Technical summary

**Management and visualization of sensor data obtained through ChirpStack and LoRa communication protocol**

PI: Walter Daniel Leon-Salas ([wleonsal@purdue.edu](mailto:wleonsal@purdue.edu))

Author: Nicolás Miguel Murillo Cristancho ([mcristan@purdue.edu](mailto:mcristan@purdue.edu)) (+57 3227524655)

Colaborators: Nicolás Rendon Arias ([nrendona@purdue.edu](mailto:nrendona@purdue.edu))

Sehyeon Lim ([lim465@purdue.edu](mailto:lim465@purdue.edu))

**Abstract:** Obtaining and storing a large amount of historical data about sensors automatically is important to achieve efficiency in agricultural industry and electrical networks with solar panels in Perú, so is desirable to have a graphical interface accessible from the web that show current sensor’s data and analyze information over a time interval to make predictions. Displaying a large amount of data through the web and in an easy way is a challenge. Is needed to consider many devices, each one with a specific location and many sensors, and each sensor with many measurements per minute. The Heltec Wireless Stick device and LoRa communication protocol were used due to these technologies that allow to send data without needing internet connection to ChirpSatck, an IoT platform that allows manage sensors information. The http protocol was used to connect Chirpstack with a Web application developed in Angular (a NodeJS framework). Ngx-chart (an Angular open-source library) and google maps API have also been used to create graphs with historic data and display the locations of the devices. The combination of these technologies has made it possible to create a website where gauges, line graphs and maps interact with each other achieving an important objective: display current and historical data from any type of sensors (voltage, temperature, etc.) for monitoring and future predictions. The dashboard will be used for multiple IOT projects, in this case, electrical networks and agriculture. In fact, it can be used for any project that uses ChripStack, regardless of context.

**Introduction**

This project aims to show sensor data previously received through the LoRa communication protocol, which is a simple communication protocol, without the need for the internet and ideal for small packets of information, such as sensor data.

Low-power hardware such as arduinos, raspberry pi or mini pcs are convenient due to low costs, low power consumption and the software they will run will not be very demanding in terms of processing.

Web development is convenient because of its versatility to create programs accessible from the web and even locally. In this case it is convenient because the hardware used for the project could have an internet connection or not depending on the location chosen by the end user, so a single application in Angular that can run in the browser locally and remotely is a convenient tool.

Angular is a framework that uses 2 essential languages, html and typescript, its main advantage being the ease of communication between the front end and the back end of the code. In this project, we'll mainly use instructions in the html code to call back functions. We will also use the Angular components, each of these has its html file connected to its component.ts file that contains the logic of the program.

**Source code:**

The code of the entire project can be reached at github, in the following link: <https://github.com/nicolas23589/DashBoard.git>

Also can be reached at tinyLab hard disk if you had the mentor’s authorization.

This documentation can be also reached in the documentation folder of the github repository.

Remember that the source code is highly commented for an easy understanding of every specific part of the code.

**Methods**:

Hardware:

1. We will have several types of hardware devices:
2. Sensors: they can be for different fields of knowledge, such as electricity (voltage and current), crops (humidity, CO2, etc.) or location. These connect to a board using analog, digital, UART ports, etc.
3. Board: this device is programmable in arduino, for the moment you want to use the heltec wireless stick (although it could change). Its role is to read data from sensors and use LoRa to send the data to a gateway.
4. Gateway: A device more similar to a PC, in this case, you want to use the rakpi OS OS. Your role will be to receive the data from the different Boards and use the pre-installed ChirpStack software to send the data to a Windows PC.
5. Pc: preferably a low-power mini pc will be used, this will communicate via Wi-Fi with the gateway to receive the data and display it locally in the browser with an Angular application and/or create a web server so that this application is accessible from any device.

Software:

ChirpStack: An IoT platform that can run locally once installed on the gateway

Xaamp: a server for windows, we will use it to receive data (for this, we will have a php code that will receive the data by writing it in a temporary text file)

Python: we will have two files with python code, a dataManager that from time to time (for example, daily) cleans the temporary file where the data is received and stores said data with a better organization on the hard drive. The second code file will be responsible for reading the data from the temporary file and/or the data stored by the datamanager to change its format to one readable by the angular application. Then, an API is integrated into this code that exposes these files so that the angular app can use them.

Angular: This takes the data from the API through an http call and transforms it into objects of a previously defined class, and then uses libraries such as ngx-chart and google maps to display this data.

Note: in this document we will focus on the use of the xaamp server and the PC, details about the use of gateways, sensors and other processes prior to the arrival of the data to the Windows PC will be detailed in the reports of the other members of the group, mentioned at the beginning of this technical summary.

**How to run the project:**

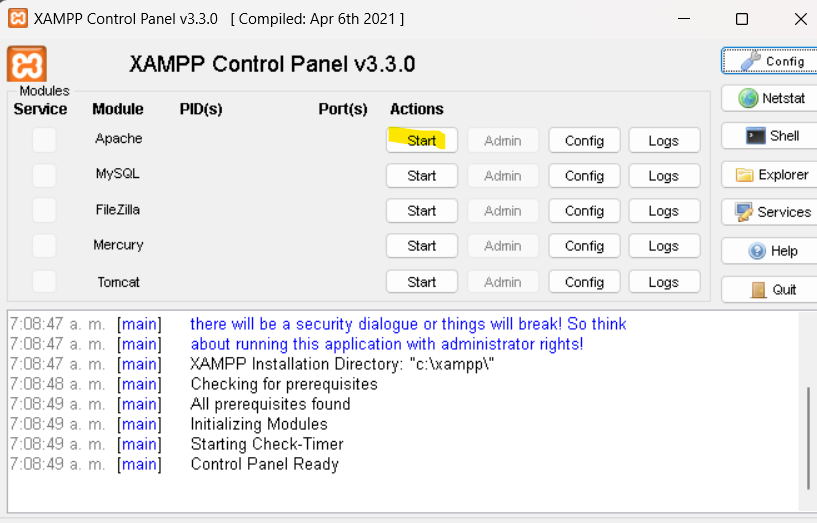
**Prerequisites:**

It is likely that you do not have most of the prerrequisites below, but don´t worry, all that are popular technologies and you can find a lot of tutorials on internet. Direct links and commands will be also provided below for faster installation but is recommended that you learn more about how these technologies works.

1. Install Node Js and npm: <https://nodejs.org/en/download/package-manager>
2. Using npm before installed, install angular with the command: npm install -g @angular/cli@17
3. Xaamp: <https://www.apachefriends.org/>
4. Python: <https://www.python.org/downloads/>

**How to execute:**

1. Run ChirpStack service (ask the asigned person specialized on chirpstack to execute the service)
2. Open xaamp and start the apache server:



1. Go to the root folder of the project and open a terminal, run the following commands:

python DataManager.py

python api.py

ng serve

1. In your browser go to http://localhost:4200/

**Tips and notes:**

-Visual Studio code is recommended as IDE to visualize and modify the code

-Remember to run the software programs in this order, otherwise the application couldn´t display any data looking like the web page is empty:

1. ChirpStack (ask the asigned person specialized on chirpstack to execute the service)
2. Xaamp
3. Python dataManager
4. Python apy
5. Ng serve command in the terminal (angular aplication)

**Results:**

Was achieved a complete and functional program that can display and storage any quantity of information comming from chirpstack, even the program was designed to be easy for modify, so it is possible to add more information sources in addition to ChirpStack in the future.

**Functional Features:**

The program can display the locations of all devices

The user can click the markrs or use a selector box to select the device.

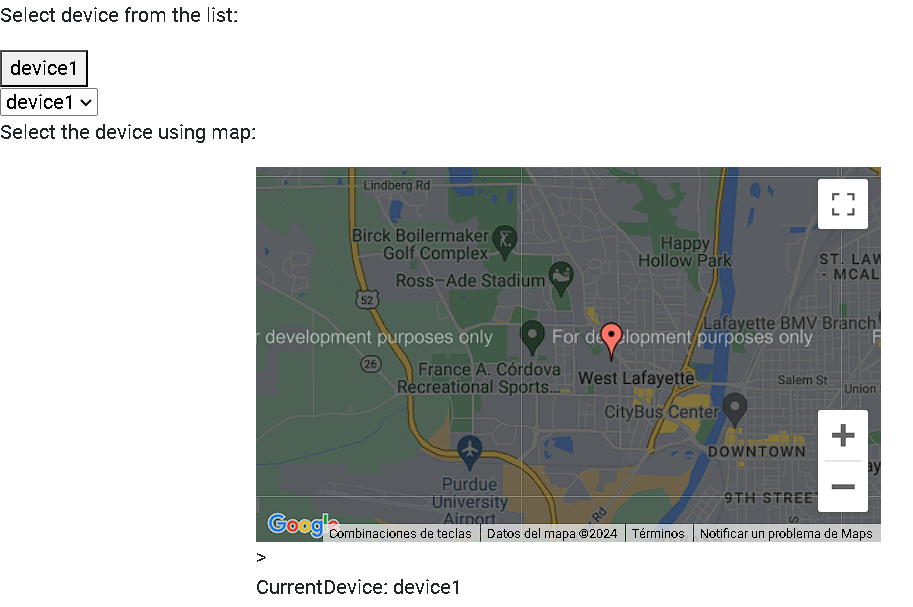
Is allowed to filter all the measurements by date

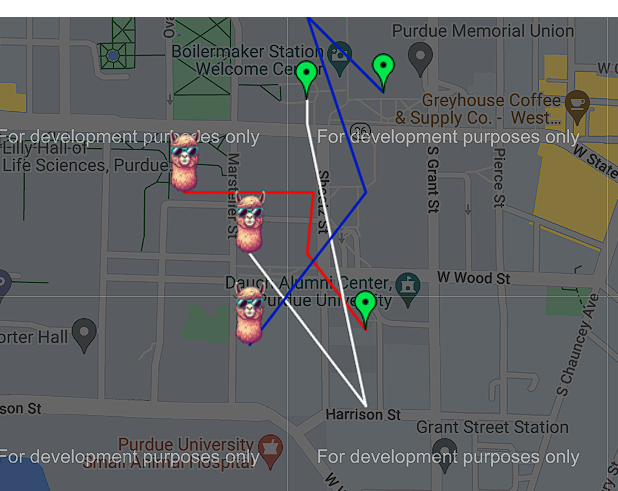
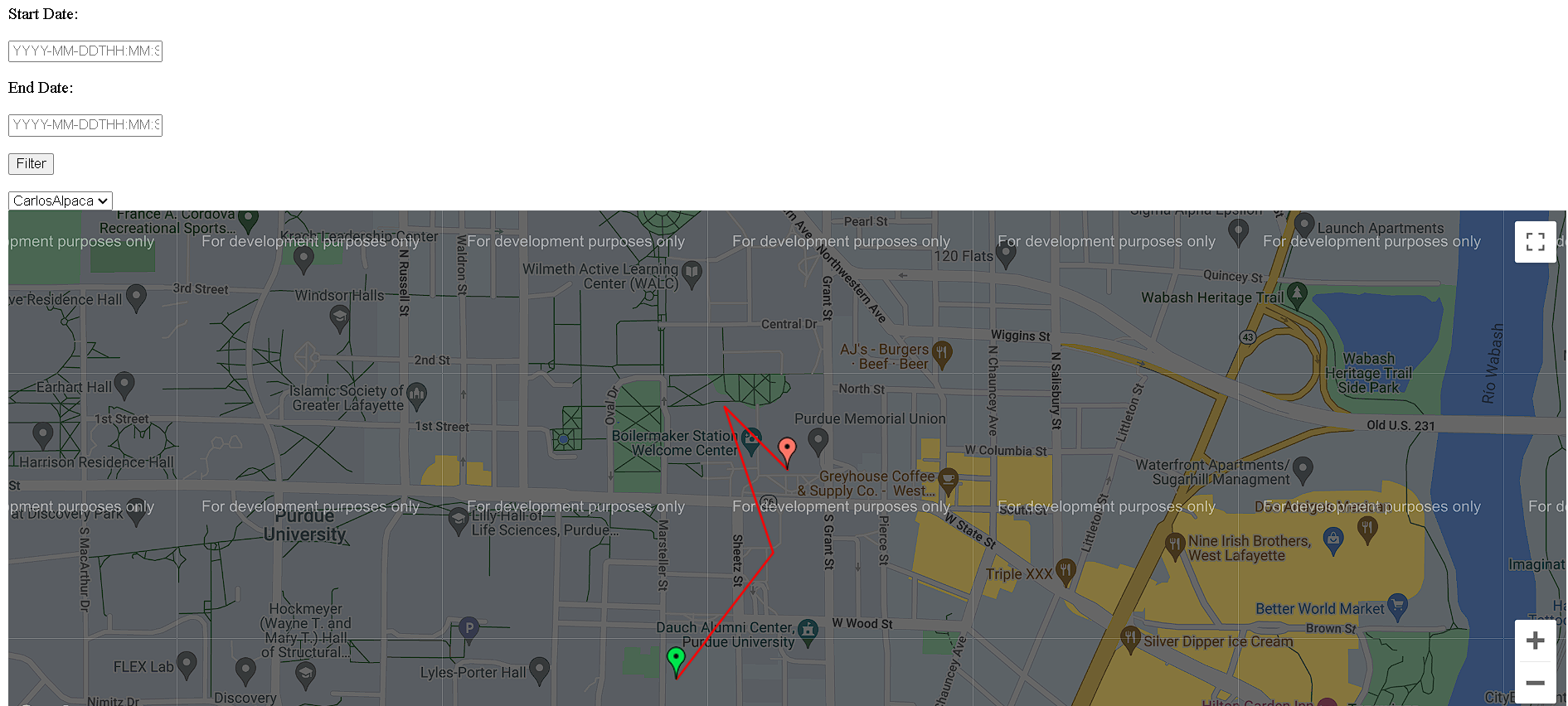
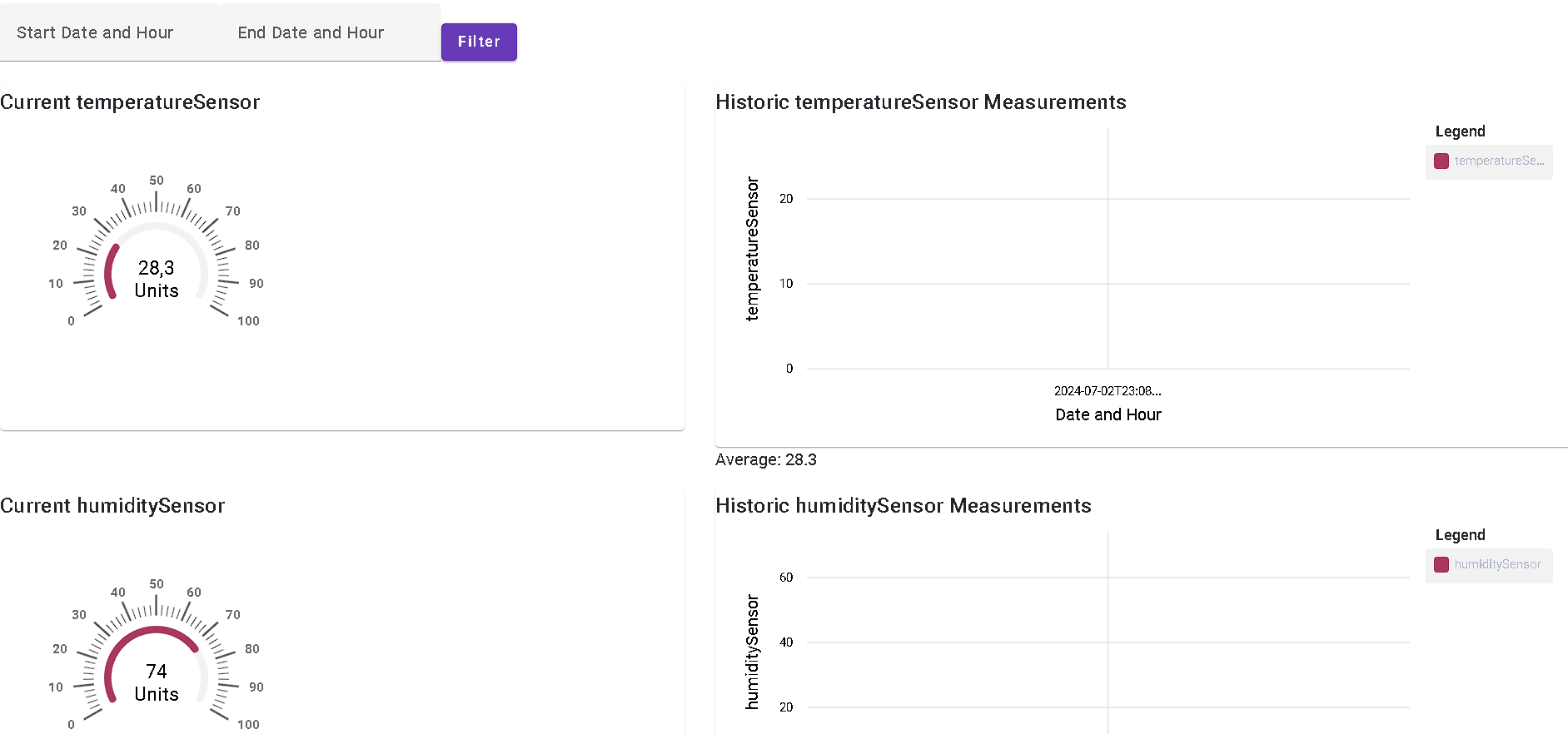
The user can see the historic measurements graph

The system saves a daily backUp of the data automatically

It shows the average measurement of every sensor in the range of dates provided by the user.

Another Version of the system was developed to display animal’s location (this is for another project, but using most of the same code)





**Quality attributes:**

-Modifiability (no big code changes or not absolutely changes required to add or remove new sensors)

-Performance (web pages loads in less than 1 second)

-Usability (Simple and intuitive interface)