MeasureMesh: An Open Source Hardware/Software Platform for Flexible Data Logging

Matt Ruffner

I. Introduction

The Lora Alliance is a not profit association promoting the adoption of a Low Power Wide Area Network (LPWAN) IoT standard knows as Long Range WAN (LoRaWAN). It is a viable platform for low bandwidth, low power remote sensing applications. Over 100 complying companies adopt hardware standards including adaptive bitrate and encryption schemes specific to the LoRaWAN protocol.

LoRaWAN is commonly utilized for remote sensing applications including environmental and livestock monitoring [1], [2]. The LoRa standard is designed to be a low power solution, with longer range than traditional wireless means of communication (WiFi, Bluetooth, Zigbee, Z-Wave, etc.). It follows that compared to other wireless protocols, LoRa has significantly lower bandwidth and throughput [1]. Performance measurements of certain LoRa chipsets have concluded that longer ranges are acheivable in a rural setting with the lowest data rates, and that range in urban settings is less, requiring lower data rates at shorter distances than in rural settings [3]. This suggests that dynamic tuning of the radio parameters is necessary based on deployment location.

MeasureMesh builds on this LPWAN technology to providing a simple, adaptable hardware and software platform that facilitates easy adoption for custom remote sensing networks. By using off the shelf radios and control units for node and gateway hardware, quick time-to-implementation is acheived, allowing for more focus and customization on sensing needs. The gateway utilizes internet protocol for communicating with a server back-end. For the purposes of this paper, a simple sensor data storage back-end and plotting front-end will be implemented.

II. Project Overview

MeasureMesh consists of a varying number of nodes which communicate to a gateway via LoRa packet radios. The gateway then logs collected data to a cloud databse via internet. Figure 1 shows the MeasureMesh Implementation of a typical LoRa network topology[1], along with chosen implementation hardware.

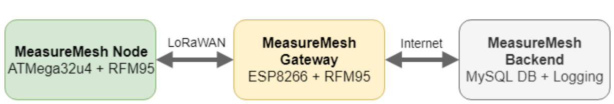

Fig. 1. Communication model showing system components and their hardware implementation.

[1]https://lora-alliance.org/about-lorawan

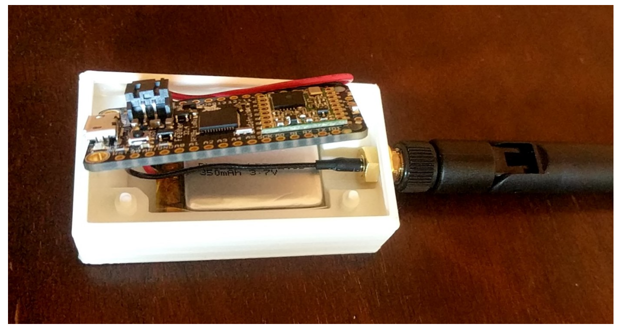

Fig. 2. MeasureMesh Node prototype, with enclosure, battery and antenna.

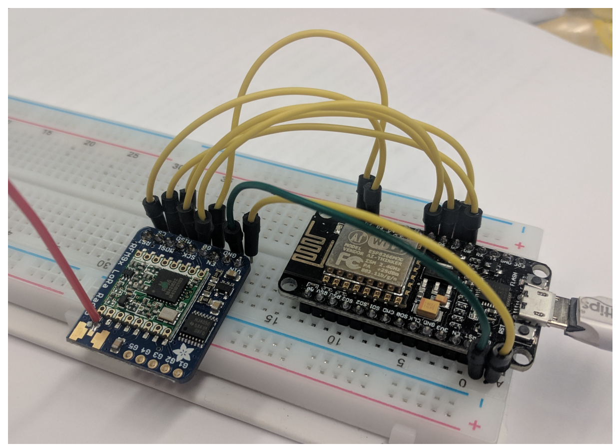

Fig. 3. MeasureMeash Gateway hardware prototype, without enclosure.

There are five major components of MeasureMesh, they can be seen listed in Table I, along with their completion status. MeasureMesh nodes consist of an Adafruit Radiofruit ATMega32u4 (Arduino programmable), with an on-board RFM95 LoRa compliant radio module [2]. This Radiofruit also includes a LiPo battery charger and USB connection for programming. In order to house these components, a 3D-printable enclosure was designed in CAD. The 3D printed version of this enclosure, with the Node hardware inside, can be seen in Fig. 2.

The same LoRa radio module is also available a a breakout board without the 32u4 processor[3]. This board is used in combination with a comodity ESP8266 development board to create the LoRa $\Longleftrightarrow$ WiFi gateway node. This assembled hardware is functional and has passed basic operations testing. It is shown in Fig. 3.

[2]https://www.adafruit.com/product/3078
[3]https://www.adafruit.com/product/3072

A MeasureMesh node reports its chosen sensor data along with certain meta information such as battery life, Node ID, and the context of the reading it is transmitting (temperature, humidity, light level, etc). MeaureMesh nodes are highly configurable in software and support varying logging intervals, sleeping in between transmitting for maximum power savings as well as multiple sensors per node.

A MeasureMesh Gateway device acts as a bridge between the LoRa radio and the MeasureMesh backend. The Gateway is always listening for LoRa packets. When one is received, it interprets the information and forms an HTTP (REST) request to be sent to the MeasureMesh database over the internet. This database address is configurable and may be a local computer or a live website.

The data is stored according to Node and Gateway ID. The user can then browse a list of available 'streams' of information that have been logged from the Nodes through the Gateway. A user can select one or more of these streams and plot them over an arbitrary timescale. The units of the plots will be according to the metadata associated with each node.

III. Projected Task Schedule

Divide your project into deliverable tasks and briefly describe the goals and outcomes of each task. For each task, describe an estimated time-line and workload division (in case of group projects). Clearly distinguish completed tasks from to-do tasks.

There are several deliverables with this project, the first of which are the hardware components of the nodes and gateway. As of now, there exists functional hardware for a single node as well as a single gateway. By the end of the semester I plan to have more than one node device for a more distributed system.

This leaves firmware and software milestones for the remainder of the semester. Table I shows a tentative schedule for the completion of remain project chunks, as well as the status of task items already finished. The back-end software depends on the completion and operation of the gateway and node firmwares, so it is scheduled to be completed last.

Component	Status and ECD
Node hardware	Completed
Gateway hardware	Completed
Node firmware	In progress, ECD 4/5/2019
Gateway firmware	In progress, ECD 4/10/2019
Backend software	In progress, ECD 4/20/2019

TABLE I
Project components and their Expected Completion Date (ECD), if not already completed.

Node firmware and Gateway firmware will be worked on in concert. Since both the ATMega32u4 and the ESP8266 are able to be programmed in the Arduino IDE, there is minimal setup required to be able to start firmware implementation. One limiting factor of the ESP8266 as a gateway as opposed to interfacing the RFM95 LoRa radio directly to say a RaspberryPi via Serial Peripheral Interface (SPI) is that the ESP8266 cannot connect to WPA2-Enterprise networks (eduroam). This makes a simple demonstration on campus more difficult (can always use phone hotspot, not a big deal.) However, in a home use scenario, where the gateway is mounted on a wall or under cover outside and constantly powered by a DC supply, the ESP8266 will draw significantly less power than a more sophisticed platform such as the RaspberryPi would. In this sense, the Raspberry Pi would be overkill and is thus not chosen as a Gateway platform.

The server back-end has the most flexibility. For initial testing purposes, a simple DB schema will be designed, and a data handling service will be written in PHP. In the long run, using a PHP framework such as Symphony or CakePHP would be much more secure, maintainable and flexible, however for the purposes of our class this semester, a simple raw PHP implementation will suffice. That being said, the planned functionality is still for the web server to be able to log information sent to it by the gateway (and only the gateway) and then store it to a database. When the site is accessed on a computer, the user will be able to recall these data and plot them.

IV. Learning Outcome

This project is providing an opportunity to explore state of the art radio communication standards that are currently being deployed on a commercial IoT scale. The improved range and lower power consumption that is offered by LoRa radios allows for an even sparser grid of sensors that can last even longer on a given battery, or more efficiently utilize solar power.

It is also interesting to experiment with actual resulting range differences between urban and rural environments, and line-of-sight versus obstructed communication paths. This will definitely come in to play when recommending Node placements to the end users of MeasureMesh.

References

[1] F. Wu, C. Rudiger, J.-M. Redoute, and M. R. Yuce, "We-safe: A wearable iot sensor node for safety applications via lora," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, vol. 2018-, pp. 144–148, IEEE, 2018.

[2] M. G. Ikhsan, M. Y. A. Saputro, D. A. Arji, R. Harwahyu, and R. F. Sari, "Mobile lora gateway for smart livestock monitoring system," in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, pp. 46–51, IEEE, 2018.

[3] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. F. Skarmeta, "Performance evaluation of lora considering scenario conditions," *Sensors*, vol. 18, no. 3, 2018.