WIRELESS & SENSING PRODUCTS

LoRa Basics™ Modem Software Development Kit User Manual

Table of Contents

List of Figures

1. Document History

Version	ECO	Date	Changes and/or Modifications
1.0	ECO-060462	2022-02-09	First release
2.0	ECO-063654	2022-10-14	Update to align with SWSD001 v2.0.0
2.1	ECO-064308	2022-11-02	Indicate WW2G4 supports only class A
2.2	ECO-064891	2023-01-31	Add LR1121 and SX1280 references

2. Introduction

This document explains how the LoRa Basics™ Modem (LBM) Software Development Kit (SDK) works and how it is configured. It also exposes usage of LoRa Basics™ Modem in end-to-end applications, with a simple Node-RED® based application server.

The original SDK (SWSD001) has been split into two SDKs to improve modularity and clarity of purpose:

- SWSD001: LBM examples https://github.com/Lora-net/SWSD001
- SWSD004: Geolocation examples and LoRa Edge Tracker demonstration https://github.com/Lora-net/SWSD004

Note: The SWSD004 SDK contains LoRa Edge Tracker (LR1110TRK1xKS) code, but this is out of scope of this document.

2.1 Scope of this document

This manual is intended for application developers and may be used in conjunction with:

- LoRa Basics™ Modem User Guide
- LoRa Basics™ Modem Porting Guide
- LoRa and LoRaWAN: A Technical Overview

This manual refers extensively to the various *README.md* files located in LoRa Basics™ Modem SDK project.

Note: Migration of application from LoRa Basics™ Modem-E to LoRa Basics™ Modem

The SWSD001 project contains details regarding the differences between LoRa Basics™ Modem-E and LoRa Basics™ Modem. It helps migrate applications from LoRa Basics™ Modem-E to LoRa Basics™ Modem, but that is out of the scope of this document.

2.2 LoRa Basics™ Modem

The LoRa Basics™ Modem products let your LoRa®-based Internet of Things (IoT) devices communicate with LoRaWAN networks, and with Semtech's LoRa Cloud™ services.

LoRa Basics™ Modem (SWL2001[1]) is an open-source software library that implements the LoRaWAN protocol stack for end devices. As well as supporting LoRaWAN regions defined in the README.md file of the LoRa Basics™ Modem library, the LoRa Basics™ Modem also proposes a WW2G4 region that takes advantage of the 2.4GHz frequency band. Only LoRaWAN class A operations are supported with this WW2G4 region.

LoRa Basics™ Modem is designed to be agnostic from the hardware architecture and from the end-device application, making it simple to integrate with a custom platform and use case.

[1] At the time of writing this document, the compatible version of LoRa Basics™ Modem is v3.2.4 available https://github.com/Lora-net/SWL2001.

2.3 LoRa Basics™ Modem Software Development Kit

The LoRa Basics™ Modem Software Development Kit (SWSD001) is a collection of LBM code examples, with accompanying README files, that expose the main LoRa Basics™ Modem features. Most examples relate to the MCU-side application, and one example is a cloud-side Node-RED® application server example:

Name	Description
Almanac update	Perform a partial almanac update
Application server	Configure the application server
DM information	Configure periodic DM message reports and trigger asynchronous report
Full almanac update	Perform a full almanac update
Large file upload	Transfer a file (up to 2048 byte long) to LoRa Cloud
LoRaWAN	Send data on a periodic basis
LoRaWAN asynchronous	React on an asynchronous event - press a button and send data
LoRaWAN asynchronous LR-FHSS	React on an asynchronous event - press a button and send data - and support LR-FHSS datarates
LoRaWAN class b	Enable unicast class B feature
LoRaWAN multicast class b	Enable multicast class B feature
LoRaWAN multicast class c	Enable multicast class C feature
Stream	Send data as a stream to LoRa Cloud
Time synchronization	Get time from the network server or from LoRa Cloud
TX RX continuous	Uses LBM test functions for TX and RX continuous operations
LoRaWAN region switching	Demonstrate switching from a LoRaWAN region to WW2G4 region, and back to initial region

The README.md file of SWSD001 details the validation status of the SDK examples.

2.4 Scope of the SDKs

The SDK is designed to:

- Help application developers understand what is required to integrate a LoRa Basics™ Modem feature
- Help application developers understand which services of LoRa Cloud are available through LoRa Basics™ Modem, and how to use it
- Provide an abstraction layer between LoRa Basics™ Modem and the host MCU
- Implement a shield abstraction layer allowing to support multiple Semtech shields

---

Note: The SDK examples should not be used for the following use cases:

- Direct integration in a final product
- Performance evaluation

---

3. SDK system

This section provides a description of the system interacting with the SDK. It covers the hardware and software used and also outlines the example code file and folder structure in the SDK.

3.1 Overview

The embedded firmware in the SDK interacts with the complete LoRaWAN system (including the *end device*).

The complete system is composed of three main elements:

- End Device
- LoRaWAN® network
- Semtech LoRa Cloud™ Modem & Geolocation Services

In the following figure, Fig. 3.1, the SDK provides a firmware collection that runs on an *End Device* and on an *Application Server*. The firmware built from the SDK is a binding of LoRa Basics™ Modem and different customer applications, and are executed by the *Host MCU* of the *End Device*.

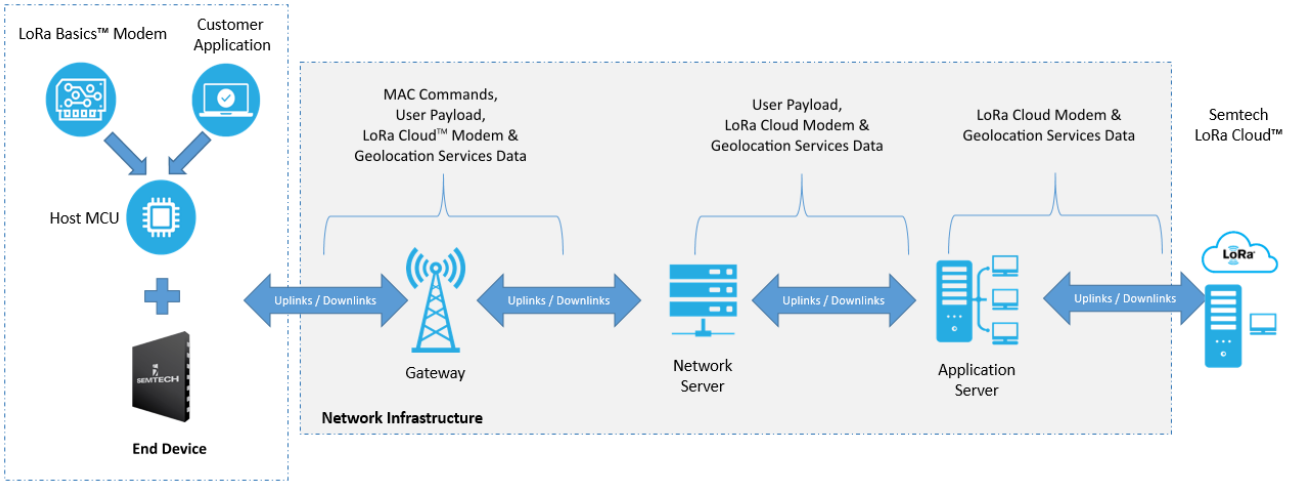

Fig. 3.1: Simplified SDK System Architecture

For more details about LoRaWAN architecture, refer to the document LoRa and LoRaWAN: A Technical Overview.

3.2 Required Hardware

The code examples included in the SDK work with the following hardware:

- Nucleo-64 circuit board L476RG
- Two LR1110 MBED Shields are supported:
  - LR1110MB1DxS: available in development kits LR1110DVK1TCKS and LR1110DVK1TBKS
  - LR1110MB1GxS: available in development kit LR1110DVK1TGKS
- Two LR1120 MBED Shields are supported:
  - LR1120MB1DxS: available in development kits LR1120DVK1TCKS and LR1120DVK1TBKS
  - LR1120MB1GxS: available in devevlopment kit LR1120DVK1TGKS

- Two LR1121 MBED Shields are supported:
  - LR1121MB1DIS available in development kits LR1121DVK1TBKS and LR1121DVK1TCKS
  - LR1121MB1GIS available in development kit LR1121DVK1TGKS
- SX126X Long Range Low-power LoRa RF transceivers:
  - SX1261MB1BAS: available in development kit SX1261DVK1BAS
  - SX1261MB1CAS: available in development kit SX1261DVK1CAS
  - SX1261MB2BAS: available in development kit: SX1261MB2BAS
  - SX1262MB1CAS: available in development kit: SX1262DVK1CAS
  - SX1262MB1CBS: available in development kit: SX1262DVK1CBS
  - SX1262MB1DAS: available in development kit: SX1262DVK1DAS
  - SX1262MB1PAS: available in development kit: SX1262DVK1PAS
  - SX1262MB2CAS: available in shield: SX1262MB2CAS
  - SX1268MB1GAS: available in development kit: SX1268DVK1GAS
- One SX1280 MBED Shield is supported:
  - SX1280RF1ZHP: available in development kit SX1280DVK1ZHP

An example of this hardware combination is shown in Fig. 3.2 (note that the touch screen is not required).

Fig. 3.2: Supported LR1110 Shields

3.2.1 Upgrading LR11XX Microcodes

The LR11XX must run the correct transceiver micocode. At the time of writing this document, the SDK is compatible with the following silicon microcodes:

- LR1110: transceiver microcode SWMC010 version 3.7.0
- LR1120: transceiver microcode SWMC020 version 1.0.1
- LR1121: transceiver microcode SWMC021 version 1.1

An updater tool is available to execute the upgrade of the LR11XX microcode. Its usage is described in the README file of the SWTL001 Github page.

The updater tool also comes in pre-compiled binaries available on the SWTL001 Wiki page.

3.3 Required Software

To compile the embedded code examples, at least one of the following components is required:

- an `arm-none-eabi-gcc` toolchain with the GNU Make tool - The SDK is known to compile with arm-none-eabi-gcc 10.3.1
- *Keil* toolchain - The SDK is known to work with µVision v5.33

To use the application server, the following are required:

- Node-RED®
- Node-Red-contrib-loracloud-utils

For hardware and software requirement details, refer to the SDK *README.md files*.

3.4 SDK Application Code Files

The *apps/common* folder provides functions that configure LoRaWAN arguments such as the join parameters (DevEUI, JoinEUI and AppKey), and initiate and handle event management. However, this event management only provides the virtual structure and it needs the specific applications to provide their own callback functions to do their specific work.

The sub-folders located in *./apps/examples* and in *./apps/demonstrations* also contain *README.md* files that provide information about each example. The *README.md* files are the primary source of information about the SDK examples. Readers looking for details concerning an example should check the corresponding *README.md* file first.

In the SWSD001 SDK, the root folder contains a *README.md* file that provides information common to all code examples. The following table details the *apps* folder layout:

- apps/common
- apps/demonstration/region_switching
- apps/examples/
  - almanac_update
  - application_server
  - dm_info
  - full_almanac_update
  - large_file_upload
  - lorawan
  - lorawan_asynchronous
  - lorawan_asynchronous_lr_fhss
  - lorawan_class_b
  - lorawan_multicast_class_b
  - lorawan_multicast_class_c
  - stream
  - time_sync
  - tx_rx_continuous

3.5 Configure, Build and Run

The following high-level steps detail the configuration, build, and run processes for an example project.

Details of the configuration examples, including the parameters that can be changed, are provided in their respective *README.md* files in the sub-folders located in *./apps*.

Note: Before starting to build an example, check the parameters in the common configuration file. Make sure that the LoRaWAN configuration is consistent with what is configured in your network server and in your LoRa Cloud account.

3.5.1 Configure

Follow these steps to configure the example code:

1. Determine which LoRaWAN network server to use, which demo to run, and which machine will run the application server (if applicable).
    - Ensure that there is a gateway capable of receiving your device's signal at the location where the example will run, and for the LoRaWAN network server selected.
    - The machine that runs the application server must be capable of exchanging data with the LoRaWAN network server, and with the LoRa Cloud™ Modem & Geolocation Services (if applicable).
    - Refer to *Node-RED Application Server Configuration* for the setup and configuration of the Node-RED Application Server example.
2. Get tokens from LoRa Cloud™ Modem & Geolocation Services. For more information, refer to the Tokens section at: LoRa Cloud™ Modem & Geolocation Services
3. Configure the application server with the tokens and URLs (if applicable). For more information, refer to the LoRa Cloud Configuration section at: LoRa Cloud Configuration
4. Register your device on the LoRaWAN network server.
    - The Device EUI, Join EUI and Application Key must be recorded as they are used later to configure the device code example.
5. Configure the LoRaWAN network server to communicate with the application server. For more information about the LoRaWAN network server, refer to the walkthrough at: LoRa Basics™ Modems: Walk-through
6. Configure the chosen example code. Refer to the corresponding *README* in the SDK.
    - Add the LoRaWAN credentials recorded before in the file *apps/common/lorawan_key_config.h*.

3.5.2 Build

Follow these steps to build the project:

1. If required, change parameters in the embedded code. Refer to the corresponding *README* to know which parameters can be changed.
2. Build the project with command `make -j` or use the Keil IDE (when applicable). The path of the compiled binary depends on the toolchain used:
    - With `make`: *apps/<EXAMPLE_NAME>/makefile/build/<EXAMPLE_NAME>.bin*
    - With Keil: *apps/<EXAMPLE_NAME>/MDK-ARM/build/lbm_example_<EXAMPLE_NAME>.hex*

Note: The build `make` call can have an `EXTRAFLAGS` argument to configure LoRaWAN and example parameters.

3.5.3 Run

The last step before running the project is to flash the build to the target. Follow these steps to flash the binary to the Nucleo board:

- If built with `make`, copy the binary file into the Nucleo board.
- If built by Keil, use the Flash feature of the IDE to flash the binary file into the Nucleo board.

All the SDK examples print log messages to the Serial port. The configuration is 921600/8-N-1.

3.6 Configure LoRaWAN Network Server and Application Servers

The section provides configuration information for the following servers:

- LoRaWAN Network Server
- Node-RED Application Server example

3.6.1 LoRaWAN Network Server

The LoRaWAN Network Server is at the core of every LoRaWAN Network that enables connectivity, management, and monitoring of devices, gateways, and end-user applications. Its main objectives are to ensure the security, scalability and reliability of data routing throughout the network. For more information on configuring the LoRaWAN network server refer to the LNS Modem Walk-Through page.

3.6.2 Node-RED Application Server Configuration

The Node-RED Application Server example provided by the SDK in folder *apps/examples/application_server* is an example of application server that interfaces between LoRaWAN® Network Server and the LoRa Cloud™ Modem and Geolocation Services. It forwards uplinks to Lora Cloud Modem & Geolocation Services server and potential downlink requests from Lora Cloud Modem & Geolocation Services server.

The details concerning the Node-RED Application Server example are provided in the README file located in *apps/examples/application_server/README.md*.

For more information on installing the Node-RED Application Server including dependency refer to the Application Server Modem Walk-Through page[1].

3.6.3 Import Application Server Code into Node-RED

After the Node-RED Application Server has been installed and is running, you need to import the application server files.

The following steps outline how to import the application server files:

1. Click the menu in the top right corner of the screen.
2. Click **Import**.
3. Choose file *apps/examples/application_server/modem.json*.
4. Select current flow, and click **Import**.

[1] The referred document addresses LoRa Basics™ Modem-E, however the configuration of LoRaWAN network server remains valid for LoRa Basics™ Modem.

3.6.4 Use the Node-RED Application Server example

Once deployed, the application server runs alone and does not need intervention. When the LoRa Network Server it is connected to receives an uplink, the application server receives a MQTT message.

This reception of the MQTT message triggers a HTTP request to the LoRa Cloud Modem & Geolocation Services. The LoRa Cloud Modem & Geolocation Services may respond with a downlink request that is propagated automatically by the application server to the LoRaWAN Network Server.

For more information on using the application server refer to the README file in the SDK *apps/examples/application_server/README.md*

4. Glossary

ADR
Adaptative Data Rate

ALC sync
Application Layer Clock synchronization

API
Application Programming Interface

DM
Device Management

GNSS
Global Navigation Satellite System

GPIO
General Purpose Input/Output

MCU
MicroController Unit

SDK
Software Development Kit

SPI
Serial Peripheral Interface

Index

Important Notice

Information relating to this product and the application or design described herein is believed to be reliable, however such information is provided as a guide only and Semtech assumes no liability for any errors in this document, or for the application or design described herein. Semtech reserves the right to make changes to the product or this document at any time without notice. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Semtech warrants performance of its products to the specifications applicable at the time of sale, and all sales are made in accordance with Semtech's standard terms and conditions of sale.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS, OR IN NUCLEAR APPLICATIONS IN WHICH THE FAILURE COULD BE REASONABLY EXPECTED TO RESULT IN PERSONAL INJURY, LOSS OF LIFE OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK. Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

Contact Information

Semtech Corporation
Wireless & Sensing Products
200 Flynn Road, Camarillo, CA 93012
Phone: (805) 498-2111, Fax: (805) 498-3804