Internship first week report
30/06/25 - 04/07/25

SUPERVISORS :
Bruno QUOITIN
Aqeel AHMED

AUTHOR :
Maxime BARTHA

1 Initial objectives

My initial objectives during this first week were to :

- set the goals for the internship
- look at what as been done last year in details
- decide on what to work on

Last year, the method to capture LoRa frames was to detect them by implementing an algorithm found in [1].

This year, we decided that I will be implementing a time synchronisation method to classify the captures of LoRa frames.

2 Material and software used

Material :

- sdr : pluto, usrp, rtl
- arduino MKR1310

Software :

- gnu radio companion : a block based programming software for signal processing.
- inspectrum : an open source spectrogram

3 Summary

I've managed to implement a proof of concept of the time synchronisation method explained in section 4. I've learned about TCP socket connections, Serial Arduino communication and gnu radio programming.

4 Time synchronisation implementation

There is basically 3 parts to this implementation:

- The receiver side capturing the LoRa frames
- The transmitter side in charge of controlling which device should transmit
- The Arduino implementation handling the transmission

The transmitter is in charge of taking all the transmission parameters such as frequency, bandwidth, spreading factor, number of frames to transmit per devices (cycles) and synchronisation parameters such as phase and period.

The receiver is in charge of capturing the LoRa frames and classify them by device.

The arduino devices transmit LoRa frames when the transmitter tells them to.

4.1 Transmitter-arduino relation

Let's first take a look at the relation between the transmitter and the arduino devices :

Its first job is to communicate all the thransmission parameters to all the arduino devices connected to it by sending an InitFlag followed by the parameters.

Then the when an arduino is initiated, it waits for a SendFlag for it to transmit a single LoRa frame. After Sending it, the arduino goes back to waiting for a SendFlag until it sent all the frames it had to.

A state machine showing this relation can be found in the figure 1.

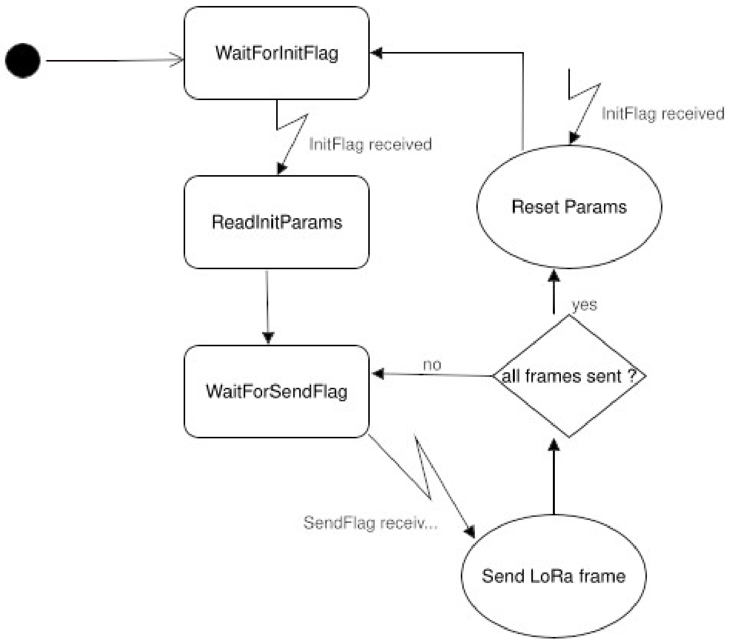

Figure 1: arduino state machine

4.2 Transmitter-receiver relation

First, the receiver sets itself as a socket server on a certain port. The transmitter connects to this socket using the port and the receiver's IP address. Then when the connection is done, the transmitter sends all the synchronisation parameters to the receiver such as :

- the starting transmission time in absolute time minus half a period
- a list of the devices ordered by device id
- the period between each transmission

Then on both sides, after the starting time is reached, the transmitter sends LoRa frames each period and the receiver saves each capture based on the device id.

To know which device is currently sending, the frames are sent by cycling through the device id list.

4.3 Saving structure

A saving directory and scenario name have to be given. The structure can be seen in figure 2

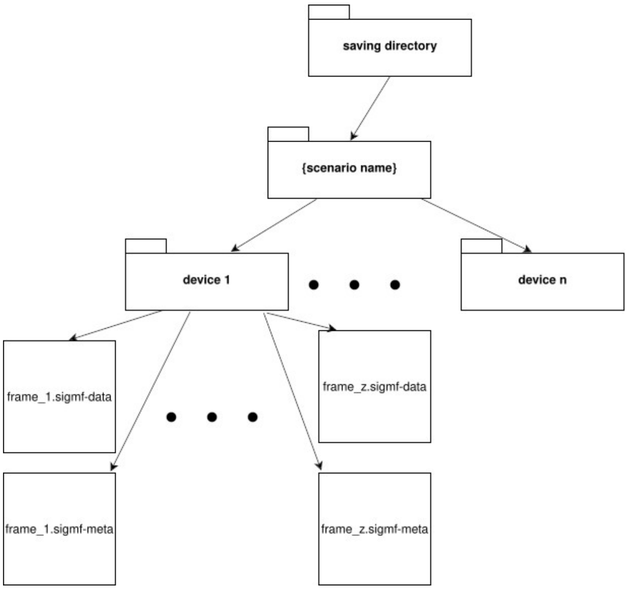

Figure 2: saving structure

Each .sigmf-meta files contains :

- the data type (c32_le)
- the frequency
- the sample rate
- the capture time
- a description taking the distance, outdoor/indoor and other possible parameters

4.4 Improvements

Some improvements can be made in this implementation :

- use implicit header for the LoRa frame
- check when receiving a frame that the expected device id is the device id sent in the frame (needs decoding)
- make the arduino state machine and the error handling more robust

5 Problems encountered

Serial ports issues when connecting certain devices

6 Conclusion

In conclusion, the implementation of the time based synchronisation is not yet robust and still needs some attention. The most important aspect is "is the data collected reliable". It's the main question I will be trying to answer in the second week.

References

[1] Zhenqiang Xu et al. "From Demodulation to Decoding: Toward Complete LoRa PHY Understanding and Implementation". In: *ACM Trans. Sen. Netw.* 18.4 (Jan. 2023). ISSN: 1550-4859. DOI: 10.1145/3546869. URL: https://doi.org/10.1145/3546869.