LPWAN 2

Denise Yap,
Dominic Heaton,
Pawel Kostkowski,
Samuel Wong

Overview

- Project Brief
- System Components
- LoRa
- LoRaWAN
- Development
- Testing
- Demonstration

Project Brief

- Design a Low Power Wireless Area Network (LPWAN) that:
    - Collects data from various sensors
    - Transmits the data using the LoRa network
    - Displays the data on a website

- Use Cases of the project:
    - Noise pollution monitoring in a urban environment
    - Basic meteorological data collection system

System Components

LoRa

- LoRa vs. LoRaWAN
    - LoRa = PHY Layer
    - LoRaWAN = MAC, NWK and APP build on LoRa.

Some Definitions...

- Symbol : discrete RF energy state that represent some quantity of information.

- Spreading factor: number of bits encoded per symbol
  - [7-12] bits per symbol

LoRa’s Proprietary PHY

- Modulations: Chirp Spread Spectrum (CSS)
  - A type of digital modulation.
  - Define how digital values are mapped into RF energy.
- What’s a chirp?
  - A signal whose frequency continuously increasing or decreasing.
  - Cyclically-shifted

CSS Chirps

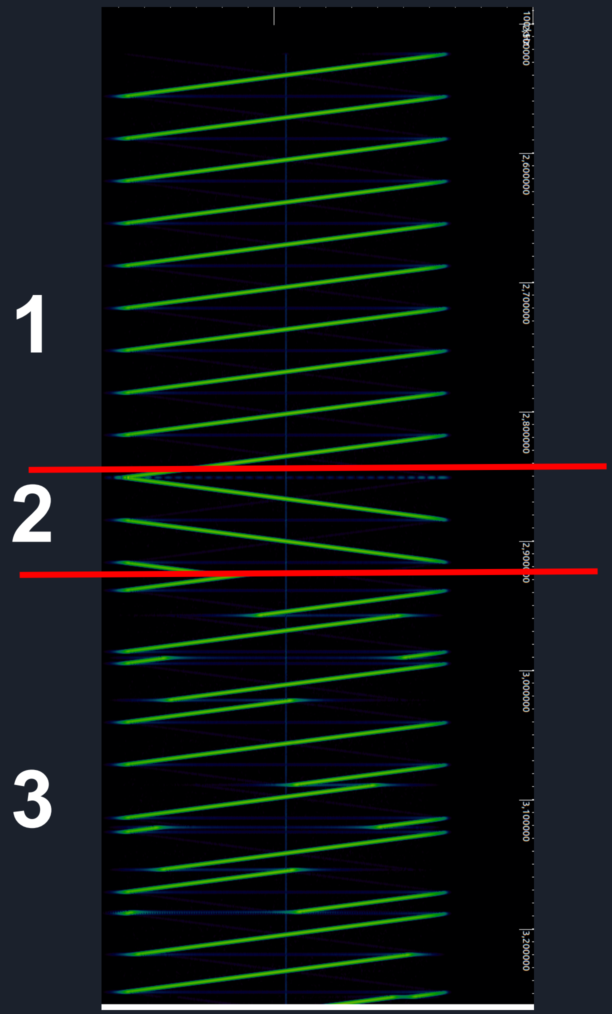
- Up chirp (top)
    - Increasing frequency
- Down chirp (bottom)
    - Decreasing frequency

LoRa PHY Frame

1. Repeated up chirps
    a. Preamble/Training Sequence

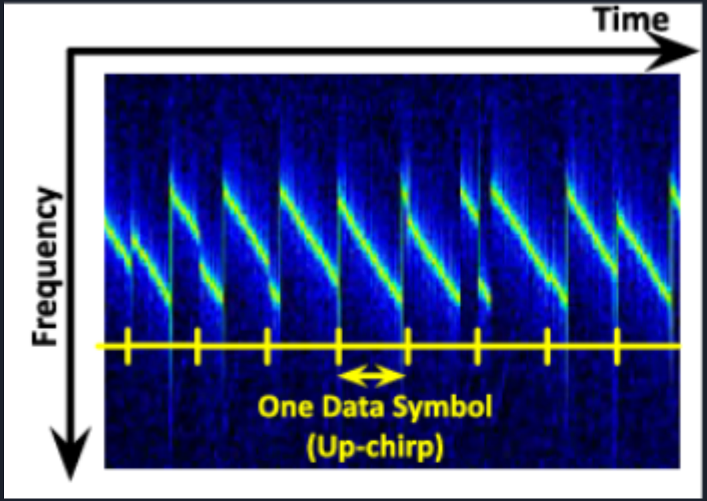
1. Two down chirps
    a. Start of frame delimiter (SFD)

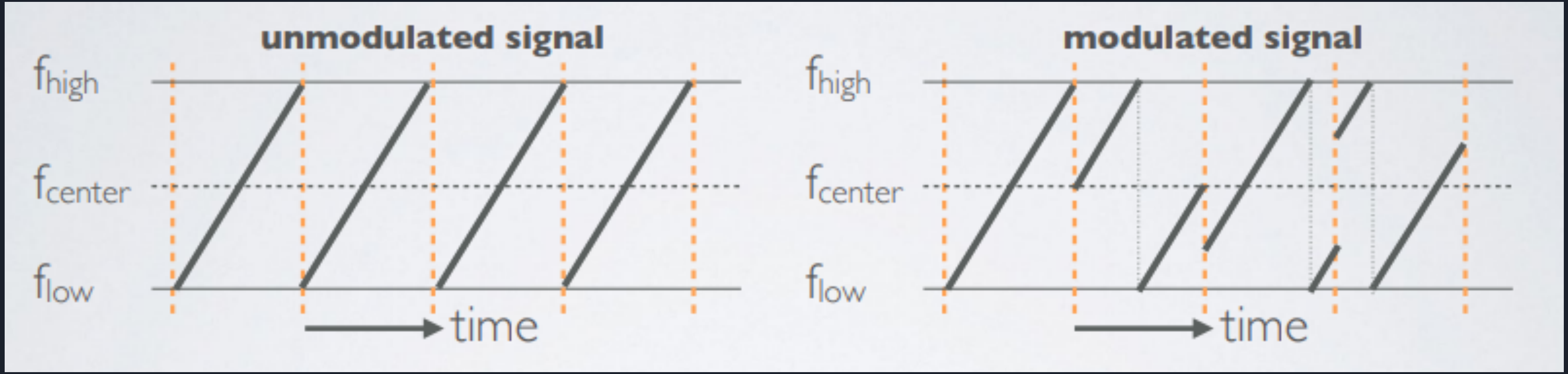
1. Choppy up chirps of varying length.
    a. Data



PHY Data Unit Structure

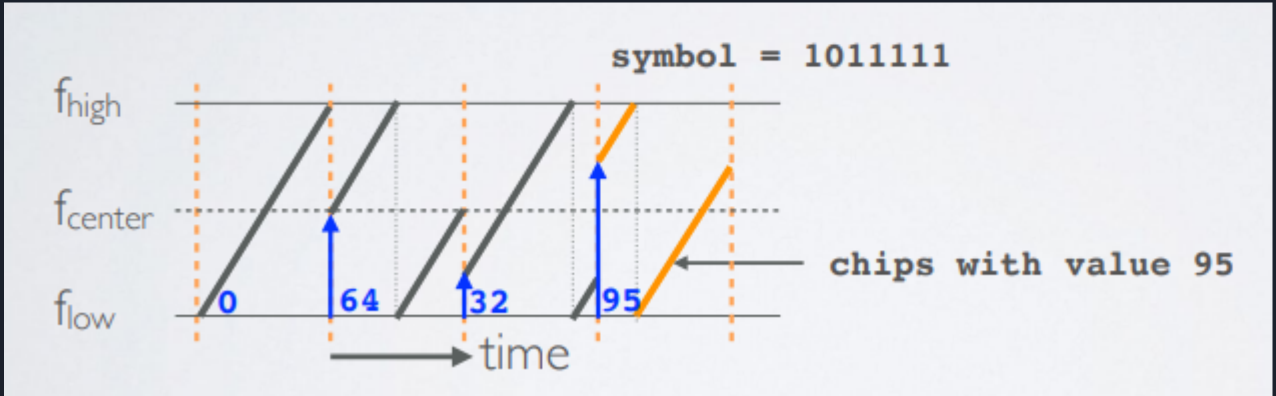
- Chirp undergo instantaneous frequency changes and “jump” throughout band.
- These are result of data being modulated onto the chirps.
- Symbol: a RF energy state that represent some data.





For Example:

- Symbol : 1011111 (decimal value = 95)
- Spreading Factor : number of bits encoded per symbol.
  - SF = 7
- The sweep signal is divided into 2^SF chips.
  - 2^7 = 128 chips
  - == chop the sweep signal into 128 pieces and call each piece a chip.
- Each sweep arrangement represent a value.

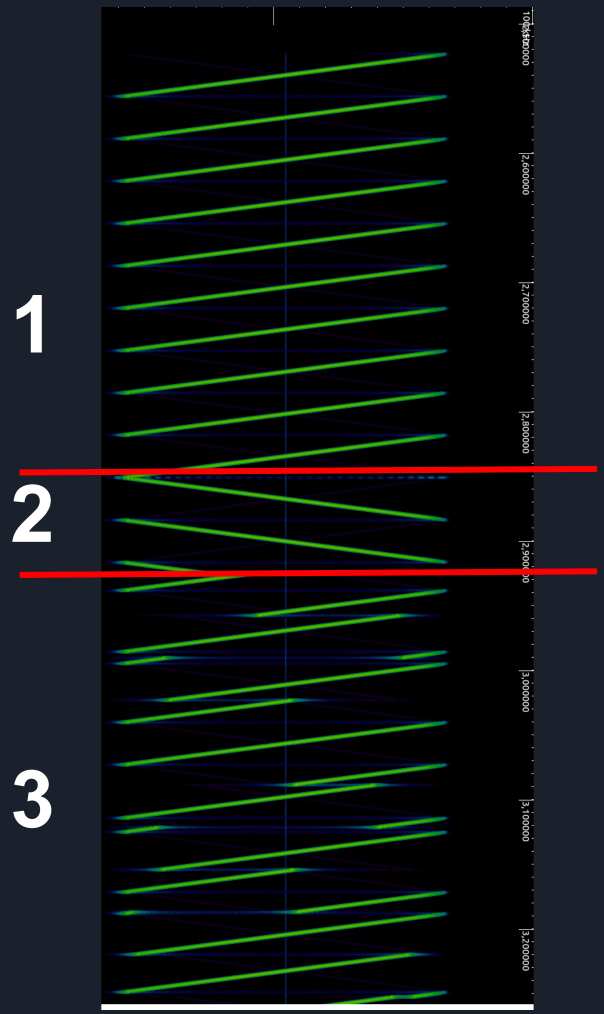


Demodulating the PHY

1. Identify the beginning of a frame.

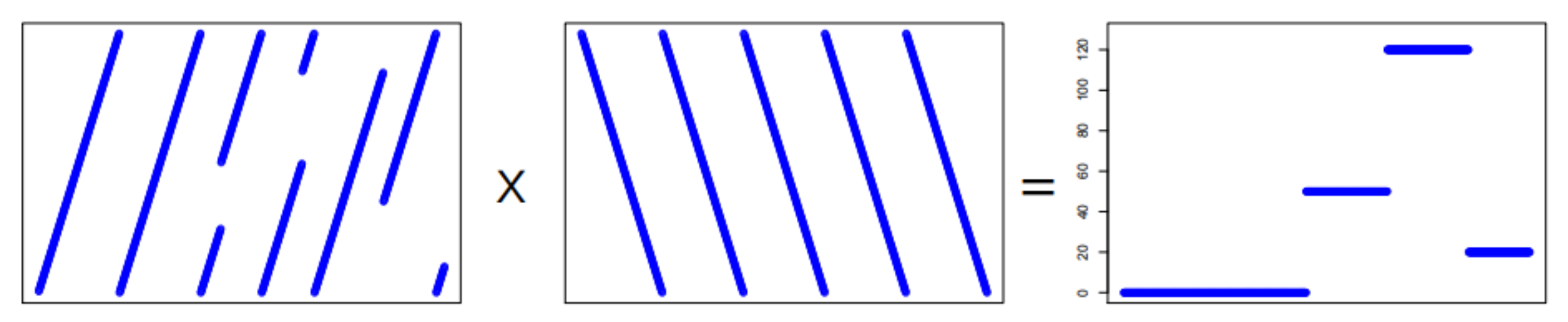
1. Find the beginning of the PHY data unit (SFD).

1. Extract data from instantaneous frequency transitions .



Transforming the Signal

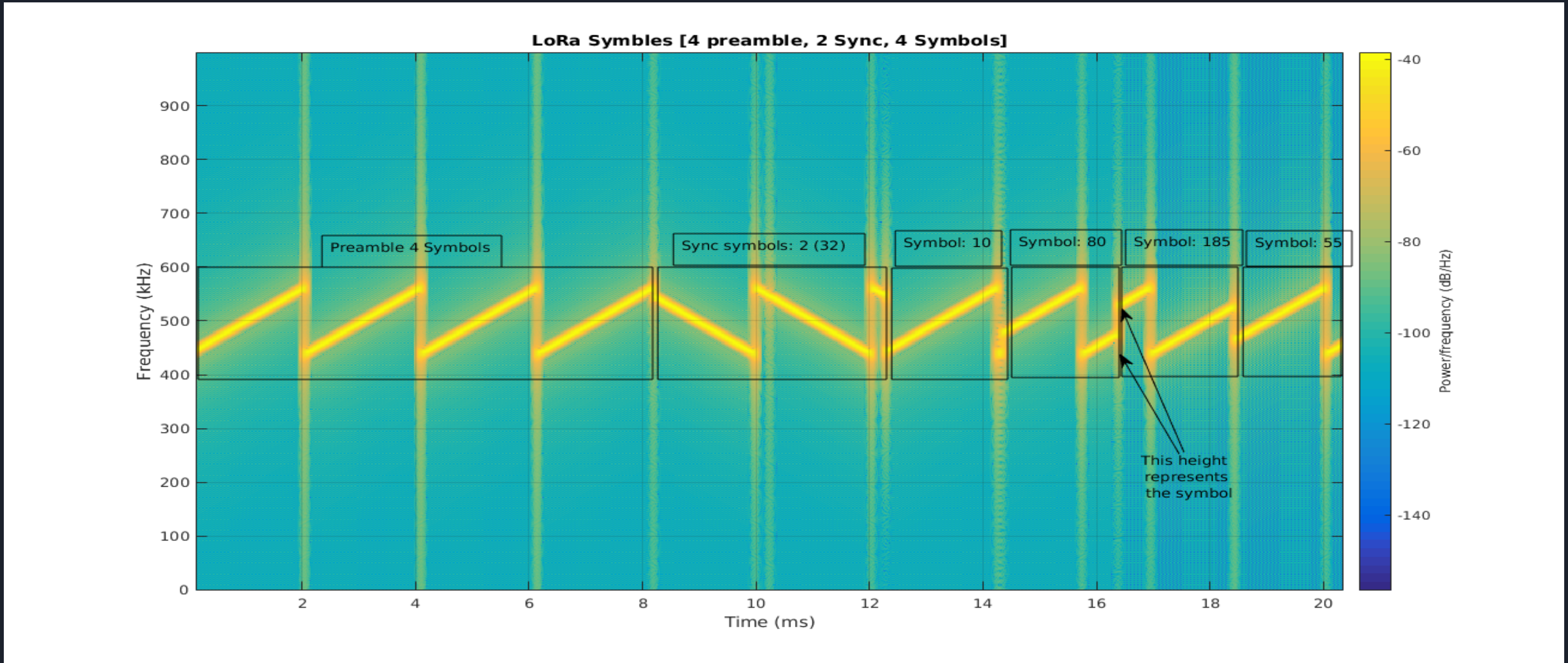
- Step 1: De-chirping the signal
  - While encoding LoRa, we cyclic shift the frequency of the LoRa signal.
  - So, if we multiply the received 'frequency shifted LoRa chirp' with 'inverse chirp' we will get the constant frequency signal, with a specific frequency characteristic of the transmitted signal.
- Step 2: Fast Fourier Transform
  - Take FFT of de-chirp signal and the symbol can be resolved by finding the argmax, (the strongest component) of each FFT



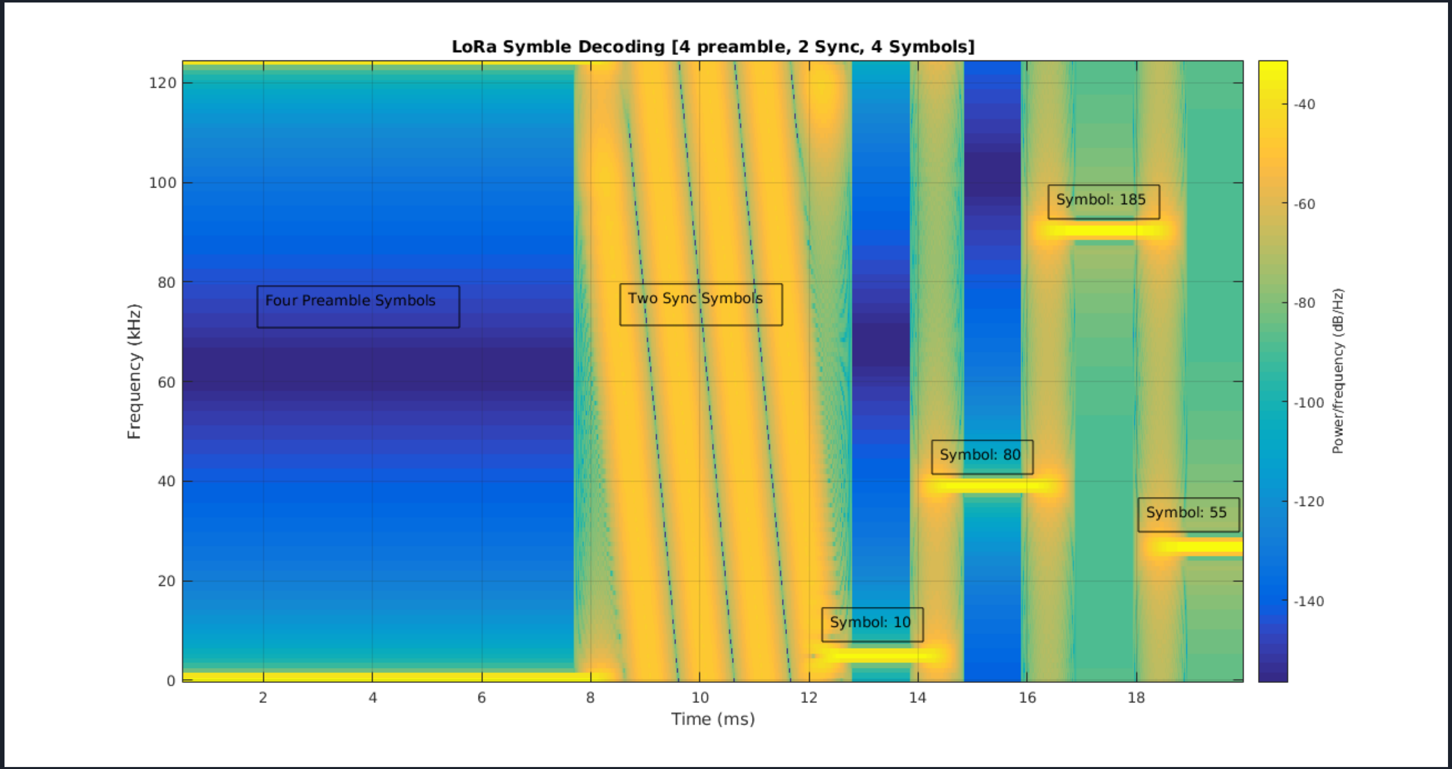
Received Signal **x** **Inverse chirp** **=** **Constant Frequency**

Example

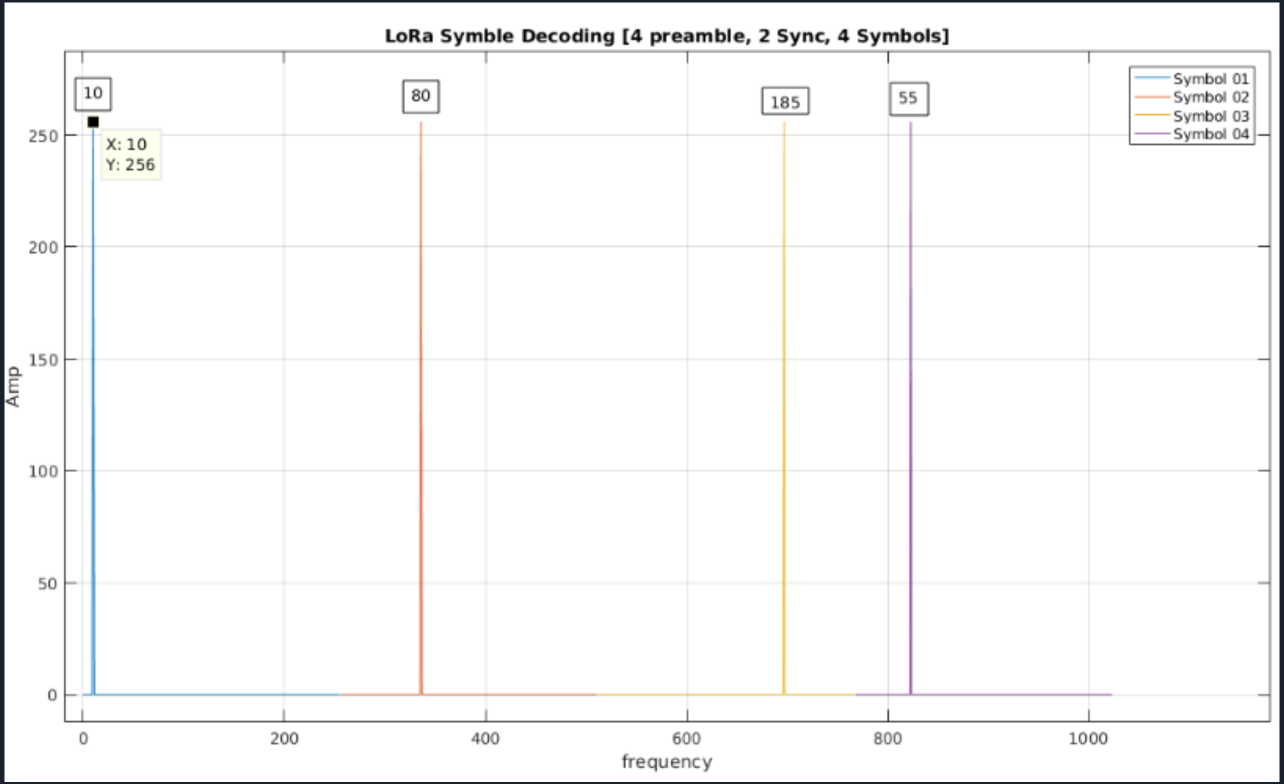
A LoRa packet which has 4 preamble symbols, 2 SFD symbols (32) and 4 message symbols (10,80,185,55) , with SF=8, is shown below:



At the receiver side, this message is multiple by inverse chirp and the spectrogram is shown below:



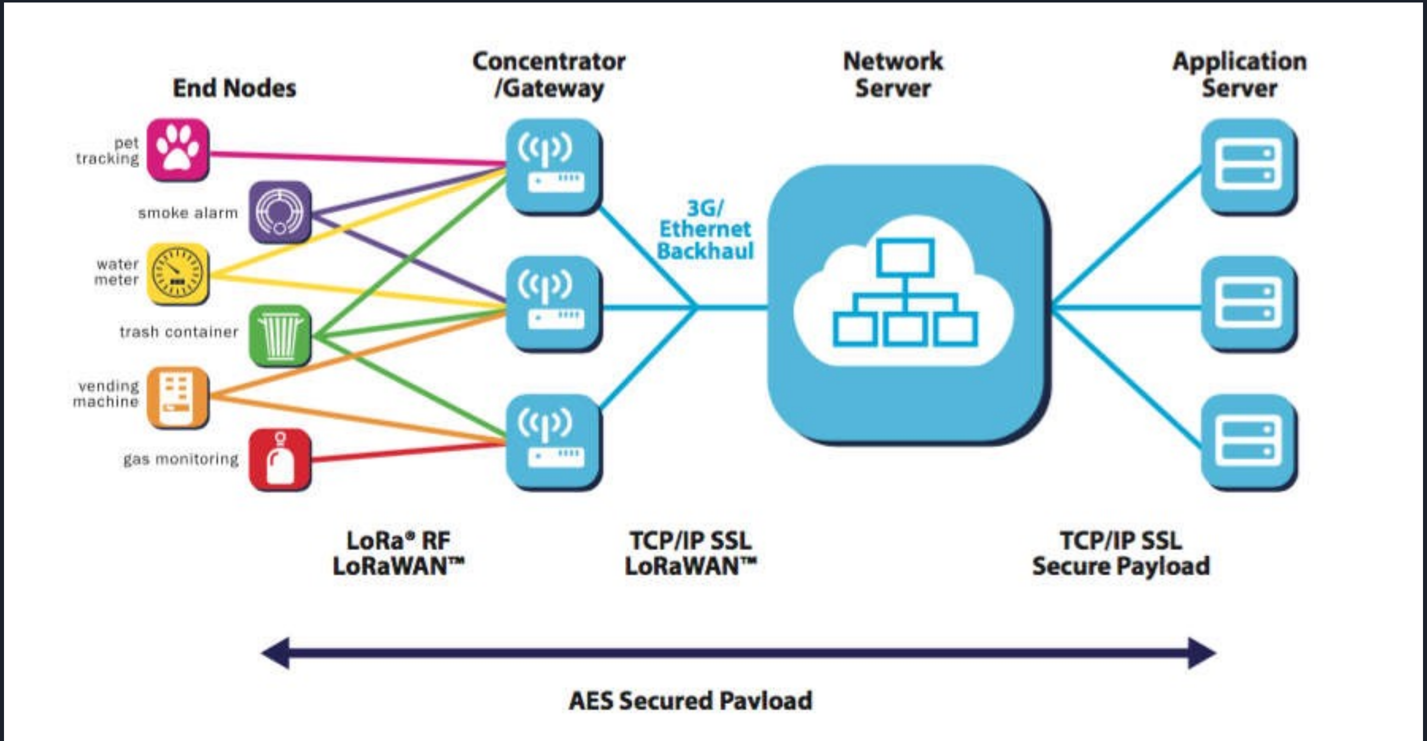
- Take FFT of de-chirp signal and check which frequency bin has highest energy.
- In the below graph the frequencies having highest energy in a symbol (FFT size 256) represents the data. This way the data can be extracted easily.



LoRaWAN

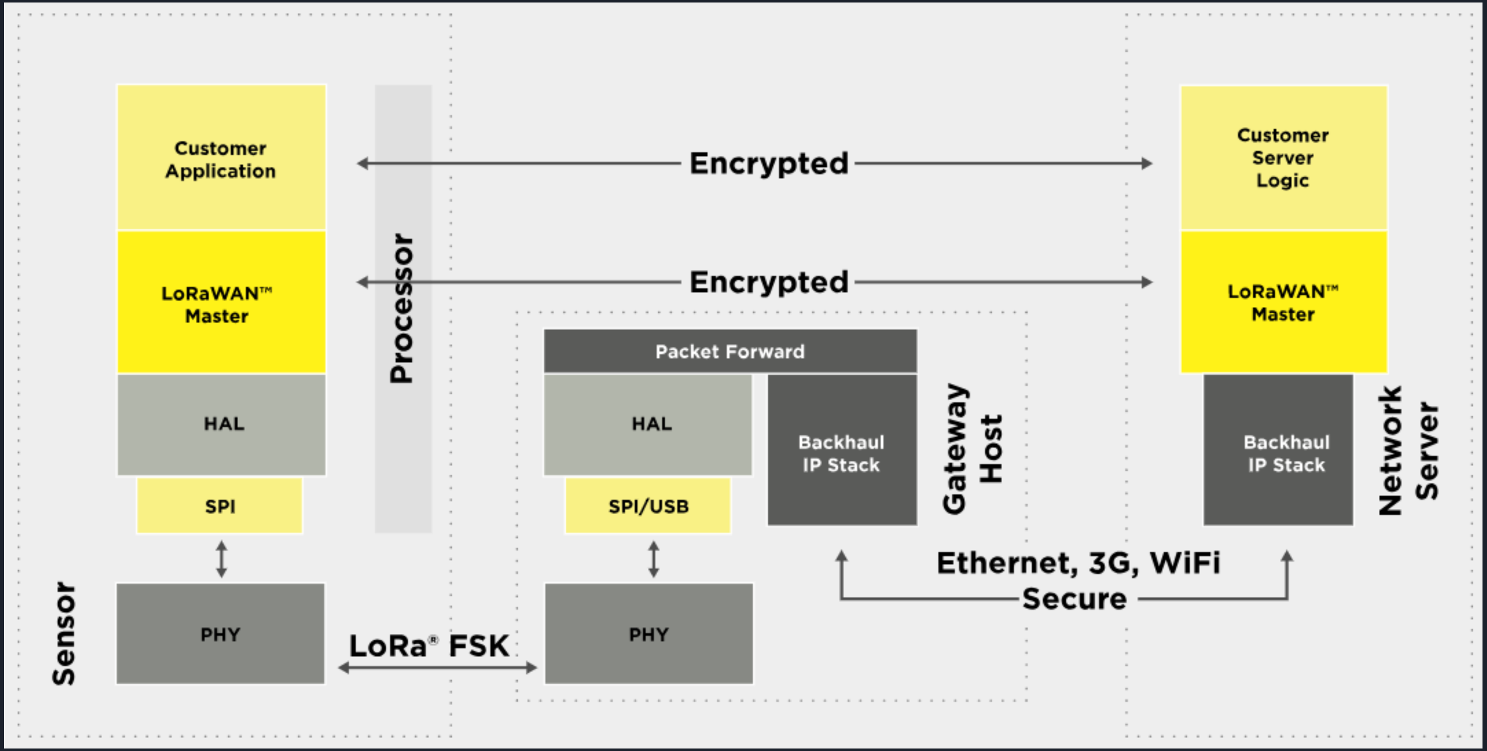
- Low Power, Wide Area Networking (LPWAN) Protocol
  - Open-standard
  - Maintained by the LoRa Alliance
  - Proprietary alternative: Symphony Link
  - defines device-to-infrastructure parameters and network protocol
- Components in star-of-star topology
  - End-devices , Gateways, Network servers
  - Gateways as link layer relay only
  - One-to-one relationship between end-device and network server

LoRaWAN



End Nodes
Concentrator /Gateway
Network Server
Application Server
pet tracking
smoke alarm
water meter
trash container
vending machine
gas monitoring
3G/ Ethernet Backhaul
LoRa® RF LoRaWAN™
TCP/IP SSL LoRaWAN™
TCP/IP SSL Secure Payload
AES Secured Payload

LoRaWAN

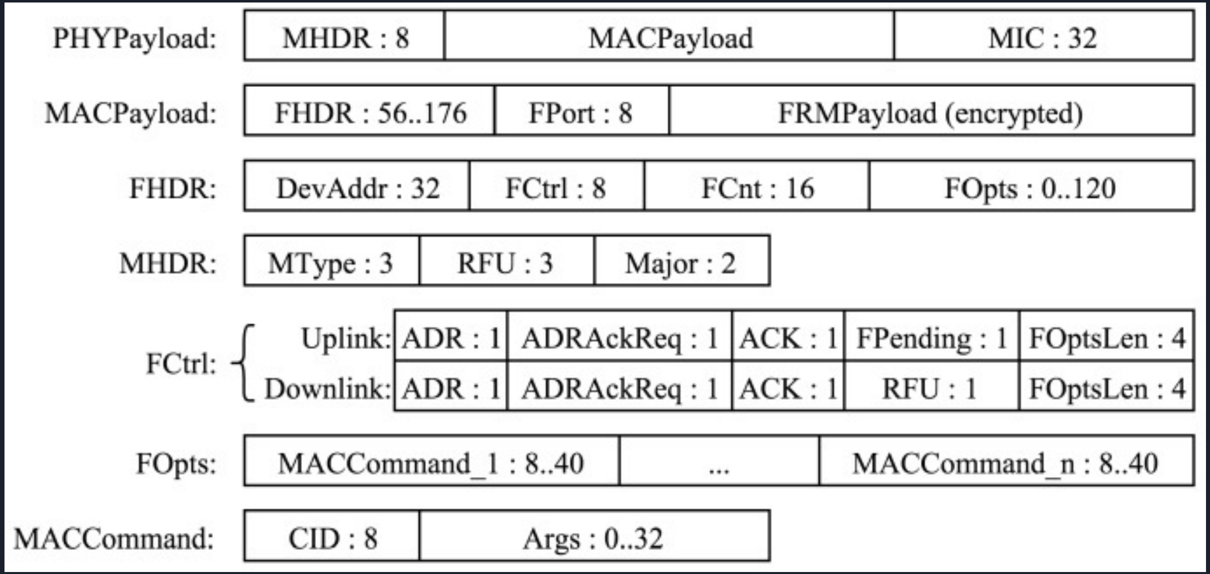


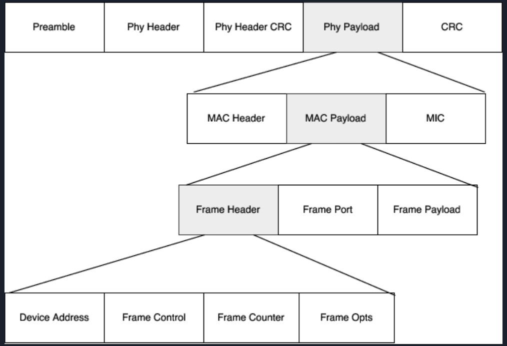
Customer Application
LoRaWAN™ Master
HAL
SPI
PHY
Sensor
Processor
Encrypted
Encrypted
LoRa® FSK
Packet Forward
HAL
SPI/USB
PHY
Backhaul IP Stack
Gateway Host
Ethernet, 3G, WiFi
Secure
Customer Server Logic
LoRaWAN™ Master
Backhaul IP Stack
Network Server

LoRaWAN - Device Classes

Class A	Mandatory; Free to schedule uplink subject to constraints; Two downlink windows following uplink; Lowest power consumption; High Downlink Latency;
Class B	Optional; Extends Class A; Beacon-synchronised downlink windows; Deterministic Downlink Latency ;
Class C	Optional; Extends Class C; Always listening, downlink window open whenever not transmitting; Lowest Latency;

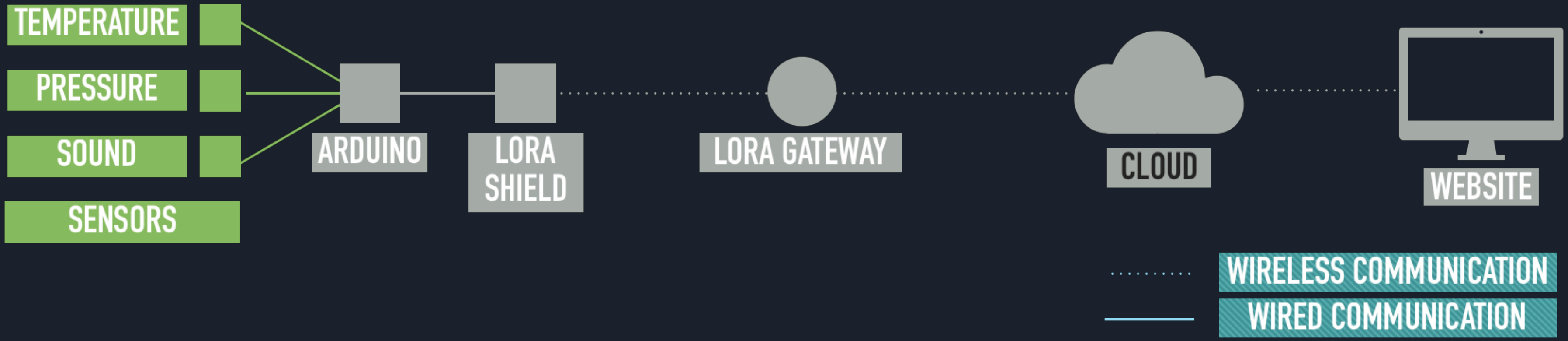
LoRaWAN - Message Format





Development

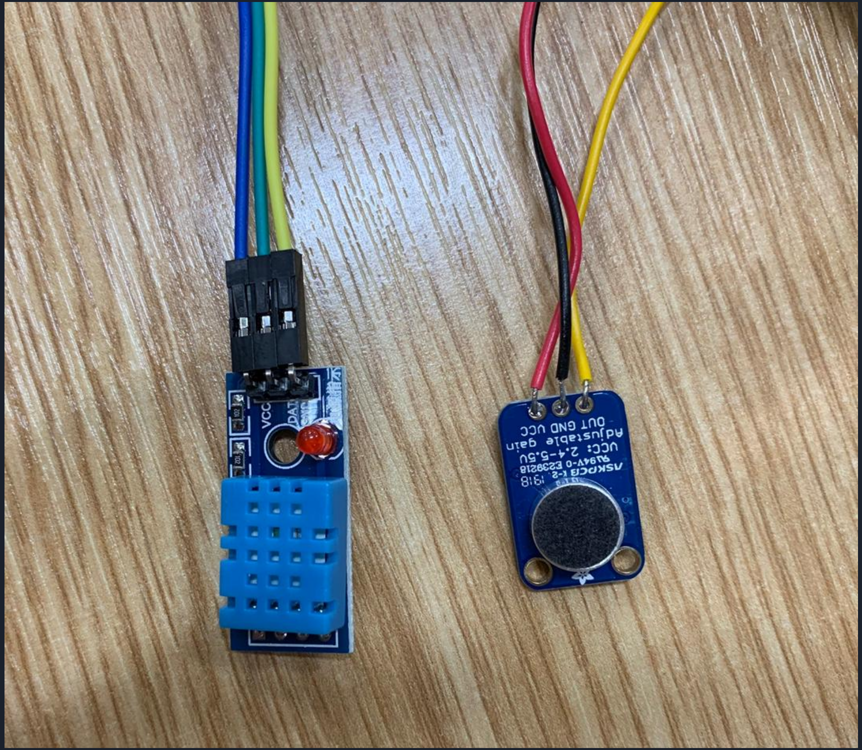
Development: Sensors



Sensors

- DHT11 Humidity & Temperature Sensor

- Adafruit 1063 Electret Microphone Amplifier



Development: Arduino & LoRa

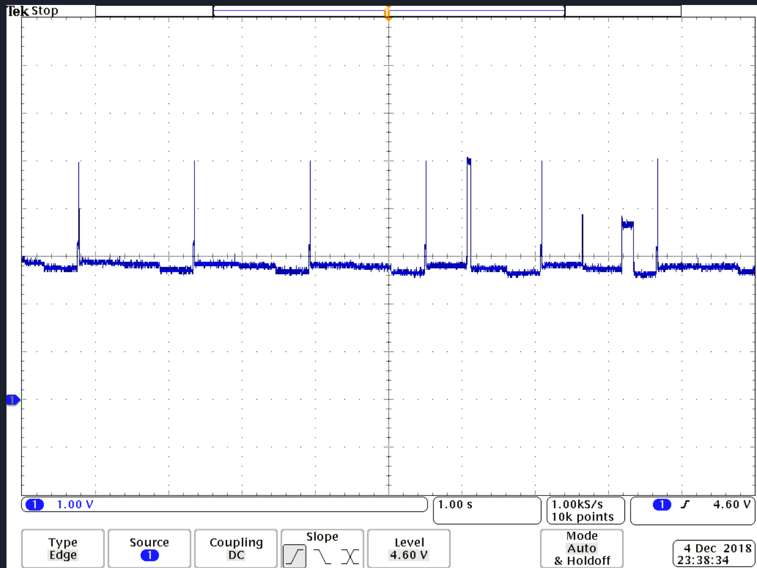
Arduino & LoRa

Dragino LoRa Shield v1.4

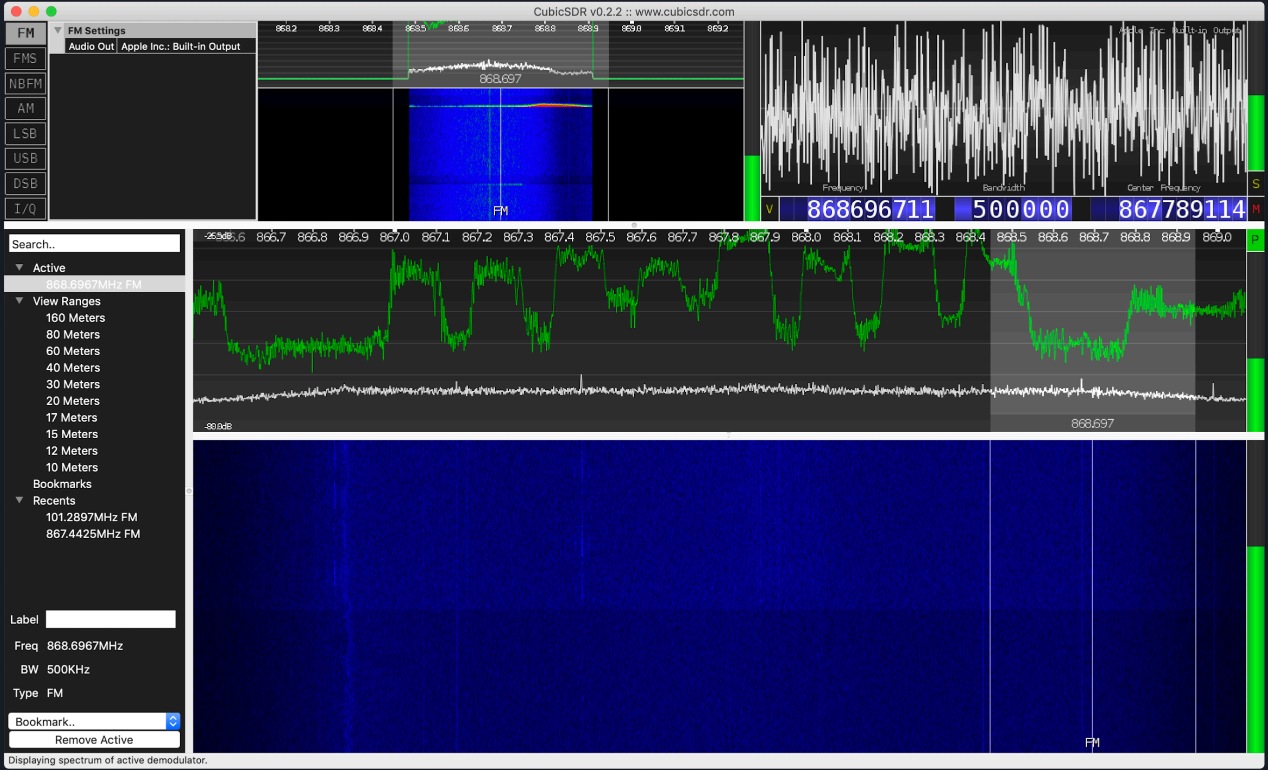
UNO Board

Arduino & LoRa

- OS with scheduler
  - Sensor reading every 1.5 seconds
  - LoRa transmission every 10 minutes (now 10 seconds)
- Use on campus TTN gateways
- Network servers by TTN
  - Device Address - Identification
  - Network Session Key - MIC
  - App session Key - AES



Arduino & LoRa



Development: LoRa & Cloud

LoRa & Cloud

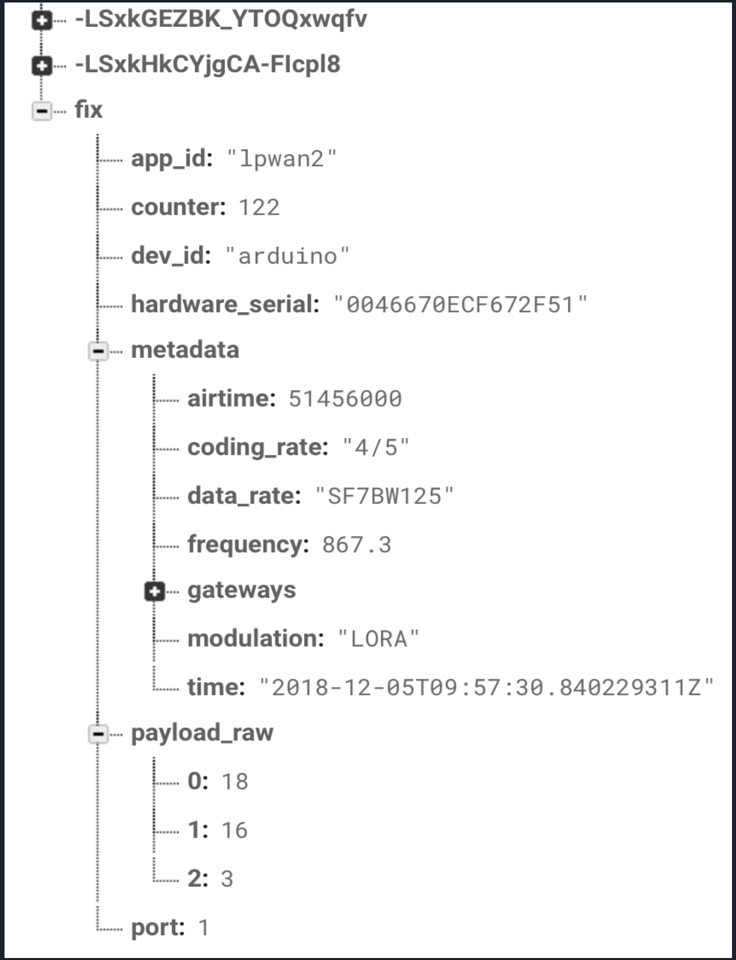
- LoRa transmissions received by gateways
- Data stored in the ‘Console’ of ‘The Things Network’ (TTN)
- Send data from TTN Console to Cloud
- Cloud service selected was Google’s Firebase
  - Real-time database
  - Hosting platform

LoRa & Cloud

- Each transmission was being routed into the TTN Console
- NodeJS was chosen to forward data to Cloud
- Simple script to collect new data from the Console and forward to Cloud

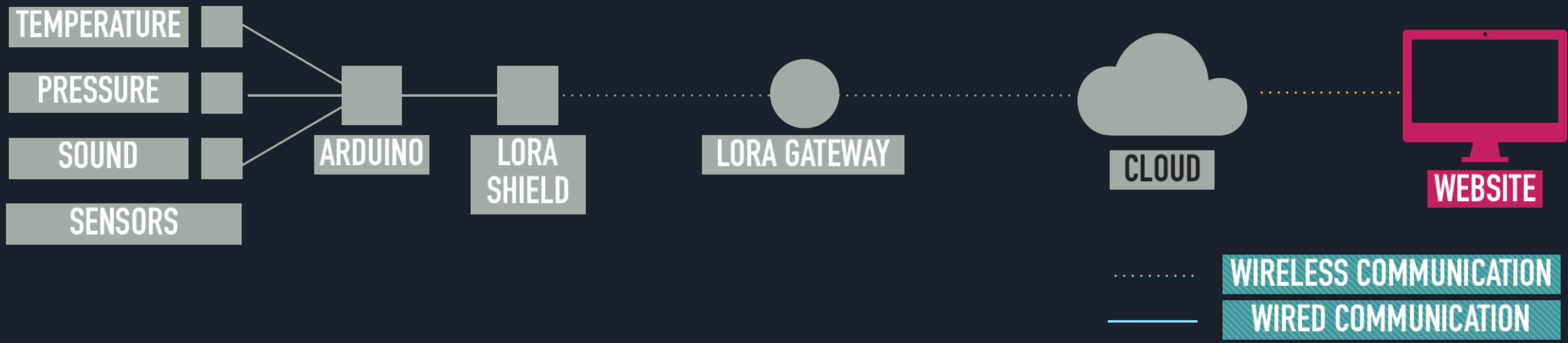
LoRa & Cloud

- Data now reaching Firebase
- Stored in JSON format
- Latest data stored under ‘fix’ branch
- Each transmission received contains:
  - Device that sent data
  - Timestamp
  - No. of gateways receiving data
  - RSSI and SNR
  - Payload



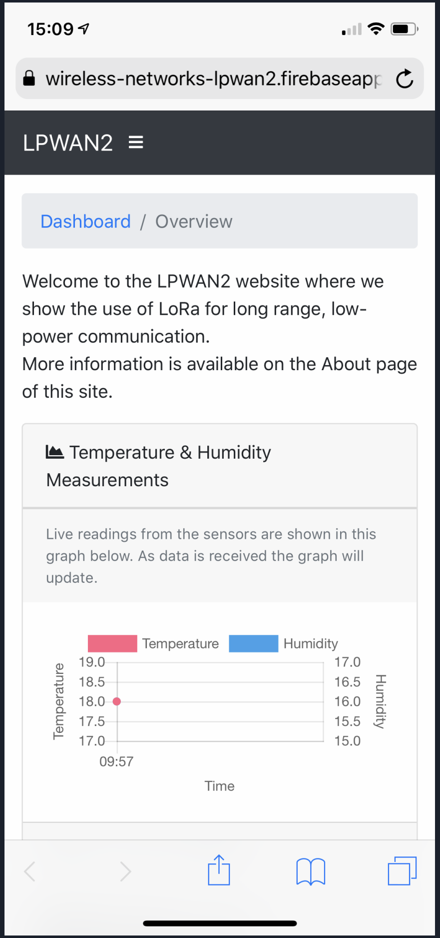
Screenshot from Firebase

Development: Website

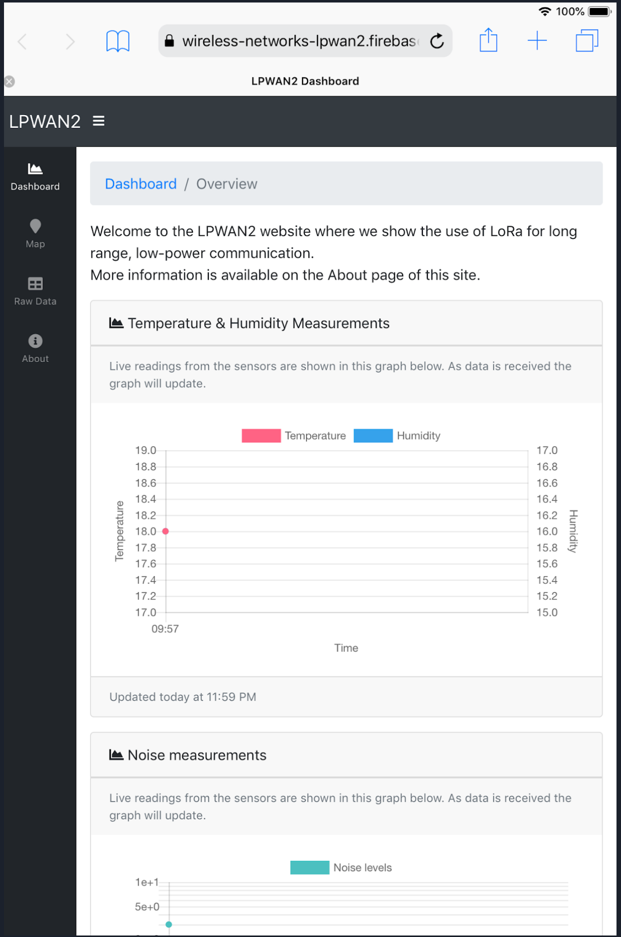


Website

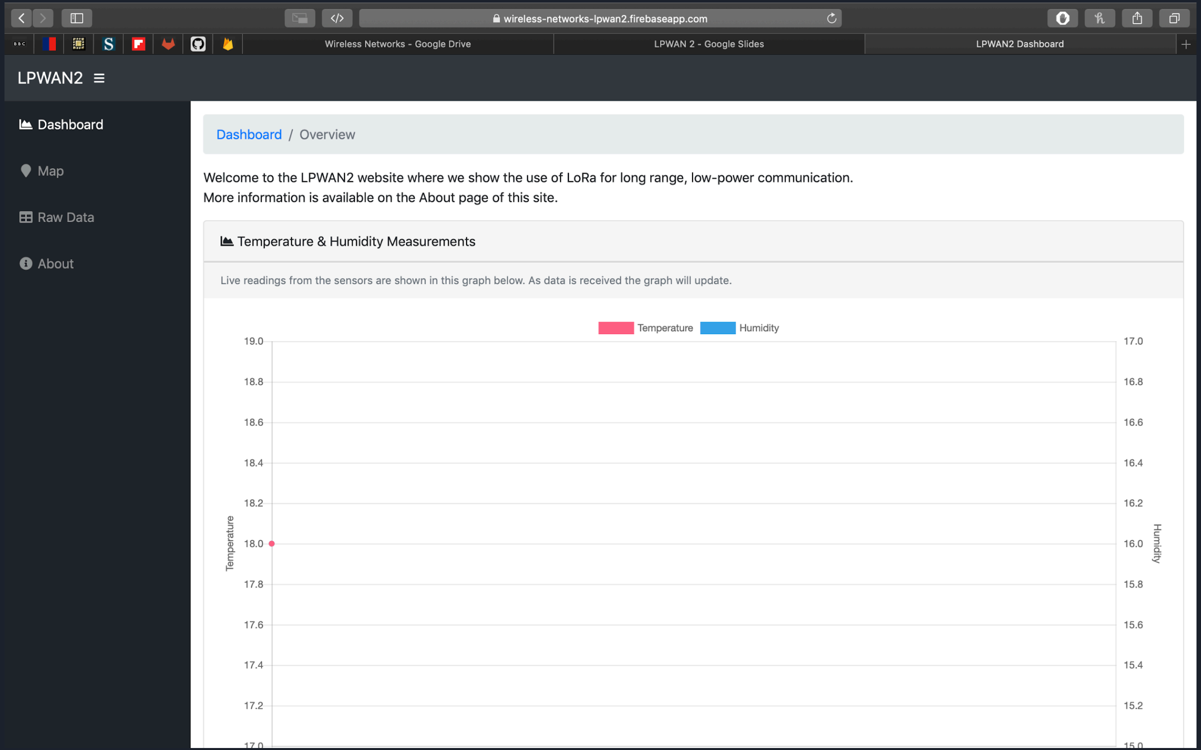
- Designed using Bootstrap
- Combination of HTML, CSS and JavaScript
- Using JavaScript to read/write from Firebase
- Responsive design for use on devices of all screen sizes



Mobile



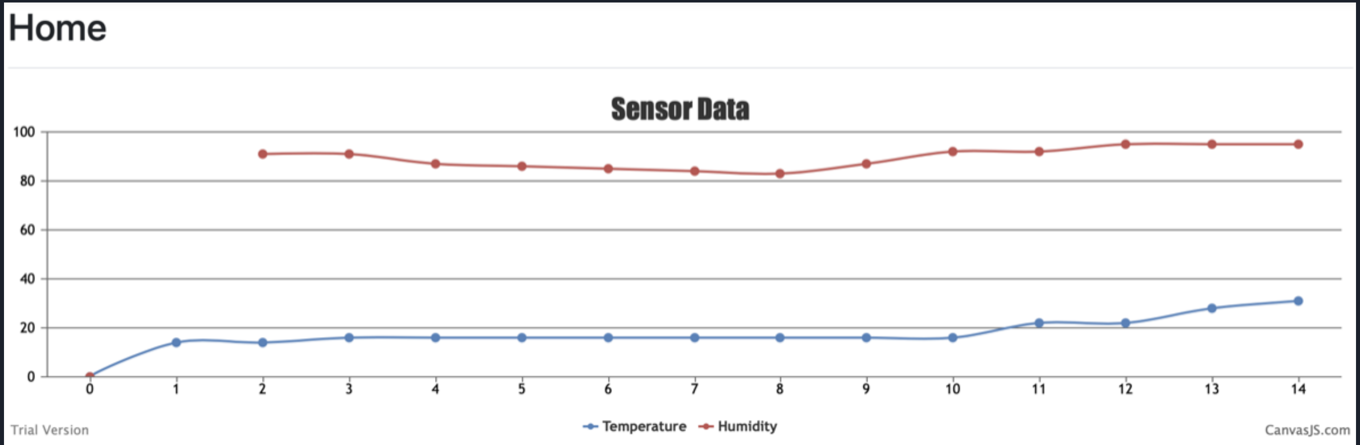
Tablet



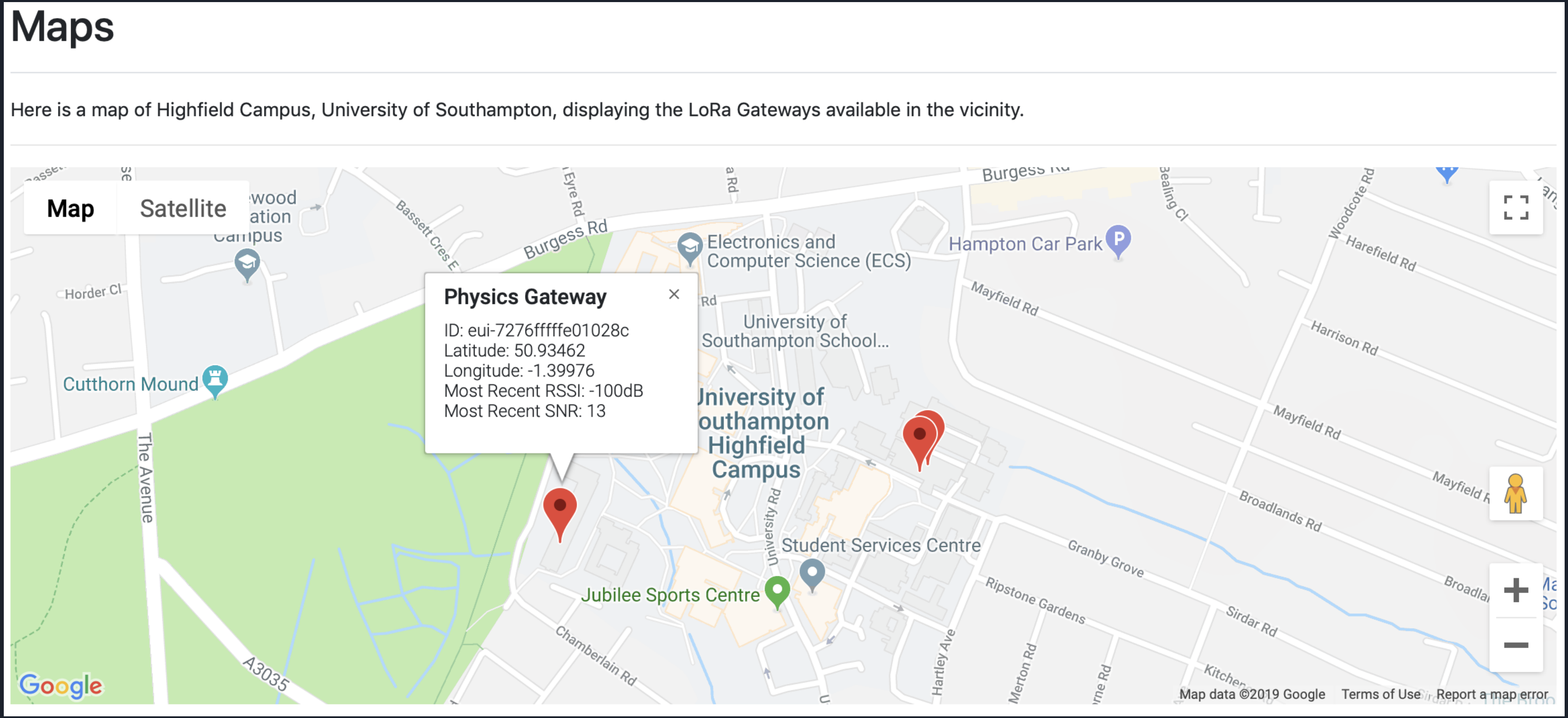
Desktop

Website

- Data is visualised through the use of charts and maps
- The raw payloads are also available through the site
- Dynamic data rendering using angularJS



Live data being streamed on the website as new readings become available

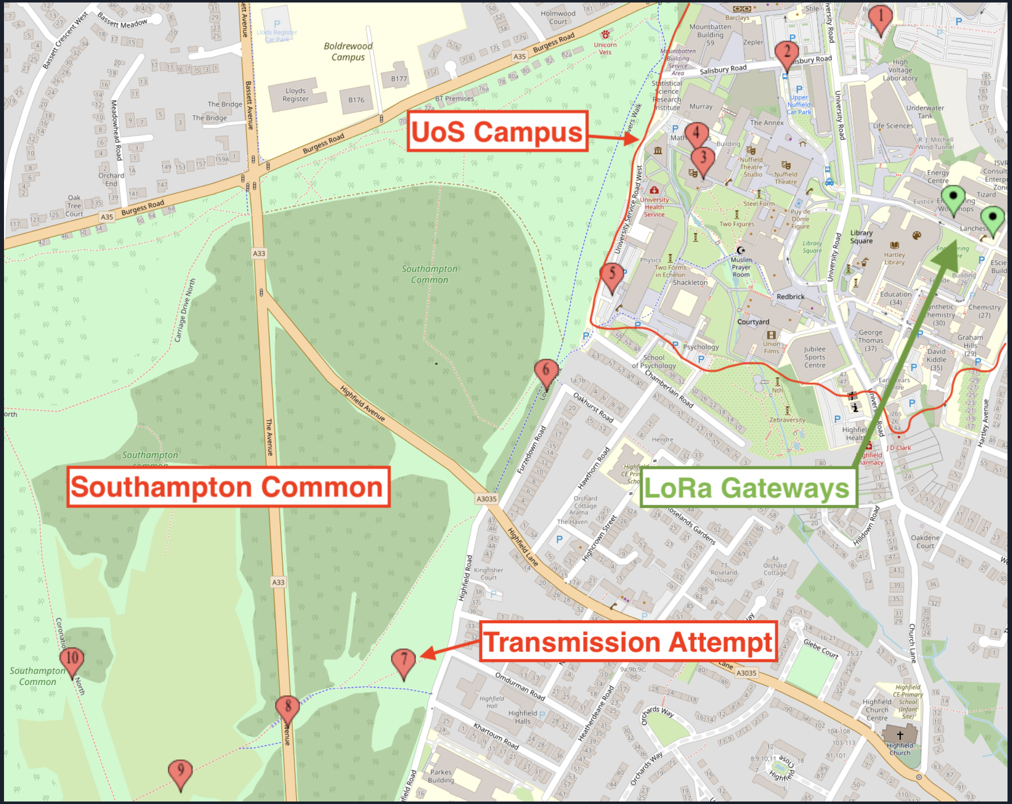


Google Maps embedded in the website showing gateways where data has been received

Testing

- Testing had three focal points:
  - Test of transmission reliability in different environments
  - Compare transmission reliability of different spreading factors
  - Test different Gateway antennas
- During testing two devices were used at each transmission location
- Identical except for the spreading factor used
- One used SF7, the other SF12
- SF12 should be received for a greater distance than SF7 due to the duration of transmission

Testing: Locations



- Environments used:
  - Line-of-sight of gateway antenna
  - Surrounded by multi-storey buildings
  - Underground (subway)
  - Open planes
  - Densely populated area of trees

Testing

- At each location for transmission these events occurred:
  - One transmission using SF7
  - One transmission using SF12
  - RSSI and SNR values recorded from Console on TTN
  - Measurements for the two Gateways on Building 7 were taken

Testing: Results

- Results show variations in signal strength due to various factors
- General Trends:
  - RSSI dropped with distance
  - SNR dropped to lowest values due to environment (buildings, trees etc.)
  - SF7 failed to send a message from Southampton Common when amongst trees
  - Signals received by the DIY Raspberry Pi gateway were lower strength than those on the multi-channel Kerlink gateway

Demonstration & Questions

Website: `https://wireless-networks-lpwan2.firebaseapp.com`

`https://tinyurl.com/lpwan2019`