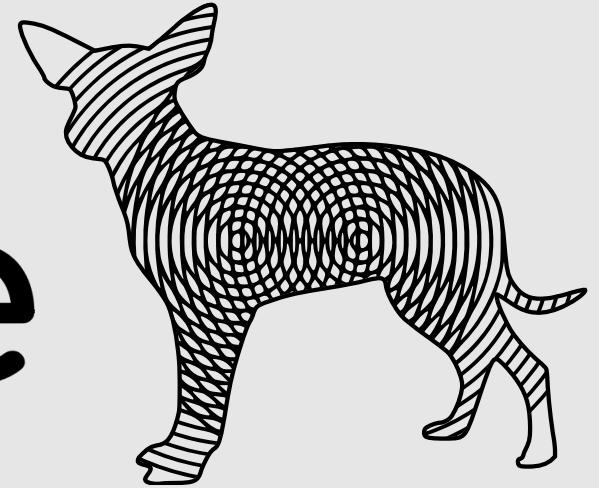


LORaware



Optimized Wireless Patient Monitoring

Master of Science in Electrical Engineering Capstone Project

Babatope Erinfolami, Antwane Green, David Rodriguez, Jorge Quiroga

Dr. Mohammad Amin, National University

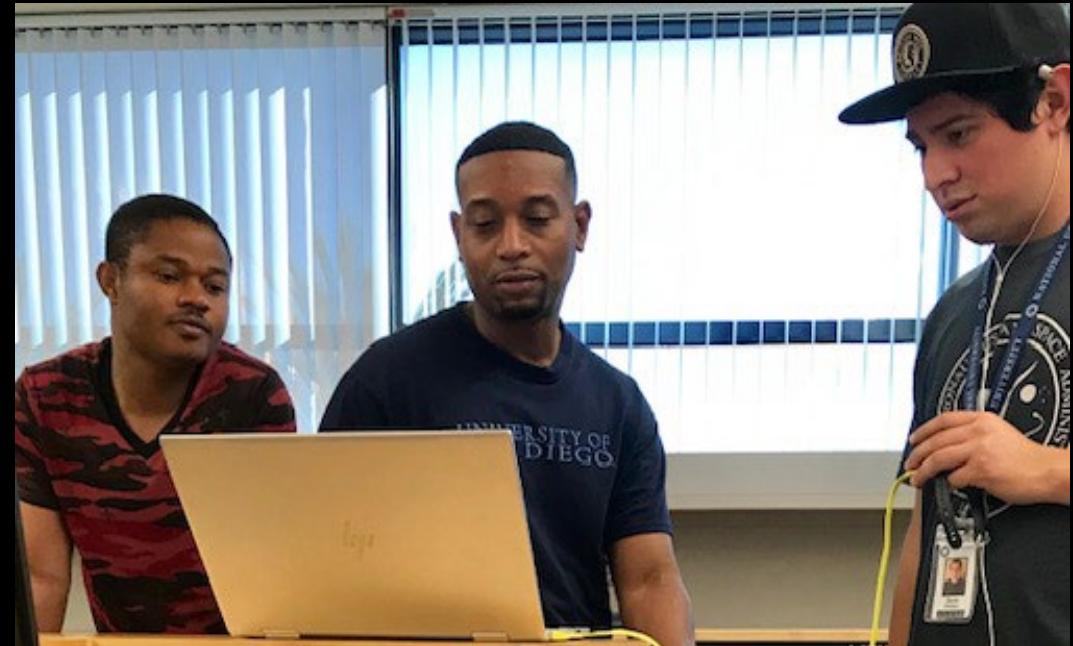
Introduction and Content

- Project Team
- Research
- Project Objectives
 - System Requirements
 - System Design
 - System Components
- Development
 - Spectrum
 - Component Integration & Assembly
 - System Management
 - Tools (HW/SW)
- Testing and Analysis
- Project Management
- Future Capabilities & Research
- Closing Points & Demo



Project Team

- Babatope (Sam) Erinfolami
 - Graduate of the Federal Polytechnic Ado-Ekiti, Nigeria, Higher National Diploma (BSEE equivalent)
- Antwane Green
 - Graduate of the University of San Diego (USD) with a BSEE
- David Rodriguez
 - Graduate of New Mexico State University with a BS in Engineering Physics and Mechanical Engineering
- Jorge Quiroga
 - Graduate of CSU San Jose, BSME; Naval Postgraduate School with a MS Operations Research; University of Phoenix , MBA



Research - Project Selection Background

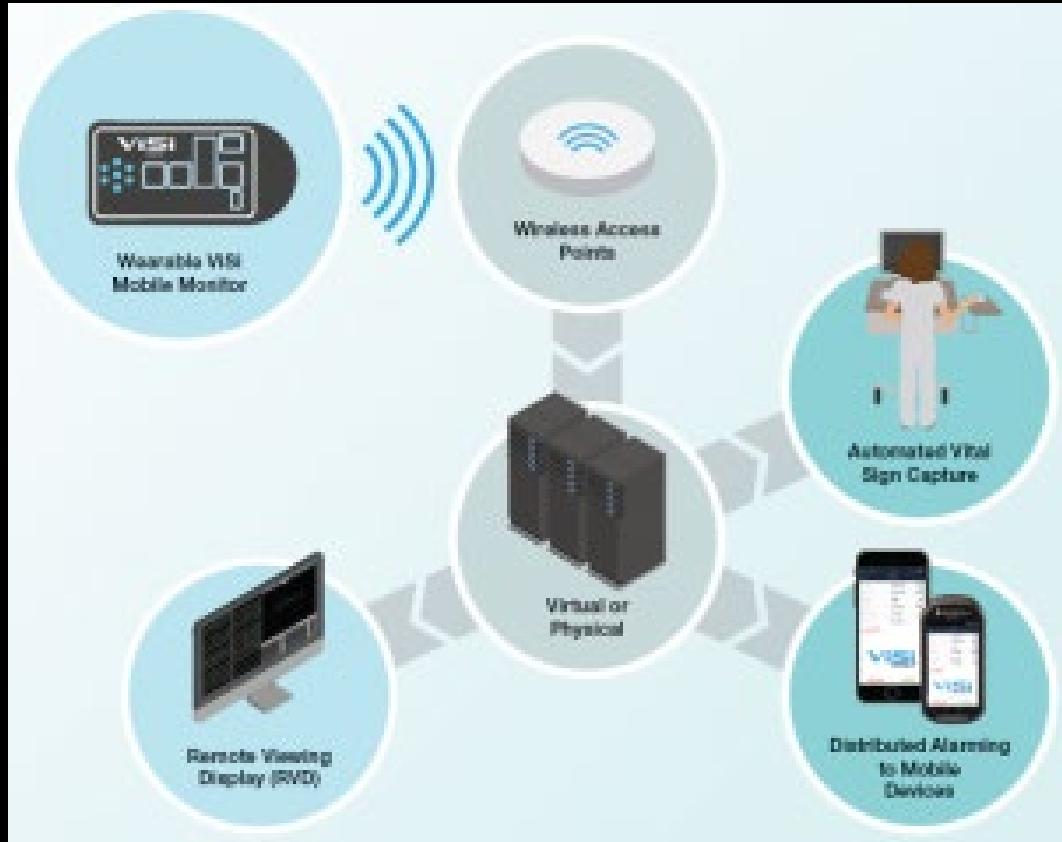
- Many Individuals are admitted to hospitals reporting minor to medium ailments or symptoms that are later discovered to be indicators of more severe underlying problems.
- At any point within their stay, these patients initially deemed “low-risk”, will be left unattended and waiting
 - To meet with a Doctor, Nurse, or Specialist
 - For Test Results to be Processed and Evaluated
- 75% of preventable hospital deaths outside of the ICU, occur while unmonitored (AHA Database, 2013)

Research – Other Related Applications

- According to the World Health Organization (WHO)
 - In 2015
 - over 17.7 million people died from cardiovascular diseases
 - 31% of the global deaths
 - Raised emphasis on health awareness, prevention of various vascular diseases and fitness
 - Increase in use of wearable medical devices and fitness devices, results in increased demand for these types of devices
 - The global Heart rate monitors market is expected to grow at a Compound Annual Growth Rate of 13.50% (According to the Market Research Future MRFR press release) during forecast period 2017-2023

Research - Commercially Available Products

- Wireless Device Power Consumption Benchmark Based on Sotera Wireless ViSi Patient Monitoring System: 14 – 16 hours



Research - Wireless Patient Monitoring

- Commercially Available Solutions show Room for Optimization
- Problems stem from using Wi-Fi on Wireless Wearable Sensor Devices
 - Bulky devices due to Large Batteries
 - Relatively Short Battery Life
 - Built on Expensive Proprietary Tools, Utilities, and Applications
 - More Devices on Existing Networks Compete for Available Bandwidth
- The Use-Case of monitoring low-risk Patients (not admitted to ICU) suggests a greater importance on Reliability and Security, rather than High Speed and High Data Transfer Rates.
- End-Users of these products
 - Hospitals and clinics
 - Sports medicine centers
 - Medical Professionals & Researchers

Project Objectives

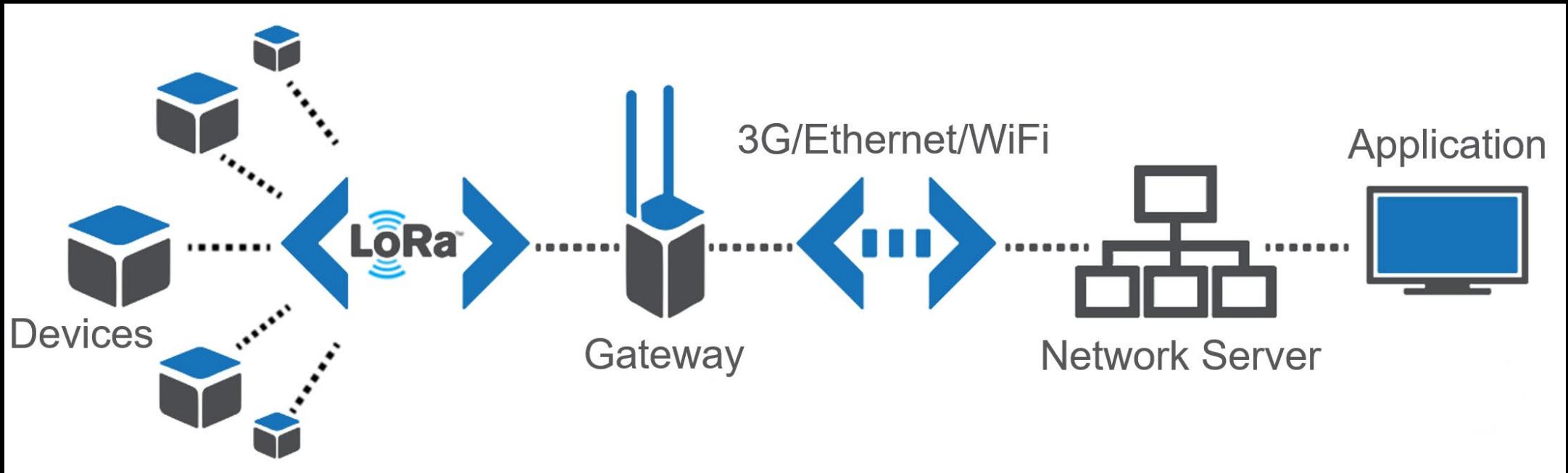
- Design, prototype, test, and manufacture an industry-competing wireless patient monitoring system including
 - 4 Custom Wearable Sensor Node Devices
 - Database Hosted on Webserver
 - Data Visualization Platform Hosted on Webserver
- Use low cost components, free & open source software
- Use Commercially Available Wireless Patient Monitoring System as Benchmark in terms of
 - Power Consumption / Battery Life (14 – 16 Hours)
 - Size & Formfactor
 - Cost of System Deployment
- Test Wireless Device Performance, Compare Estimated Results with Measured Results

Objectives - System Requirements

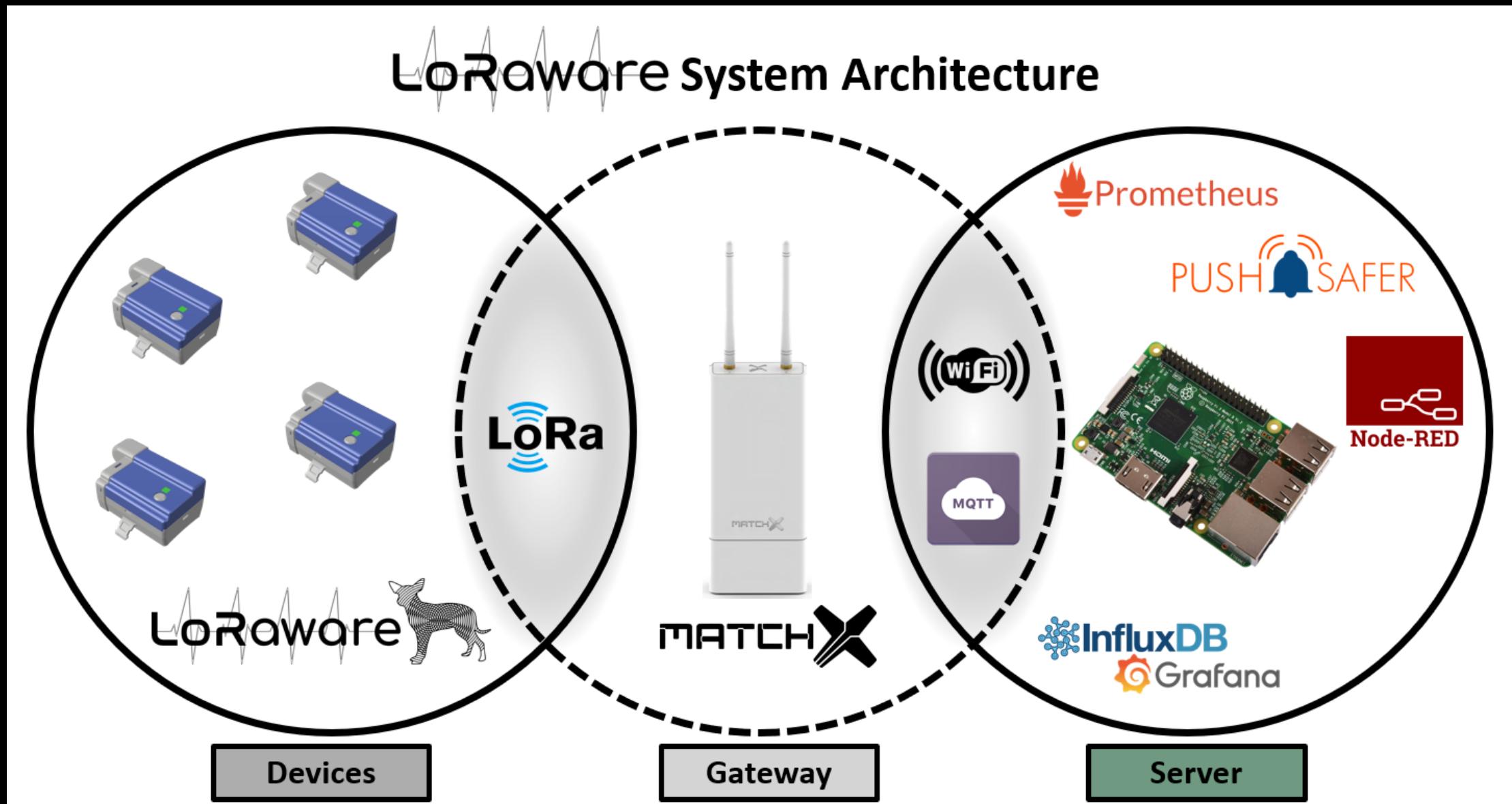
- Reliable Storage, Monitoring, and Data Visualization Access
- No Reliance on pre-existing Wi-Fi Infrastructure
- No Interference with pre-existing Wi-Fi Infrastructure
- Secure Transfer of Patient Biometric Data
- Battery Life exceeding 16-hours

Objectives – Employ LoRaWAN Architecture

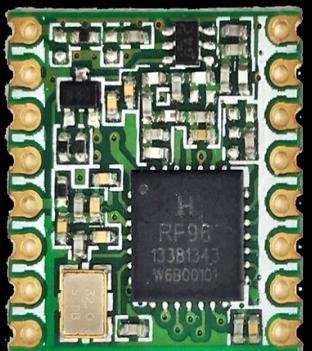
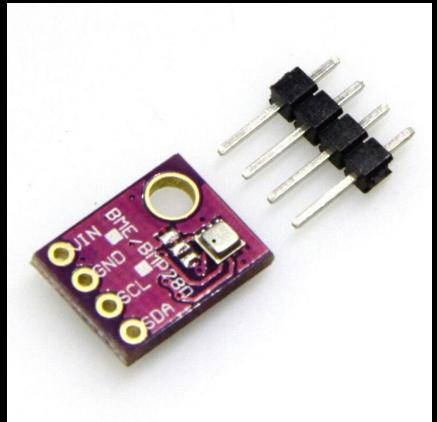
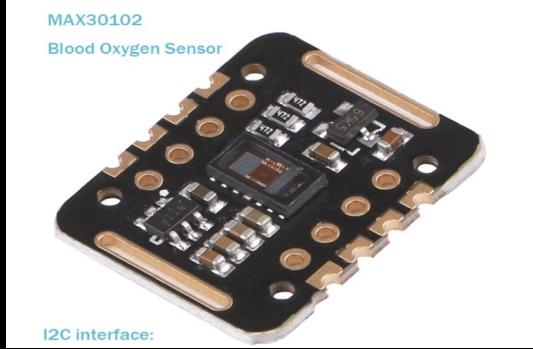
- Standard Network Topology



Objectives - System Layout



Objective – Low Cost Device Components



Control:

- ESP32 Microcontroller

Transducers:

- MAX30102 Heart Rate Sensor and Blood Oximetry Sensor
- BME280 Temperature and Humidity Sensor

RF:

- RFM95W RF Module
- LoRa 915 MHz Antenna

Power:

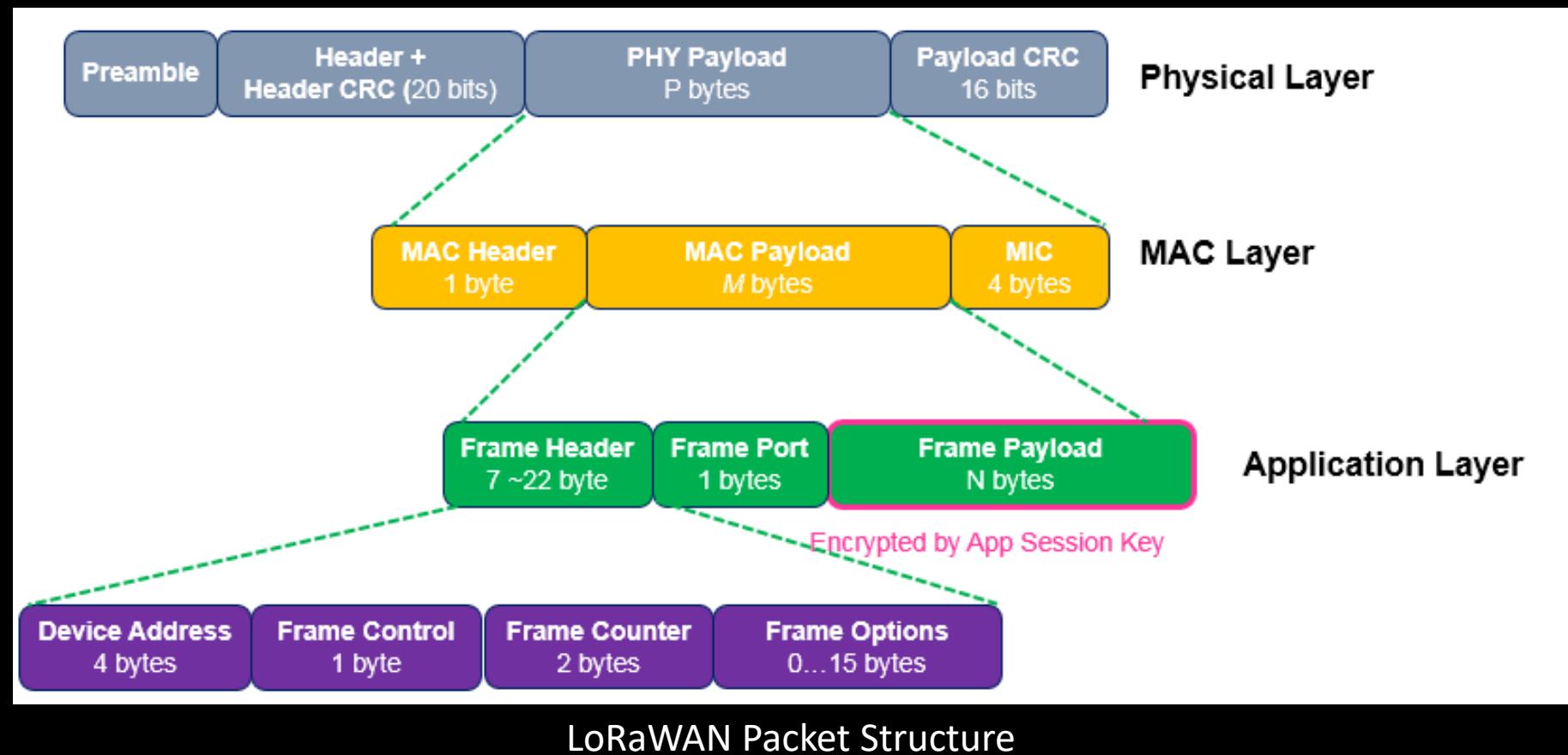
- 2000 mAh Lithium (Li) Ion Polymer Battery
- TP4056 Li-ion Charge Protection Integrated Circuit (IC)

Integration and Structure:

- Printed Circuit Board (PCB)
- Case

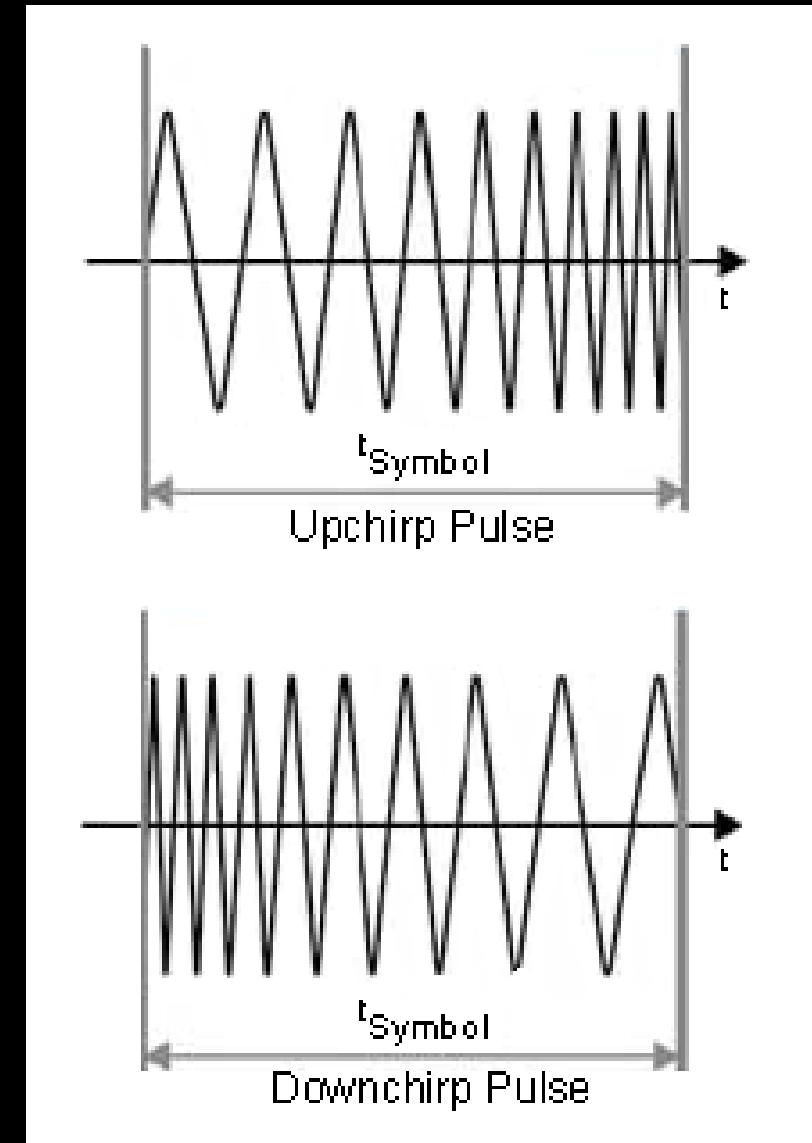
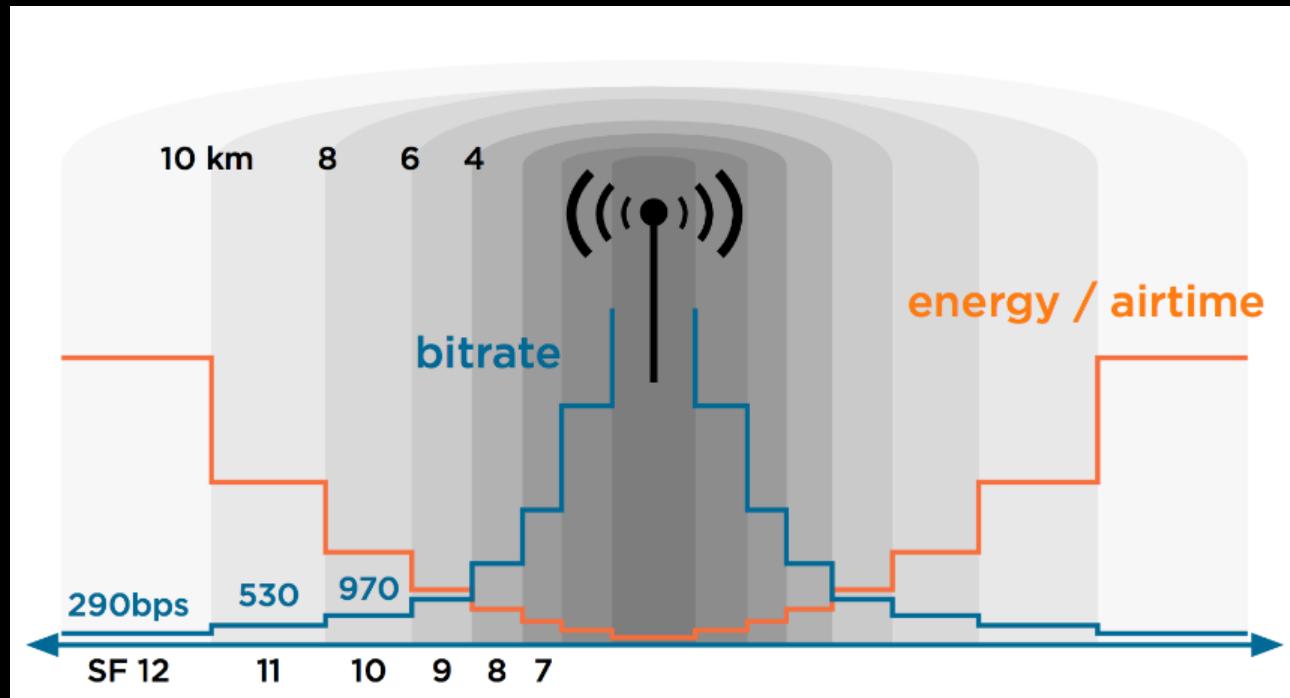
Objectives – Network Security

- LoRa Physical Layer Encryption
 - Network Session Key – 128-Bit AES Encrypted Frame (Header + Payload), used to sign the message for Network Server to verify ID of sender
 - Application Session Key – 128-Bit AES Encrypted Payload Value in Frame, only decrypted by the Application running in Gateway
- Server-Side Encryption
 - TLS Encryption via SSL Certificate



Development - LoRa “Physical Layer”

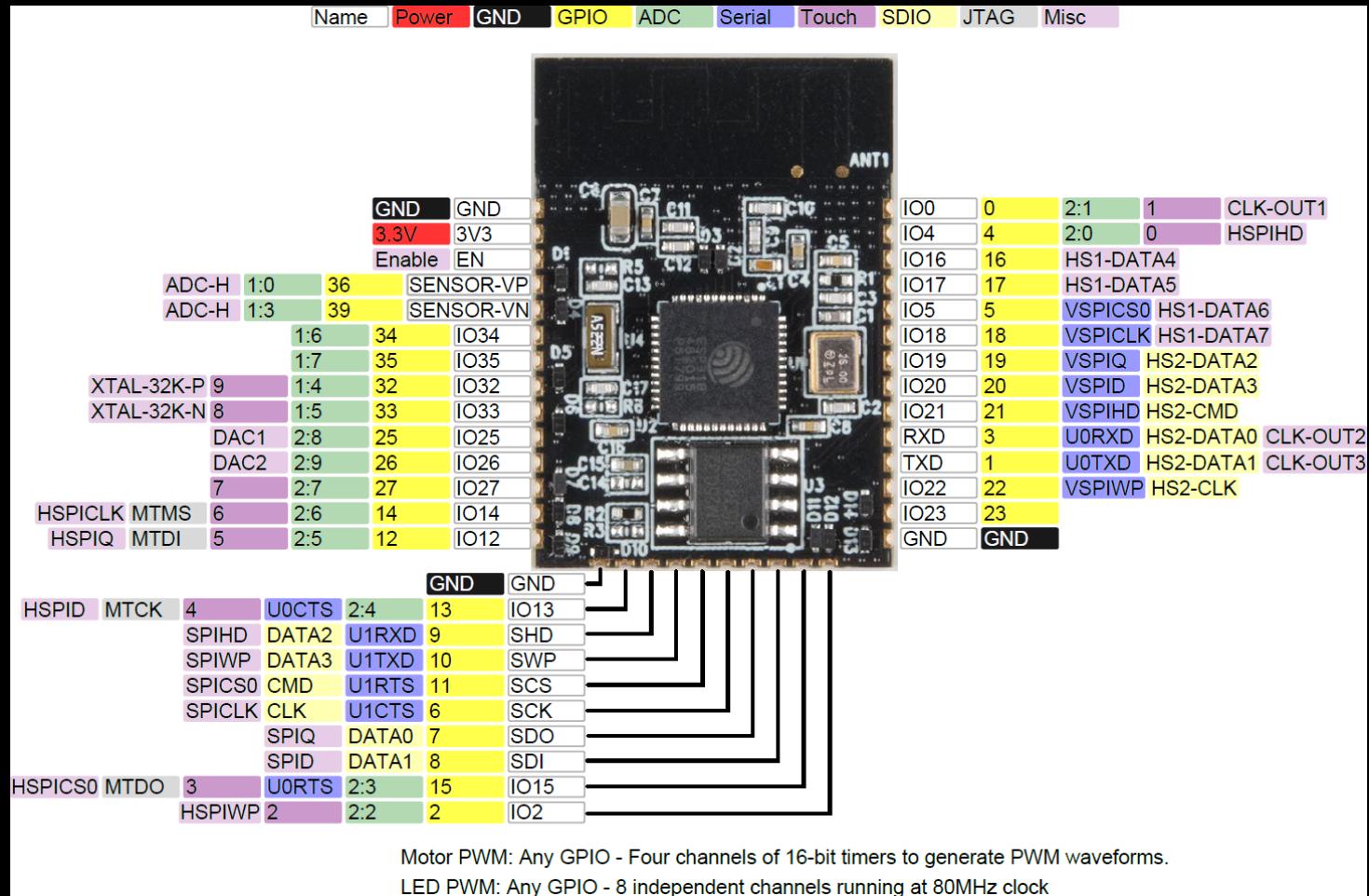
- Uses a modified Chirp Spread Spectrum Modulation Technique borrowed from IEEE 802.15
- Spread Spectrum AKA Spreading Factor – used to improve range and data rate by adjusting transmission power to distribute data across wider bandwidth of the spectrum
- Forward Error Correction



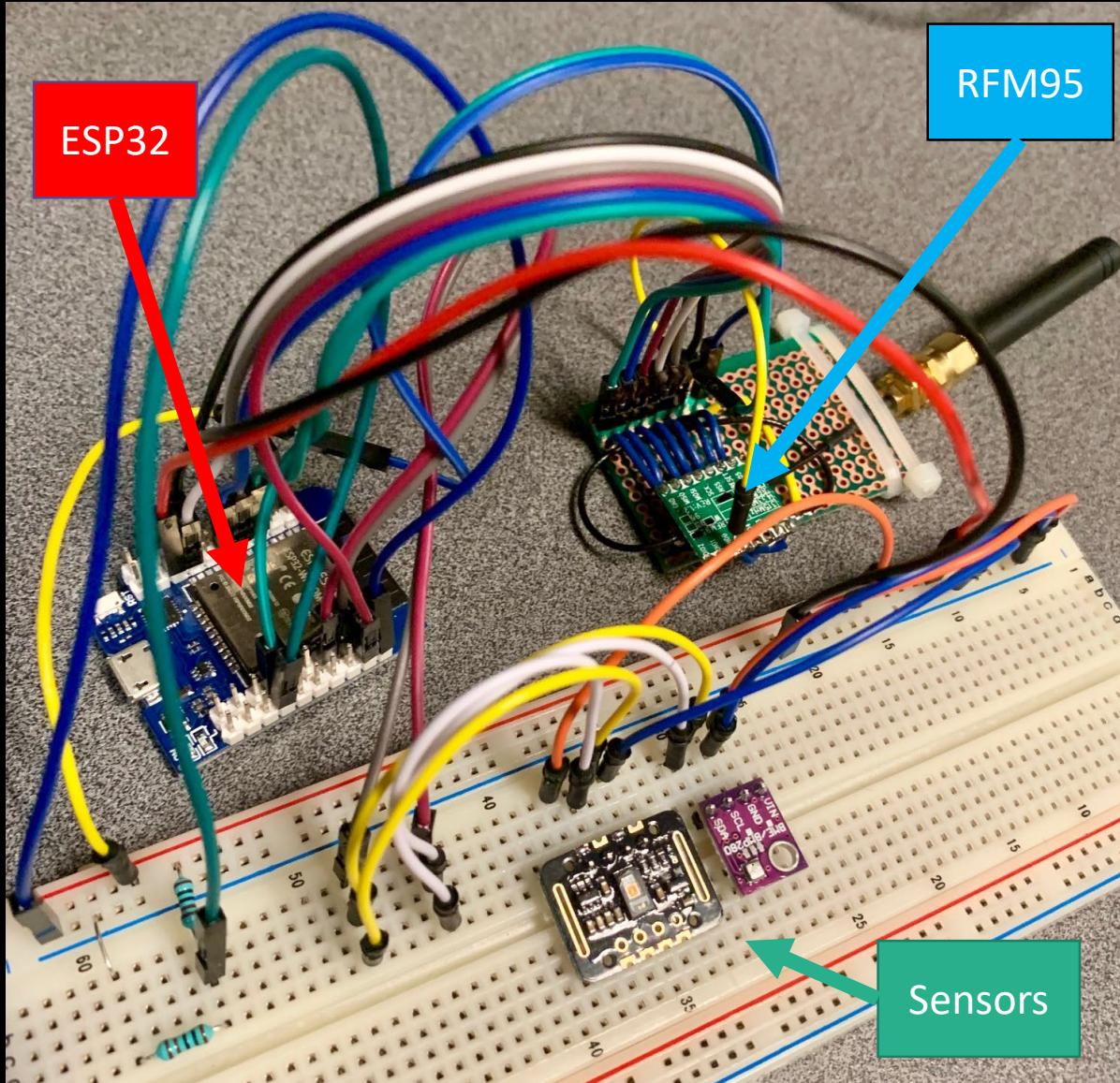
Development - Why use the ESP32 as main processor in LoRaware wearable devices?

- Function

- Low Cost
- Low Power
- Wi-Fi
- Bluetooth 3.0 & 4.0

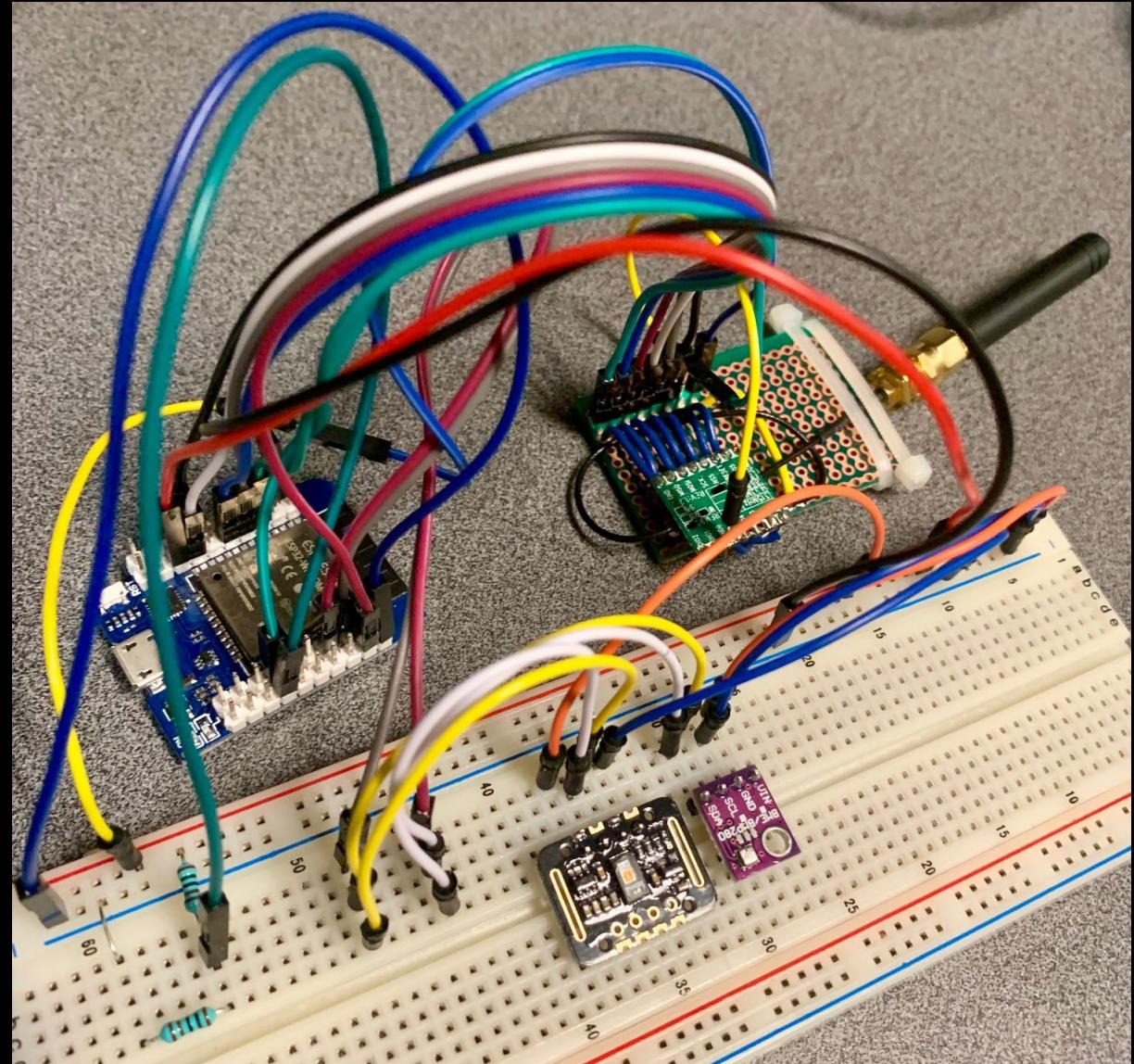


Development – Device Proof of Concept



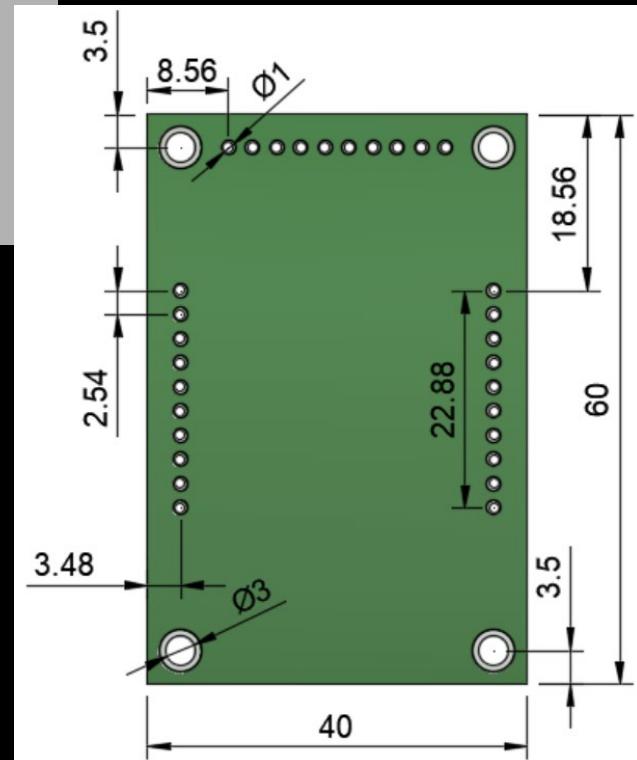
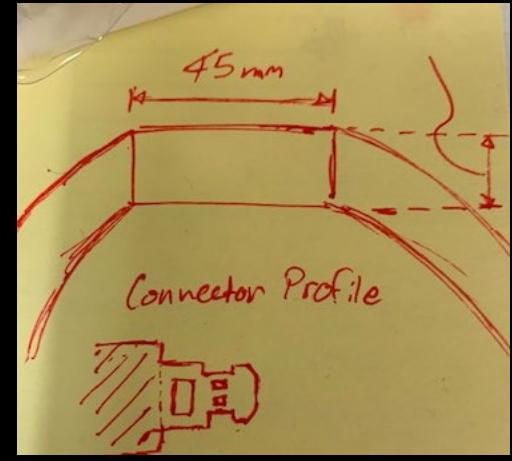
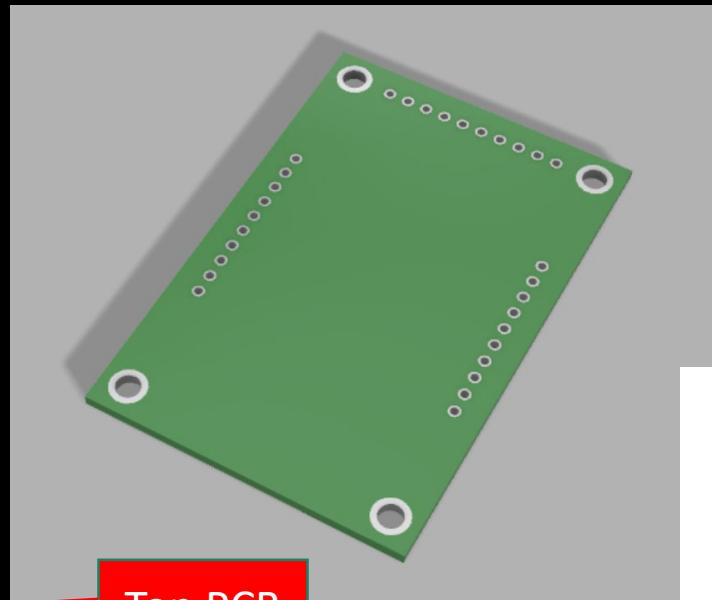
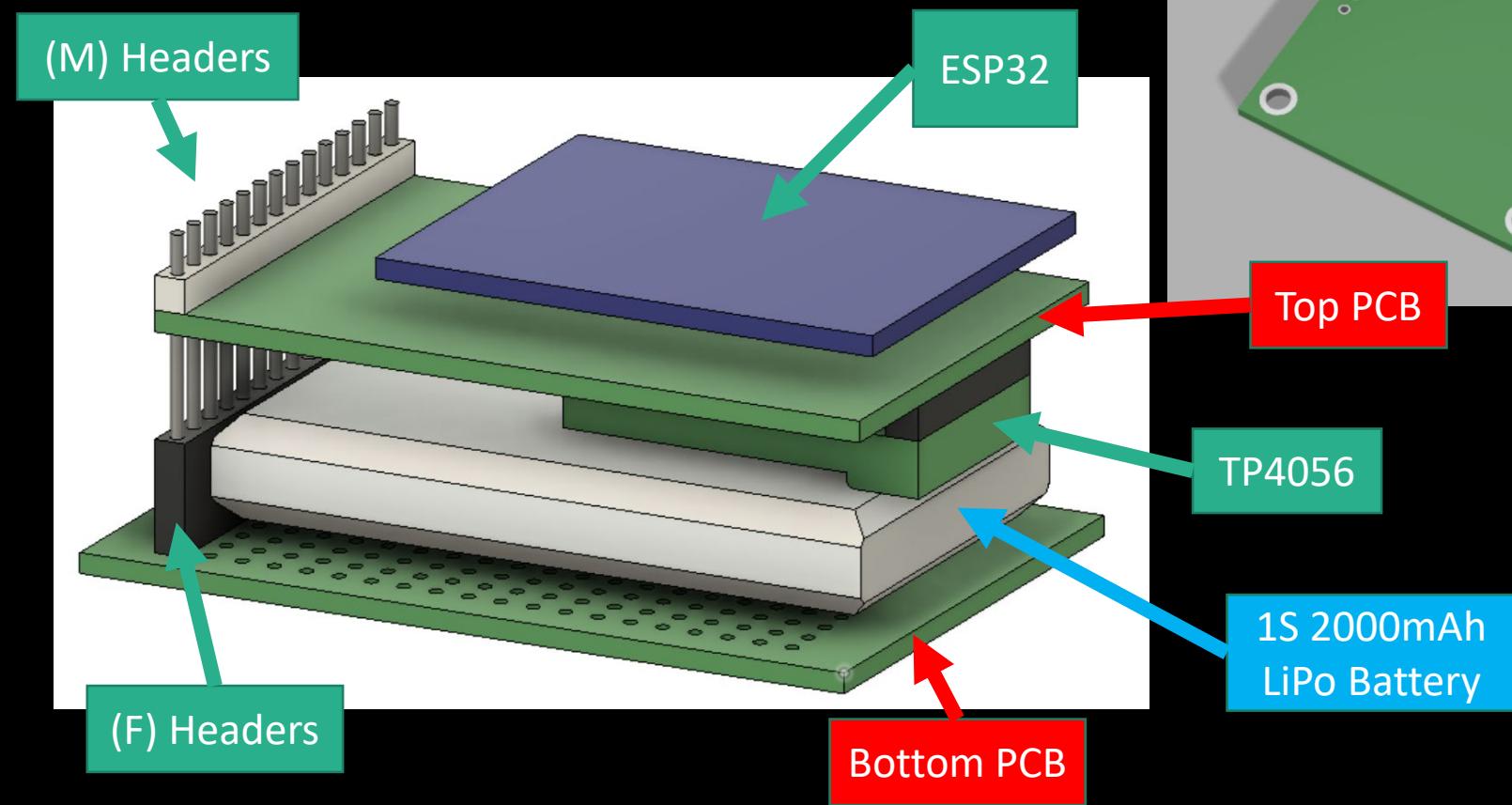
Development – Device Proof of Concept Results

- Sensor Measurements facilitated by processor
- Packet transmitted in 915 MHz ISM Band with Unencrypted Payload
- LoRa Point-to-Point Configuration with single Receiver
- Did not use LoRaWAN



Development – Device Assembly Concept

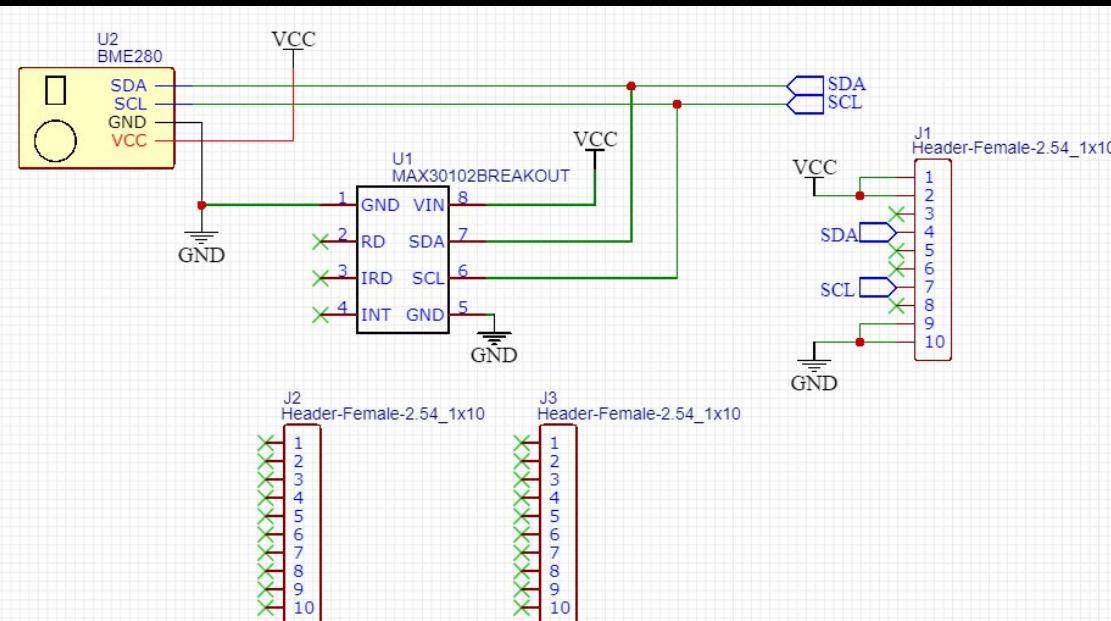
- Rudimentary Board Planning
 - Determined Board Final Dimensions (40mm x 60mm)
 - Two Stacked PCBs joined by 2.54mm pitch M/F Headers



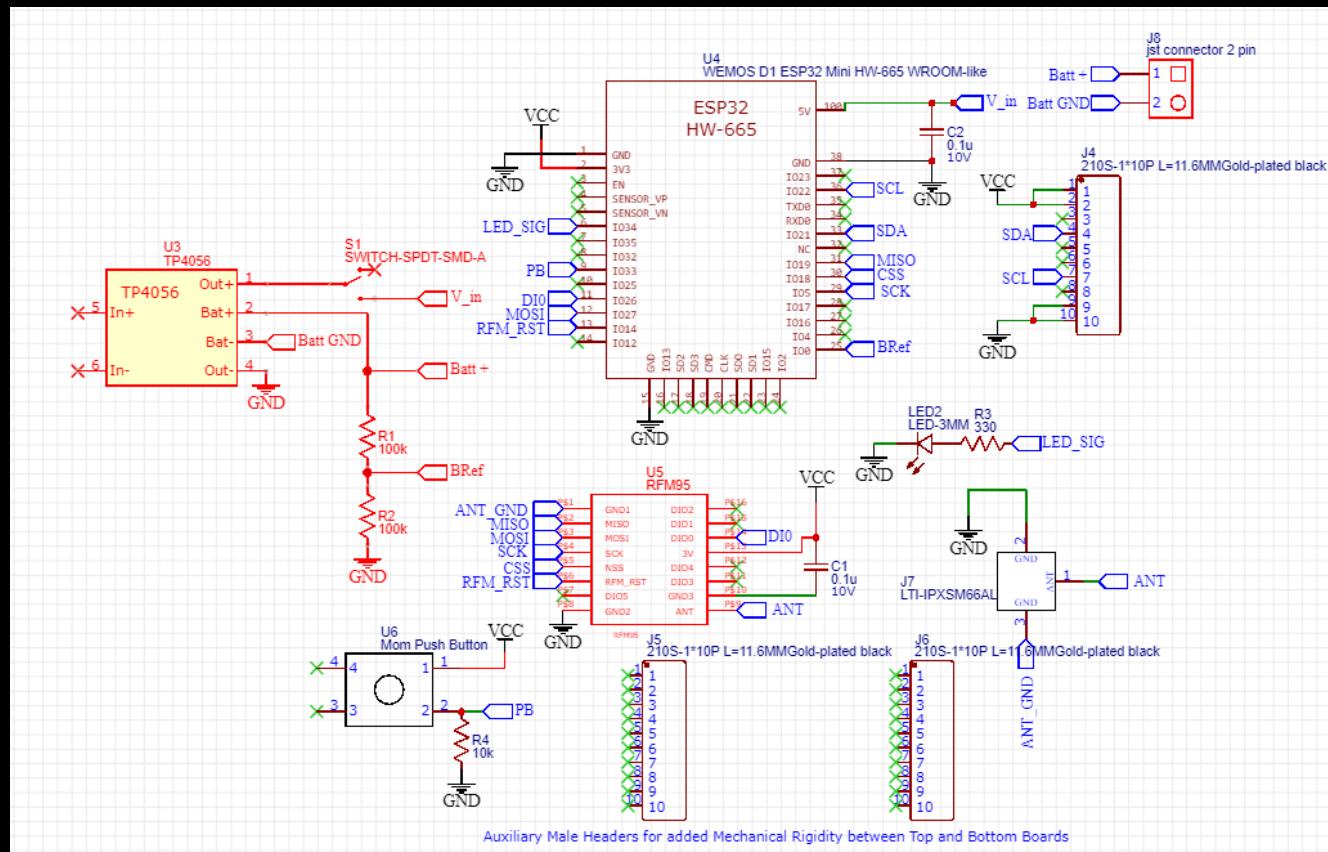
Development - Schematic Design

- Schematic made with EasyEDA

Bottom PCB



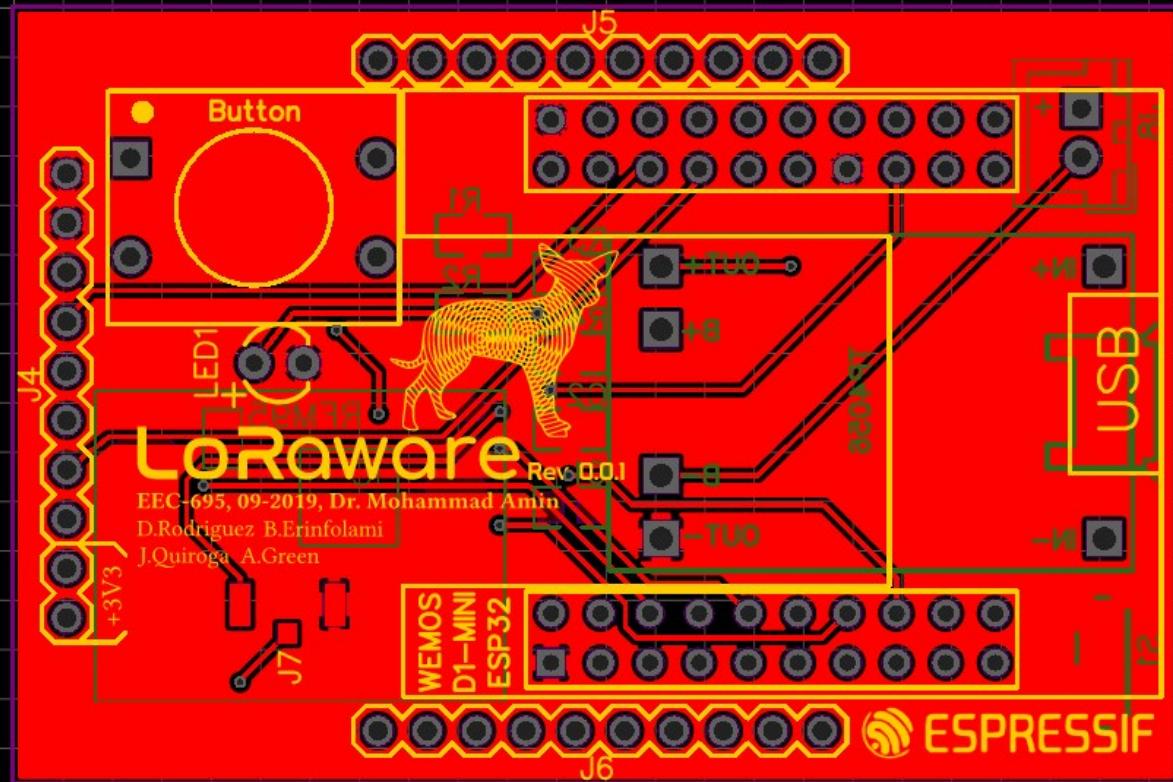
Auxiliary Female Headers for added Mechanical Rigidity between Top and Bottom Boards



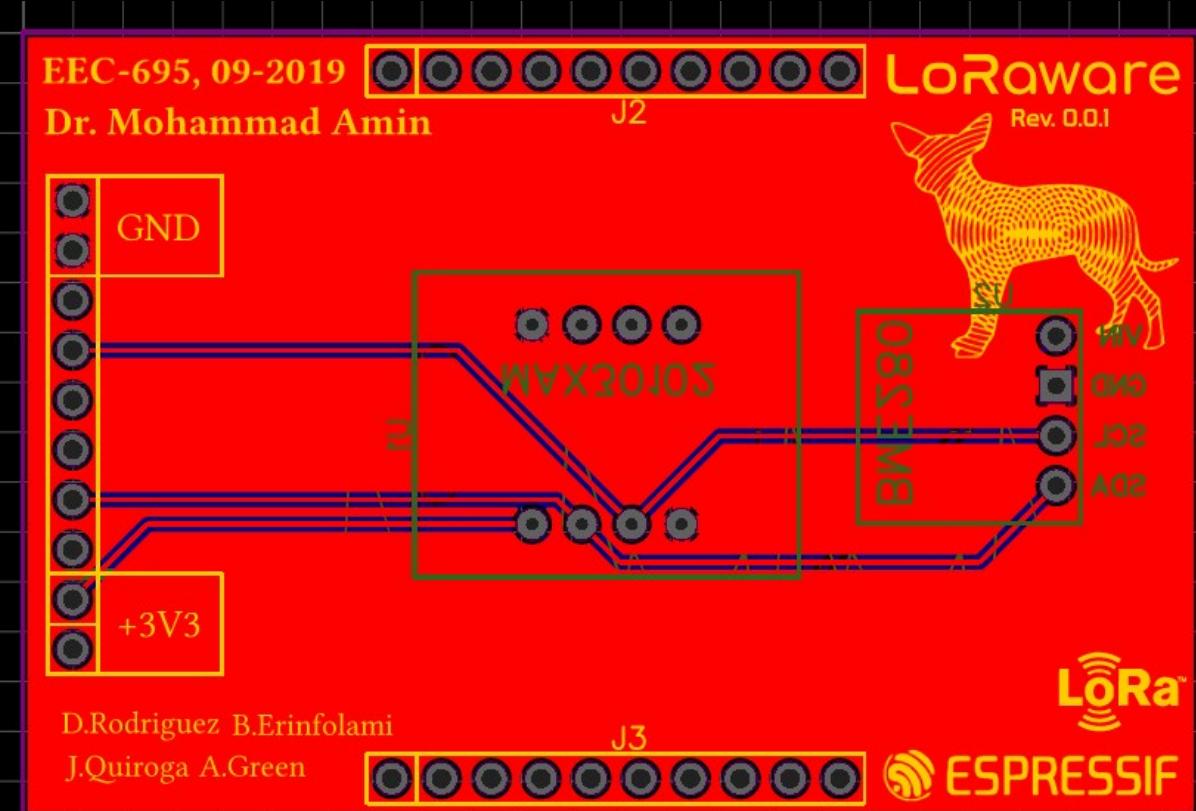
Top PCB

Development – PCB Pre-Fab Design (EasyEDA)

Top PCB



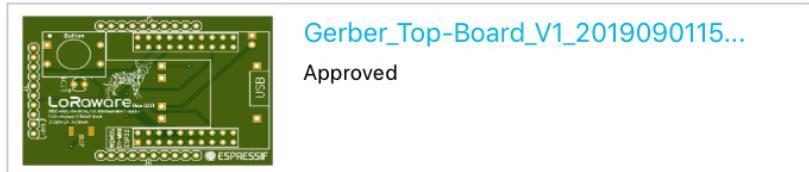
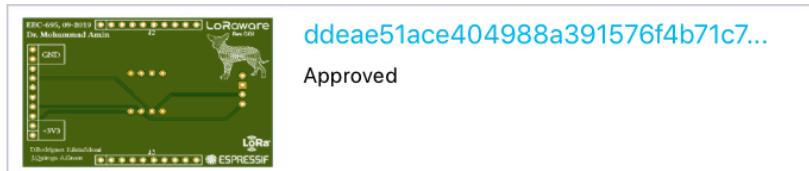
Bottom PCB



Development – PCB Fabrication (JLC PCB)

Thanks for placing your order on [JLCPCB.COM](https://www.jlcpcb.com).

Your order has been approved and put into production.
You can review your order status at any time by logging
into your JLCPCB Account and viewing your order history.

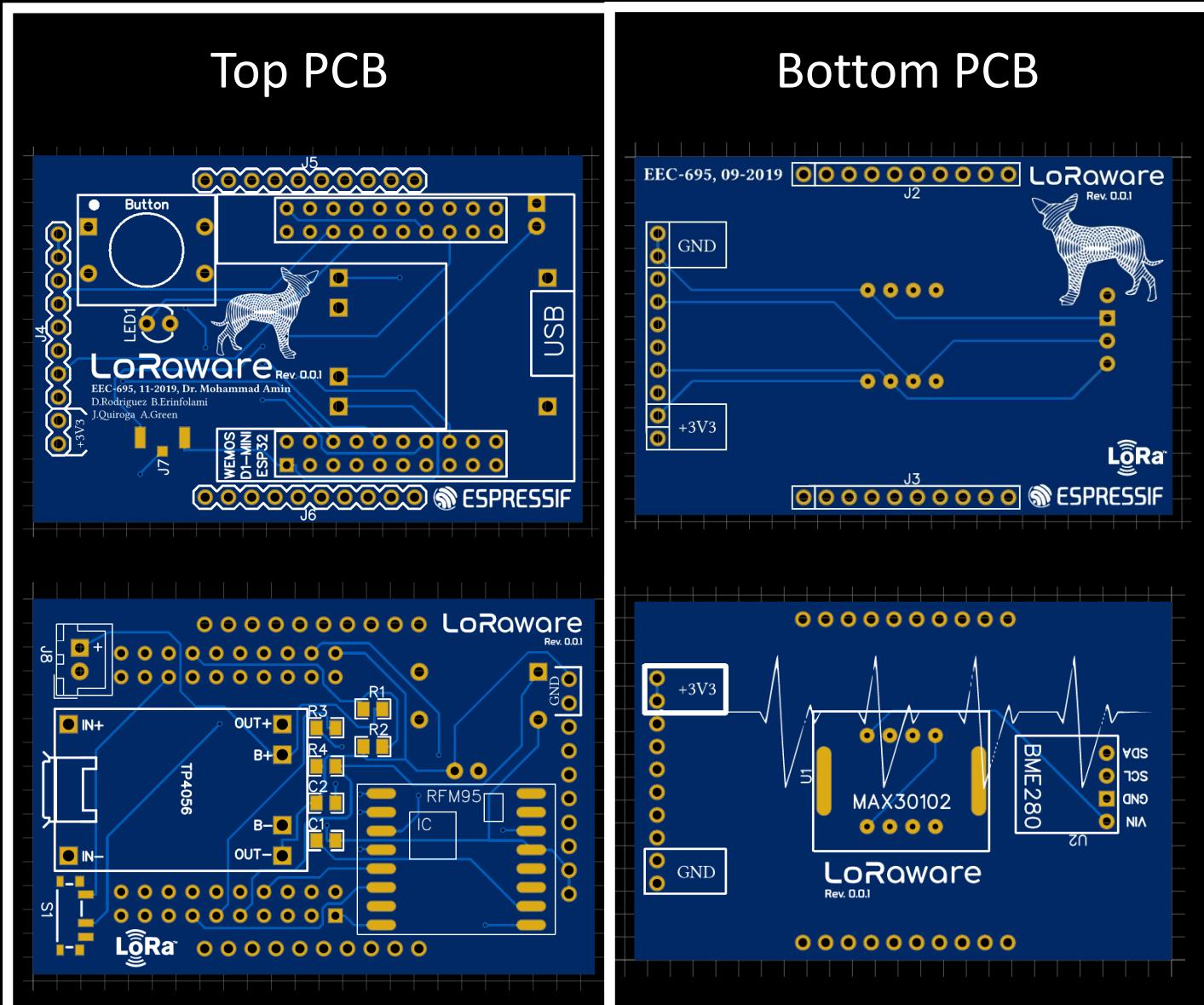


[View order details>>](#)

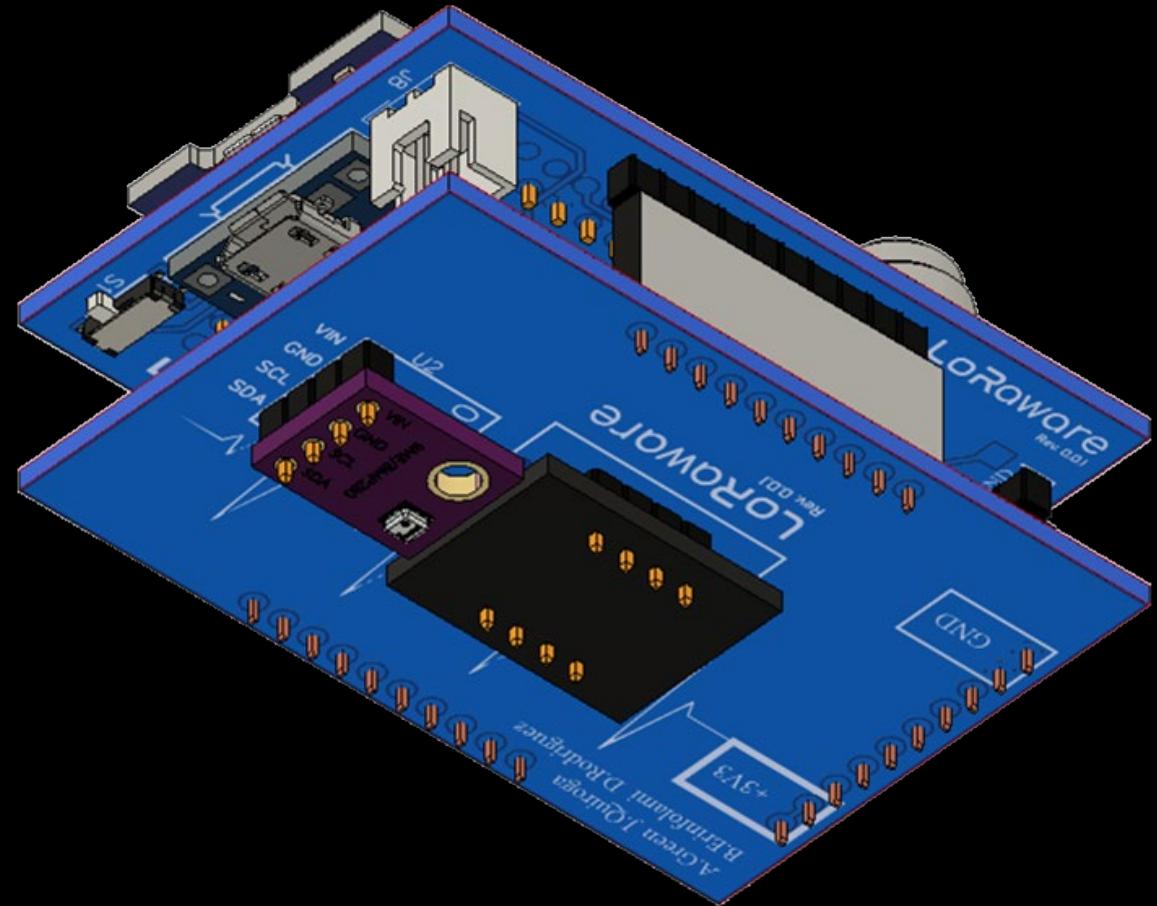
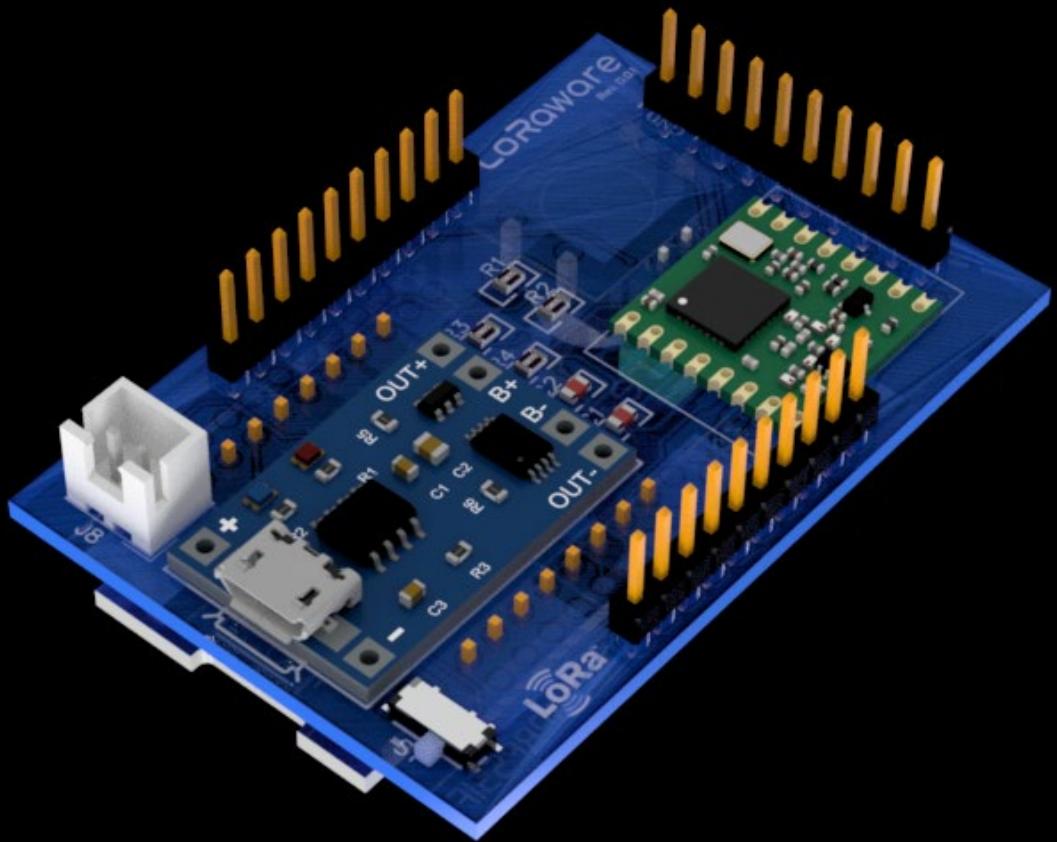
This is an automatically generated email – please do not reply to it. If you have any queries regarding your order please email support@jlcpcb.com.

JiaLiChuang (HongKong) Co., Limited

w: [JLCPCB.COM](https://www.jlcpcb.com)



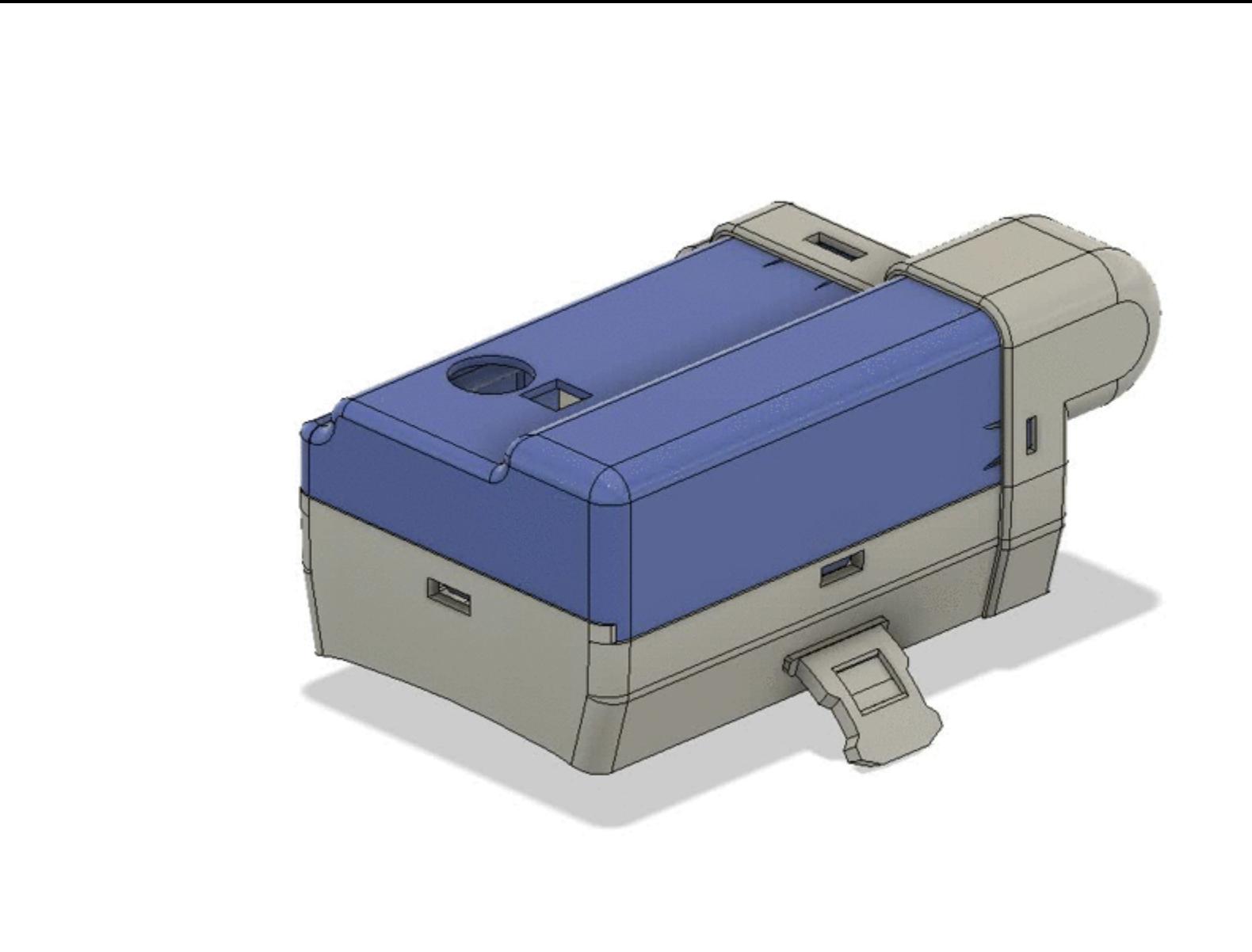
Development – Component Assembly



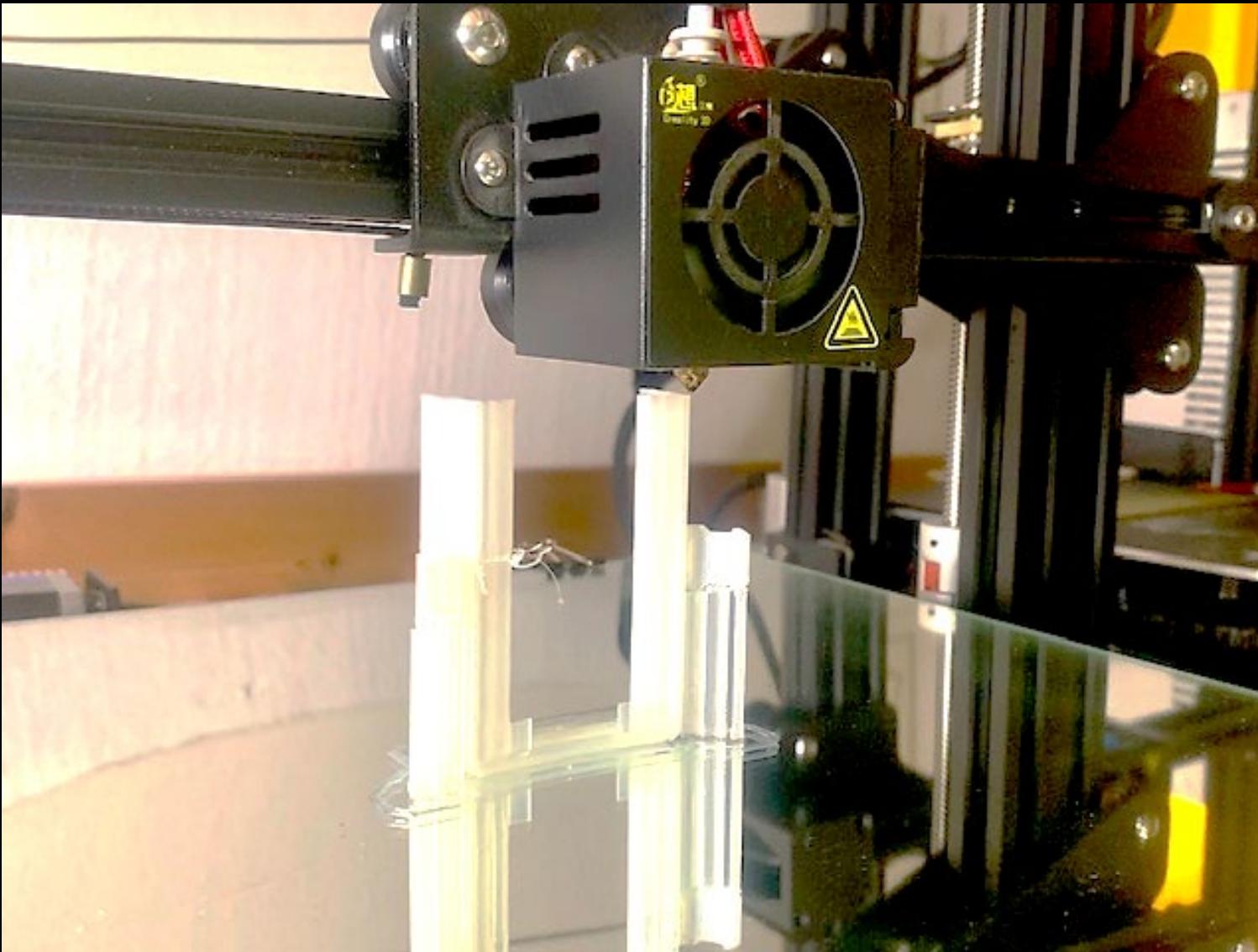
Development – Board Assembly Day



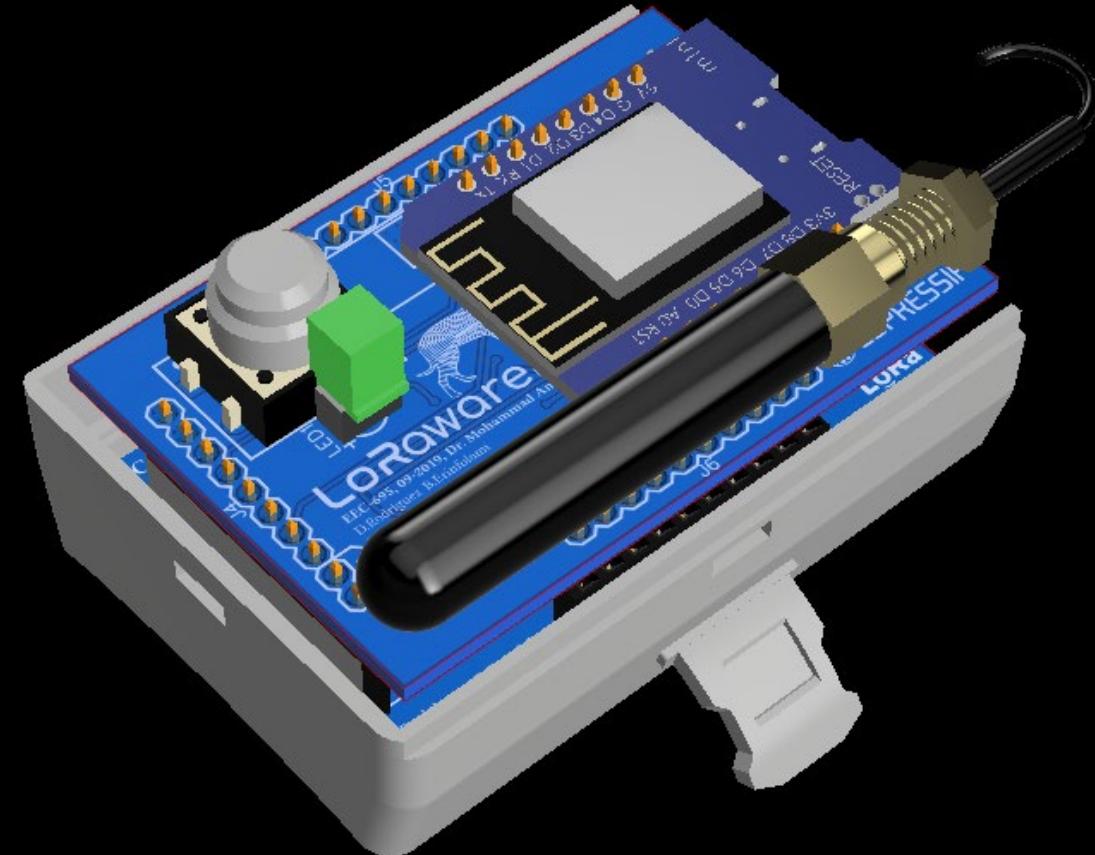
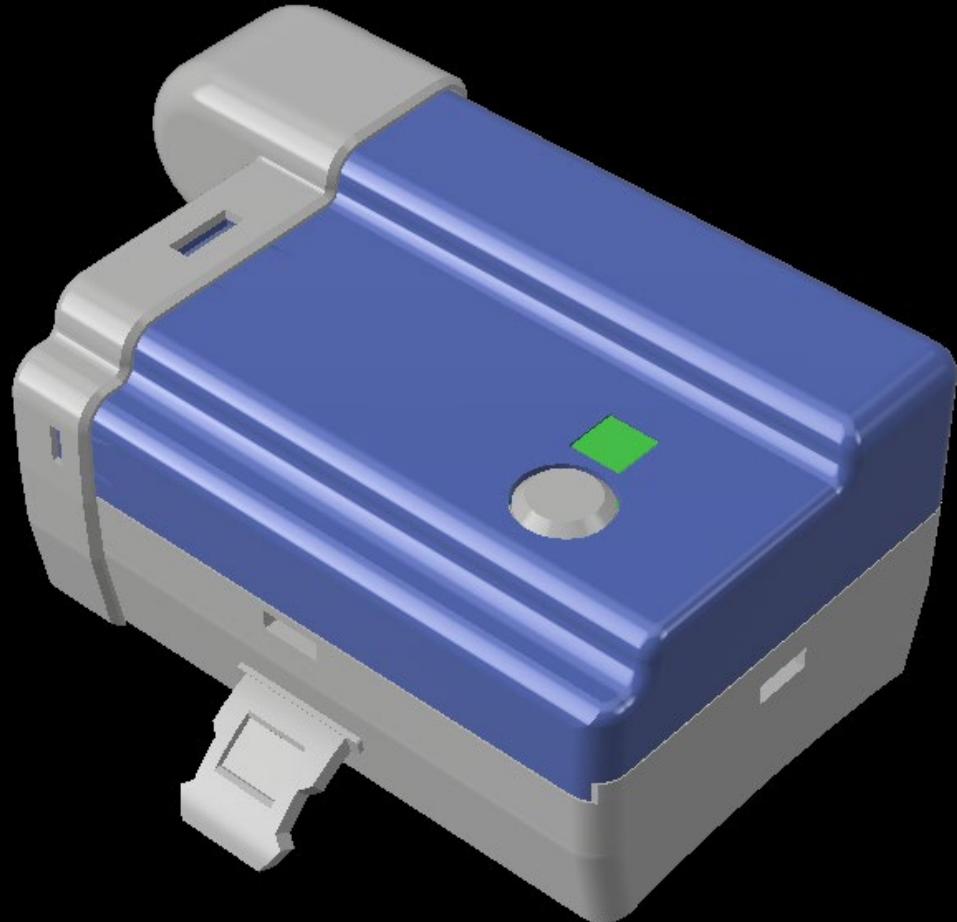
Development - Enclosure CAD Design



Development – Enclosure Manufacturing



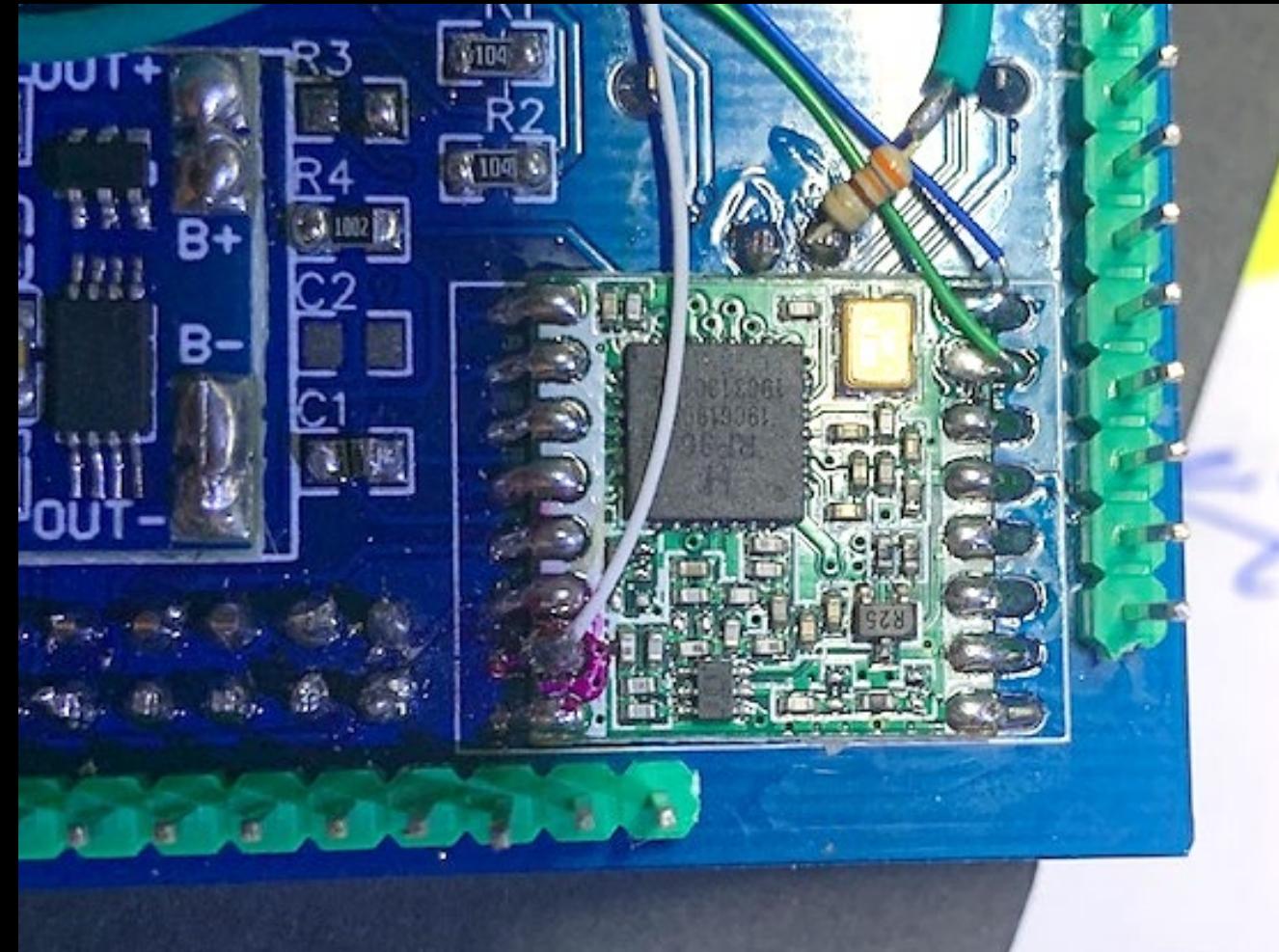
Development – Enclosure Assembly



Development – LoRaWAN Roadblocks

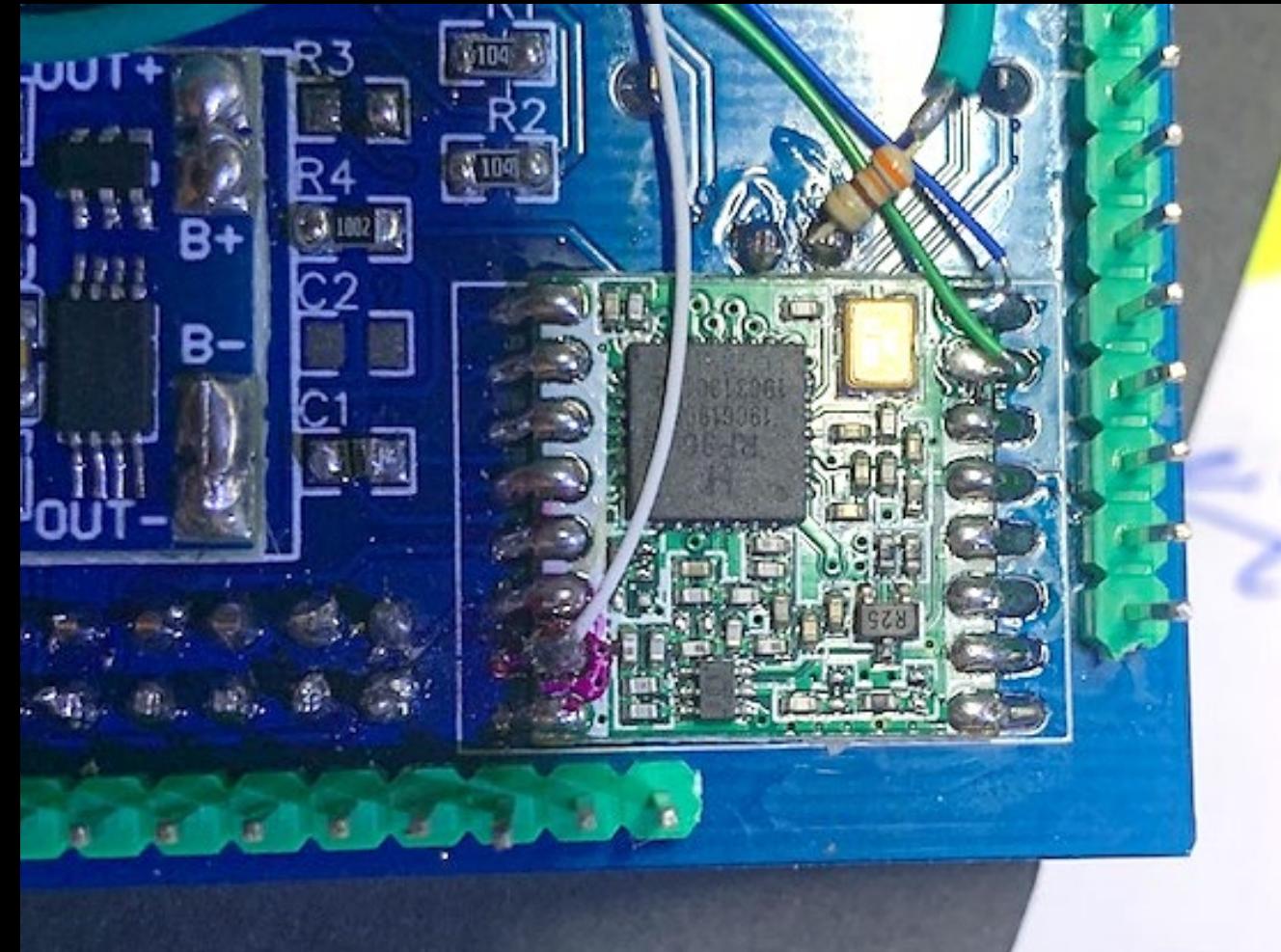
- Small Board Errors

- RFM95 required three more Interrupt Pins to enable Full LoRaWAN Capabilities
- Onboard voltage regulator dropout when Battery Voltage Applied



Development – LoRaWAN Roadblocks

- Small Board Errors
 - Resolved by cutting 5V trace on PCB and bypassing with external 30 Gauge Kynar Wires



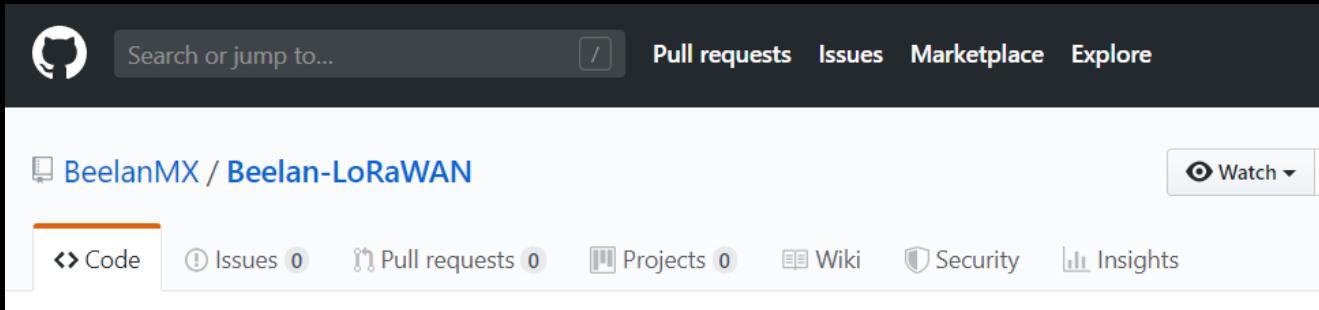
Development – LoRaWAN Roadblocks

- Arduino LoRaWAN Library for US-915 MHz was incompatible with ESP32 Architecture
 - Produced Errors when Flashing LoRaWAN Communication Firmware to LoRaware Devices

The screenshot shows a GitHub repository page for 'BeelanMX / Beelan-LoRaWAN'. The repository title is 'BeelanMX / Beelan-LoRaWAN'. Below the title, there are tabs for 'Code' (which is selected), 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Security', and 'Insights'. A description below the tabs states: 'A LoRaWAN library for compatible arduino board <https://www.beelan.mx>'. Below the description are several tags: 'lora', 'lorawan', 'arduino', and 'lorawan-library'. At the bottom of the main content area, there are statistics: '102 commits', '4 branches', '3 releases', and '7 contributors'. Below these statistics, there are buttons for 'Branch: master ▾', 'New pull request', 'Create new file', and 'Upload files'. The commit list starts with a commit from 'sabas1080' titled 'Add support ESP32 README'. The list continues with other commits related to documentation, examples, source code, tests, and file handling.

File / Action	Description
doc	Add library from https://git.antares.id/lorawan-loraid/arduino-loraid
examples	update travis
src	Added #include "Arduino.h"
test	modify examples
.gitignore	add setChannel
.travis.yml	Update .travis.yml
LICENSE.txt	Fix autor beelan
README.md	Add support ESP32 README
library.properties	Added esp32 to library.properties

Development – LoRaWAN Roadblocks



Commits on Oct 9, 2019

Added #include "Arduino.h" ...

 DavidRodrii committed 17 days ago

Verified

Added esp32 to library.properties

 DavidRodrii committed 17 days ago

Verified

└── SRC	Added #include "Arduino.h"
└── test	modify examples
└── .gitignore	add setChannel
└── .travis.yml	Update .travis.yml
└── LICENSE.txt	Fix autor beelan
└── README.md	Add support ESP32 README
└── library.properties	Added esp32 to library.properties

Development – LoRaWAN Roadblocks

The screenshot shows a GitHub repository page for [BeelanMX / Beelan-LoRaWAN](#). The repository has 8 stars. The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore.

The repository has 0 issues, 0 pull requests, 0 projects, and 0 wiki pages. The security and insights sections are also present.

The current branch is master. A dropdown menu shows "Branch: master".

The commit history for Oct 14, 2019, includes:

- Add support ESP32 README by [sabas1080](#) committed 12 days ago. This commit is verified.
- Merge pull request #22 from [DavidRodrii/master](#) by [sabas1080](#) committed 12 days ago. This commit is highlighted with a red box and is also verified.

The file changes listed at the bottom are:

- LICENSE.txt: Fix autor beelan
- README.md: Add support ESP32 README
- library.properties: Added esp32 to library.properties

Development – LoRaWAN Roadblocks

- Resolved after editing source files with incremental changes and monitoring C++ Compilation Errors
- Our edits were pushed to the Library's Github Repository in the form of a Pull Request
- Our changes received approval from Repository Maintainers and were merged into the Master Build
- <https://github.com/BeelanMX/Beelan-LoRaWAN>

The screenshot shows the GitHub repository page for 'BeelanMX / Beelan-LoRaWAN'. The repository description is 'A LoRaWAN library for compatible arduino board' with a link to <https://www.beelan.mx>. The repository has 102 commits, 4 branches, 3 releases, and 7 contributors. A pull request is visible, showing changes made by 'sabas1080' to support ESP32. The changes include updates to README, doc, examples, src, test, .gitignore, .travis.yml, LICENSE.txt, README.md, and library.properties.

File	Description
doc	Add library from https://git.antares.id/lorawan-loraid/arduino-loraid
examples	update travis
src	Added #include "Arduino.h"
test	modify examples
.gitignore	add setChannel
.travis.yml	Update .travis.yml
LICENSE.txt	Fix autor beelan
README.md	Add support ESP32 README
library.properties	Added esp32 to library.properties

Development – Gateway Configuration

- Supports entire LoRaWAN Network
- Operates as Wi-Fi Access Point
- Gateway Default Application – Forwards LoRaWAN Packets to Published MQTT Messages within IEEE 802.11 Network



Development - LoRaWAN Gateway Management

The screenshot shows a web-based management interface for a LoRaWAN gateway. On the left is a vertical sidebar with icons for Home, Nodes, Application configuration, Application users, and Help. The main area shows the navigation path: Organizations > drodrii > Applications > LoRaware-LoRaWAN-1. A red "Delete application" button is in the top right. Below it are tabs for Nodes, Application configuration, and Application users, with Nodes selected. A "Create node" button is in the top right of the main content area. The main content displays a table of nodes:

Device name	Device EUI	Device description	Frame Logs	Activation
LoRaware_1	7615064722431272	LoRaware Wearable Device 1	View	ABP
LoRaware_2	8999333684176860	LoRaware Wearable Device 2	View	ABP
LoRaware_3	2493300901472720	LoRaware Wearable Device 3	View	ABP
LoRaware_4	4969177764010230	LoRaware Wearable Device 4	View	ABP

Development – LoRaWAN Proof of Concept

```
pi@raspberrypi: ~/MatchX
,"codeRate":"4/5"}, "fCnt":532, "fPort":1, "data": "Q291bnRlcio1MzI=")
^C
pi@raspberrypi:~/MatchX $ mosquitto sub -h usx.matchx.io -p 8883 -t "#" -v --caf
ile /home/pi/MatchX/ca.crt
ostname`-$$
application/24/node/2687614411400960/rx {"applicationID": "24", "applicationName": "LoRaWAN-Test1", "nodeName": "TBEAM-NoOLED", "devEUI": "2687614411400960", "rxInfo": [{"mac": "40d63cfffe021652", "rssi": -51, "loRaSNR": 12, "name": "MX5JS4UYZ", "latitude": 32.94986, "longitude": -117.01875, "altitude": 186}], "txInfo": {"frequency": 904100000, "dataRate": {"modulation": "LORA", "bandwidth": 125, "spreadFactor": 8}, "adr": false, "codeRate": "4/5"}, "fCnt": 534, "fPort": 1, "data": "Q291bnRlcio1MzQ=")
application/24/node/2687614411400960/rx {"applicationID": "24", "applicationName": "LoRaWAN-Test1", "nodeName": "TBEAM-NoOLED", "devEUI": "2687614411400960", "rxInfo": [{"mac": "40d63cfffe021652", "rssi": -46, "loRaSNR": 11.5, "name": "MX5JS4UYZ", "latitude": 32.94986, "longitude": -117.01875, "altitude": 186}], "txInfo": {"frequency": 905100000, "dataRate": {"modulation": "LORA", "bandwidth": 125, "spreadFactor": 8}, "adr": false, "codeRate": "4/5"}, "fCnt": 535, "fPort": 1, "data": "Q291bnRlcio1MzU=")
application/24/node/2687614411400960/rx {"applicationID": "24", "applicationName": "LoRaWAN-Test1", "nodeName": "TBEAM-NoOLED", "devEUI": "2687614411400960", "rxInfo": [{"mac": "40d63cfffe021652", "rssi": -46, "loRaSNR": 10.5, "name": "MX5JS4UYZ", "latitude": 32.94986, "longitude": -117.01875, "altitude": 186}], "txInfo": {"frequency": 903900000, "dataRate": {"modulation": "LORA", "bandwidth": 125, "spreadFactor": 8}, "adr": false, "codeRate": "4/5"}, "fCnt": 536, "fPort": 1, "data": "Q291bnRlcio1MzY=")
application/24/node/2687614411400960/rx {"applicationID": "24", "applicationName": "LoRaWAN-Test1", "nodeName": "TBEAM-NoOLED", "devEUI": "2687614411400960", "rxInfo": [{"mac": "40d63cfffe021652", "rssi": -49, "loRaSNR": 12, "name": "MX5JS4UYZ", "latitude": 32.94986, "longitude": -117.01875, "altitude": 186}], "txInfo": {"frequency": 904900000, "dataRate": {"modulation": "LORA", "bandwidth": 125, "spreadFactor": 8}, "adr": false, "codeRate": "4/5"}, "fCnt": 537, "fPort": 1, "data": "Q291bnRlcio1Mzc=")
application/24/node/2687614411400960/rx {"applicationID": "24", "applicationName": "LoRaWAN-Test1", "nodeName": "TBEAM-NoOLED", "devEUI": "2687614411400960", "rxInfo": [{"mac": "40d63cfffe021652", "rssi": -49, "loRaSNR": 12.3, "name": "MX5JS4UYZ", "latitude": 32.94986, "longitude": -117.01875, "altitude": 186}], "txInfo": {"frequency": 904700000, "dataRate": {"modulation": "LORA", "bandwidth": 125, "spreadFactor": 8}, "adr": false, "codeRate": "4/5"}, "fCnt": 538, "fPort": 1, "data": "Q291bnRlcio1Mzg=")
application/24/node/2687614411400960/rx {"applicationID": "24", "applicationName": "LoRaWAN-Test1", "nodeName": "TBEAM-NoOLED", "devEUI": "2687614411400960", "rxInfo": [{"mac": "40d63cfffe021652", "rssi": -47, "loRaSNR": 10.3, "name": "MX5JS4UYZ", "latitude": 32.94986, "longitude": -117.01875, "altitude": 186}], "txInfo": {"frequency": 904900000, "dataRate": {"modulation": "LORA", "bandwidth": 125, "spreadFactor": 8}, "adr": false, "codeRate": "4/5"}, "fCnt": 539, "fPort": 1, "data": "Q291bnRlcio1Mzk=")
application/24/node/2687614411400960/rx {"applicationID": "24", "applicationName": "LoRaWAN-Test1", "nodeName": "TBEAM-NoOLED", "devEUI": "2687614411400960", "rxInfo": [{"mac": "40d63cfffe021652", "rssi": -49, "loRaSNR": 12, "name": "MX5JS4UYZ", "latitude": 32.94986, "longitude": -117.01875, "altitude": 186}], "txInfo": {"frequency": 905100000,
```



debug

msg.payload : Object

```
▶ { applicationID: "24",
  applicationName: "LoRaWAN-Test1",
  nodeName: "TBEAM-NoOLED", devEUI:
  "2687614411400960", rxInfo: array[1]
  ... }
```

9/3/2019, 8:45:10 PM node: be860696.207518

application/24/node/2687614411400960/rx :

msg.payload : Object

```
▶ { applicationID: "24",
  applicationName: "LoRaWAN-Test1",
  nodeName: "TBEAM-NoOLED", devEUI:
  "2687614411400960", rxInfo: array[1]
  ... }
```

9/3/2019, 8:45:20 PM node: be860696.207518

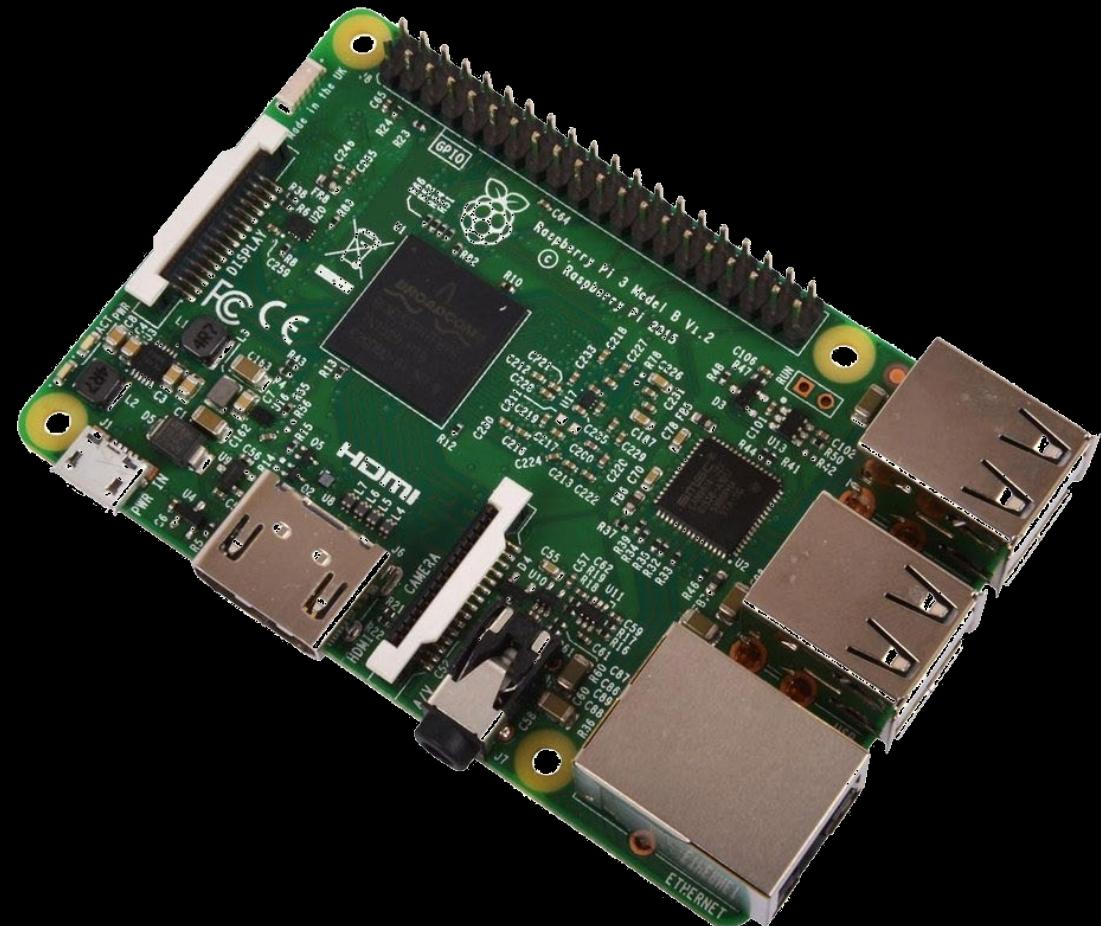
application/24/node/2687614411400960/rx :

msg.payload : Object

```
▶ { applicationID: "24",
  applicationName: "LoRaWAN-Test1",
  nodeName: "TBEAM-NoOLED", devEUI:
  "2687614411400960", rxInfo: array[1]
  ... }
```

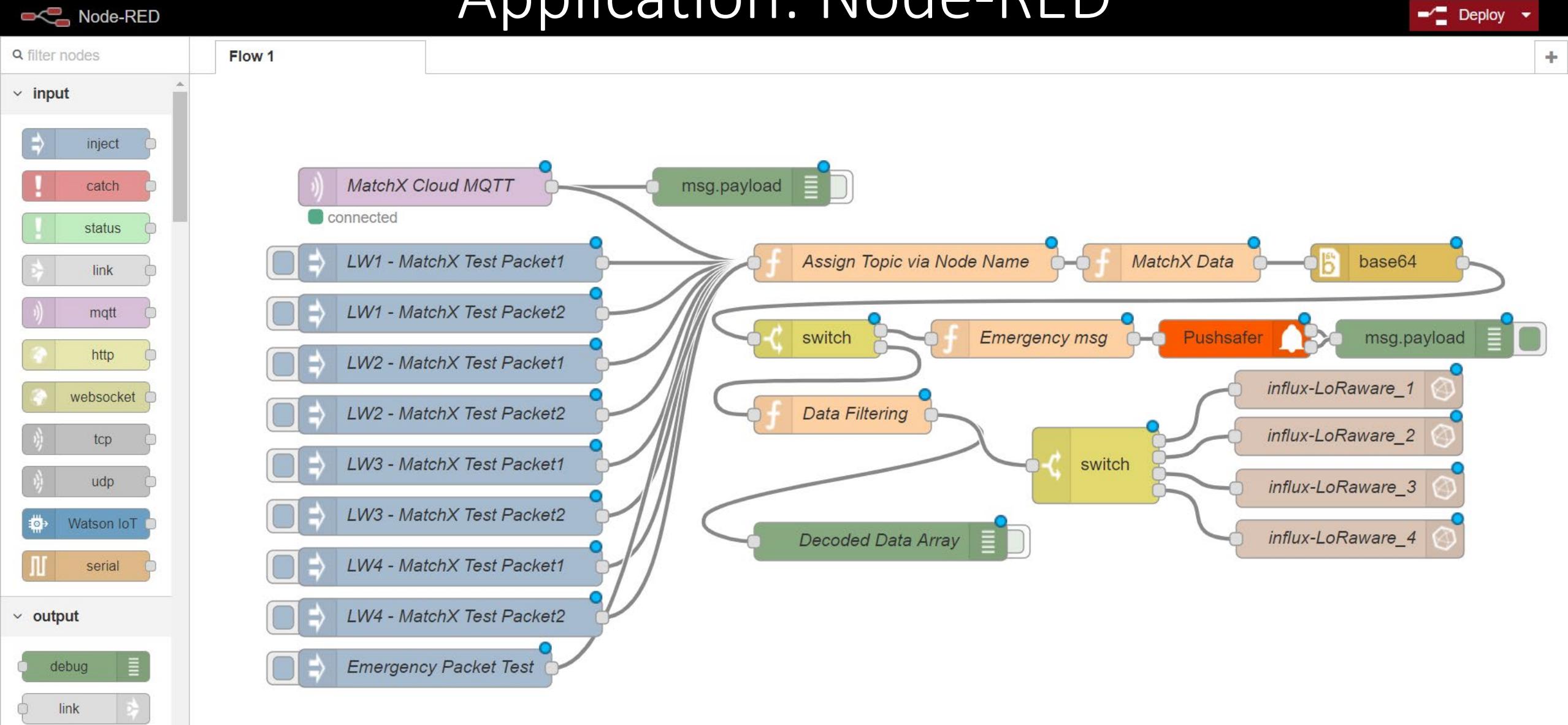
Development – Server & Applications

- Connected to Gateway's Independent Wi-Fi Network
- Subscribed to Gateway's MQTT Messages
- Hosts Node-Red for Data Parsing, conversion, and Database Queries
- Hosts InfluxDB Database
- Hosts Grafana Dashboard



Development – Server-Side Data Parsing

Application: Node-RED



Development – Server-Side Data Parsing

Application: Node-RED

The screenshot shows a Node-RED flow titled "Flow 1". On the left, the main canvas displays a "Data Filtering" function node (orange) with multiple input and output ports. An arrow points from this node to the right, where a detailed view of the function's code is shown in a modal window titled "Edit function node".

Properties:

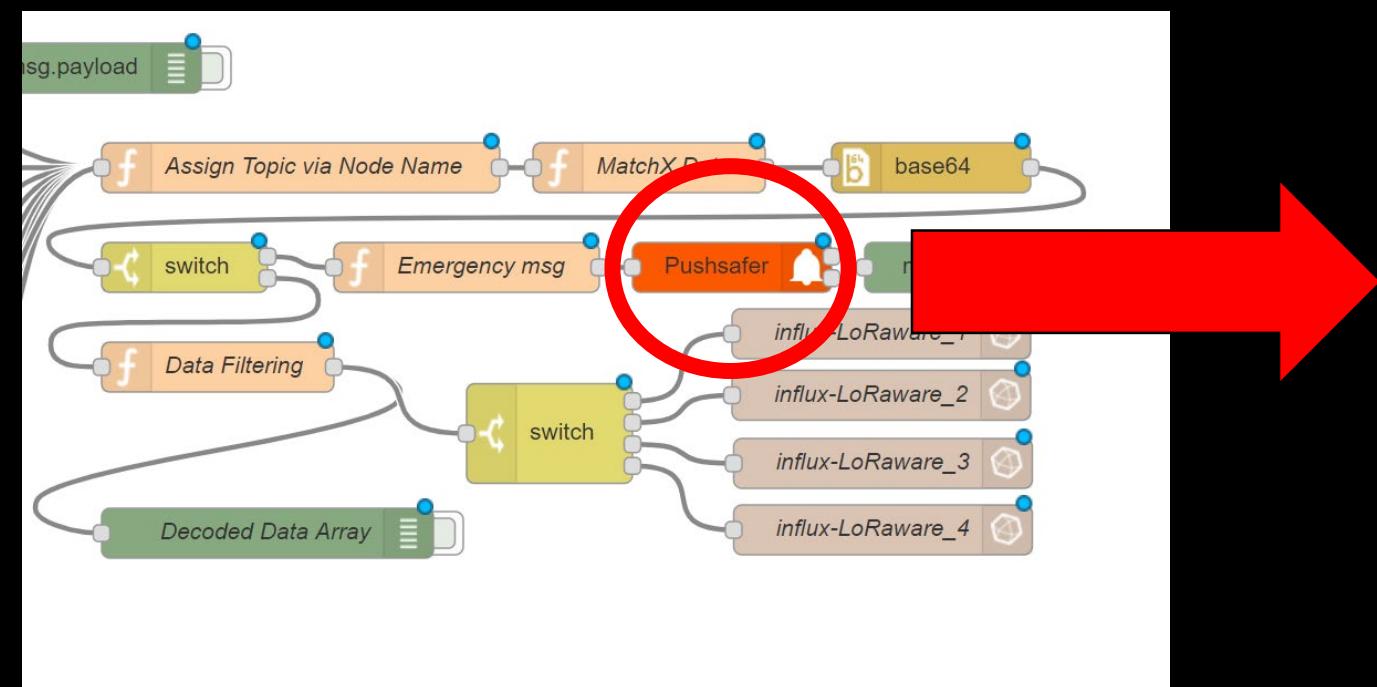
- Name: Data Filtering

Function:

```
1 // Converts Decoded String into Array of Values
2 // Then converts Array into ordered JSON Object
3 // Note: Conditionals omit any inconclusive HR or O2 readings
4
5 // to be sent to InfluxDB with Measurement = Node Name
6
7 var dataArr = msg.payload.dataEnc.split(',');
8 var SNR = msg.payload.SNR;
9
10 // var BattPercent = (parseFloat(dataArr[0])*0.0055)+3.5952;
11
12 if ((dataArr[3] == "-999.00") && (dataArr[4] == "-999.00")){
13   msg.payload = {
14     //numValue: 123.0,
15     //strValue: "message",
16     //randomValue: Math.random()*10
17     BattVolt: parseFloat(dataArr[0]),
18     BattPercent: Math.abs((parseFloat(dataArr[0])-3.5952)/0.0055),
19     Temp: parseFloat(dataArr[1]),
20     Hum: parseFloat(dataArr[2]),
21     //HR: parseFloat(dataArr[3]),
22     //O2: parseFloat(dataArr[4]),
23     SNR: SNR
24   }
25 }
26
27 else if (dataArr[3] == "-999.00"){
28 }
```

Development – Server-Side Notifications/Alerts

Application: Pushsafer



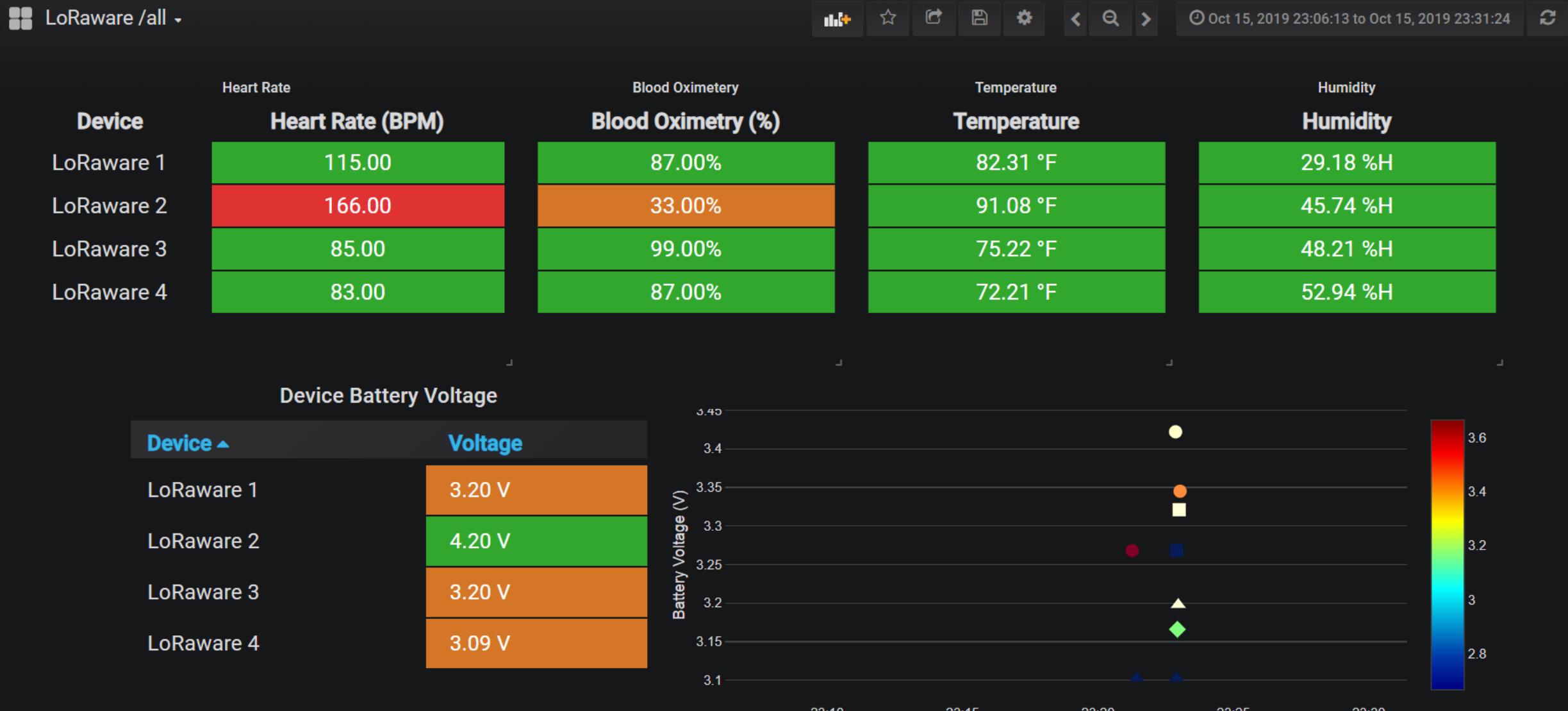
The screenshot shows the Pushsafer application interface. The top navigation bar includes links for "DASHBOARD", "Private Key", "Registered Devices", "Device Groups", "Send a Notification", "E-mail Alias", "ACCOUNT PROFILE", "API-LOG", "CLIENTS / APPS", "API", "EXAMPLES & PLUGINS", "TOOLS", "DIY", "FAQ", "SUPPORT / FORUM", "CHANGELOG", "LANGUAGE English", and "LOGOFF". The sidebar on the left lists "Your private key" (redacted), "Your Devices" (showing 25 entries), and "Search" fields. The main content area displays a table titled "Your Devices" with columns for "ID", "Name", "Status", and "Message". Two devices are listed:

ID	Name	Status	Message
19856	J-dawg (Guest)	OFF	
19855	Anwtane (Guest)	OFF	

At the bottom, there is a logo for "SOFTWARE MADE & HOSTED IN GERMANY" with the German flag.

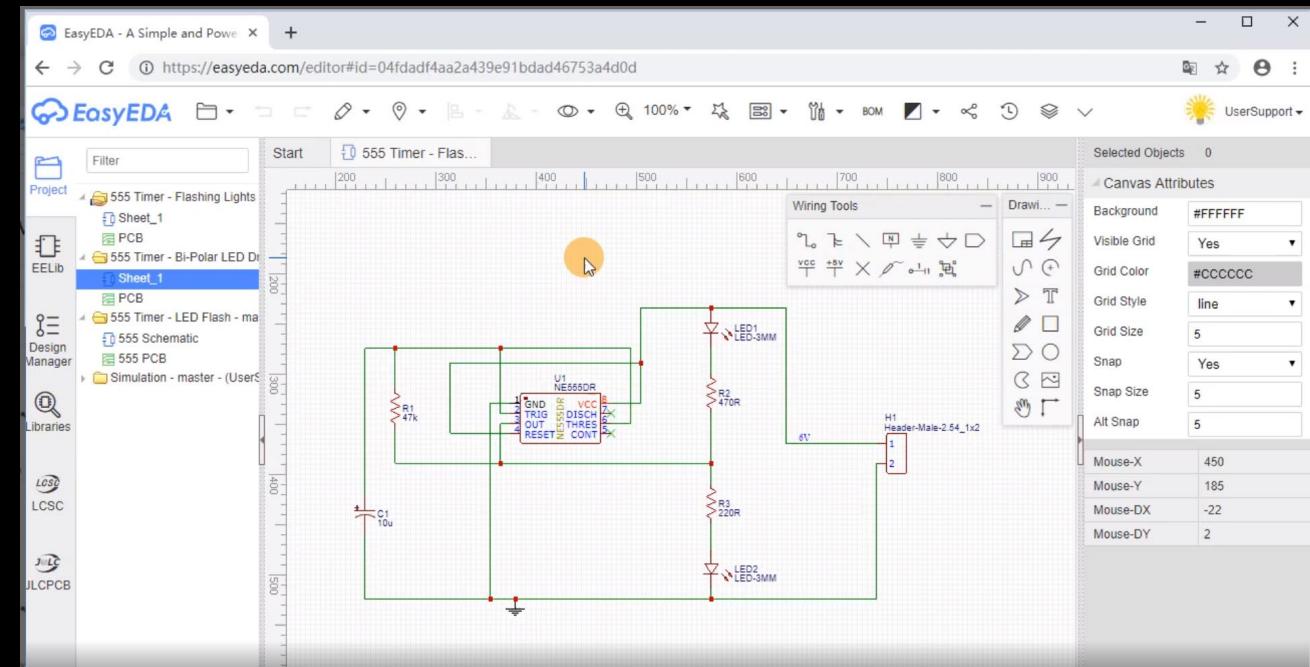
Development – Server-Side Metrics

Monitoring Application: Grafana Dashboard

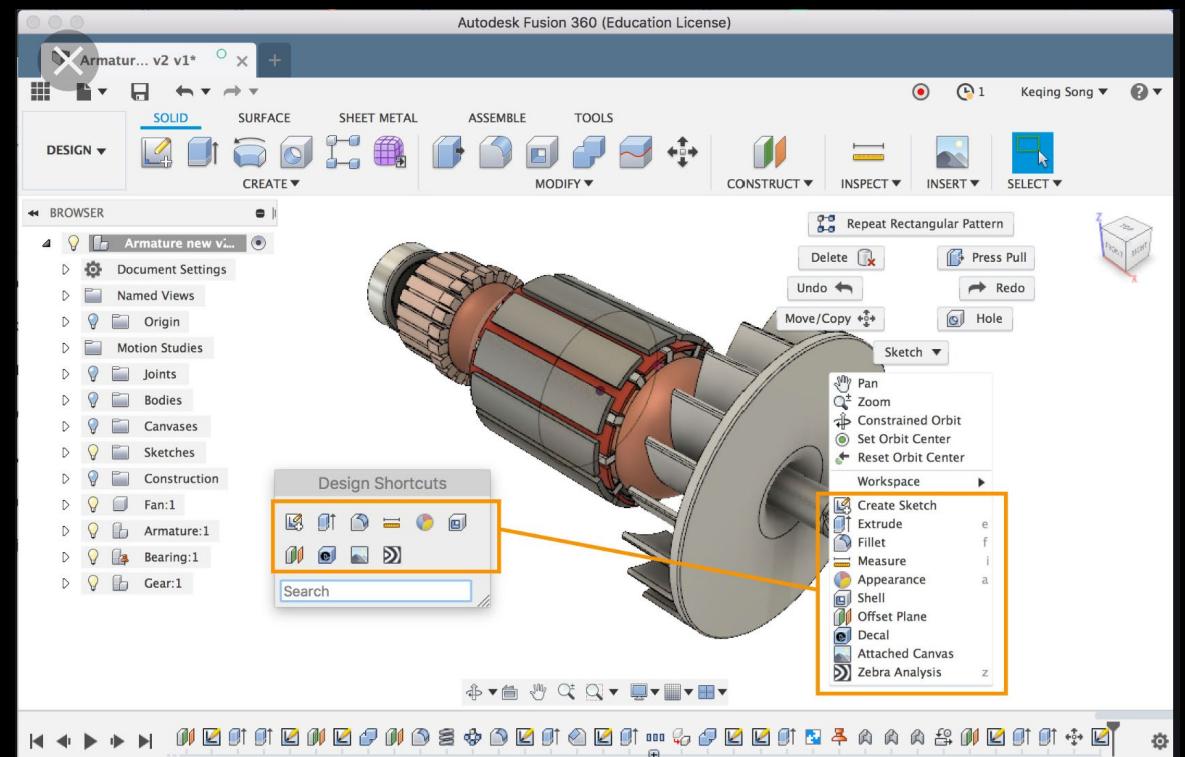


Development – Tools (Software)

- EasyEDA – Schematic & PCB Design for Manufacturing



- AutoDesk Fusion 360 – CAD Design for Board Layout Planning & 3D Printed Enclosure



Source:

https://www.google.com/search?q=fusion+360+images&rlz=1C1JZAP_enUS836US837&tbo=isch&source=iu&ictx=1&fir=_mdYx_vx_9qjiM%253A%252C6rRNQFv-1dVM%252C_-&vet=1&usg=AI4_-kSW7pLaBY_QAXRN8cKCO9IK3v9p0g&sa=X&ved=2ahUKEw9s9yk9qvkAhUKPa0KHYXWBsAQ9QEwAXoECAUQCQ#imgrc=RJdmrm7LvTjwyM&vet=1



Development – Tools (Software)

- Arduino IDE – C++ Development Environment used for ESP32 firmware

The screenshot shows the Arduino IDE interface with a code editor window titled "LW_Device1_V6 §". The code is written in C++ and performs several functions:

- It includes a function `LEDbreathe()` that uses an analog write to control an LED over 360 degrees.
- It includes a function `LoRaware_norm()` which performs the following steps:
 - Wakes up the particle sensor.
 - Reads 100 samples from the sensor.
 - Calculates heart rate and SpO2 after the first 100 samples.
 - Converts battery voltage and sensor data to strings.
 - Sends the concatenated string via LoRaWAN to a gateway.
- It handles battery voltage conversion and displays sensor data (temperature, humidity, heart rate, SpO2) on the serial monitor.

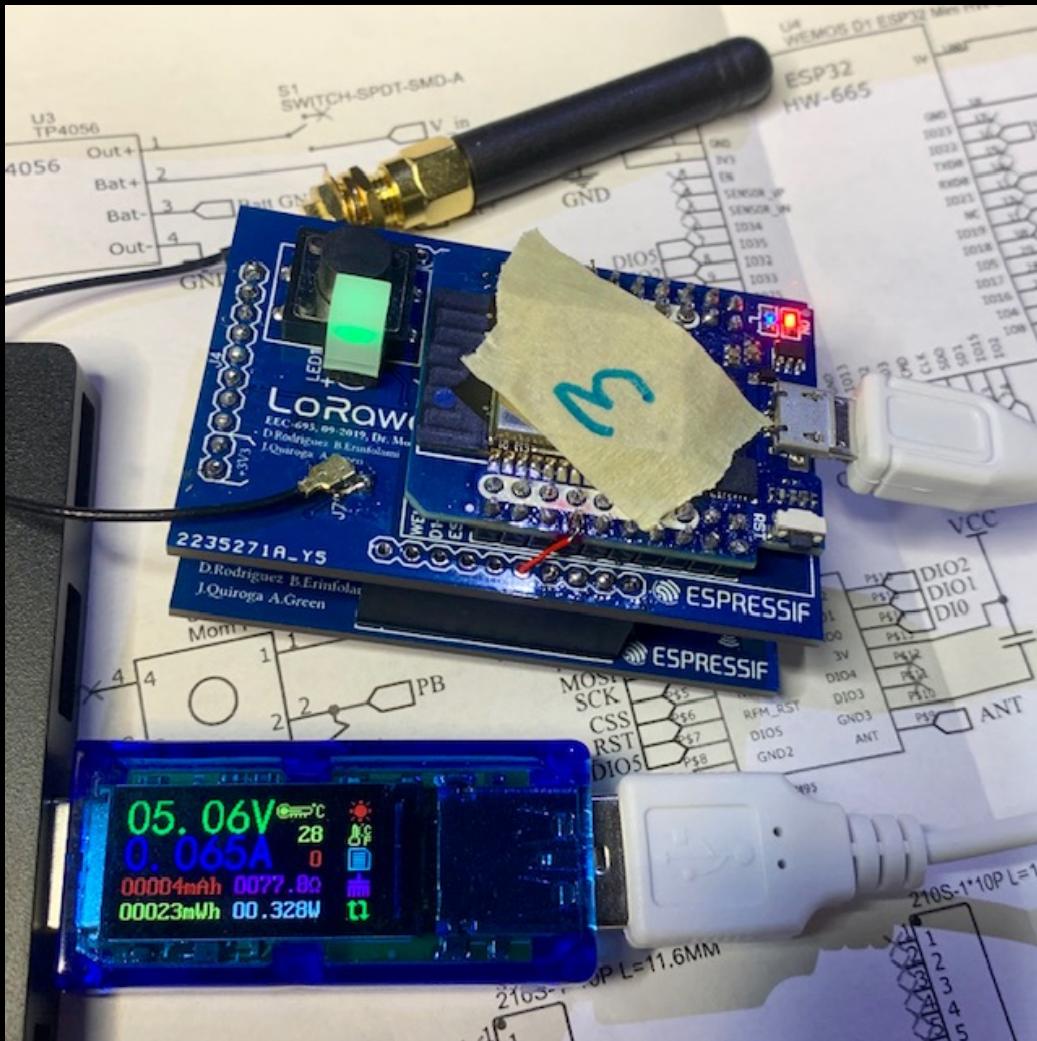
```
70
71 void LEDbreathe() {
72     for (int i = 0; i < 360; i++) {
73         analogWrite(readLED, (sin(i * 0.0174533) + 1) * (127/16));
74         delay(6);
75     }
76
77 void LoRaware_norm() {
78     LEDbreathe();
79     particleSensor.wakeUp();
80     bufferLength = 100; //buffer length of 100 stores 4 seconds of samples running at 25sps
81     //reads| the first 100 samples, and determine the signal range
82     for (byte i = 0 ; i < bufferLength ; i++)
83     {
84         while (particleSensor.available() == false) //do we have new data?
85             particleSensor.check(); //Check the sensor for new data
86
87         if (eState == true) {
88             break;
89         }
90         else {
91             redBuffer[i] = particleSensor.getRed();
92             irBuffer[i] = particleSensor.getIR();
93             particleSensor.nextSample(); //We're finished with this sample so move to next sample
94         }
95     }
96     //calculate heart rate and SpO2 after first 100 samples (first 4 seconds of samples)
97     maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2, &validSPO
98     float measuredvbat = analogRead(battPin);
99     float battV;
100    battV = (((measuredvbat)*(2.1) / (4096)))*2;
101    dtostrf(battV, 6, 3, myStrVB);
102    dtostrf((bme.readTemperature()*(1.8) + 32), 6, 3, myStrT);
103    dtostrf(bme.readHumidity(), 6, 3, myStrH);
104    dtostrf(heartRate, 6, 2, myStrHR);
105    dtostrf(spo2, 6, 2, myStrO2);
106    String str;
107    str = (String(myStrVB) +","+ String(myStrT) +"," + String(myStrH) +"," + String(myStrHR) +",
108    unsigned int str_len = 37;
109    str.toCharArray(payload,str_len);
110    Serial.println(payload);
111    // LoRaWAN Data Send Uplink to Gateway
112    lora.sendUplink(payload, strlen(payload), 0); // 0 = UNCONFIRMED Message
113    analogWrite(readLED,0);
114    particleSensor.shutDown();
115    delay(5000);
```

System Testing

- Proof-of-Concept Testing
 - Breadboard circuitry
 - Transducer Operation
 - RF Transmissions
 - Packet Management
 - Data Display
- Prototype Testing - Static and Dynamic
 - Power Usage Measurement and Battery Life Calculations
 - Comparison against 16 hour Battery Life Benchmark
 - Single (wired/wireless) and multi-node data captures (wireless)
 - Comparison against benchmark values
 - “calibrated” sensors – Fingertip Oximeter/HR, Laser Temperature Gun



Development – Power Usage



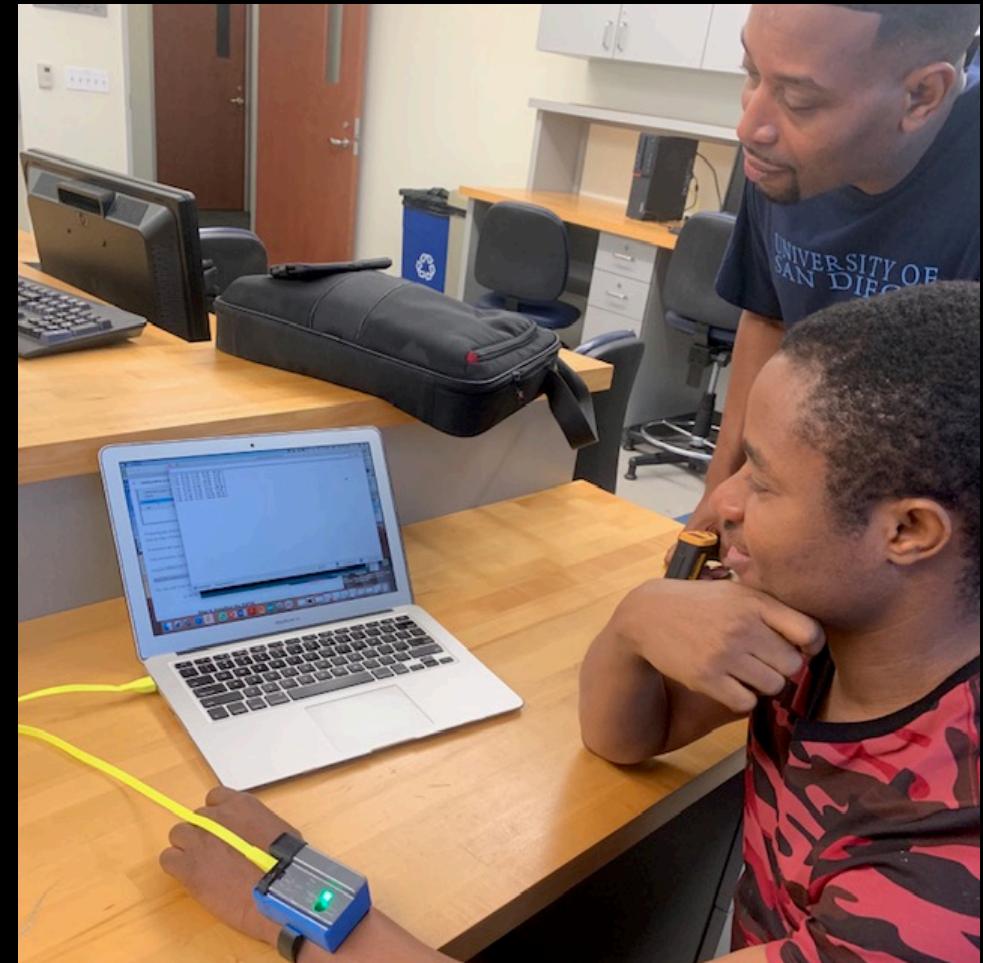
Power Usage Testing Results

Li-Po Battery Capacity (mAh):	2000
Target Battery Life (hrs.):	16

USB Power Meter 1 (Blue)				USB Power Meter 2 (Black)			
Device	Current Draw (mA)	Power (W)	Battery Life (hrs.)	Device	Current Draw (mA)	Power (W)	Battery Life (hrs.)
LoRaware_1	66.0	0.333	30.30	LoRaware_1	50.0	0.253	40.0
LoRaware_2	63.0	0.319	31.75	LoRaware_2	50.0	0.253	40.0
LoRaware_3	60.0	0.303	33.33	LoRaware_3	50.0	0.253	40.0
LoRaware_4	58.0	0.298	34.48	LoRaware_4	40.0	0.202	50.0
Average Battery Life (hrs.):		32.47		Average Battery Life (hrs.):		42.5	
Average Percent Improvement (%):		102.91		Average Percent Improvement (%):		165.6	

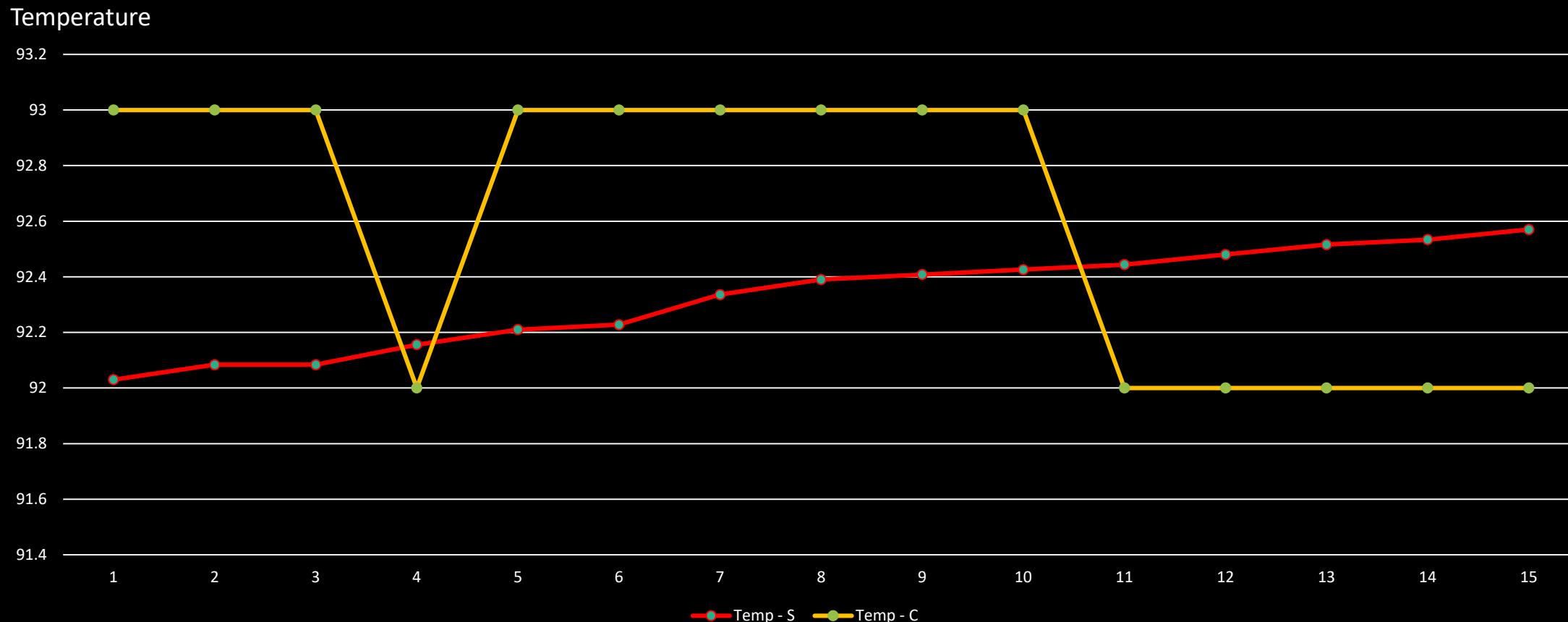
Overall Average Battery Life (hrs.)	37.48
Overall Average Percent Improvement (%)	134.27

Testing – Sensor Data Captures



Static Testing – Temperature Sensor (S) vs Calibrated (C)

Averages
Calibrated Sensor 92.6° F
Sensor 92.3° F



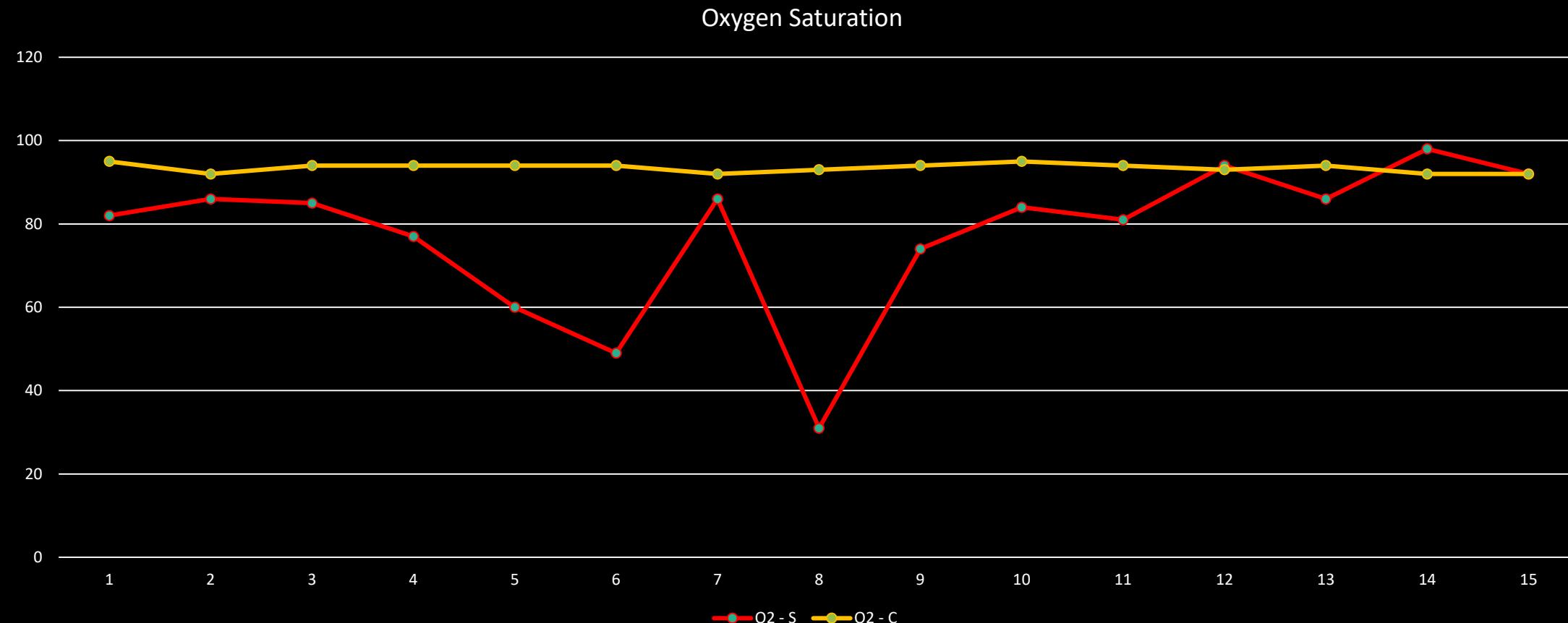
Static Testing – Heart Rate Sensor (S) vs Calibrated (C)

Averages
Calibrated 120.7 BPM
Sensor 72.7 BPM



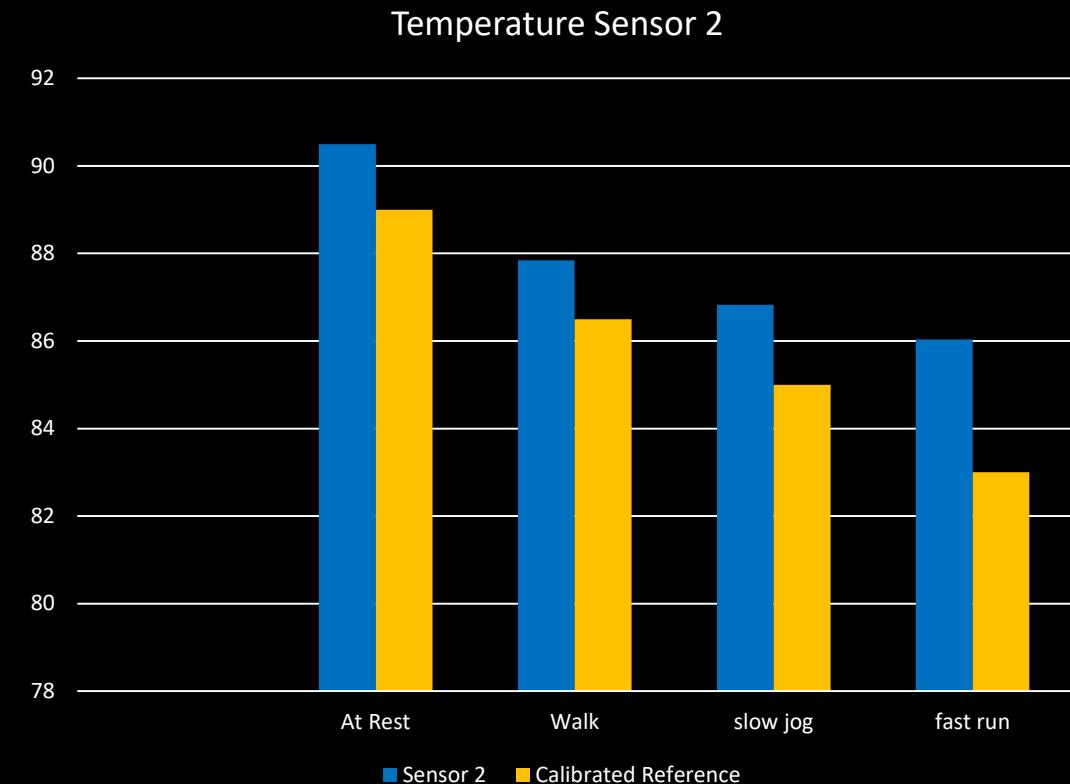
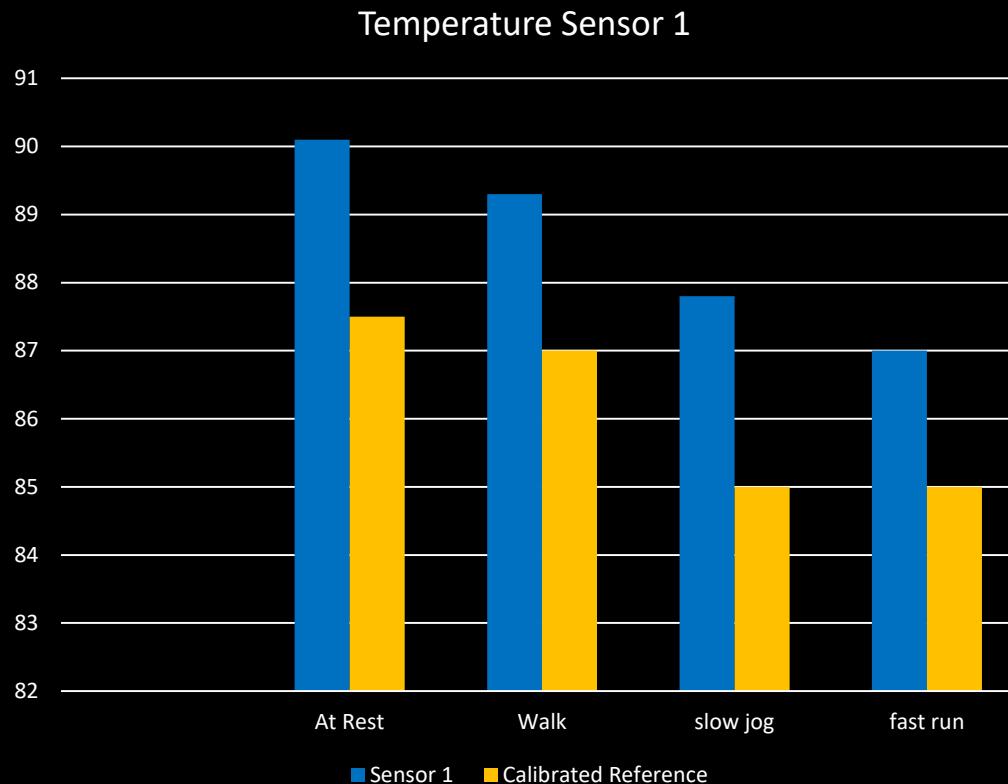
Static Testing – Oxygen Saturation Sensor (S) vs Calibrated (C)

Averages
Calibrated Sensor 93.5 %
Sensor 77.6 %



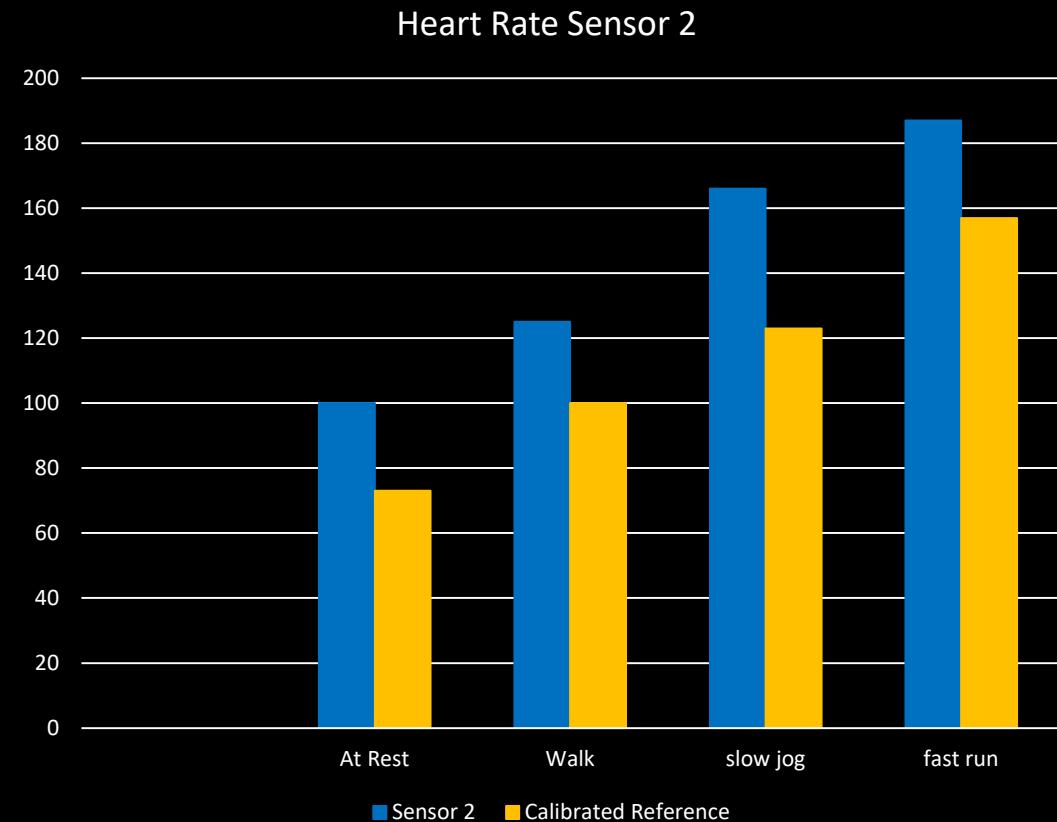
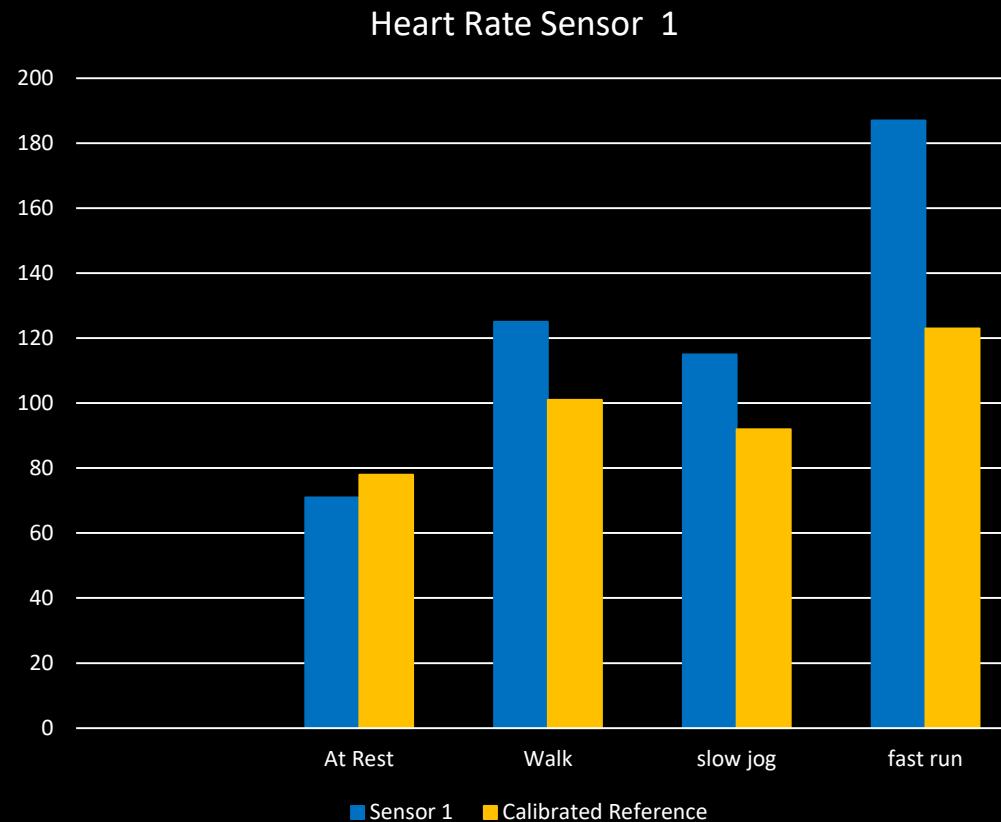
Dynamic Testing – Temperature

Average Difference
Sensor 1 2.42° F
Sensor 2 1.92° F



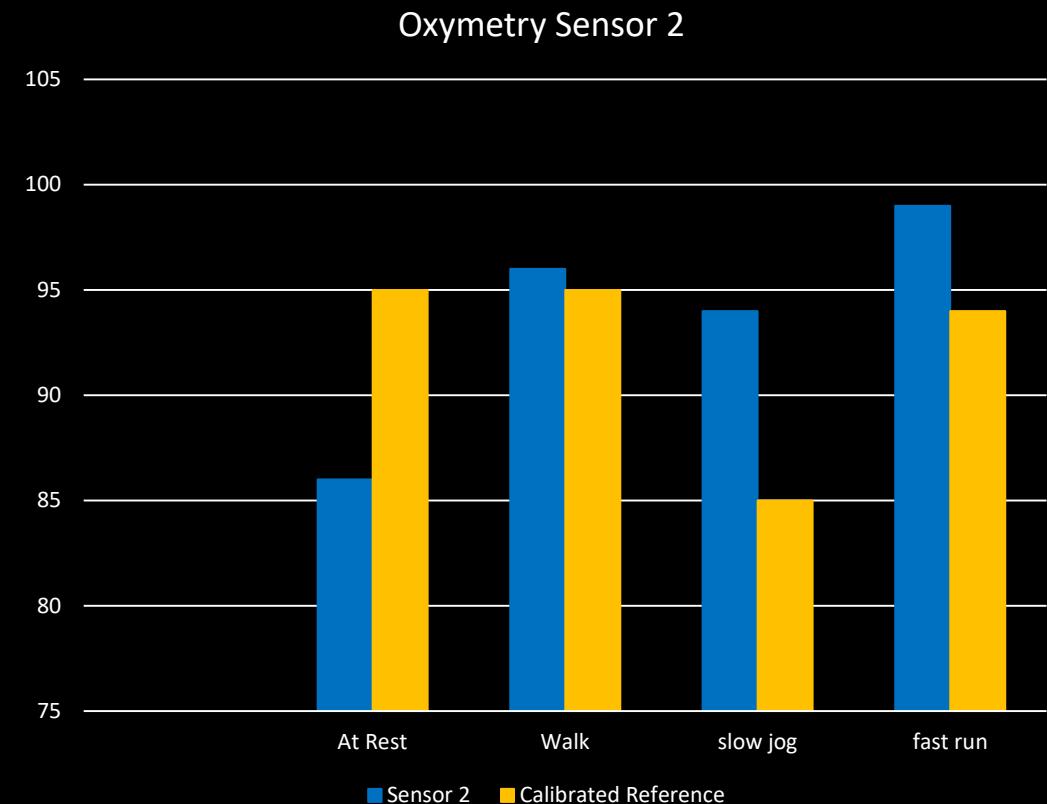
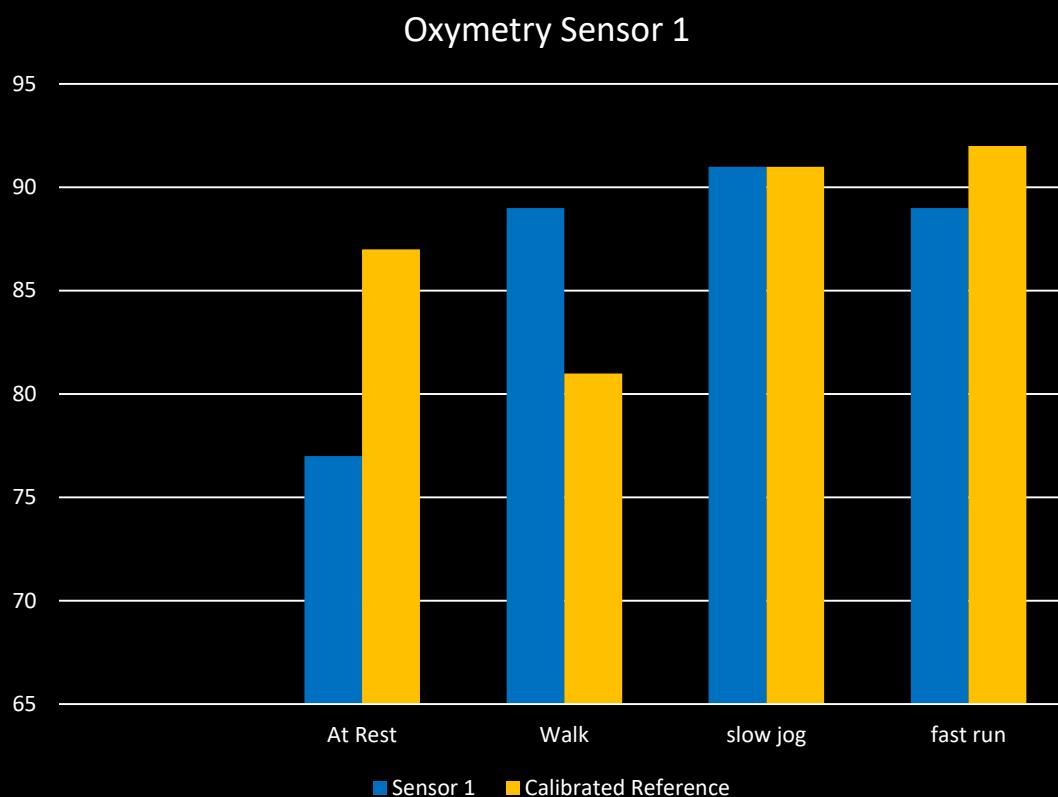
Dynamic Testing – Heart Rate

	Average Difference
Sensor 1	26 BPM
Sensor 2	31 BPM



Dynamic Testing – Oxygen Saturation

	Average Difference
Sensor 1	1.25 %
Sensor 2	1.50 %



Testing - Summary

- Additional Sensor Calibration is crucial for future development
- LoRaware Device performance in Power Usage was able to surpass our projected benchmark value

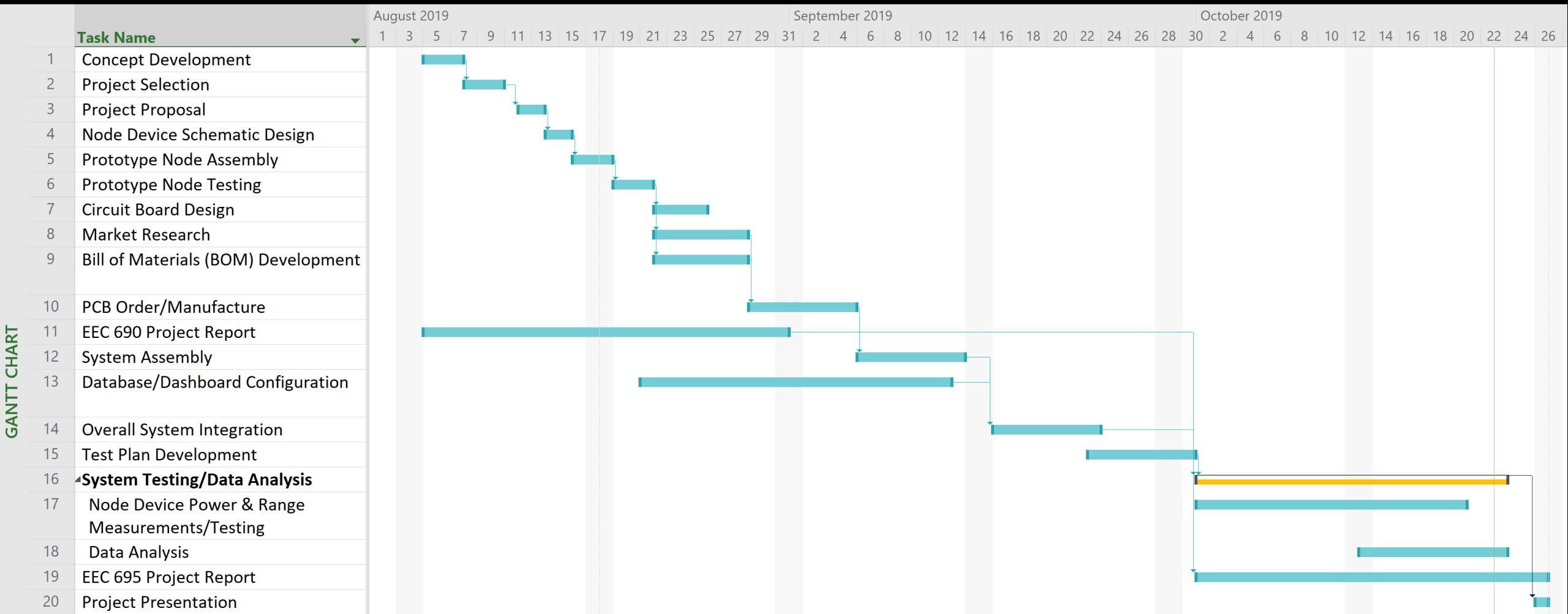


Project Management – Cost

- Handling of resources to attain intended goals:
 - Cost - Labor/Material resources
 - Schedule -Time
 - Performance – Working product

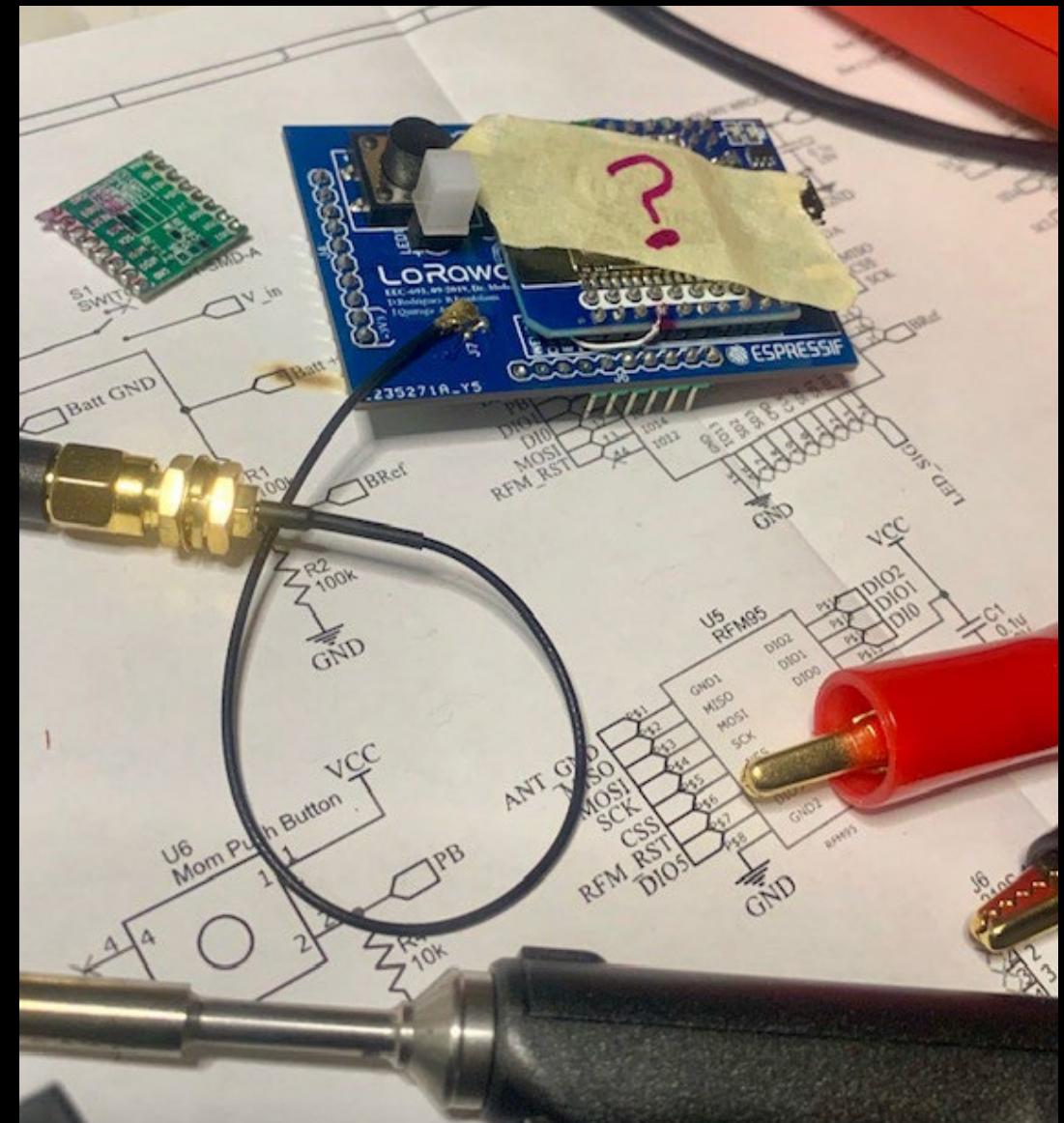
MATERIALS	Unit Cost	# Units	Total Cost
MatchX LoRa Gateway	\$ 450.00	1	\$ 450.00
RFM95 LoRa Module	\$ 20.00	4	\$ 80.00
TP4056 Li-Ion Charge Protection IC	\$ 1.20	4	\$ 4.80
2000 mAh Lithium Polymer Battery	\$ 10.00	4	\$ 40.00
ESP32 – Microcontroller	\$ 10.00	4	\$ 40.00
MAX30102 HRT/O2LT Sensor	\$ 8.00	4	\$ 32.00
BME208 Temperature/Pressure Sensor	\$ 2.00	4	\$ 8.00
Raspberry Pi 3B+	\$ 30.00	1	\$ 30.00
LoRa 915 MHz Antenna	\$ 10.00	4	\$ 8.00
IPX ULF Connector	\$ 0.60	4	\$ 2.40
PCB's manufactured by JLC PCB	\$ 2.00	5	\$ 10.00
TOTAL MATERIAL			\$ 705.20

Project Management – Schedule



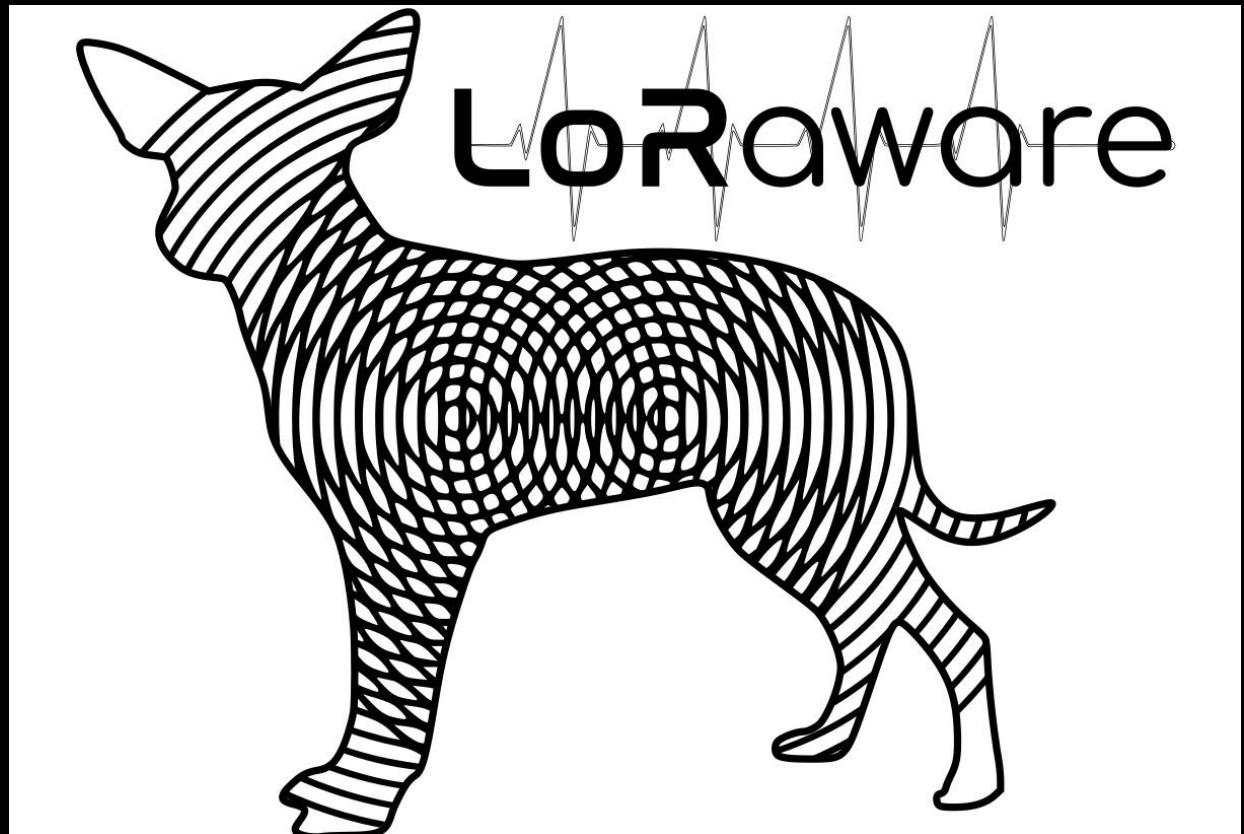
Future Capabilities & Continued Research

- Cloud Integration / Web Deployment
- ESP32 Power Saving Sleep Modes (Light Sleep & Deep Sleep)
- Dynamic Wireless Protocol Hybridization (Wi-Fi, Bluetooth 4.0, LoRa)



Closing Points & Demo

- Team Achieved Objectives
 - Successfully Tested Design
 - Systematic Approach
 - Applied Knowledge Gained
 - Proved Concept Works
 - Added ESP32 Compatibility to Arduino LoRaWAN Library
 - Demonstrated Significant Power Usage Improvements over competitor product
 - Design can be Further Developed
 - Used Available Resources at Minimal Cost
 - Showcased Free Software Tools for Future Academic utilization



References

- <https://www.hoperf.com/modules/lora/RFM95.html>
- https://github.com/SeeedDocument/RFM95-98_LoRa_Module/blob/master/RFM95_96_97_98_DataSheet.pdf
- http://surehobby.com/desc/HR/HR5512/BN_220_GPS_Antenna_datasheet.pdf
- https://www.amazon.com/gp/product/B07TGJTCTB/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1
- https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf
- https://www.soterawireless.com/wp-content/uploads/2016/03/14_SOT_005_ViSi_Brochure_8.5x11_Digital.pdf
- <http://www.techplayon.com/lora-long-range-network-architecture-protocol-architecture-and-frame-formats/>
- <https://www.maximintegrated.com/en/products/sensors/MAX30102.html#tab2>
- <https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf>
- <https://www.marketresearchfuture.com/reports/heart-rate-monitor-market-1370>
- <http://www.easyeda.com>
- <https://ru.wikipedia.org/wiki/ESP32>
- <https://socialcompare.com/en/comparison/espressif-esp8266-vs-esp32-vs-realtek-rtl8710>
- <https://www.soterawireless.com/resources/qa/>
- https://www.soterawireless.com/wp-content/uploads/2019/08/9-004020_A-ViSi-Mobile-Brochure.pdf
- <https://easyeda.com>
- <https://www.labnol.org/tech/types-of-wireless-networks/13667/>
- <https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf>
- https://www.amazon.com/gp/product/B07GGSM8SP/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1
- https://www.mouser.com/new/bosch/bosch-bme280/?gclid=CjwKCAjwkqPrBRA3EiwAKdtwkyOZ-eQqDwHZ76aTq2MBWNVu5_Nky2wK5seoHWPx5W8nIIyH63w6BoC_CkQA_vD_BwE
- https://www.mouser.com/ds/2/621/BST-BME280_DS001-09-521021.pdf
- <https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/>
- <https://blog.ampow.com/lipo-voltage-chart/>
- https://www.amazon.com/gp/product/B07BTV4J26/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
- <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf>
- https://www.amazon.com/gp/product/B01DRT4PWY/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
- <https://www.labnol.org/tech/types-of-wireless-networks/13667/>
- <https://www.semtech.com/lora/what-is-lora>
- <https://www.link-labs.com/blog/sigfox-vs-lora>
- <https://www.matchx.io/product/matchx-gateway/>
- https://nanotron.com/EN/CO_techn-css.php/
- <https://www.maximintegrated.com/en/products/sensors/MAX30102.html#tab0>
- <https://www.link-labs.com/blog/sigfox-vs-lora>
- <https://www.semtech.com/lora/what-is-lora>
- <https://www.soterawireless.com/why-visi/>