.com/