Sensors in Snowy Alpine Environments: Progress Report

Martha Apple*, James Gallagher†, Kevin Negus*, Samuel Croft*, Tyler Vendetti*, and Carson Fiechtner*

*Montana Tech, †OPeNDAP

<mapple@mtech.edu>, <jgallagher@opendap.org>

Alpine plants act as sentinels of climate change.

Overview

- About Open Source Hardware
- Our designs for an in-situ sensor network
- LoRa Background
- Test implementation

Open Source Hardware

- Closely aligned to the Open Source Software movement
  - "Open hardware," or "open source hardware," refers to the design specifications of a physical object which are licensed in such a way that said object can be studied, modified, created, and distributed by anyone.[1]
- A few companies that support OSH:
  - Adafruit, SparkFun, Seeed, Mouser, OSHPark, …
- Tools:
  - Arduino (IDE, MCUs), Fritzing (CAD), GitHub, BitBucket (code repositories)
- Umbrella organizations:
  - OSHWA, opensource.com

1. https://opensource.com/resources/what-open-hardware

Most Importantly…

- Open Source Hardware provides an easy way to share ideas
- And learn about techniques
- So we can build devices we need that (may) lack mass-market appeal

Overview

- About Open Source Hardware
- Our designs for an in-situ sensor network
- LoRa Background
- Test implementation

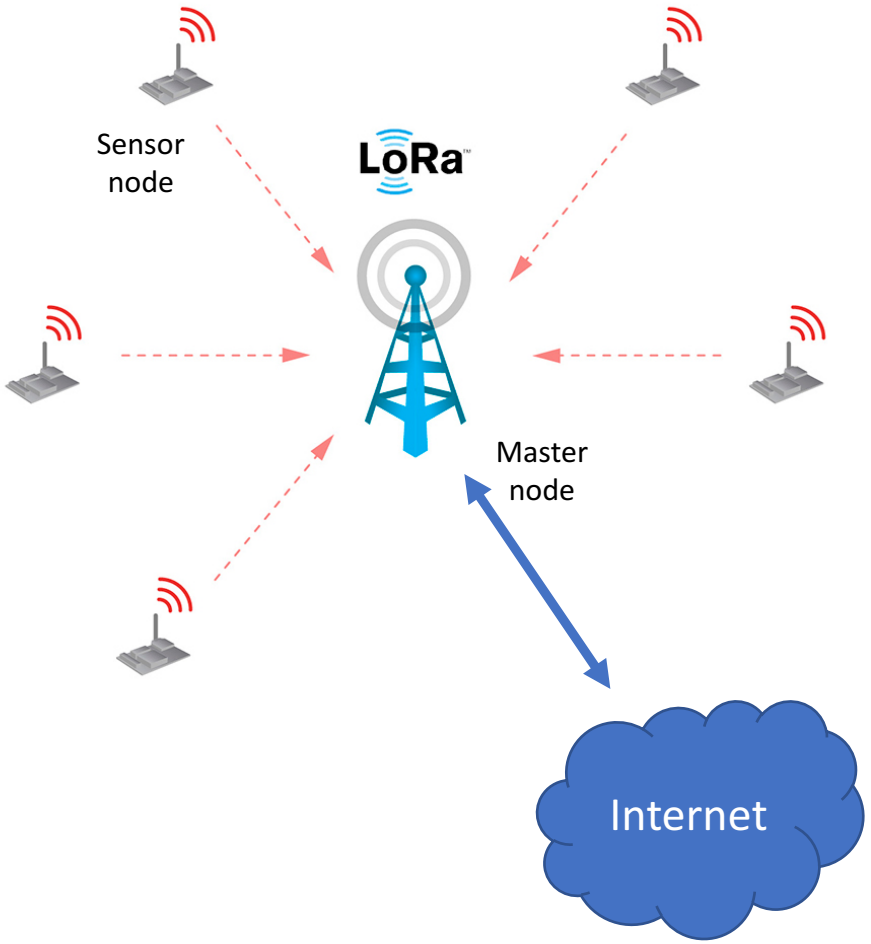
Requirements

- Soil temperature and moisture monitoring
- Operation unattended in remote, harsh environments
- Remote connectivity
- Low cost

Sensor Network: Basic Design

- Deploy sensors at research sites (initial site: Goat Flat, ~9,000 ft).
- Sensor nodes should run on battery power for 2+ years.
- Use LoRa radio to connect sensor nodes to a master node.
- Use the master node to forward data to the Internet.
- The master node will use ~~cell phone~~ Satellite technology for an Internet uplink.
  - Subsequent implementations could use other techniques.
- Data from the master node will be sent to an Application Server that can be accessed from browsers and analysis tools.
- The Master node can sense a wide(r) variety of conditions such as snow pack presence.
- Calibration: use reference instruments including a portable soil temperature and moisture measurement device.

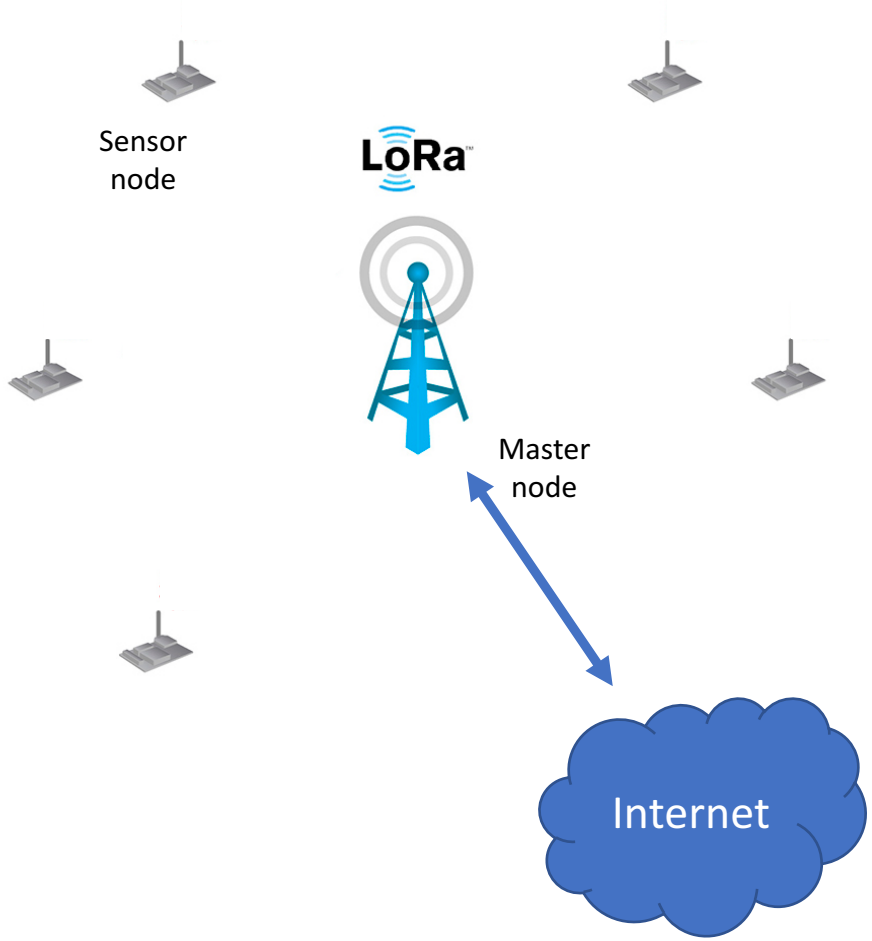
Sensor Network: Topology



- Master/Sensor star topology
- Sensor nodes
  - Collect data (temperature, soil moisture, etc.)
  - Use LoRa radio to send data to Master
- Master
  - Receives data from sensors using LoRa
  - Forward data to ‘Application Server’ via Internet

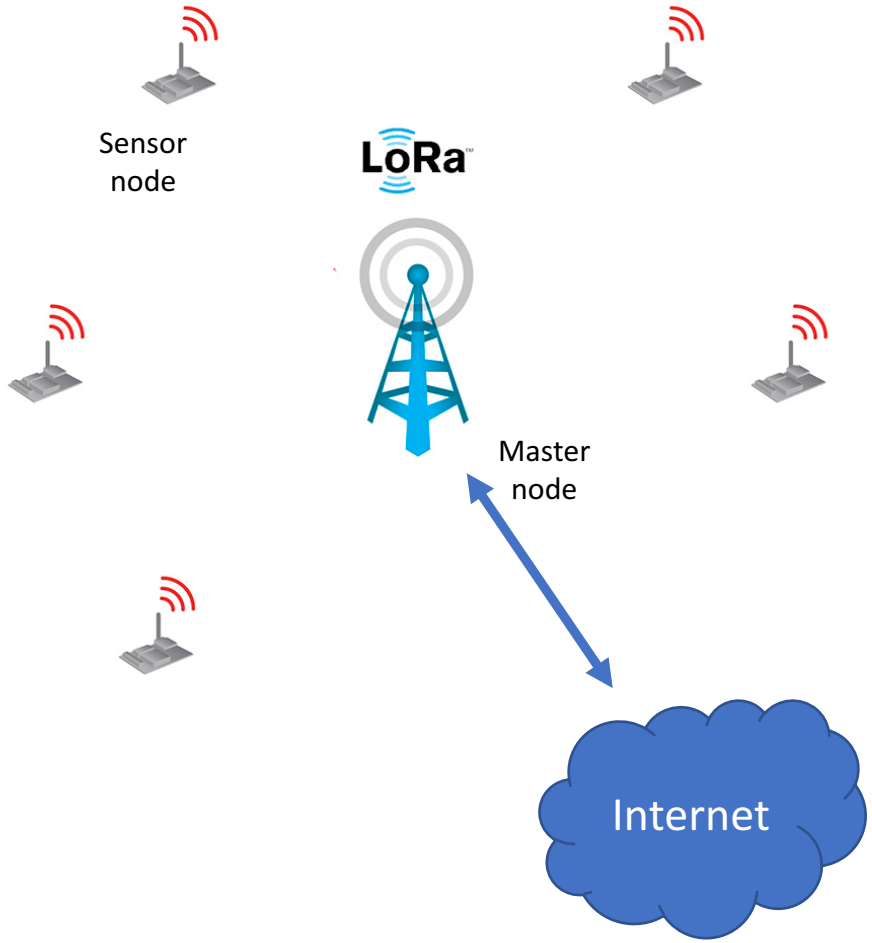
Sensor Network: Operation

Sensors sleep most of the time to conserve power



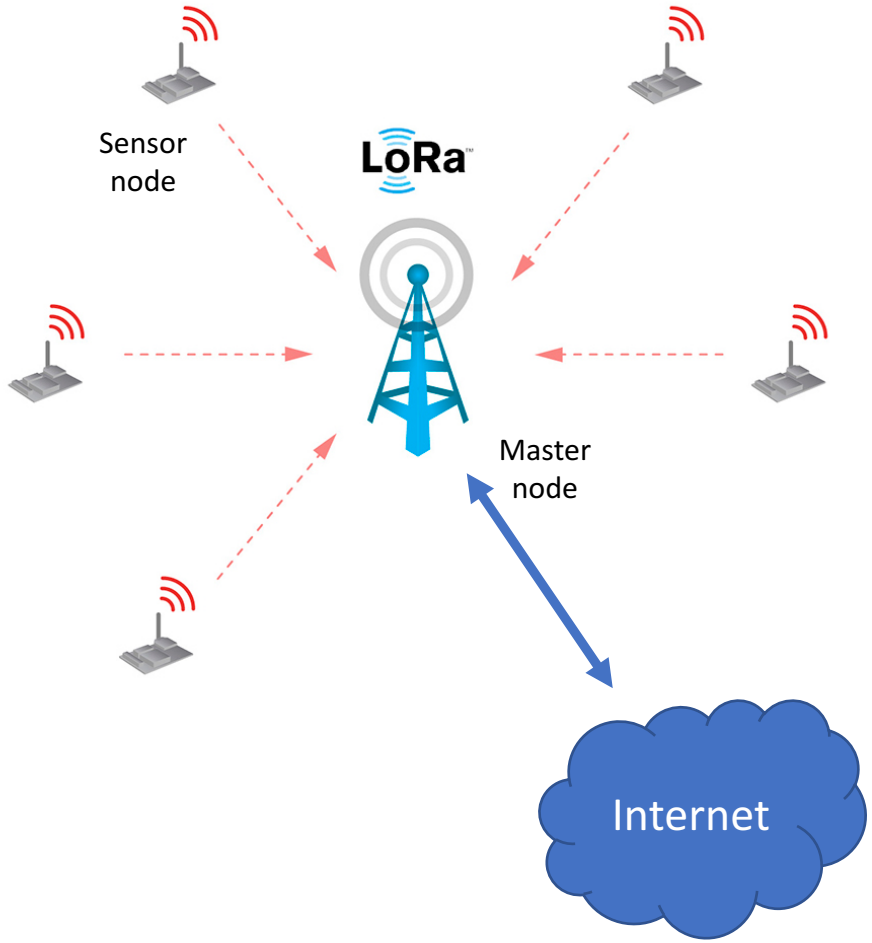
Sensor Network: Operation

Sensors wake up and measure environmental parameters



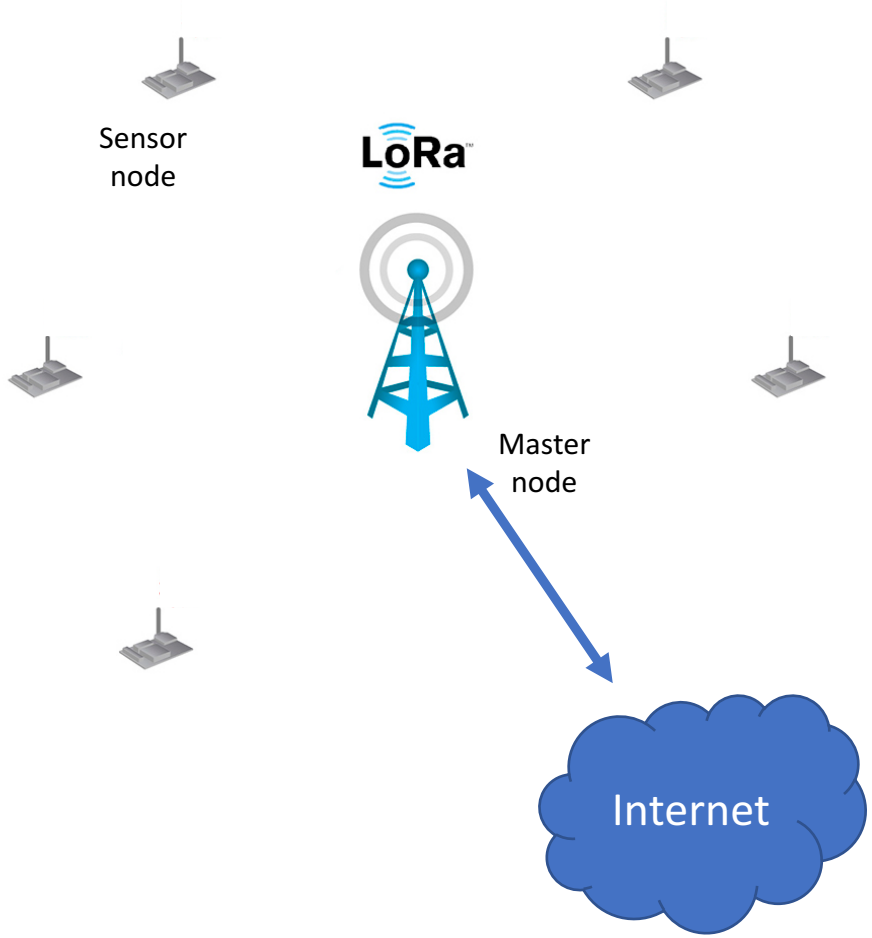
Sensor Network: Operation

Sensors transmit data to the Master



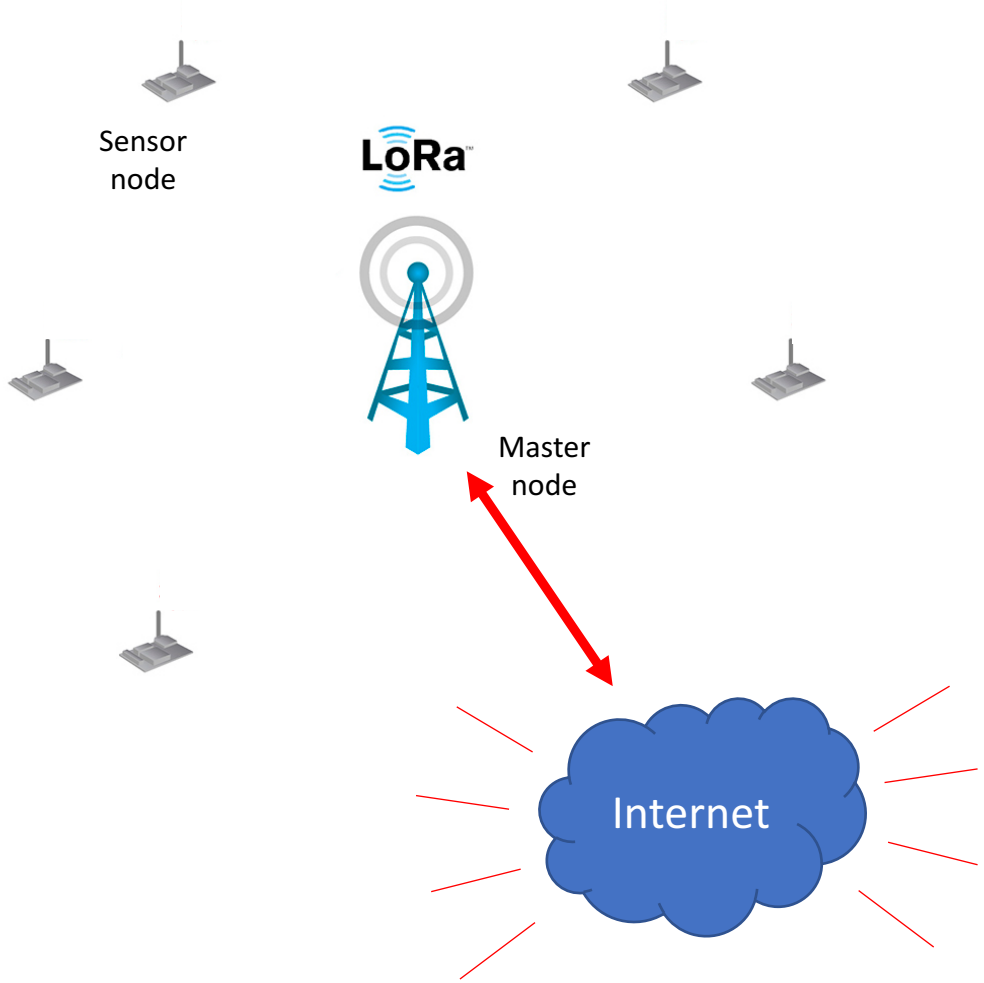
Sensor Network: Operation

Sensors go back to sleep



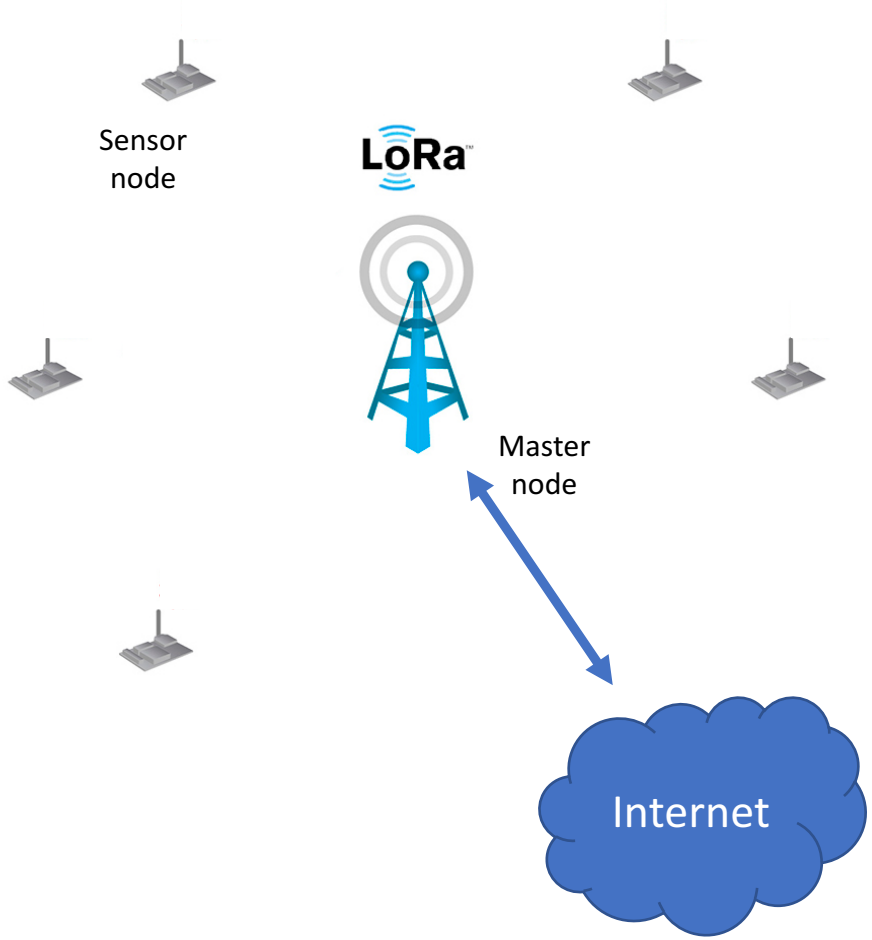
Sensor Network: Operation

The Master node forwards the collected data to an Application Server in the Internet



Sensor Network: Operation

The Master returns to its nominal mode

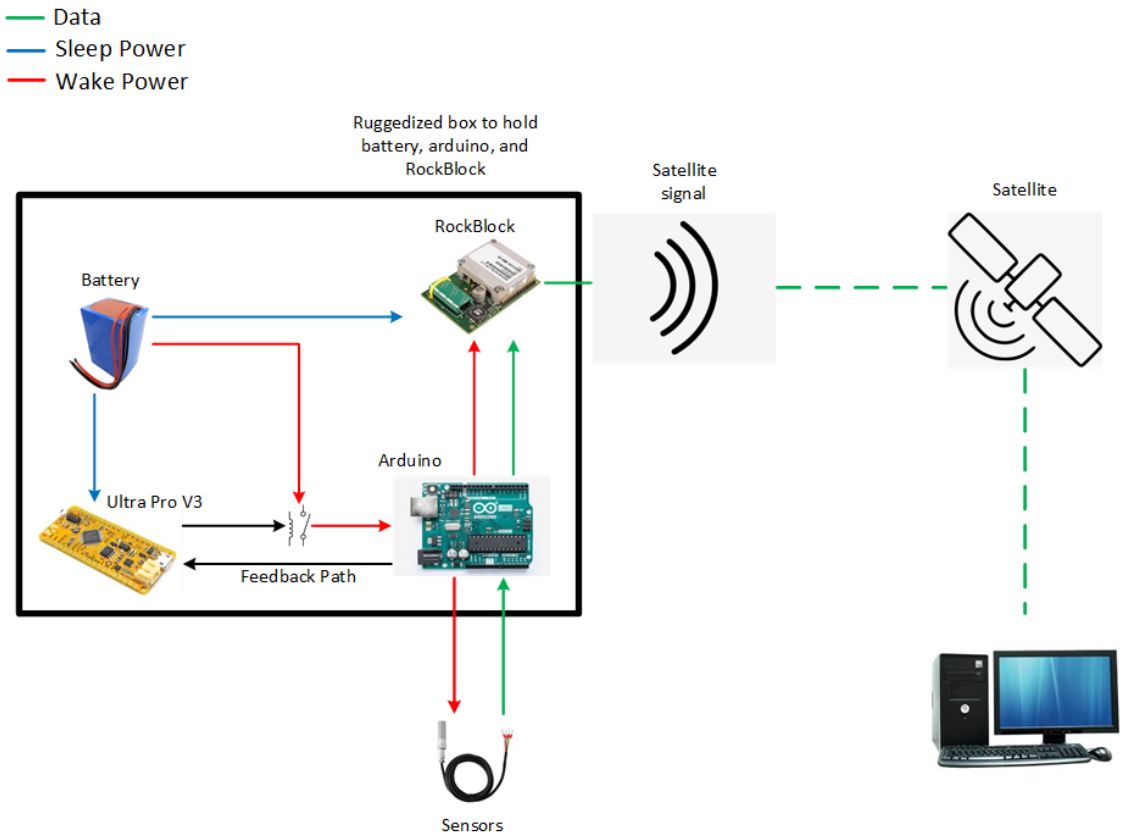


Overview

- About Open Source Hardware
- Our designs for an in-situ sensor network
- LoRa Background
- Test implementation

Test Implementation – Master Node

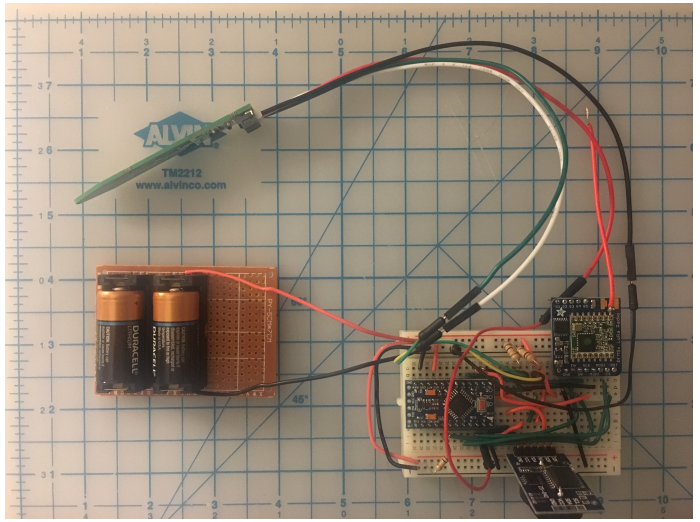
- Design work started in Fall 2019 by Mtech Students with faculty mentor
- Designed to function standalone
- Satellite module for connection to the Internet
- MCU/RTC to control power
- 3.3Ah/year power requirement*
- Add LoRa for Sensor Nodes



*Apple 2019 laptop has a 4.5Ah battery

Test Implementation – Sensor Node

- Sensor node
  - LoRa + Arduino + Temperature and soil moisture sensor
  - $LiMnO_2$ batteries (-50° C)
  - Prototypes operated at -25°C and were run buried in both soil and snow as well as in open air
  - Total parts cost < $50; options exist to reduce that cost further
  - Dropped MCU sleep in favor of a power latch
  - Estimated 1 Ah/year power requirement @ 3.0v
- Design info:
  - github.com/jgallagher59701/Soil_moisture



Implementation Notes

- Satellite uplink chosen by Mtech Master Node developers to broaden deployment locations (over cell technology)
- Both the Mtech and OPeNDAP developers independently decided on ‘power latch’ technique to control power consumption, esp. the power used by the communication technology
  - A better option than the peripheral’s sleep modes
  - The Master Node design uses a second MCU to toggle power (using a built in clock).
  - The Sensor Nodes uses discreet components in combination with a real-time clock
- Master/Sensor LoRa protocol
  - Reliable Datagram – was previously an ad hoc protocol – concerns regarding collisions
- Application Server
  - CHORDS or ERDDAP server running on Amazon Web Services EC2 instance

Questions?

Bonus material on LoRa

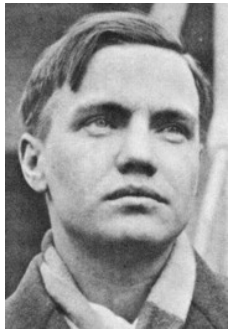
Overview

- About Open Source Hardware
- Our designs for an in-situ sensor network
- LoRa Background
- Test implementation

LoRa Background: What is LoRa?

- “LoRa (short for Long Range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology.”*
- Spread Spectrum: To spread an information signal over a (wide) band of radio frequencies.
- Robust transmission technique; also used in WiFi, Cell phones, Bluetooth, etc.
- Hedy Lamarr and George Antheil are credited with its invention

Lamarr[1]



Antheil[2]

*https://www.semtech.com/lora/what-is-lora

LoRa specifics

- Widespread use by Internet of Things applications.
- Inexpensive (~$5 eBay; $20 Adafruit/SparkFun)
- Small (16mm x 16mm)
- Range: 1–10km
- Adjustable bit rate: 293 bps to 37.5 kbps
- Adjustable error correction, etc., parameters
- Works well with Arduino and similar MCUs
- Good library support (e.g., RadioHead; TinyLora)
- US/EU/AS frequency bands (915/868/430 MHz)
- Low power – 20mA receive/100mA transmit

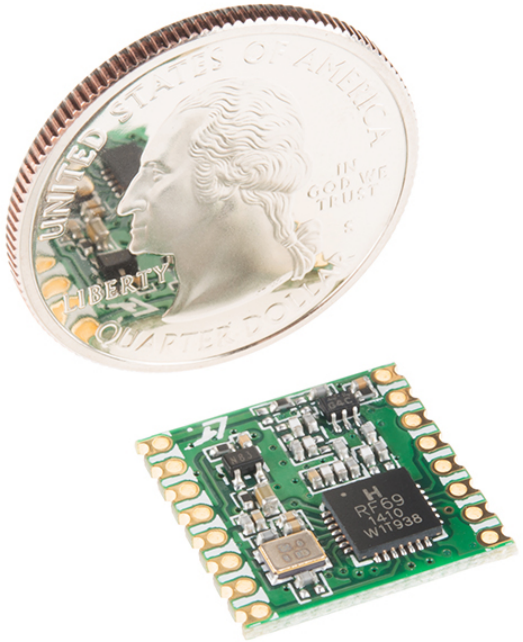
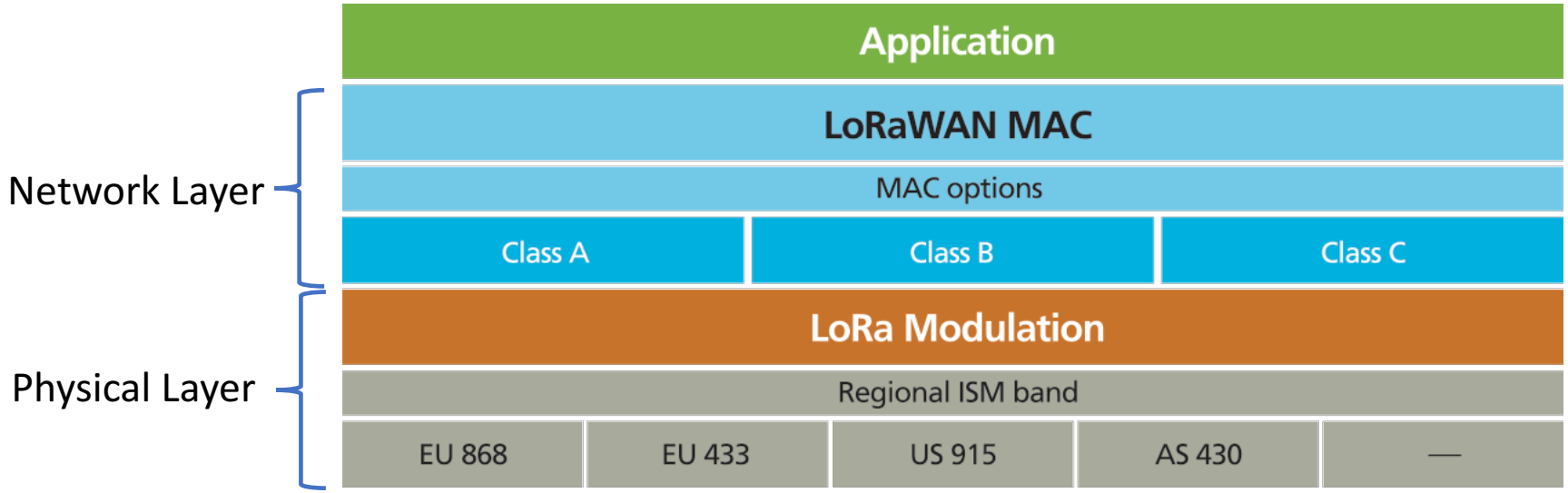


Image from the SparkFun website

How LoRaWAN Relates to LoRa

- Network layer defined by the LoRa Alliance
- Physical layer defined by the radio hardware



Alternatives to LoRaWAN: LoRa Physical link

- MCU software library: RadioHead.[2]
  - We're using RadioHead, but there are other libraries for LoRa.
- How the physical link can be configured
  - Power level (5 to 23 dBm), Bandwidth, Coding rate, Chips/symbol, CRC[3]
  - The last four determine data rate and noise rejection
  - Supports hardware Channel Activity Detection (CAD)
- Using the SPI bus
  - Many LoRa boards use the SPI bus and signal reads and writes using interrupts
  - If there are other SPI-based devices, disable interrupts while using another device
  - Other SPI devices include SD memory cards

2. http://www.airspayce.com/mikem/arduino/RadioHead/; or https://github.com/adafruit/RadioHead
3. https://en.wikipedia.org/wiki/Cyclic_redundancy_check

RadioHead Architecture

- Driver + Manager: Implements the physical and logical network layers
- Driver
  - C++ object that supports a particular transceiver
  - Can be used alone, without a Manager
  - Suitable for early prototypes and/or very simple uses
  - Various Drivers support different max packet sizes (28 to 255 octets)
  - RFM95W → 251 octets (i.e., HopeRF and Semtech LoRa boards)
- Manager
  - C++ object that uses a Driver to implement a logical network
  - Objects exist for Datagram, Reliable Datagram, Router and Mesh protocols

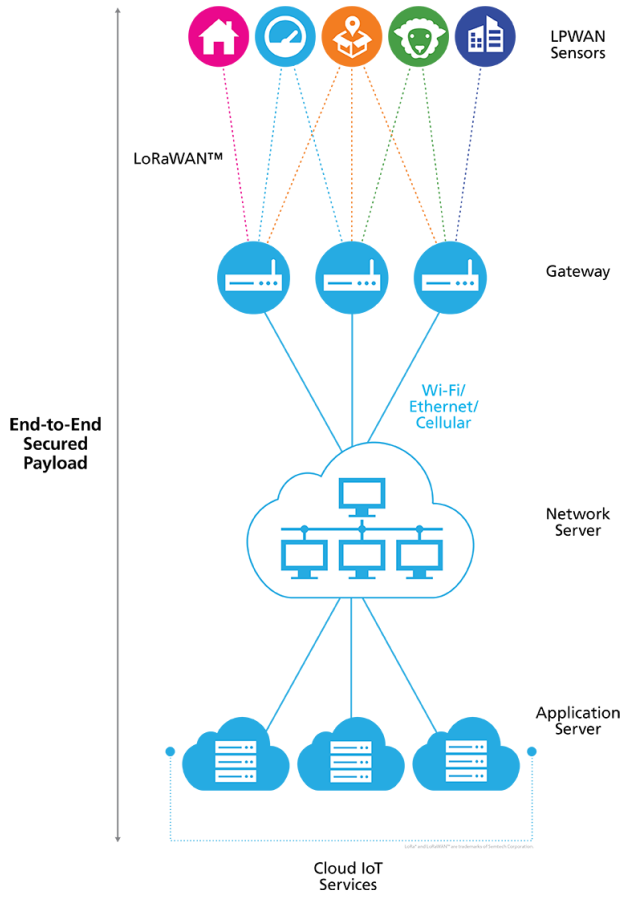
RadioHead Manager Classes

- These are based on well-known networking principles
- Addressing: Assigning an address to each device helps maintain reliability when scaling up a network to more nodes
- Datagram:
  - A simple addressed protocol. Every packet has an 8-bit SRC and DEST address
  - Also supports a Broadcast ‘address’ (0xFF)
  - No error detection
- Reliable Datagram:
  - Addressing + simple error recovery: messages are acknowledged by the recipient, and unacknowledged messages are retransmitted until acknowledged or the retries are exhausted
  - Broadcast does not support error detection

Routing and Meshes

- **RHRouter** enables multi-hop transmissions
  - LoRa networks can be designed to move information between star nodes
  - Must be deployed with care since intermediate nodes will need more power
  - Can support larger topological deployments with fewer Internet uplink sites
  - Routing tables must be manually managed in software (using 8-bit addresses)
- **RHMesh** enables automatic route discovery
  - This supports dynamically adding and removing both leaf and internal nodes
- Router and Mesh are useful for the ’star’ node in our design, but not for the leaf nodes, which will be powered down for all but the few seconds they sample and transmit data.

Generic IoT/LoRa Topology



Picture: Semtech[1]

- Autonomous sensors communicate with LoRa Gateways
- LoRa Gateways transfer information to the Internet
- Once on the Internet the information can be routed to data servers, etc.
- This can use LoRaWAN (a kind of Low Power Wide Area Network – LPWAN) or Datagrams

More information

- What is LoRa, Semtech website: https://www.semtech.com/lora/what-is-lora
- Comparison of LoRa with FSK (frequency Scaled Keying): Jesus Sanchez-Gomez, Ramon Sanchez-Iborra, and Antonio Skarmeta, Transmission Technologies Comparison for IoT Communications in Smart-Cities**.** 2017. DOI: 10.1109/GLOCOM.2017.8254530.
- The Things Network (non-profit) : https://www.thethingsnetwork.org/
- MathWorks ThingSpeak: https://thingspeak.com/
- LoRa Alliance (non-profit) – LoRaWAN information source: https://lora-alliance.org/
- LoRa Datagram open source implementation: https://www.airspayce.com/mikem/arduino/RadioHead/classRHReliableDatagram.html
- LoRaWAN open source implementation: https://github.com/adafruit/TinyLoRa
- RadioHead library: http://www.airspayce.com/mikem/arduino/RadioHead/; or https://github.com/adafruit/RadioHead