

# BLE-Based Mobile App for Raspberry Pi Management

---

Tyler Collins

# Introduction

- What is AppMAIS?
- What problem does my project address?
- How does my project solve this problem?

# Project Overview

This project combines a mobile application with a GATT server on the Raspberry Pi to enable interaction with the Pi.

- **GATT Server:** Deployed on Raspberry Pi, facilitates BLE communication and serves as the central hub for accessing and modifying characteristics on the Pi.
- **Mobile Application:** React Native app designed to connect with GATT server, providing a way to manage variables and verify sensor functionality.

# Part 1: BLE Connection



# Bluetooth Low Energy (BLE)

A wireless communication technology designed for short-range communication between devices with low power consumption.

## **BLE**

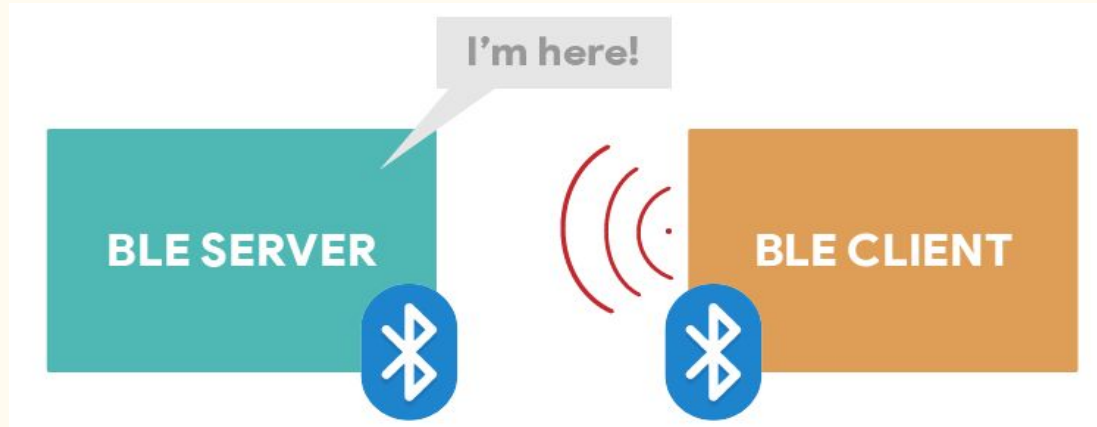
- Intermittent data transmissions
- Low power consumption
- Shorter-range
- Low data transfer rate

## **Bluetooth**

- Continuous data transmission
- Longer range (up to 330ft)
- Higher data transfer rate

# Advertisement and Connection

- The Generic Access Profile (GAP) controls connection and advertisement.
- Peripheral device (GATT server) advertises its presence using GAP. This includes the device's address as well as available services. The central/client device then scans for these advertisements and connects upon discovery.



# GATT Servers

- A GATT (Generic Attribute Profile) Server is a BLE feature which organizes and provides access to data for wireless communication.
- Serves as a framework for clients to interact with data via attributes.
- Data is read and modified through services and characteristics.

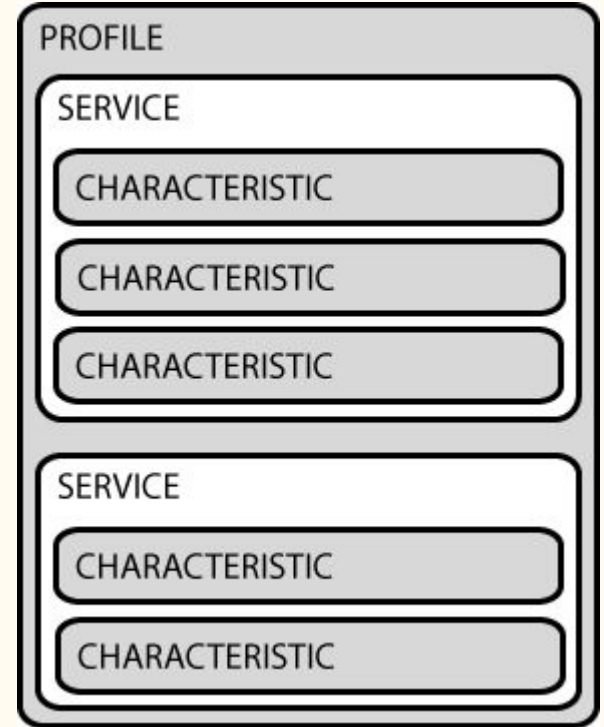
# GATT Services and Characteristics

## Services

- A logical grouping of related characteristics.

## Characteristics

- An individual data point or function within a service.
- Consists of value and properties (read/write)
- Ex. Characteristic providing access to variable in appmais config file.





**Central**

**GATT Client**



- ➡ Discover services
- ➡ Discover characteristics
- ➡ Read characteristic
- ➡ Enable notifications

Connected

**Peripheral**

**GATT Server**

Heart Rate Service

Heart Rate Measurement **N**

Body Sensor Location **R**

Heart Rate Control Point **W**

Device Information Service

Manufacturer Name String **R**



**Heart Rate  
Sensor**

- R** Read
- W** Write
- N** Notify

# Part 2: Mobile Application (Demo)



# Device List

- List of Raspberry Pi devices which have been registered in the application.
- Makes the connection process quicker.
- If a Pi has not been registered yet, we can still use the ‘scan for devices’ button to find it.

## Device List

### Registered Devices:

Below is a list of registered devices. Tap on a device to connect. If connection fails, try the "Scan for Devices" button to ensure the device is advertising.

rpi4-60

---

rpi4-30

---

rpi4-11

---

6RC

---

6L

---

**Scan for Devices**

---



# Password Verification

- So any person who happens to come across this app can't automatically get into the hives and tamper with them.
- A characteristic exists on the Pi with both read and write permissions. We write the user entered password to the Pi where it is checked with the correct password and then a true or false response is returned.

**Password Verification**

Disconnect

**Submit**

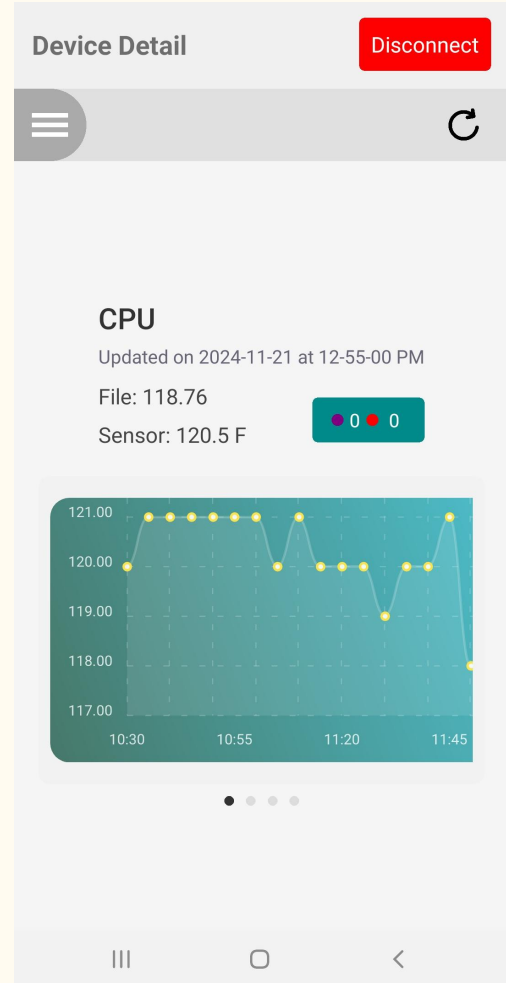
|||

○

<

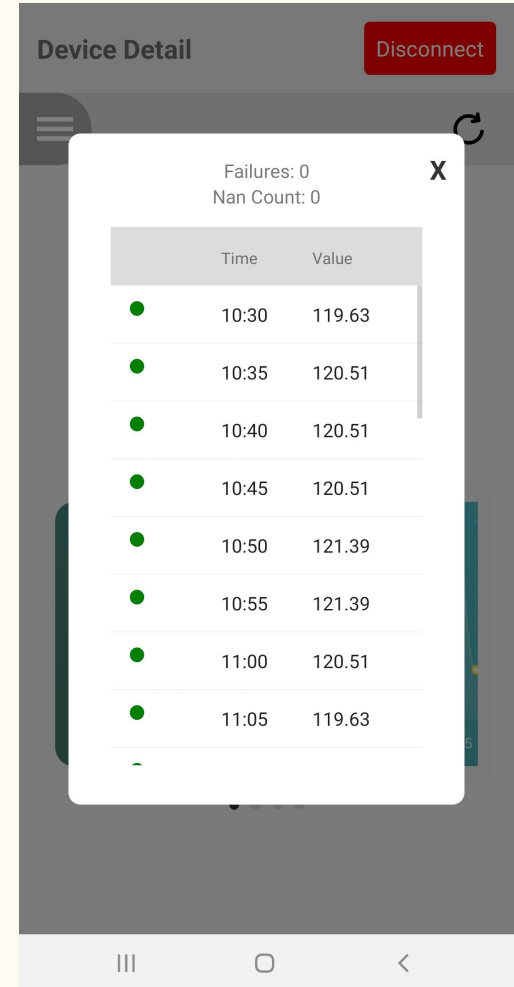
# Device Details

- Home screen of the application.
- Provides information on each of 4 sensors including:
  - Most recent recording from saved AppMAIS file.
  - Line chart showing all recordings from file.
  - Failure and nan value counts



# Details Modal

- Gives details on each point in the selected line chart.
- Data includes time of recording and the value recorded.
- Failure count = 3 or more nan recordings in a row.
- Nan Count = Amount of nan recordings.
- Dot color meanings:
  - Green: Non-nan value recording
  - Red: Nan recording
  - Yellow: Interpolated value



# Variables Tab

- Where variables contained in the config file located on the Raspberry Pi can be modified.
- Modifiable variables include:
  - capture\_window\_start\_time
  - capture\_window\_end\_time
  - capture\_duration\_seconds
  - capture\_interval\_seconds

←

Modifications

VARIABLES

SENSORS

COMMANDS

Basic Data

Device ID: E4:5F:01:5F:AF:73

Device Name: rpi4-60

Below, you can modify the variables in the Pi configuration file to customize their values.

All Other Sensors

capture\_window\_start\_time:0800

capture\_window\_end\_time:2300

capture\_duration\_seconds:60

capture\_interval\_seconds:300

Submit

Refresh

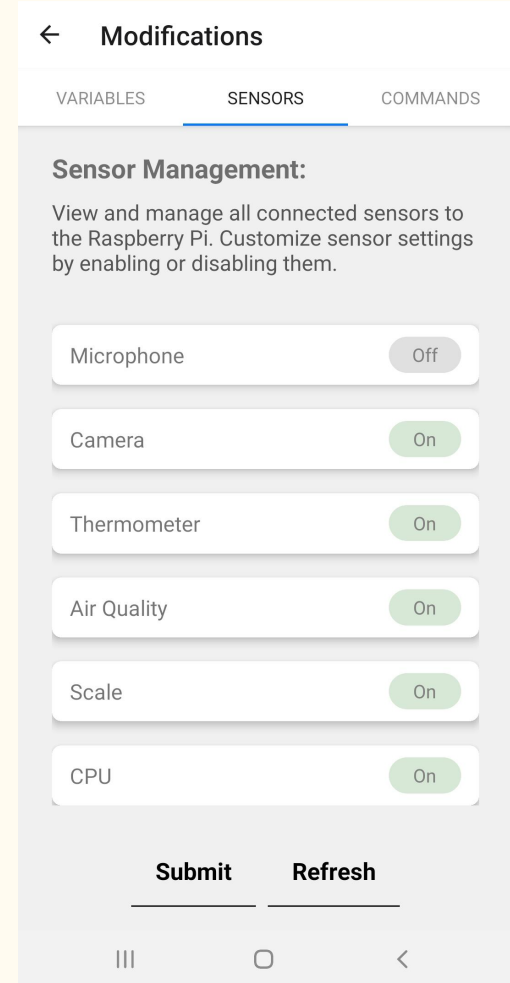
III

O

<

# Sensor States

- Where the state of sensors can be modified (on/off)





# Commands

- Used to send commands to be run on the Raspberry Pi.
- Contains list of common quick commands.
- Won't allow user to run a command if recording is currently taking place.

← Modifications

VARIABLESSENSORSCOMMANDS

Commands

Device ID: E4:5F:01:5F:AF:73  
Device Name: rpi4-60

Type your command here...

Clear

Quick Commands

sudo systemctl start appmais

sudo systemctl stop appmais

sudo systemctl restart appmais

Send Command

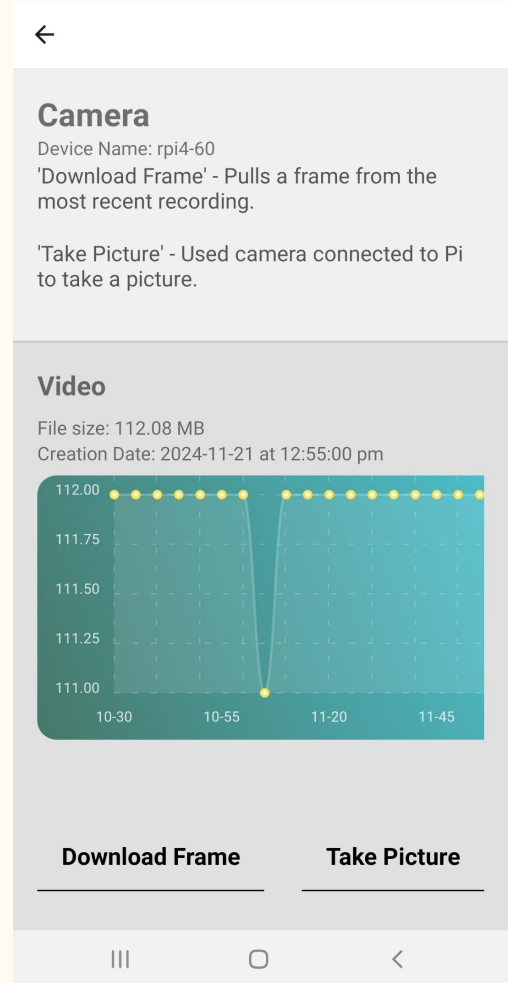
III

O

<

# Camera

- Contains information on the most recent recording by the Pi including file size and the date of creation.
- Contains line chart showing file sizes of recordings for the day.
- Includes button used to download a frame from the most recent recording.
- Includes button used to take picture using the camera connected to the Pi.



# Conclusion

—

# Recap

## Problem

- Needed simple way to modify variables on the Pi
- Needed simple way to verify sensor functionality.

## Solution

- Developed mobile application for BLE communication with Pi.
- App allows users to modify configuration variables on the Pi as well as access sensor data allowing for sensor functionality verification.
- Wireless communication with Pi without need for internet access.

# Future Work

- Support for new sensors can be added
- UI improvements
- Services and characteristics can be added to support new applications which require BLE communication with Pi's.

Questions?

