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 achievable 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 achieved, 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. All work is contained in the GitHub repository at http://github.com/ruffner/measure-mesh.

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

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

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

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 commodity 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.

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

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

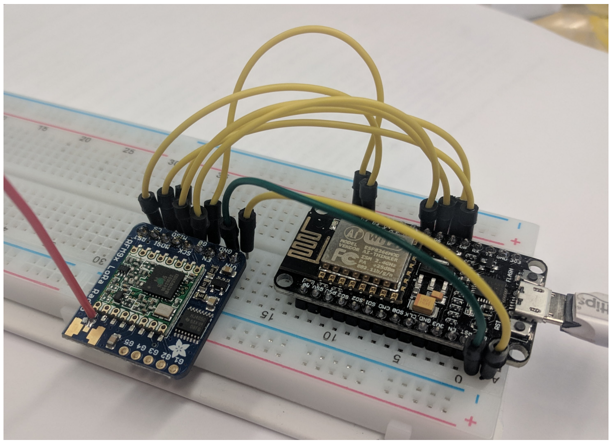

Fig. 3. MeasureMesh Gateway hardware.

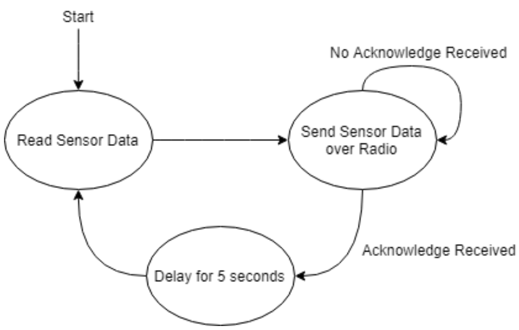

Fig. 4. Operational state diagram of the MeasureMesh Node firmware.

configurable in software and support varying logging intervals, sleeping in between transmitting for maximum power savings as well as multiple sensors per node.

The firmware for the Node device can be found in the `NodeFirmware/` subdirectory of the project repository. This Arduino script requires a version of the Arduino IDE that supports the Board Manager, and the Adafruit SAMD21 board package must be downloaded. The RadioHead library by Paul Stoffregen is used to simplify radio communications. Other than that the Node is pretty straight forward. The firmware currently sends its battery voltage every 5 seconds to the gateway as a form of test telemetry. A state diagram showing nominal operation of the Node device can be seen in Fig. 4.

A MeasureMesh Gateway device acts as a bridge between the LoRa radio and the MeasureMesh back-end. The Gateway

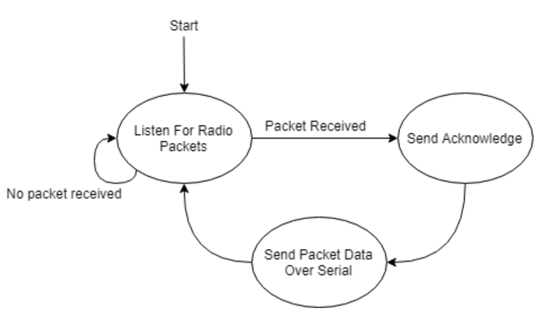

Fig. 5. Operational state diagram of the MeasureMesh Gateway firmware

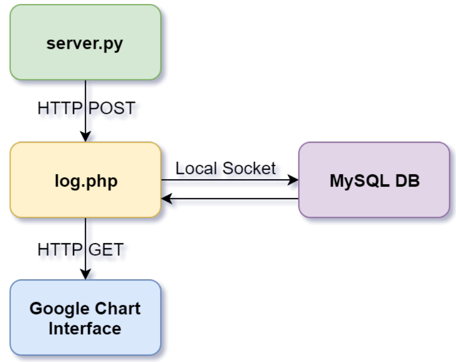

Fig. 6. MeasureMesh Server diagram.

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. Currently, the Gateway device is connected to the Server via USB. A CP2104 USB to serial converter on the ESP8266 breakout seen in Fig. 3 populates a virtual com port on the Linux server and is used instead of WiFi for testing purposes. The state diagram in Fig. 5 shows the nominal operation of the Gateway device. Gateway firmware can be found in the `GatewayFirmware/` subdirectory of the project repository.

Since the logging back-end is meant to be initiated with a HTTP GET request, a tool was needed to convert the serial messages incoming from the Gateway to GET requests for the logging back-end. To do this, the script `Server/server.py` was written to serve this purpose. Most of the work that is done by this script is interpreting the 6 lines of information that is dumped by the gateway to just extract the data value. This could obviously be streamlined as the gateway prints verbosely and not all 6 lines of information are pertinent to the logged data. Truly, this script isn't necessary if the Gateway firmware is modified to preform the GET request itself. This just shows the flexibility of MeasureMesh. A diagram of the server operations and component interaction is shown in Fig. 6.

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 of these streams and plot them.

The MeasureMesh back-end is implemented with a MySQL database that stores all information logged from nodes. Tables support metadata describing the context of the information logged. The two table schema can be found in the `Server/` subdirectory of the repository. I wrote the graphing and most of the logging front end a few years ago for a different project and was able to reuse portions of code for the purposes of MeasureMesh. A live demo of the logging and data visualization front end can be seen hosted on my home Linux box at http://ruffner.ddns.net/measure-mesh/Server

III. 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.

The most enjoyable part of this project is the number of different techniques and tactics that I employed to successfully build an end-to-end system. I had to draw on a wide array of my skill set including 3D modeling, circuit design, web programming and embedded firmware design. I enjoyed this class and being able to create MeasureMesh as a final project.

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.