Project Report
in the context of Internet & Security

LoRaSense

Gathering and Distribution of Environmental Data

Group 9: Leonardo Salsi & Patrice Delley

July 19, 2020

Abstract

Environmental data can be collected in various ways. Within the scope of the project the goal was defined to collect weather data at several locations. Normally this is done at weather stations with the nodes not interacting with each other thus making it hard to collect all data. We created a network where at every so-called node all the information is present. We did this with the help of LoRa. The data from the sensors is read and sent to a central node where it is evaluated. Furthermore, the nodes should be able to receive commands and process them accordingly, such as the time span between the individual measurements making it easy to change settings from afar without needing to reprogram and upload to single nodes. Our project is able to measure temperature, humidity, air pressure and light intensity. The communication between the nodes is stable thanks to LoRa and a very small loss of data was noticed. The environmental data can then be visualized with the help of software "SensUI".

1 Introduction

LoRaSense is the collection of nodes that are capable of collecting environmental data and sharing information with each other via a LoRa network. Each node should not only gather data and distribute, but also receive information from other nodes, such that all nodes hold the same set of data. This leads to a decentralized system. Furthermore, LoRaSense must allow to extract said data in order to sort and visualize it without needing to visit or have a direct connection to every node.
The central goal of our contribution can be summarized in

Can we create a network where environmental data is collected and shown from multiple locations and distributed in order to guarantee data consistency over the entire network?

This question was too open in its formulation for the beginning, which is why we decided to split it into sub-questions to which we also referred as milestones. By answering one of the following question we made a significant step towards the realization of the project.

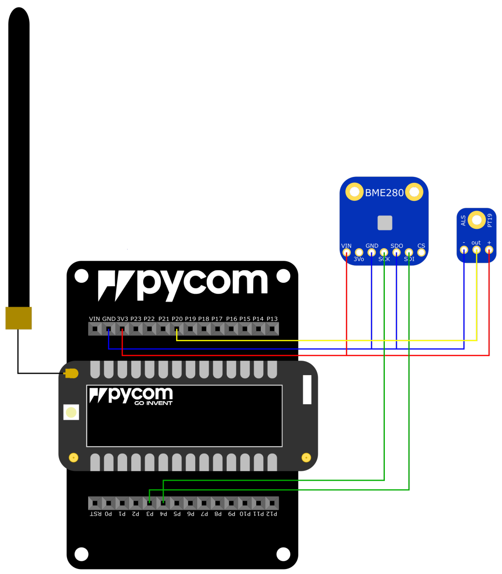
1. What information do we want to collect?
2. What hardware is suitable for our project?
3. How can we represent the data?
4. Is LoRa the suitable transmission technology for us?
5. What are the necessary interfaces in order to allow easier implementation in other projects?
6. How can data consistency be guaranteed?
7. In the case that a node shuts down, what steps should be taken?
8. How scalable is LoRaSense?
9. How can we extract the data to a PC and visualize it?

In the early stage of development our group started collaborating with LoRaLink and with SensUI. LoRaLink focused on the interconnectivity between the nodes while SensUI developed a program that displayed the collected data in a detailed and comprehensive manner.

2 Background

Hardware

- *Pycom's LoPy 4* is a microcontroller that is able to run *Micropython* and has a native interface for LoRa connectivity. Along with the expansion board it provides pins allowing to connect sensor chips to the LoPy. It also included the LoRa antenna.
- The *BME280* is a chip that is able to gather information about temperature, humidity and air pressure. It provides a connection to the board either via *I2C* or *SPI*. In our project the first connection variant was chosen.
- The *ALS* is a chip that is capable to read the light intensity and translates it into a voltage that can be directly read from the pin.
- *Heltec LoRa 32* is a development kit similar to the LoPy 4 that is controlled by the arduino programming language. By default this module has a small display attached to it. We used this only at the beginning testing how to work with LoRa and grasping the general idea, since it was easy to debug due to the small display.



How we connected the BME280 and the ALS to the LoPy 4's expansion board.

Software

- The IDE used for development was *Visual Studio Code* since it was compatible with *Pymakr*, a tool that uploads to and runs code on LoPy.
- Since LoPy is not able to understand the entirety of Python, the project was written in an abridged version called *MicroPython* which is an abbreviated version of Python.

3 Implementation

3.1 Stages of Development

The development of LoRaSense can be split up into 5 phases. Each had a different main focus.

Phase 1: Early Stage

The early stage was mainly used for planning. We first needed to set up a concept to decide what kind of parameters we were going to measure. According to our decision, we chose the sensor chips that can be connected with our LoPy-modules. Due to LoPy only being able to work with MicroPython, we had to set up the environment and develop a sense for how it works and write first samples, such that when the parts arrive we could test the hardware.
At the end of the early stage, we knew *how* the tools are to be used.

Table of labor: Table 1 in the appendix.

Phase 2: Hardware Testing Stage

As soon as all hardware was delivered, we had to make sure that all parts were working according to our expectations. Since we both had to work remotely, one of us had to take upon the first hardware-related tests and implementations. Leonardo mostly worked on the LoPy 4 where Patrice worked with the Heltec LoRa 32. After this phase, Patrice also started working on the LoPy as the Heltec devices were not a promising option aside from debugging.
At the end of the testing stage, we were certain that all components were working as expected.

Table of labor: Table 2 in the appendix.

Phase 3: Prototyping

First code samples were written in this stage in order to make measurements of temperature and humidity only for testing purposes. LoRaLink was not implemented in this stage yet, which is why we sent data in raw format from one node to another via LoRa. This is also where we decided what form the data should take on when working with LoRaLink. With that, a protocol was formed that would later prove important for SensUI as well. This included all measurements as well as a time-stamp for each round of measurements. A library was created that serves as the interface for the BME280 sensor. Also, the network we developed was more extensive, since now connecting more than 2 nodes were possible.
At this point, we knew how the chips are connected to the LoPy and how LoPy sends data via LoRa and is able to be read on the console of a connected computer.

Table of labor: Table 3 in the appendix.

Phase 4.1: Implementation of LoRaLink

LoRaLink was responsible for distributing data through a network. We were provided an interface allowing us to store data on each node and thereafter it would update the other nodes with said information.

After this stage of development, each node of LoRaSense sent data to each other such that all nodes held the same dataset.

Table of labor: Table 4.1 in the appendix.

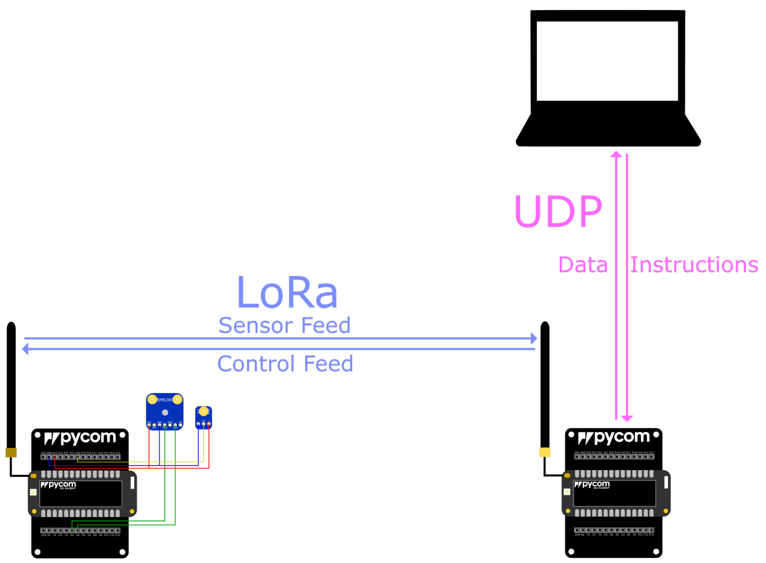
Phase 4.2: Inclusion of SensUI

We needed to send the data from the nodes to a computed and chose to use a central node, which resides in the same network as the computer SensUI is running on, that collects the data from all nodes. It also served as a sender such that commands could be sent to said nodes. We also helped with the setup for the UDP connection between LoPy and a computer, since we had some experience with it following our efforts in giving each measurement a time-stamp. The inclusion of SensUI was mostly between SensUI and LoRaLink though it was beneficial for us to be integrated into that workflow as we were able to help.
Hereafter we were able to give commands to the nodes as well as receive and display information gathered on different nodes. It was only ever tested with one node since we did not have more than 2 LoPy at any given place and with a configuration as such everyone was able to test it for themselves.

Table of labor: Table 4.2 in the appendix.

3.2 Set-Up Concept

We differentiate between sensor node and a master node. Sensor nodes collect and hold environmental data and feed them via the *sensor feed* to the master node, which in addition to the LoRa-network is also connected to the WiFi. This connection serves the purpose to forward collected data to a computer running SensUI via UDP. SensUI also allows to change the frequency in which measurements are made. If the user changes said frequency, the master node informs all sensor nodes via the *control feed* that said adjustments must be made; the sensor nodes then act accordingly.



4 Results and Discussion

We now can collect the following data:

1. Temperature
2. Humidity
3. Air Pressure
4. Light Intensity

And we decided the suitable hardware for the project would be:

1. LoPy 4
2. BME280
3. ALS

Our first prototype with Heltec modules worked faster than we expected. It did not take a long time to get it working and putting out random numbers generated on the device. The problem with Heltec LoRa 32 was that we could not connect sensors to the device. While not being the most elegant solution, we printed the data onto the serial monitor built into the Arduino IDE.

We tried around with different technologies such as WiFi as well but decided to stay with LoRa since it seemed to be best suited in our project with higher ranges, which would represent the real world.
As we started working with the LoPy 4, testing with random numbers through LoRa went well. Problems arose when we connected the sensors to the board. We created a library tasked with being the interface between the LoPy 4 and the sensors.

- LoRaLink was able to guarantee data consistency and safety through frequent updates of the nodes. It also was able to recover from a node that shut down (when the battery ran out). With only said node needing to be restarted (after swapping battery) would thereafter be functional again.
- SensUI was useful for giving us an interface with which we could issue commands to the second channel of communication. It also provided a graphical representation of the data collected.

The combination of the three individual projects was certainly time-consuming due to interface problems with connection for example. LoRaSense provides the measurements to LoRaLink which transport it to SensUI. Finally, SensUI then represents the transmitted data. We were able to achieve what we set out to do. We created a scalable network of nodes that collects weather data and is graphically represented.

5 Lessons Learned

A problem we encountered was a difficult setup of the LoPy 4 including its expansion board. Loading the original firmware onto it was even with a tutorial hard. When Patrice encountered an error not yet documented, he was able to find a solution through going on a forum and asking about it. Forums can be extremely helpful and should be encouraged to be used more.

Another aspect that stood out to us was working in groups. Now this is more or less normal but what made it interesting was multiple groups working together. Each having their vision and working on their own at first. Then coming together and working alongside each other to create something everyone can be proud of.
It has to be mentioned though that it is not always easy! Sometimes you have questions and want answers fast but we were lucky since all groups participating were always easily reachable and were always willing to go into a quick zoom-session.

Organization for the whole project was a little chaotic at first. The COVID-19 situation certainly did not make this any easier. For quite a long time it was unclear with which groups we would be working together. We believe that a complete overview from the beginning would have been helpful. Still given the special circumstances it was a really interesting experience with lots of problems that needed solving.

Not being able to meet up was quite a challenge, since we were unable to link up more nodes than shown in our set up. We ordered more breadboards needed to connect all sensors to the expansion board, but they failed to arrive in time.

6 Appendix

Table 1

	Leonardo	Patrice
Conception	x	x
Determining hardware	x	x
Raw connection scheme		x
First code samples	x	

Table 2

	Leonardo	Patrice
Component testing	x	x
First wiring	x	

Table 3

	Leonardo	Patrice
Hardware prototyping	x	
Environmental data formatting		x
Programming	x	x

Table 4.1

	Leonardo	Patrice
Inclusion of LoRaLink	x	x

Table 4.2

	Leonardo	Patrice
Interface programming	x	
Programming interpreter for commands		x

Here a few photos taken of the project.

Figure 1: Heltec LoRa 32 receiver display

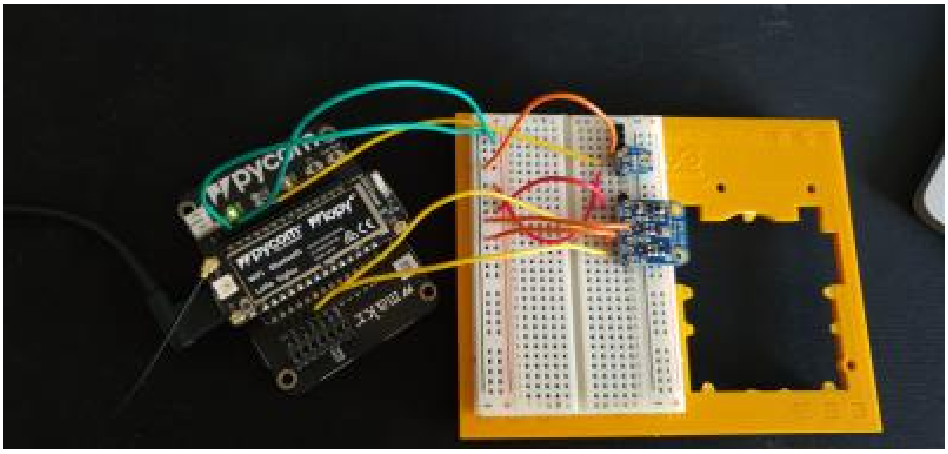


Figure 2: Setup for a measuring node