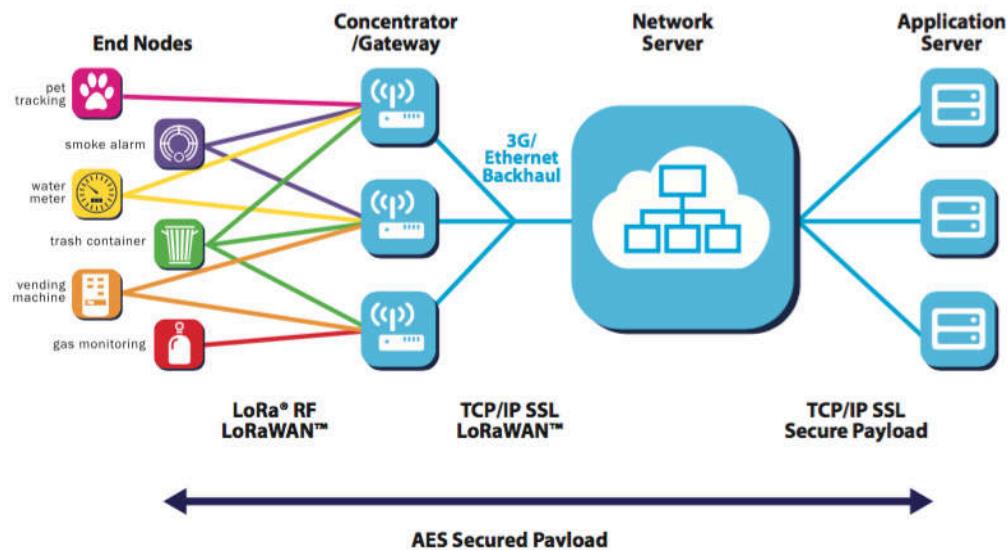


LoRaWAN Overview: End nodes are the sensors. The sensors read data and then send that data to the LoRaWAN gateways via the LoRa protocol. The gateways then send this data to the network server where it can then be used in the final application. The gateway must be connected to WiFi so that it can communicate with the server. Our end nodes are Heltec development boards (which can be integrated with other MCUs), our gateway is the SX1303 Gateway, and our server is TheThingsNetwork. We use ThingSpeak as our application server.



End Nodes

Our end nodes: Heltec Development Boards

- We use the CubeCell (ASR-650X) series, specifically the AB02S [2]
- Other Heltec devices should work as well (such as ESP32 series) - the main advantage of the Heltec development boards are that there are Arduino framework for them that allow for easy LoRaWAN integration and development

Setting up the Heltec (specifically the AB02S):

- Install the USB driver [3]
- Install the correct Arduino framework by following the steps in the Heltec documentation [4]

To establish LoRaWAN communication:

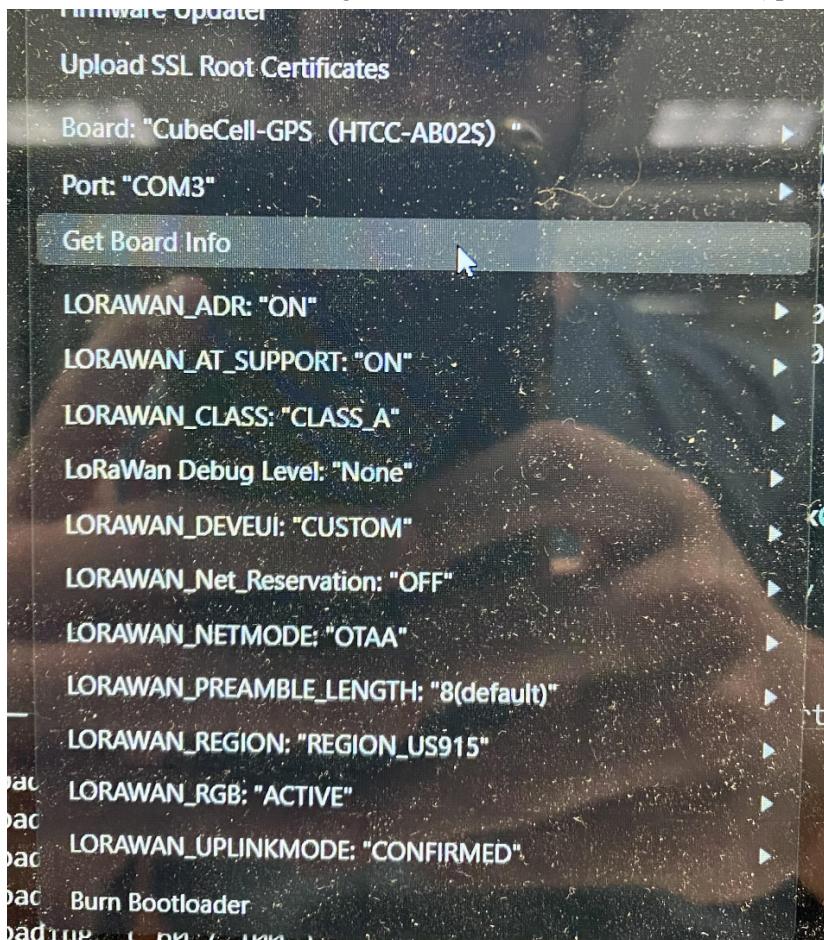
- Open the LoRaWAN example project

The screenshot shows the Arduino IDE interface with the following details:

- File Menu:** Contains options like New Sketch, Open, Examples, Save, Preferences..., Advanced, and Quit.
- Sketchbook:** Shows the current sketch name: GPS (HTCC-A...).
- Examples:** A large list of built-in examples categorized by number and topic. The category "LoRa" is currently expanded, showing sub-options like AT_Command, LoRaBasic, and LoRaWAN. The "LoRaWAN" option is highlighted with a blue selection bar.
- Code Area:** Displays the source code for the GPS (HTCC-A...) sketch. The code includes sections for battery voltage handling, appData processing, and various sensor and communication setup functions. It also includes comments about Heltec and Third Party sensors.

- Change the following line
- `uint16_t userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };`
- To
- `uint16_t userChannelsMask[6]={ 0xFF00,0x0000,0x0000,0x0000,0x0000,0x0000 };`

- So that the Board sends info on Sub-band 2, which is what TheThingsNetwork (our server) uses
- Ensure correct setting in Arduino under the “Tools” tab (specifically the REGION):



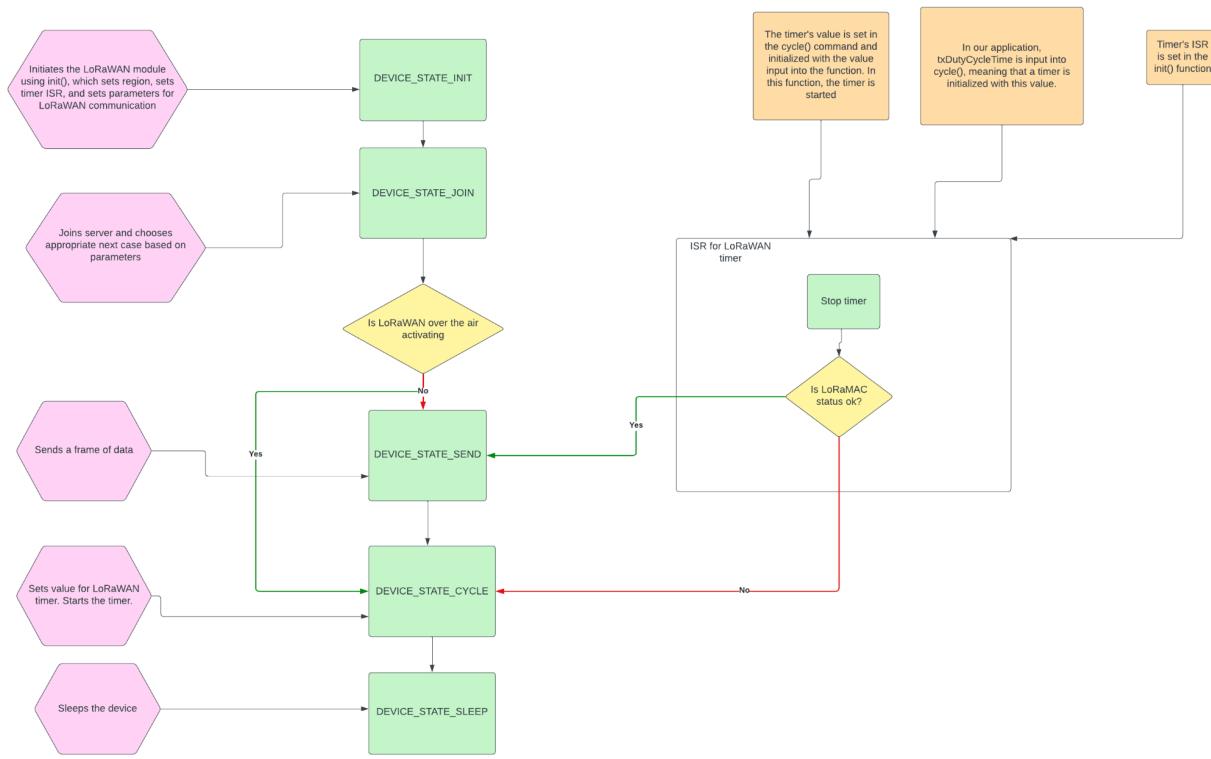
The main things to pay attention to in this code are:

-OTAA parameters, which we will need to change according to configurations on TheThingsNetwork (TTN). We use OTAA for communication rather than ABP because it is more secure

-the “appTxDutyCycle” variable, which specifies the approximate time in ms between transmissions to the server (approximate because there is a random small addition to this number to stop two devices from sending info at the same time). The default value is 15 seconds.

-the “prepareTxFrame” function, which prepares data to be sent to the server. To send the information that you want to the server, just change the appDataSize to the number of bytes that you want to send, and then send individual bytes of your sensor data. You will need to configure a payload formatter on TTN to decode this information.

-That is pretty much it, the rest is done by the program’s code. Below is a flowchart of the code:



Server

We use TheThingsNetwork as our server. We use a shared account that other teams can use. The gateway is already configured on this account. Link:

<https://nam1.cloud.thethings.network/console/applications>

Login Info:

User: lora23

PW: IoT4Ag

You will need to create a new application for your team. Once you do so, you can register your devices.

- Click register end device

Register end device

Does your end device have a LoRaWAN® Device Identification QR Code? Scan it to speed up onboarding.

[Scan end device QR code](#) [Device registration help](#)

End device type

Input method ⓘ

- Select the end device in the LoRaWAN Device Repository
- Enter end device specifics manually

End device brand ⓘ *	Model ⓘ *	Hardware Ver. ⓘ *	Firmware Ver. ⓘ *	Profile (Region) *
HelTec AutoMation	HTCC-AB02S(Cl...	Unknow...	1.0	US_902_928

HTCC-AB02S(Class A ABP)

LoRaWAN Specification 1.0.2, RP001 Regional Parameters 1.0.2 revision B, Activation by personalization (ABP), Class A



The Heltec HTCC-AB02S (Class A ABP) / CubeCell GPS-6502 is a LoRaWAN® development board based on ASR605x (ASR6501, ASR6502) that integrates a GPS module. The ASR605x chip is integrated with the PSoC® 4000 series MCU (ARM® Cortex® M0+ Core) and SX1262. HTCC-AB02S allows connecting various sensors and supports the Arduino® development environment.

[Product website](#) | [Data sheet](#)

Frequency plan ⓘ *

United States 902-928 MHz, FSB 2 (used by TTN)

Provisioning information

- Fill in the following information, though your model may differ - make sure you select your model with (Class A - OTAA) (more on the different classes in [5])
- Input all 0's for JoinEUI
- Have TTN Generate the DevEUI and AppKey for you (the End device ID will be autopopulated)
- Go back to the LoRaWAN.ino script and paste the TTN generated information into the devEui and appKey variables
- Register the device in TTN

The device will now show up in the application. If you click on the device, you will see that you can create payload formatters for each device so that the received information can be reformatted. We use the Custom Javascript formatter. Within the payload formatter and TTN, you can easily integrate with an IoT analytics platform called ThingSpeak, which has MATLAB integration and allows you to create custom visualizations and data analysis.

Gateway

The gateway is a SX1303 which was setup by a previous team. All that you need to know how to do with the gateway is to connect it to the server, assuming that it isn't already. The gateway will appear as Connected on TTN once the gateway's packet forwarder has been started.

LOCK	Overview	Applications	Gateways	Organizations
Gateways (1)				
ID	Name	Gateway EUI	Status	Created at
waveshare-sx1303-eedesign2	Gateway Attempt 3	00 16 C0 01 F1 60 F3 E5	Disconnected	Nov 2, 2023

Two options for starting the gateway:

1. SSH in
2. Start packet forwarder directly on gateway

Option 1:

- Gateway and computer need to be on the same WiFi network. You cannot use eduroam on the gateway, so you will either need to use your hotspot, or connect to the same wifi as in NEB212 that Prof Eisenstadt has provided. If you are using your hotspot, the gateway will not automatically join the first time, so you will need to go onto the gateway similar to in Option 2 so that you can manually connect to your hotspot.
- Once gateway and computer are on same WiFi:
 - Open up command prompt
 - Paste “ssh lora23@raspberrypi”
 - Enter the password “IoT4Ag”
 - Paste:
 - cd ~
 - cd ~/sx1302_hal/
 - cd packet_forwarder
 - sudo ./lora_pkt_fwd -c test_conf

The packet forwarder will now be running. When you look on TTN, the Gateway will be listed as Connected.

Option 2:

- Connect HDMI connector to gateway and connect it to monitor. Connect mouse and keyboard to gateway.
- Connect to hotspot or other WiFi (not eduroam)
- Open command prompt
- Paste
 - cd ~
 - cd ~/sx1302_hal/
 - cd packet_forwarder
 - sudo ./lora_pkt_fwd -c test_conf

The packet forwarder will now be running. When you look on TTN, the Gateway will be listed as Connected.

The gateway need to be connected to WiFi in order to communicate with the server and receive information from the end nodes.

References:

- [1] <https://docs.heltec.org/>
- [2] <https://docs.heltec.org/en/node/asr650x/index.html>
- [3] https://docs.heltec.org/general/establish_serial_connection.html
- [4] https://docs.heltec.org/en/node/asr650x/htcc_ab02s/quick_start.html
- [5] <https://www.thethingsnetwork.org/docs/lorawan/classes/>