# BLIPS
# Fall Review

Matthew Speck, Amber Du, Ahmed Saied, Kevin La

# Introduction

What is BLIPS?

- BLIPS stands for Bluetooth Low energy indoor positioning system. Our goal is to track the movement of doctors and nurses and equipment in an operating room using Bluetooth Low Energy devices that are placed in the employee's ID cards.

How does BLIPS work?

- 3 - 6  Bluetooth beacons will be placed on the ceilings of the operating room to accurately triangulate the employee's position using RSSI values.  An IMU is used to keep our processor in a low power state until it is needed (Eg. when the wearer is moving).

# The Team

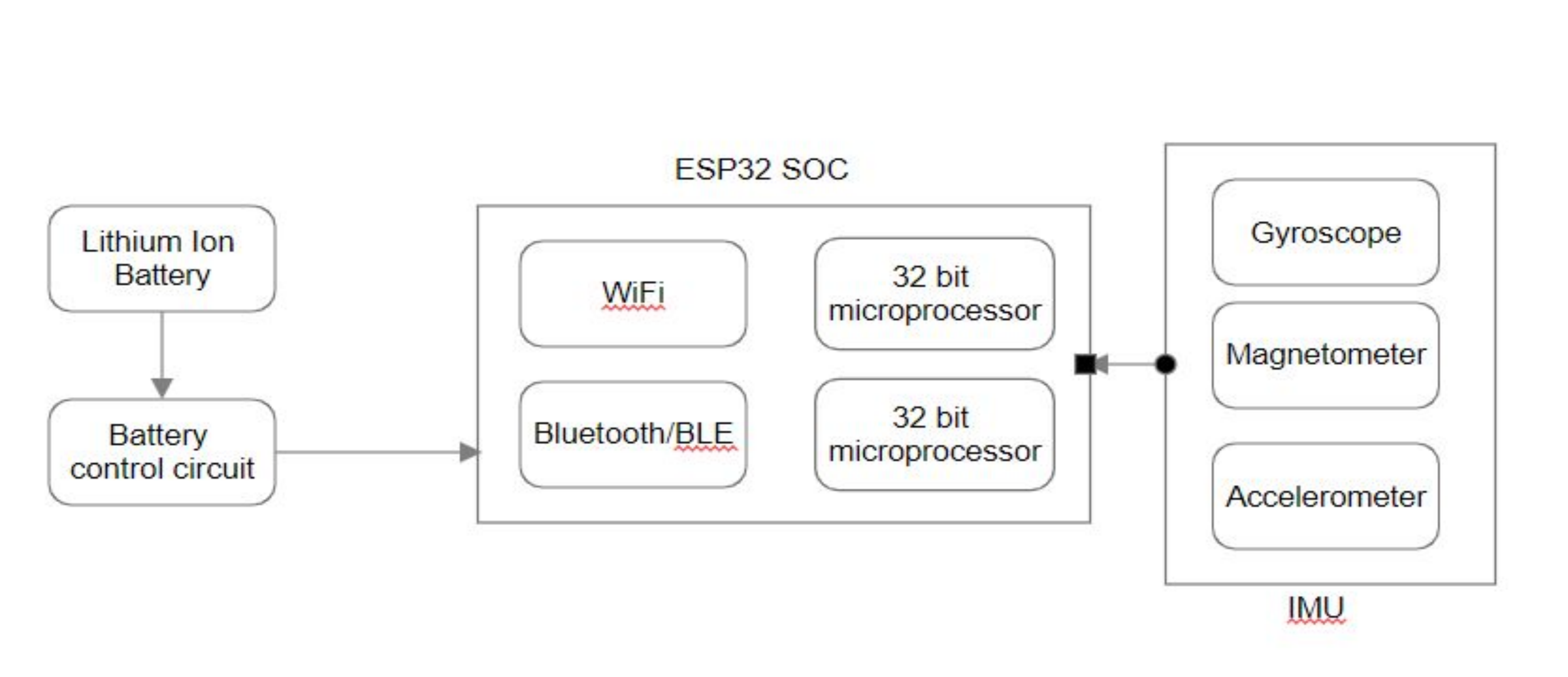
Matthew Speck - Project Lead, Parts selection, PCB/Schematic

Amber Du - Data Collection,  Signal analyst

Ahmed Saied - Software Development

Kevin La - Network Development

# High level Block Diagram



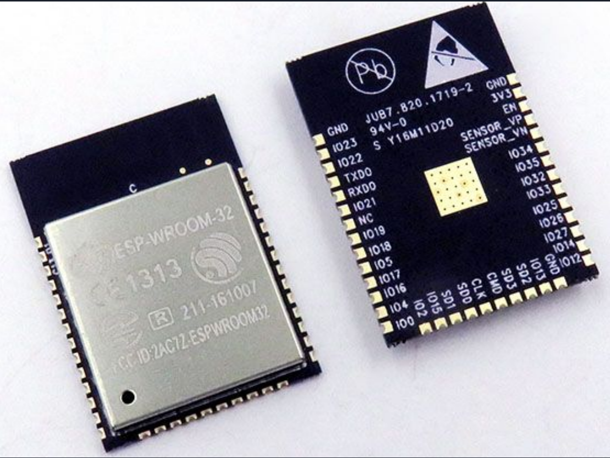
# Bill of Materials

| 1 | Ref Des | Manufacturer Part Number | Units per Board / System | Unit Price | Description |
|---|---|---|---|---|---|
| 2 | | | | | |
| 3 | Board Mounted Components | | | | |
| 4 | | esp32 WROOM32 | 1 | $8.95 | ESP32 SOC Micro controller |
| 5 | U1 | LSM9DS1TR | 1 | $6.33 | IMU ACCEL/GYRO/MAG I2C/SPI 24LGA |
| 6 | U2 | AP2112K-3.3TRG1 | 1 | $0.49 | Linear Voltage Regulator IC Positive Fixed 1 Output 3.3V |
| 7 | Q3 | DMG3415U-7 | 1 | $0.55 | MOSFET P-CH 20V 4A SOT-23 |
| 8 | D4 | NRVB120VLSFT1G | 1 | $0.43 | 1A Schottky Diodes |
| 9 | R1,R3,R7,R12 | SG73G1JTTD1003D | 4 | $0.15 | 100K ohm smt resistor |
| 10 | R13G$3,R13G$2,R13G$4 | CN1J4TTD103J | 1 | $0.10 | 10K resistor network |
| 11 | R2 | CHP0603-FX-1001ELF | 1 | $0.36 | 1K resistor |
| 12 | R4 | RR0816P-472-D | 1 | $0.10 | 4.7K resistor |
| 13 | R5,R6 | RR0816P-103-D | 2 | $0.10 | 10K resistor |
| 14 | C9 | 885012207074 | 1 | $0.11 | .22uF capacitor |
| 15 | C1,C2,C3,C5,C6,c8 | 885012107010 | 6 | $0.19 | 10uF capacitor |
| 16 | C7,C4 | 885012106010 | 2 | $0.10 | 1uF capacitor |
| 17 | x1 | | 1 | $0.75 | battery connector |
| 18 | SW2 | KMR221GLFS | 1 | $0.52 | Tactile reset switch |
| 19 | Q1,Q2 | MMBT2222A-7-F | 2 | $0.14 | Bipolar Transistors - BJT 40V 300mW |
| 20 | CHG | LO R976 | 1 | $0.38 | orange led |
| 21 | D | LS R976 | 1 | $0.35 | red led |
| 22 | U3 | MCP73831T-2ACI/OT | 1 | $0.58 | Lithium ion charger |
| 23 | X4 | 10103594-0001LF | 1 | $0.82 | micro usb |
| 24 | IC1G$1 | CP2104-F03-GMR | 1 | $1.65 | IC SGL USB-TO-UART BRIDGE 24QFN |

# Parts

Microprocessor: ESP32 BLE and WiFi module

- Dual core • Xtensa 32 bit microprocessor
- 160 MHz
- 448 Kb of ROM, 520 Kb SRAM
- 2x I2C & I2S
- 3 UART
- Ultra low power co-processor for sleep @ 8Mhz
- 150 mA max current draw



# Parts

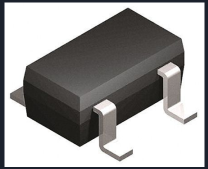
LSM9DS1

- IMU chip, contains gyroscope, accelerometer, magnetometer
- Connects through I2C
- 9 Degrees of freedom
- Used to wake Cpu from low power state
- 15 mA max current draw

# Other parts

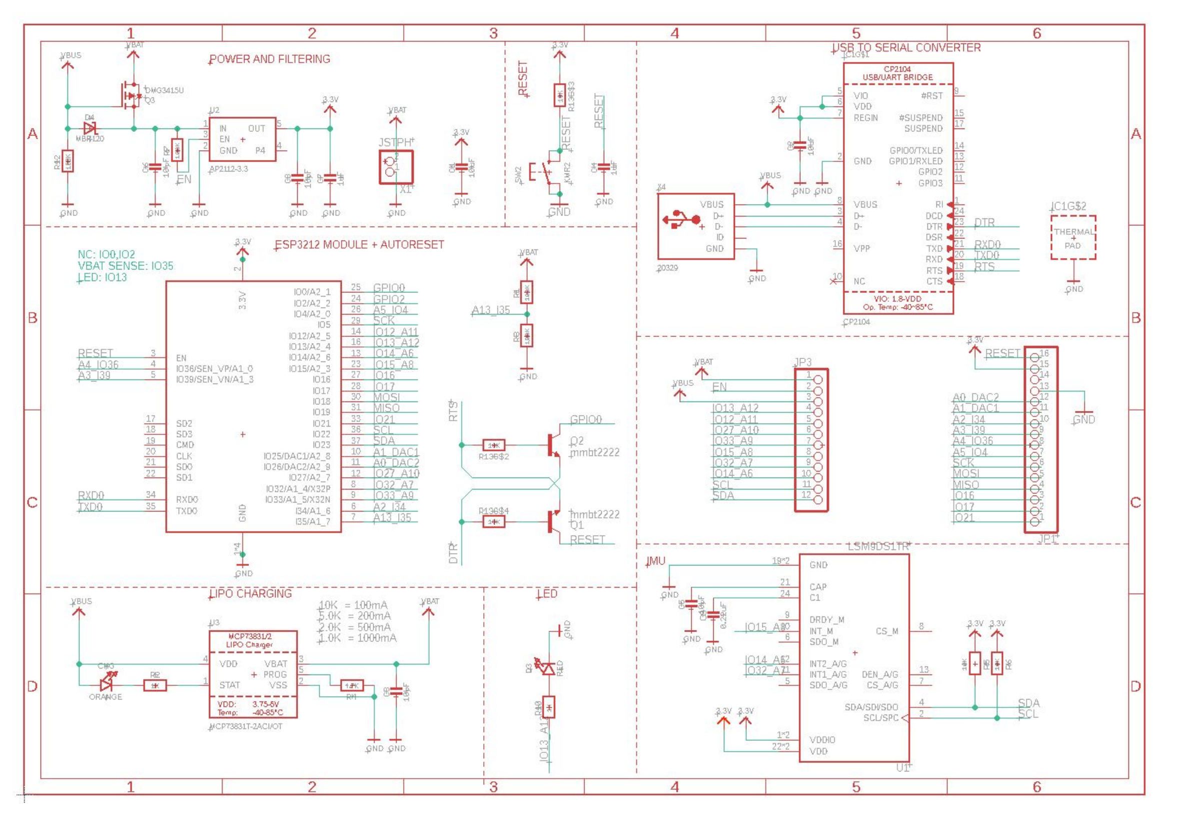

- Battery - Adafruit Lithium-Ion battery pack
  - either 2000 or 2500 mAh


- Battery Charger - MCP73831T



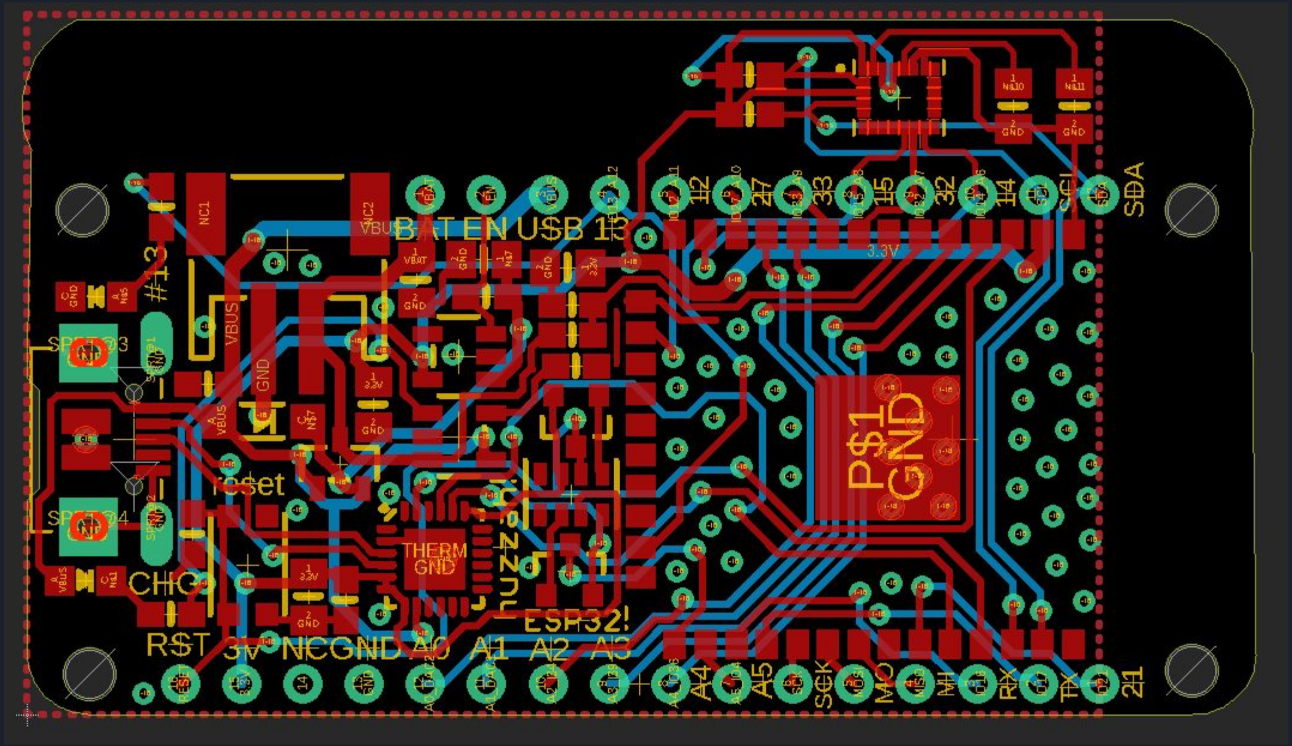
- Voltage Regulator- AP2112K

# Schematic



POWER AND FILTERING
ESP3212 MODULE + AUTORESET
USB TO SERIAL CONVERTER
LIPO CHARGING
LED
IMU
NC: IO0,IO2
VBAT SENSE: IO35
LED: IO13
CP2104
USB/UART BRIDGE
MCP73831/2
LIPO Charger
10K = 100mA
5.0K = 200mA
2.0K = 500mA
1.0K = 1000mA
LSM9DS1TR

# PCB



BAT EN USB 13
12
27
33
15
32
14
SCL
SDA
VBUS
GND
reset
CHG
THERM
GND
ESP32!
RST 3V NC GND A0 A1 A2 A3
A4
A5
SCK
MO
MI
RX
TX
21

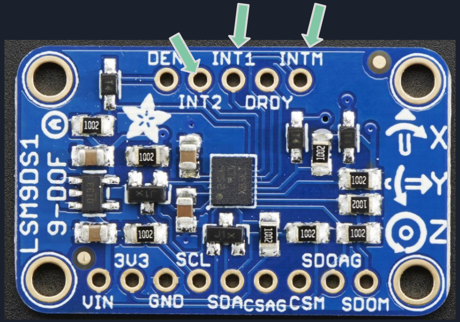
# Power Distribution

- Battery Charging:  MCP73831T
  - Lithium ion battery Charger
  - 5 volt input, drops to 3.3V to charge battery
  - Max charge rate: 550 mA

- Voltage distribution: AP2112K
  - Linear voltage regulator
  - 3.3V
  - 600 mA continuous load

# Software Development (Badge side)

- Badges will operate on an interrupt driven software structure
  - High energy mode is costly, so we only transmit/collect on a change in position
- This is accomplished via the inertial measurement unit (IMU) on our board
  - Using accelerometer/gyroscope, we can tell when a badge has shifted its location drastically
- Interrupt handling is done through an RTOS interrupt design
  - Basically, this means we will avoid interrupt collisions via RTOS interrupt queue
- Once an interrupt is triggered on LSM9DS1, we set a pin high, which the ESP32 can tell means to collect beacon data
- Accelerometer/gyroscope/magnetometer data will also transmitted in the case of the need for dead reckoning



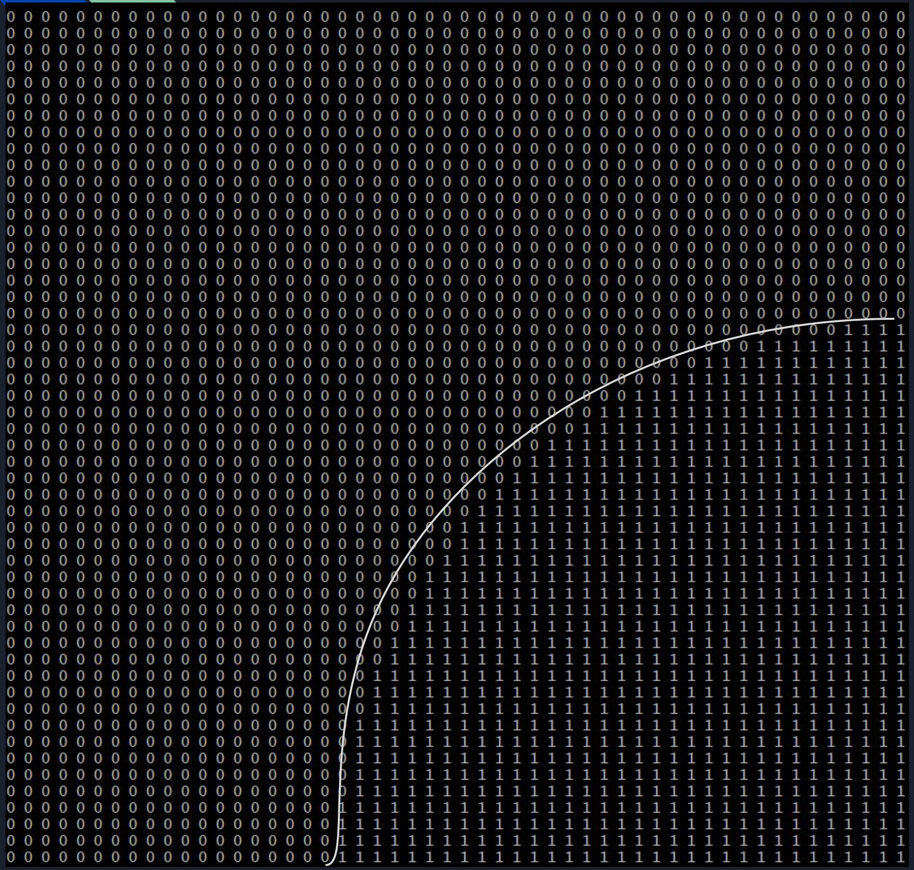
# Client - Server Communication

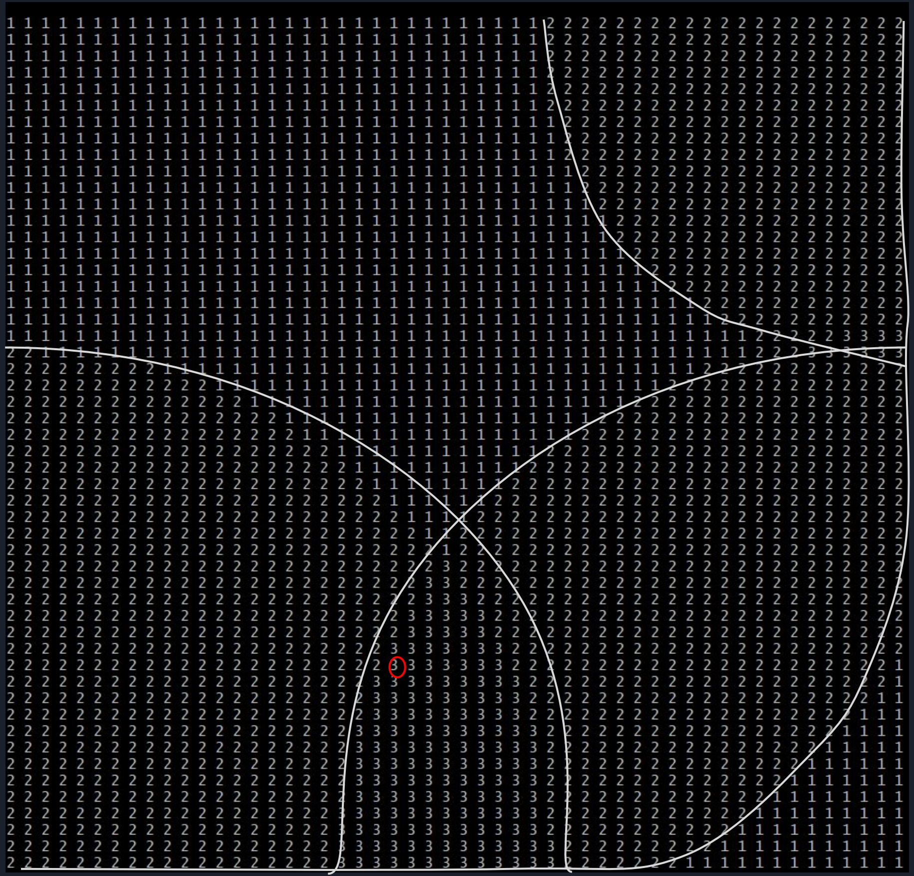
- ESP32 (badge) CPU has a built in WiFi module
  - Once data is calculated we send it off to be dealt with on the server side
  - Again, high energy state should be minimized
- Client - Side code will compartmentalize and clean up data as much as possible
  - Can handle readings from up to 10 beacons
- We will be using TCP to transmit data from client to server
  - Why TCP?
    - Reliability
    - Ensured data is retrieved in proper order
    - Maintained connection b/w client and host
- Each TCP packet should include data for every beacon it can see (less transmissions)

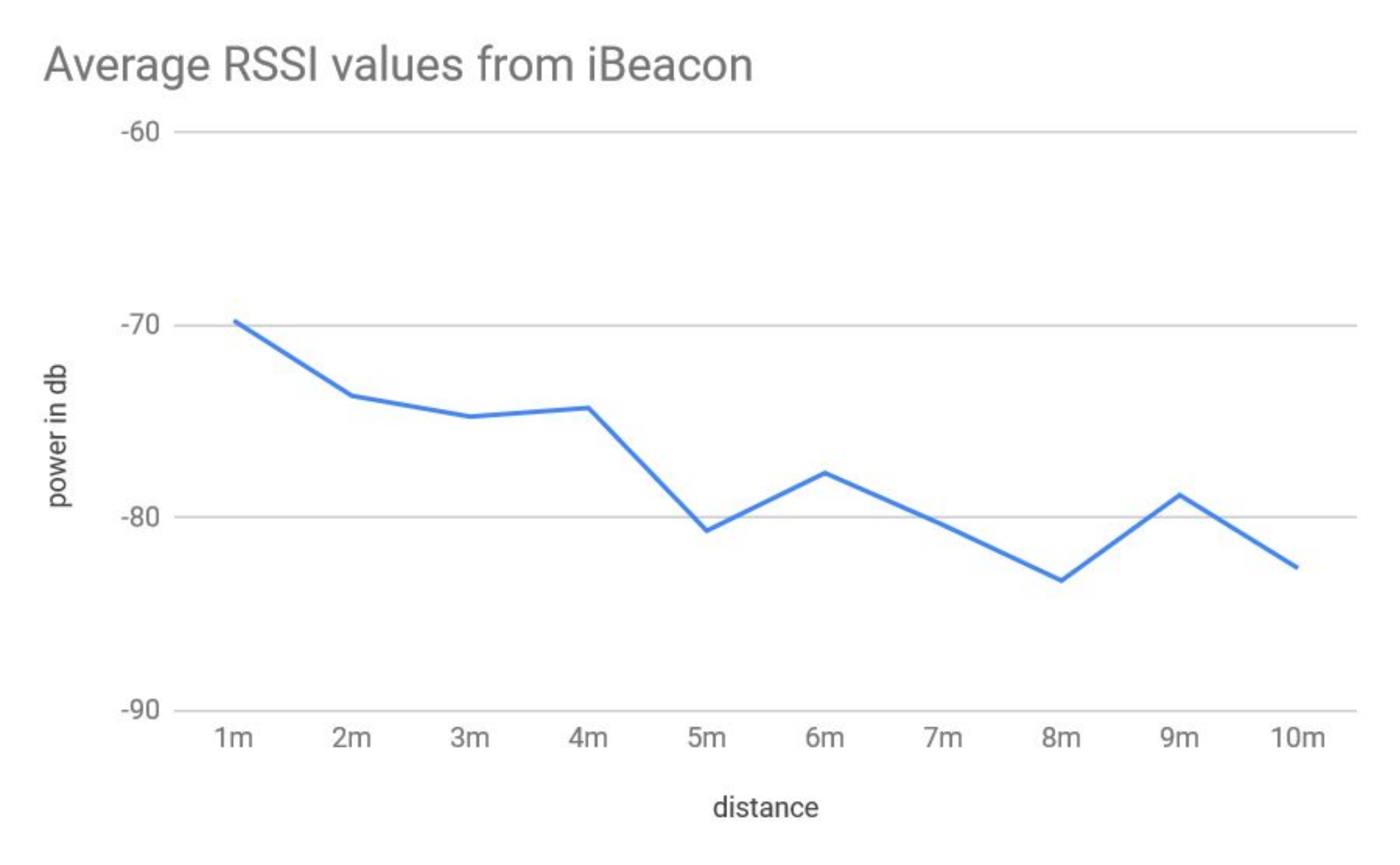
# Software Development (Server Side)

- Where we are now
  - Beacons at a preset location
  - Each beacon has a value applied to it that matches its Minor value
  - By transmitting minor to map to a specific beacon,  we can map values to locations
  - Uses square array to roughly transmit transmit distance
- Goal: Configurable Room Design
  - Meteor.js web application
    - Built in MongoDB database
    - Simplified Web Hosting Application
  - Beacons can be placed and located via reverse triangulation
  - Based on readings, we can draw circles with defined radius' that configure to the distance
    - Three intersecting circles will allow us to achieve triangulation

# Current triangulation






Average RSSI values from iBeacon
-60
-70
-80
-90
power in db
1m
2m
3m
4m
5m
6m
7m
8m
9m
10m
distance

# Software Development (Beacons)

- Estimote beacons have their own API and programmable software
- If we do use these, there is the option to configure how we'd like
    - Proposed idea: Transmissions at different signal levels
        - At high, medium, and low energy levels, we can get mitigate some error
        - This would require slight changes on the ESP32 code (handling different power levels from the same beacon)

# Things to Consider

- Precise distance measurements with BLE is difficult
  - Signal is easily blocked or reflected causing mismeasurement
    - Accelerometer/Gyroscope in IMU could help with this
- Battery Life must be as long as possible
  - Keep the processor in a low power state as often as possible through the IMU
  - Minimal processing and communication

# Conclusion

- Plans for Winter and Spring
  - We will start working on the server side of this project
  - Look into Dead Reckoning and WiFi tracking as backups
- The prototype board design complete will be sent in for production, once completed we can start testing with the IMU.
- Will further improve the conversion from RSSI values to distance measurements
  - Using a data driven equation instead of the definition of RSSI conversion would provide a more accurate representation of distance measured
  - The farther away the fewer packets received. Beacons with more packets received in a time period

# Thanks to:

- Our Sponsor: Arthrex
- Our Instructor : Yoganada Isukapalli
- Our TA's:
  - Brandon Pon
  - Carrie Segal