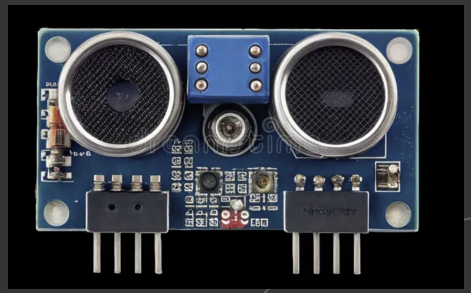
sdmay27-34
10-27-26

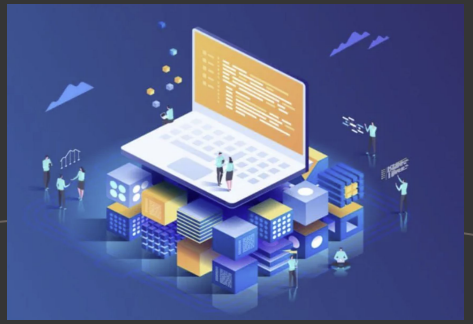
CICADA

Fire Detection System for Farm Fields

Lightning Talk

Meet the team





LoRa:
Jack Rice
Cooper Hoy
Sean Fjellman

Sensors:
Hrushi Bhatt
Caleb Kugel

Dashboard:
Jake Ebersole
Marcus Yeung

Our Project

Farmers can’t always be there to monitor their fields

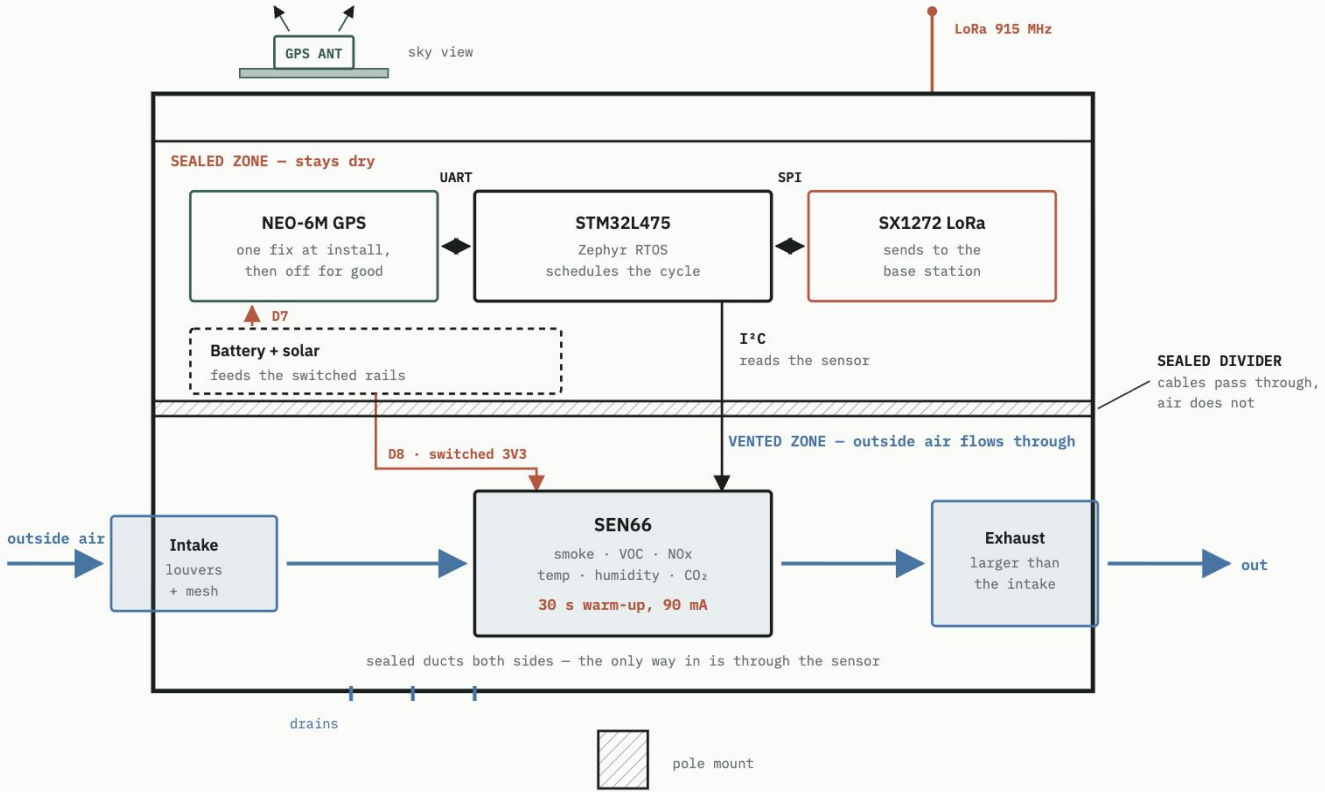
CICADA will detect a fire in the farmer’s field and notify the farmer and first responders

Works via LoRa radio eliminating the need for expensive cellular or Wifi infrastructure

Each node is independent and easily deployable in the field



Sensor Integration



Sensor Functionality:

1. Temp (°C)
2. Smoke (CO_2) and VoC
3. GPS

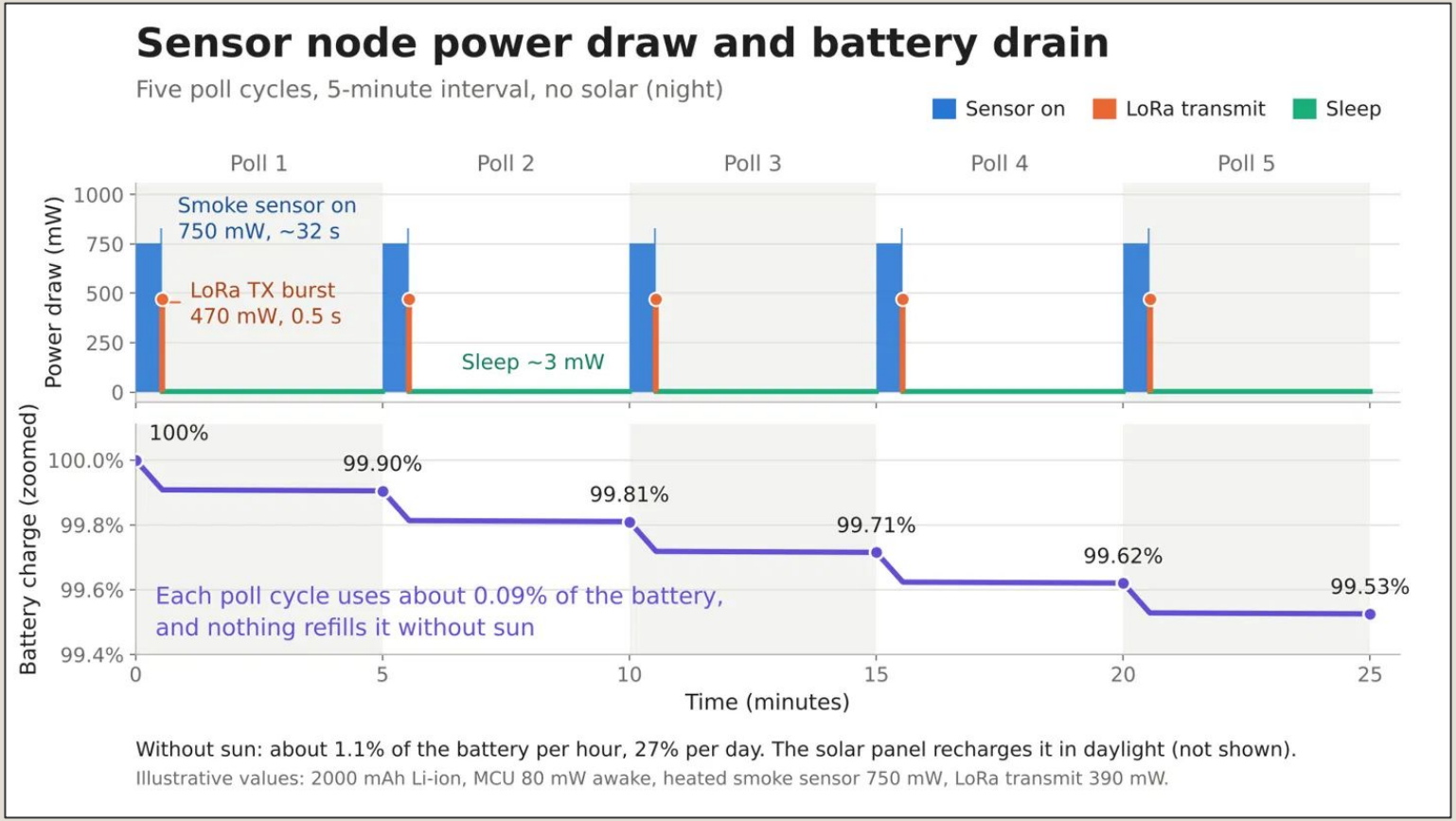
Compartment Design:

1. Weather conditions
2. Insect interference
3. Terrain

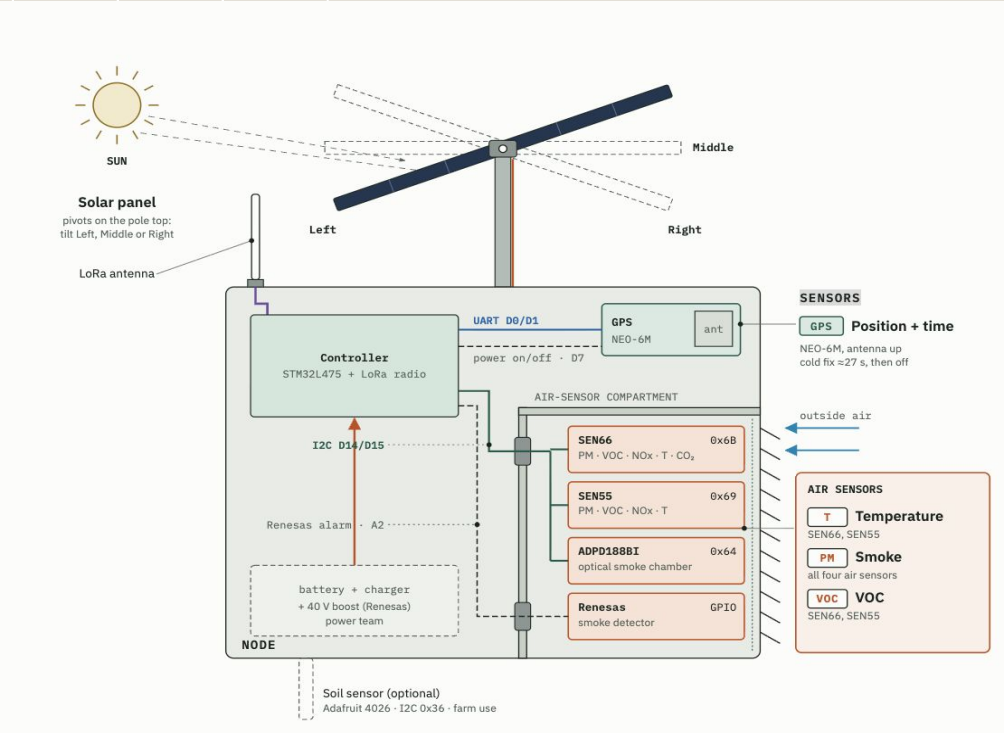
SEN66 Decision:

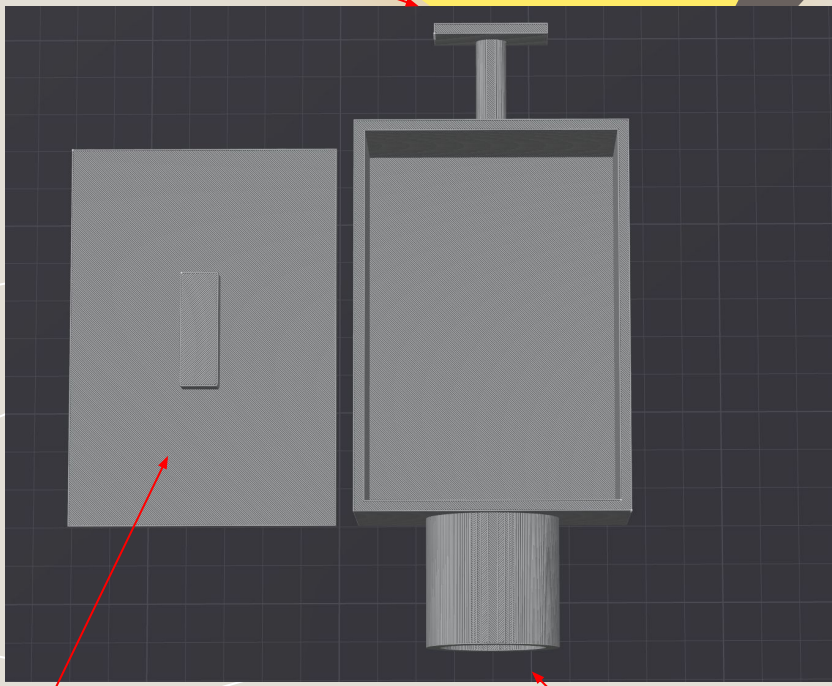
1. Singular Sensor
2. Multiple functions

Sensor Power Draw



Node Design


SUN
Middle
Solar panel
pivots on the pole top:
tilt Left, Middle or Right
Left
Right
LoRa antenna
Controller
STM32L475 + LoRa radio
UART D0/D1
power on/off · D7
GPS
NEO-6M
ant
SENSORS
GPS
Position + time
NEO-6M, antenna up
cold fix ≈27 s, then off
AIR-SENSOR COMPARTMENT
outside air
I2C D14/D15
SEN66
PM · VOC · NOx · T · CO₂
0x6B
SEN55
PM · VOC · NOx · T
0x69
Renesas alarm · A2
ADPD188BI
optical smoke chamber
0x64
AIR SENSORS
T
Temperature
SEN66, SEN55
PM
Smoke
all four air sensors
VOC
VOC
SEN66, SEN55
battery + charger
+ 40 V boost (Renesas)
power team
Renesas
smoke detector
GPIO
NODE
Soil sensor (optional)
Adafruit 4026 · I2C 0x36 · farm use


Photo Sensors(Not to Scale)
Door
Pole Mount

Solar Tracking

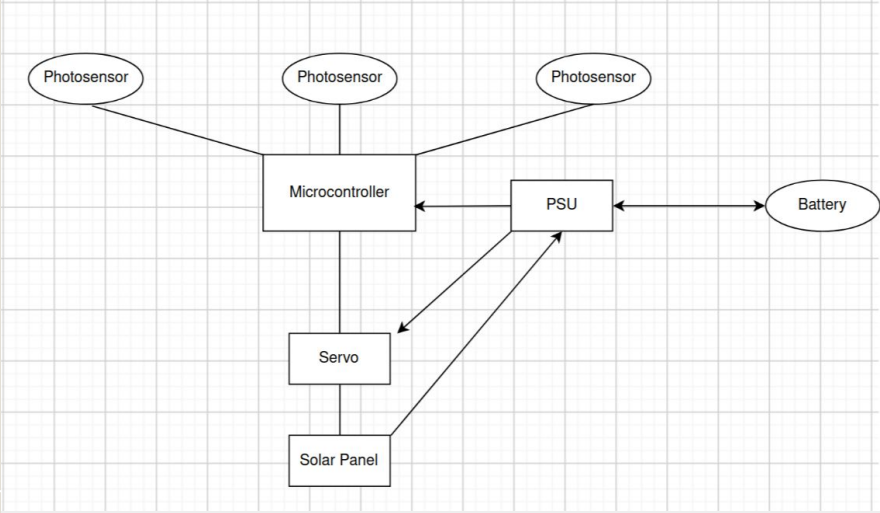
Designing: A solar tracking system that automatically adjust the panel towards the sun.

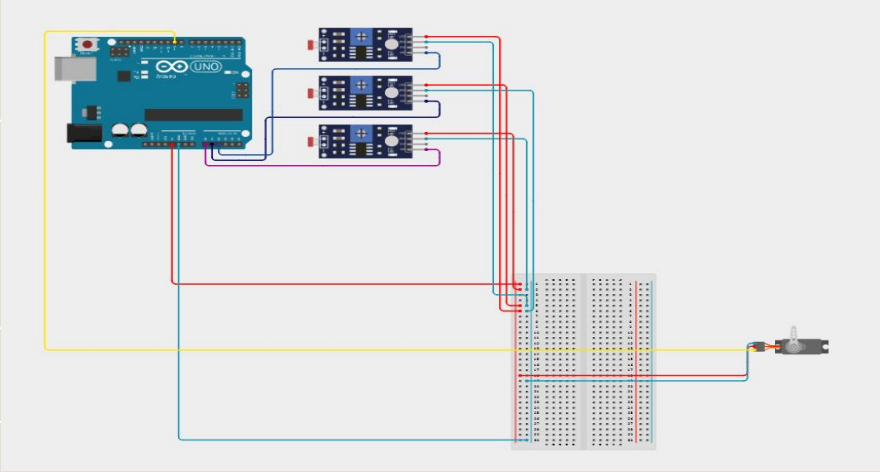
How it works: Three fixed photosensors send information to the STM32 microcontroller.

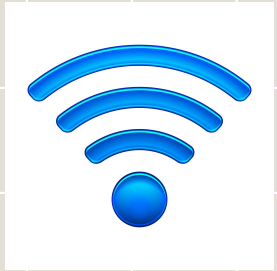
Big Idea: Is a stand alone system that needs to know when to activate / stay idle.

Challenges:

- Make sure power cost of servo does not exceed energy gained.
- Servo / locking mechanism that can resist wind / elements.
- Requires power to be sent to motor from PSU not microcontroller







LoRa Packet Structure

Packet Header

The packet header describes the “metadata” of the packet.

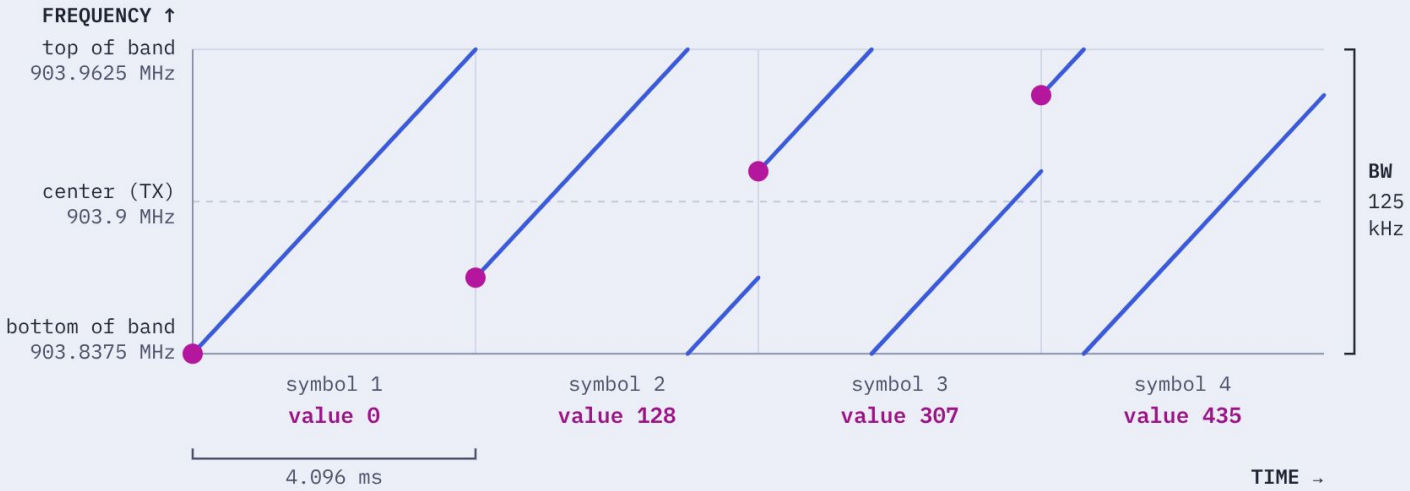
The sensor data is placed in the “payload”

Offset	Field	Size	Meaning
0	magic	1 byte	Identifies the protocol
1	version	1 byte	Protocol version.
2	type	1 byte	HELLO, DATA, ACK, etc.
3	flags	2 bytes	For example, ACK_REQUESTED
5	hops_remaining	1 byte	Maximum additional relay transmissions allowed.
6	network_id	2 bytes	To identify node subsets, if necessary.
8	origin_id	4 bytes	Node that originally created the message.
12	destination_id	4 bytes	Final recipient, usually your gateway.
16	sender_id	4 bytes	Node transmitting this particular radio hop.
20	next_hop_id	4 bytes	Node expected to process/forward this transmission.
24	session_id	2 bytes	New identifier each time the origin boots.
26	sequence_id	4 bytes	Message counter within that boot session.
30	payload_length	2 bytes	Number of bytes following the header.
32	payload	Variable	Sensor readings, acknowledgment reference, etc.

LoRa Band Specs

LoRa Channel (CH8, US915)

SF9: 512 possible start positions per symbol, each symbol lasts 4.096 ms



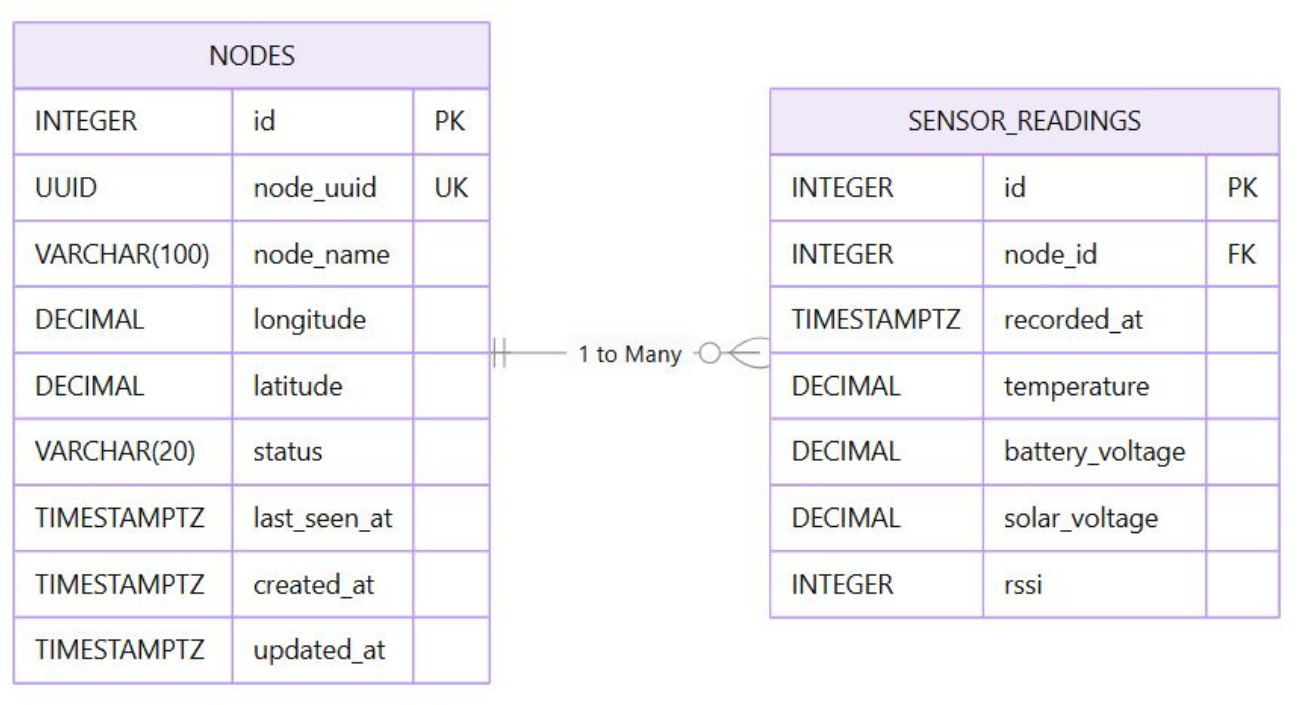
Chirp: sweeps the full 125 kHz once per symbol

Start point: the data (one of 512 values = 9 bits)

BAND	TX FREQUENCY	BANDWIDTH	MODULATION	SPREADING FACTOR
US915, SB2	903.9 MHz	125 kHz	LoRa CSS	SF9

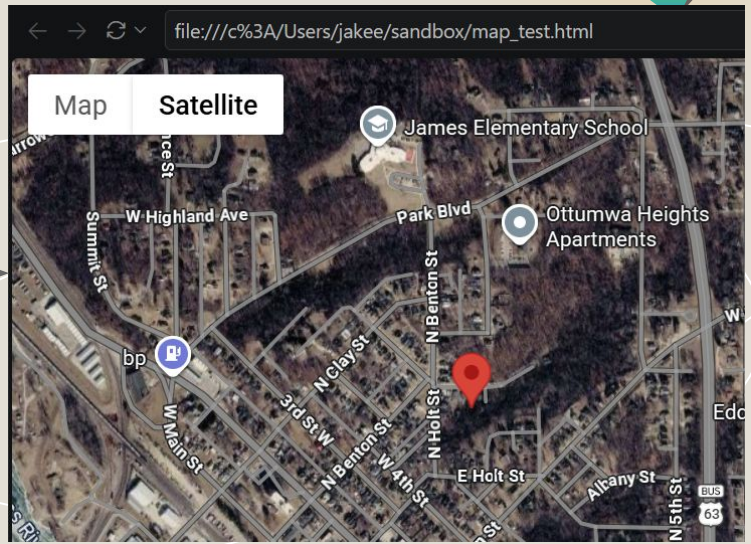
SYMBOL TIME	BITS PER SYMBOL	DATA RATE	CODING RATE	TX POWER
4.096 ms	9	1.76 kbps	4/5	+20 dBm

Backend: Database Schema

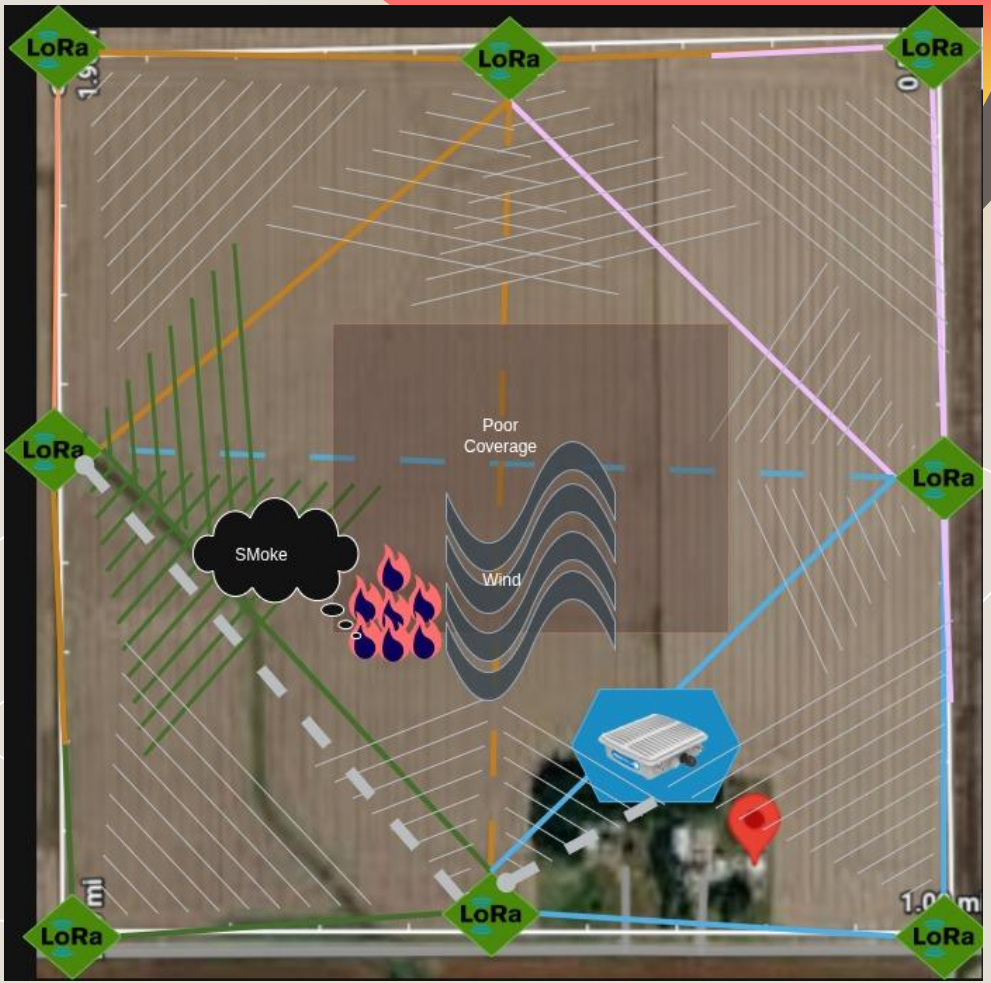


Frontend: Google Maps API

```
  <script
    async
    src="https://maps.googleapis.com/maps/api/js?loading
  ></script>
</head>
<body>
  <gmp-map
    center="41.027748173921374, -92.41852445367961"
    zoom="13"
    map-id="DEMO_MAP_ID"
  >
    <gmp-advanced-marker
      position="41.027748173921374, -92.41852445367961"
      title="Ottumwa, IA"
    ></gmp-advanced-marker>
  </gmp-map>
</body>
```



Any questions? Ask away!




Thank you