ELEC6212: Wireless Sunflower Network

Benjamin Dix-Matthews*, Georgios Psimenos*, Ganiyu AJ Ibraheem* and Aswin K Ramasubramanian*
*University of Southampton

***Abstract*—The idea of this project was to create a device capable of ultra-low power sensing and ultra-low power transmission that is capable of harvesting the energy required for its operation from solar sources. The energy harvesting nature of the device would eliminate the significant operational expenditure required to replace batteries in periphery nodes after they have gone flat. As batteries are typically known for having relatively short operational lifetimes (¡10 years), our design would instead use a super capacitor in order to store the energy required for operation. This type of Zero-Energy Sensing is a new idea with huge potential in the IoT field.**

I. Introduction

THIS product would be aimed toward outdoor sensing applications where typical wireless sensing networks could be useful. A specific example of its usage would be a soil moisture detector for monitoring crops. The idea of the Wireless Sunflower Network would be to design a cheap product that could be easily set up by someone unfamiliar with electronics in order to build the sensing network (a plug and play solution).

bdm
April 25, 2018

A. Definitions

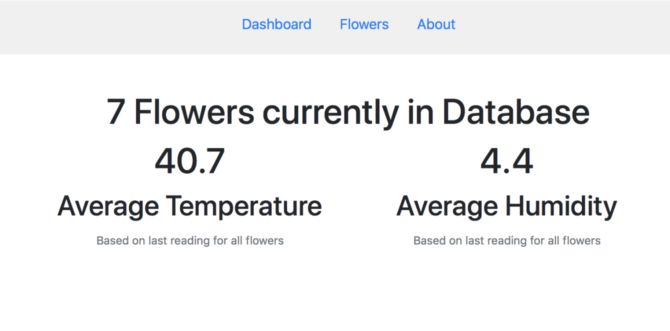
Internet of Things: IOT

II. Motivation

III. Architecture

IV. Communication

LoRa is a low power, long range telecommunication technology that is ideal for the type of energy stringent wireless networks being considered. The physical layer of LoRa (Long Range) is a proprietary spread spectrum technique derived from Chirp Spread Spectrum (CSS) and is owned by Semtech **borLORAforIOT** The standard MAC layer protocol is LoRaWAN and is an open standard being developed by the LoRa Alliance **ouluPetajajarvi** As this protocol is open source, ad hoc networks can be set up without having to wait for a network provider. The transmission distance, energy usage and data rate are all dependent on five configurable parameters. The Semtech data sheets **lora:sx1272/73** specify how these parameters affect power usage. Semtech also offer a calculator **loraCALCULATOR** that may be used to test individual parameter sets.



7 Flowers currently in Database

40.7

Average Temperature

Based on last reading for all flowers

4.4

Average Humidity

Based on last reading for all flowers

Fig. 1. Homepage for the dashboard

V. Energy Harvesting

VI. Data Handling

The data handling is organised into 3 sections: The first being a MongoDB data-store for handling all the incoming data from the flowers, the second being the dashboard for visualising the data available in the data-store and the last being a central server that communicates with a lora-module via serial-port, parsing and saving the lora packets and also acting as a web-server to the dashboard.

A. MongoDB Data store

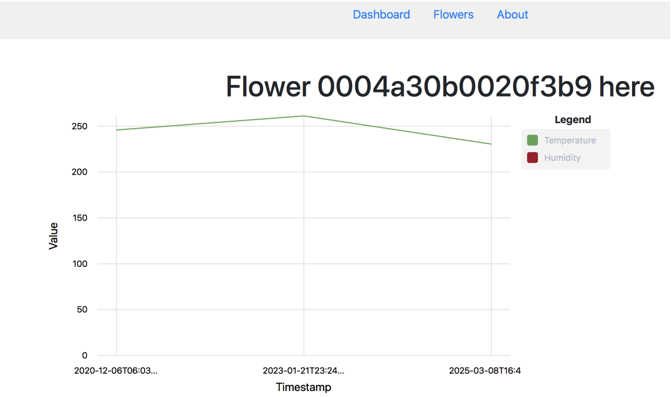
MongoDB is a No-SQL database that provides an expressive query language and flexible data-store that allows iteration of data models. It is a highly efficient database that supports millions of operations per sec and can handle petabytes of data as well being able to scale (grow) horizontally with support for database clusters.

This rich feature set provided us with a strong foundation as a data-store for the sunflowers for two main reasons. Firstly, during the development stage, it provided us with a medium for rapid prototyping as we were able to rapidly change the schema of the data that was being stored and due the distributed and highly scalable nature of MongoDB; It is theoretically possible for us to scale this data-store to support thousands of sunflowers.

B. Dashboard

The dashboard acts as an operator user-interface for navigating the data in the MongoDB data-store. One of the functionalities we encoded into it is support for analytics on the Sun flower data, this is illustrated in figure 1 where we aggregated data on the temperature and humidity.

The dashboard is built using web-technologies of which are AngularJS for the architecture and Ngx-Charts for the visualisations. The dashboard supports single-page apps and



Flower 0004a30b0020f3b9 here

Fig. 2. Sample detail page for a SunFlower

by default offline mode; highlighting the strong technologies used for the project.

Currently the dashboard only receives data from the central server as data currently flows from the sun-flowers to the server and then to the dashboard. In the future, bilateral communication could be added; In figure 2, the sample page for a sun-flower is shown there, a sample feature that could be added would be a button to shutdown the flower where this command propagates from the dashboard to the central server and then to the sun-flower.

C. Central Server

The central server is the main entry point for the data processing end. It is built using NodeJS, a highly scalable technology with first class asynchronous support; This is useful for our application where the packets are streamed as they arrive into the central server.

The pipeline of the central server from parsing the data to sending it to clients is described below and figure 3 shows the flow of data across the entire application from the lora modem to the dashboard client.

- The first thing the server does is to connect to the lora-modem via serial port and instructs it to listen for data packets from the flowers. It should be noted that the entire setup is asynchronous and data is just streamed into the server once it arrives.
- the central server receives it from the modem and parses it. The parsing is done with the assumption that the bits are stored in big endian format; Once this is completed, they are automatically saved into the MongoDB database where they update the data for an existing flower or create a store if the data is from a new flower.

The server exposes a REST(Representational State Transfer) API that enables it to communicate with any client that supports this protocol. For this project, we have built a dashboard app for the web that uses REST to communicate with the server but this could easily be adapted for another client such as an iOS mobile app. As the central server has a medium for communicating with the sun-flowers via the attached lora model, it is technically possible to enable the server pass commands on the flowers as mentioned earlier.

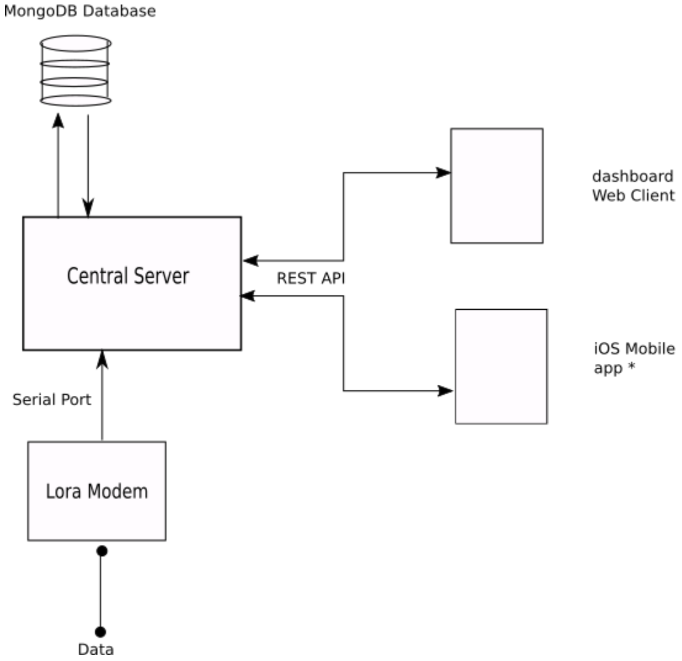


Fig. 3. Pipeline flow of data across the application

VII. Mechanical Design

VIII. Conclusion and Future Work