H2S-Dev board Tutorials

LoRaWAN Helium network sensor development board

V1.2

Initial programming environment setup

1. Abstract

H2S-Dev board is a ready-to-use, Open source hardware, Arduino compatible, LoRaWAN Helium network sensor development board. It is capable of implementing a number of sensor and positioning applications.

This tutorial will guide you through the initial setup of the programming environment to run an example code and start developing your own application with LoRaWAN Helium network sensor development board - the H2S-Dev board.

2. Install Arduino IDE

Install or update your Arduino IDE to the most recent version.

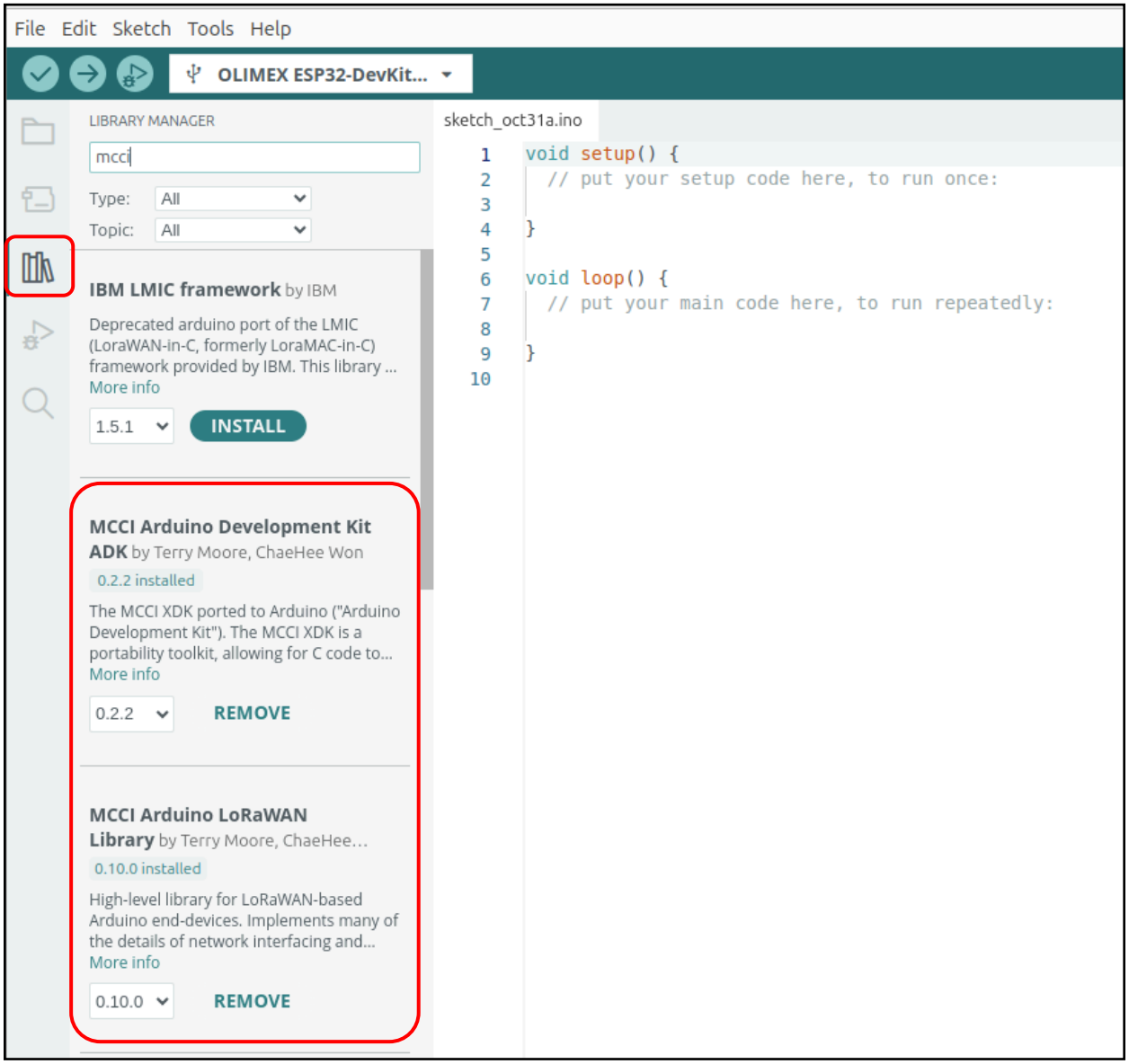
https://www.arduino.cc/en/software/

3. Install libraries required

A few libraries are required for the H2S-Dev board to start functioning.

3.1 Install LoRaWAN libraries for Arduino

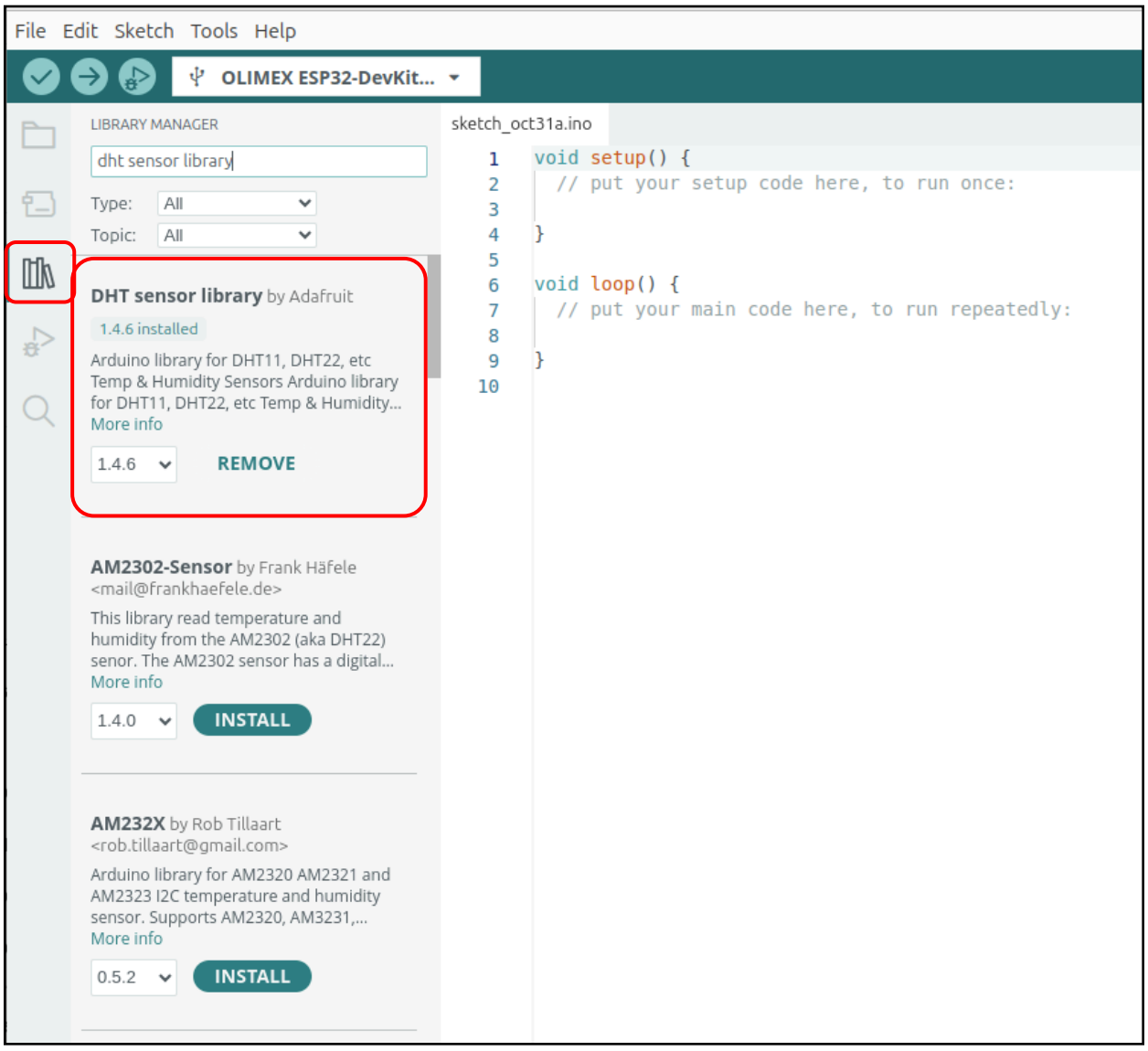
As per our experience so far, we recommend the following libraries to be installed together with all the dependencies required.



More information: https://github.com/hobbyiot/HELIUM-SENSORS

3.2 Install DHTxx library

For a specific application example - reading temperature and humidity from a DHT11 sensor, you will also need to install the corresponding library support. We recommend the THT sensor library from Adafruit. Please select the most recent version and install it.



More information: https://github.com/hobbyiot/HELIUM-SENSORS

4. Configure lmic_project_config.h

An important thing to set up before starting to use the H2S-Dev board is to correctly configure the project parameters for LoRa communication. Please find the *project_config* folder within the Arduino libraries installed as shown in the picture below and open the *lmic_project_config.h*

For a typical Linux installation the path to that file could be as follows:

/home/user/Arduino/libraries/MCCI_LoRaWAN_LMIC_library/project_config

Open the file *lmic_project_config.h* and set by comment/uncomment using '#':

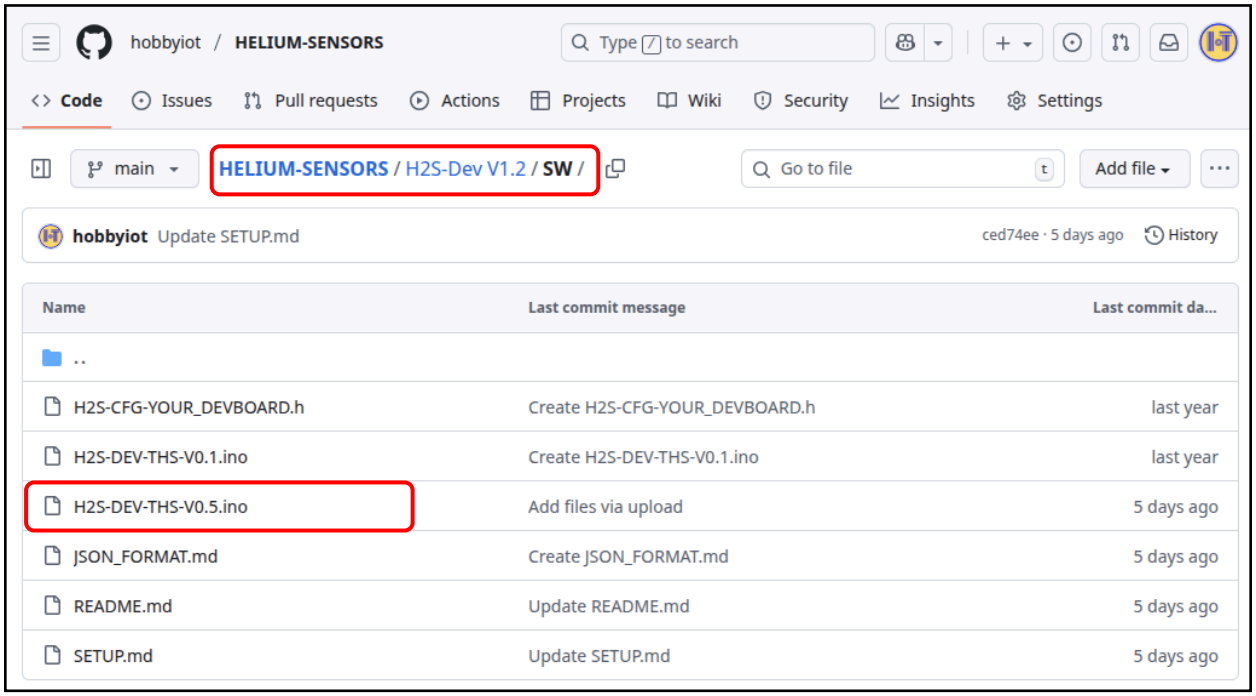
- **Frequency** set according to your country region;
- **Chipset** select *CFG_sx1276_radio*;
- **Interrupts** allow using interrupts by *LMIC_USE_INTERRUPTS*

More information: https://github.com/hobbyiot/HELIUM-SENSORS

5. Download the example code from the project Github repository

https://github.com/hobbyiot/HELIUM-SENSORS/blob/main/H2S-Dev%20V1.2/SW/H2S-DEV-THS-V0.5.ino

The example code provided uses a DHT11 temperature and humidity sensor and sends data over the Helium network in a specific format according to the Helium network requirements. DHT11 is programmed to be connected to GPIO13 in the example but you can change the configuration as per your needs.



The algorithm periodically reads the data from the connected DHT11 sensor. Temperature and humidity values are then combined into a specific JSON structure and sent over the Helium network.

The specific JSON format and various ways of receiving and using the data are all described in separate documents.

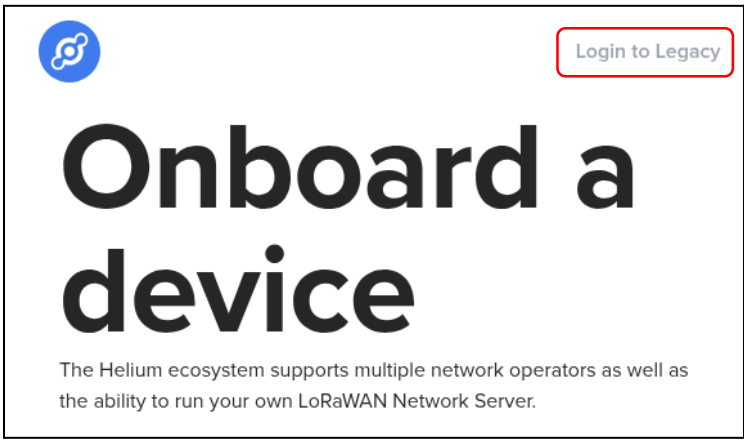
Generally, the data path is constructed as a set of logical rules (flows) within the Helium console environment and could be directed to a shared spreadsheet, MQTT broker, etc.

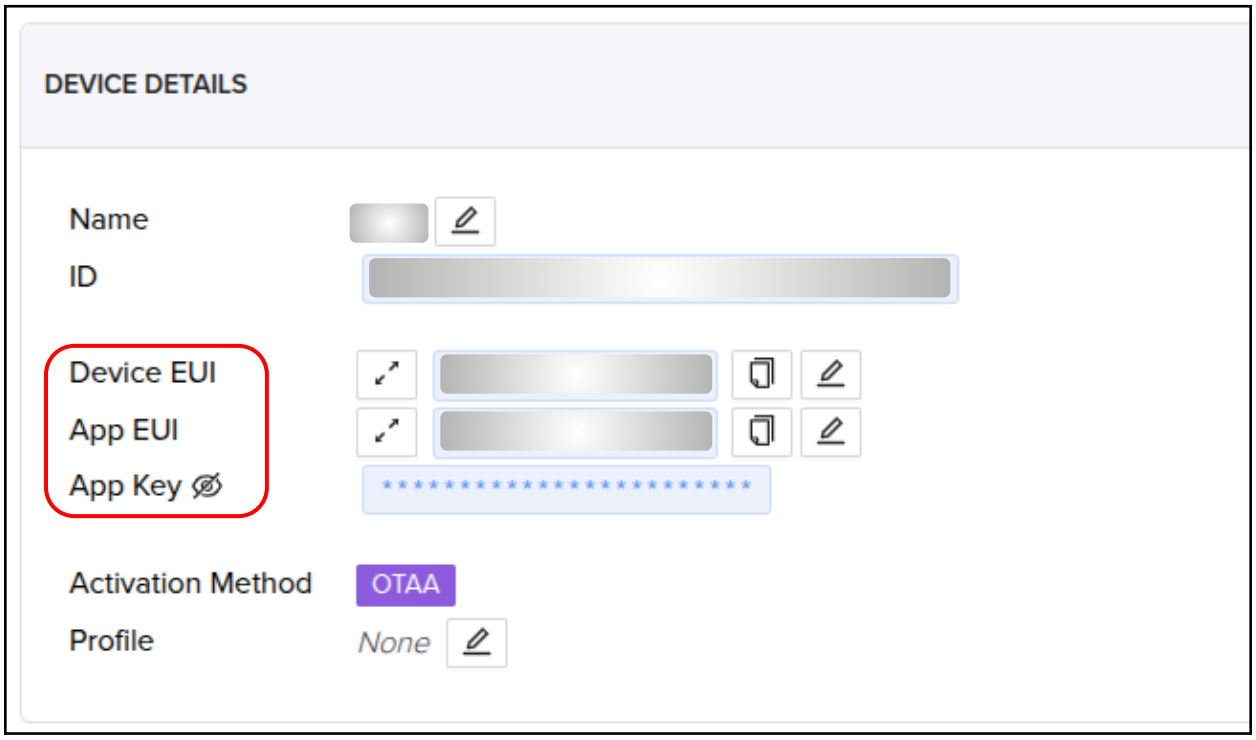
More information: https://github.com/hobbyiot/HELIUM-SENSORS

6. Set up a Helium network connection

To connect your H2S-Dev board to the Helium network, you need device credentials to be generated. Currently the system has been tested with the legacy console interface. Please use the following link to connect:

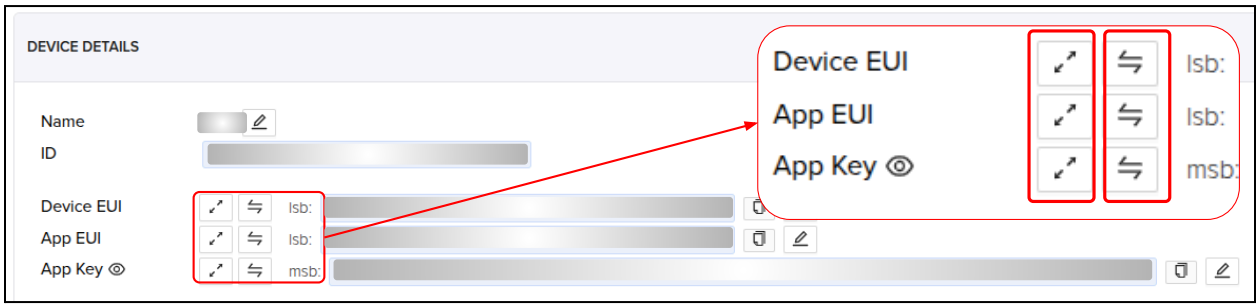
https://console.helium.com/



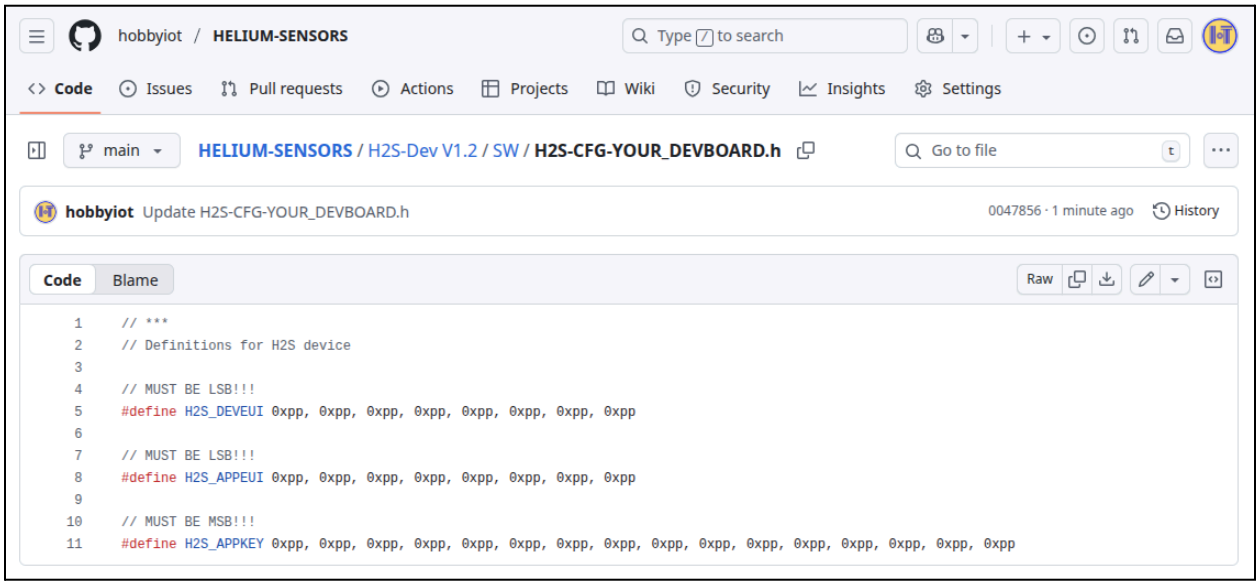


More information: https://github.com/hobbyiot/HELIUM-SENSORS

Use the “expand” key and select the proper order (MSB/LSB)

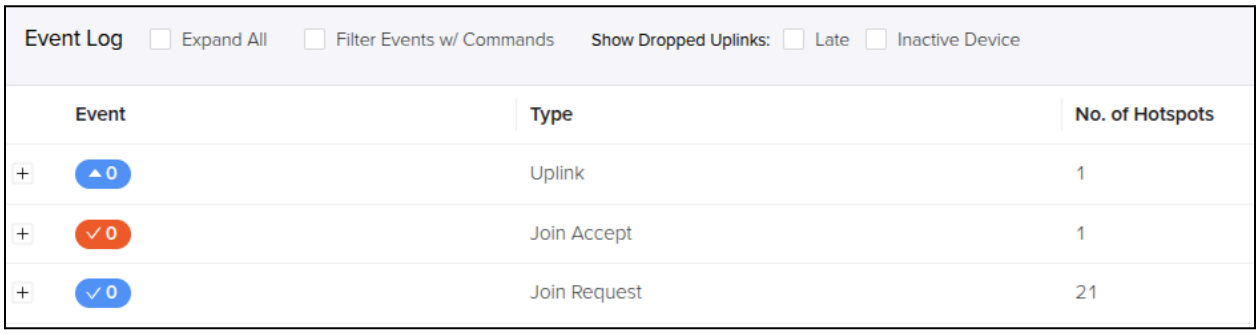


Copy and paste these values into the configuration file (H2S-CFG-YOUR_DEVBOARD.h)



7. Compile the project for ESP32_DevKit_LiPo

In case of correct settings you will see the new device join process within the Helium console.



Your H2S-Dev board is now connected to the Helium network! Good luck!

More information: https://github.com/hobbyiot/HELIUM-SENSORS