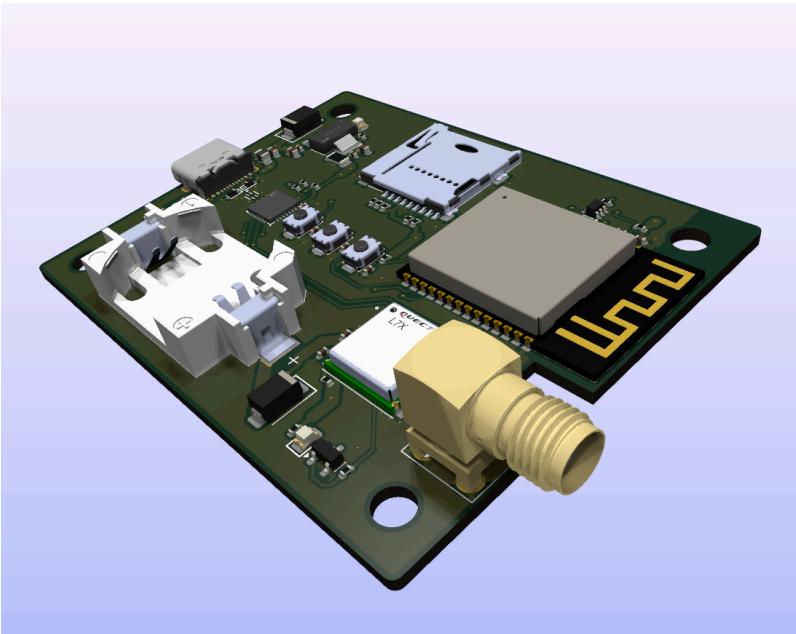


# Mileage Logger Project



## Scope

To log and display mileage for business purposes. With the ability for switching from personal journey to business journey.

## Technical Design Skills

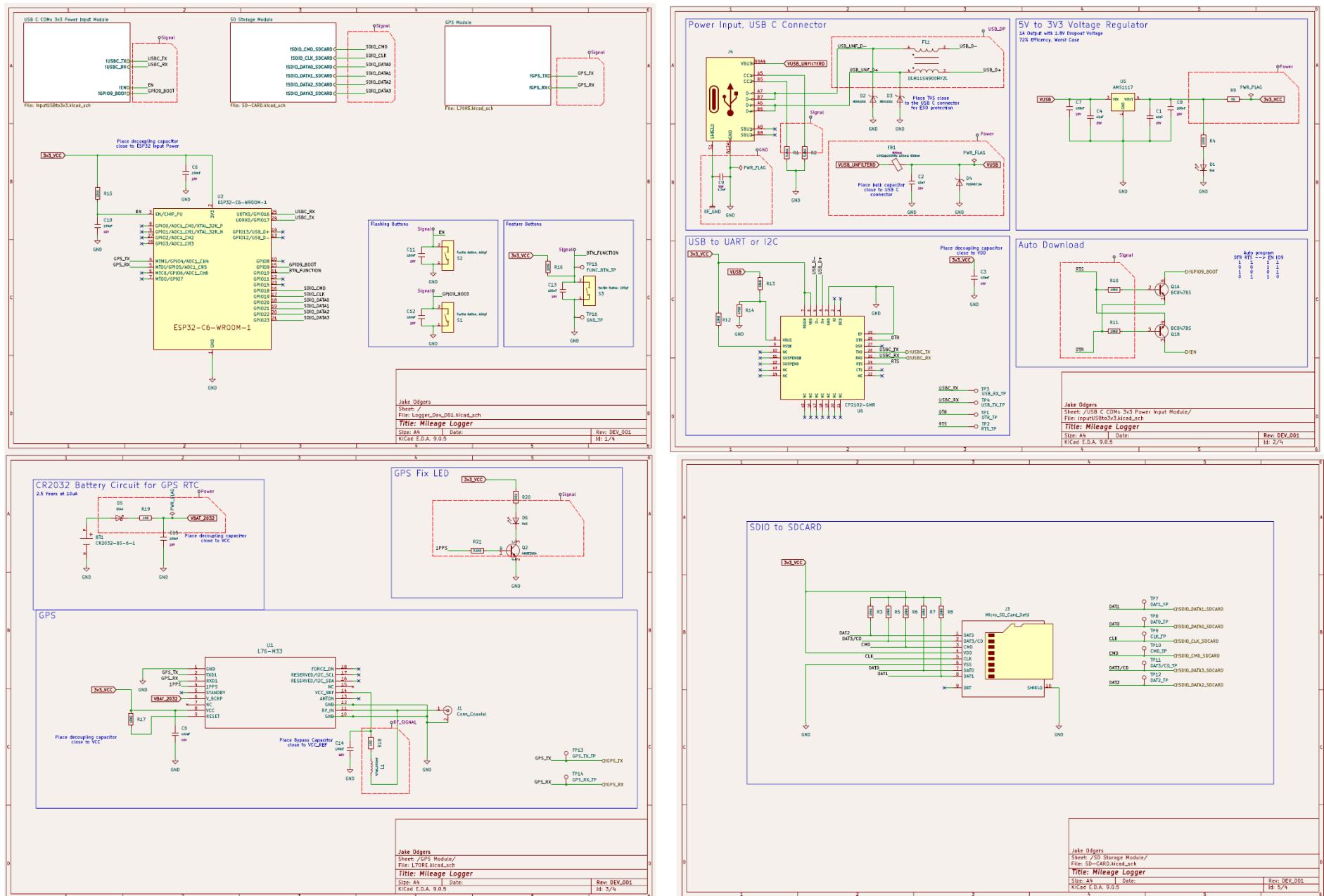
- RF waveguide design with a GCPW (Grounded Coplanar Waveguide)
- Differential pair routing for USB C datalines
- EMC and transient considerations
  - TVS (Transient Voltage Suppression)
  - Common mode chock
  - 4 layer board design with ground pours and power plane

## Display

Option 1 - Display through a GUI on a computer made through Python Tkinter.

# Mileage Logger Project

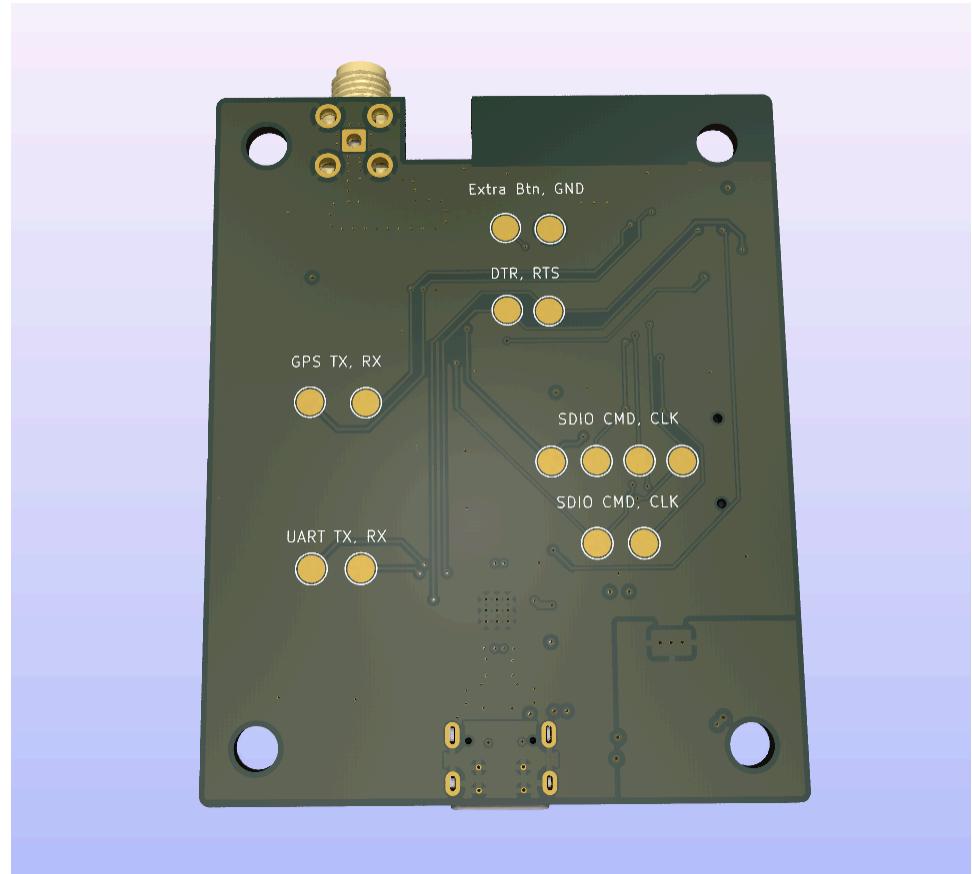
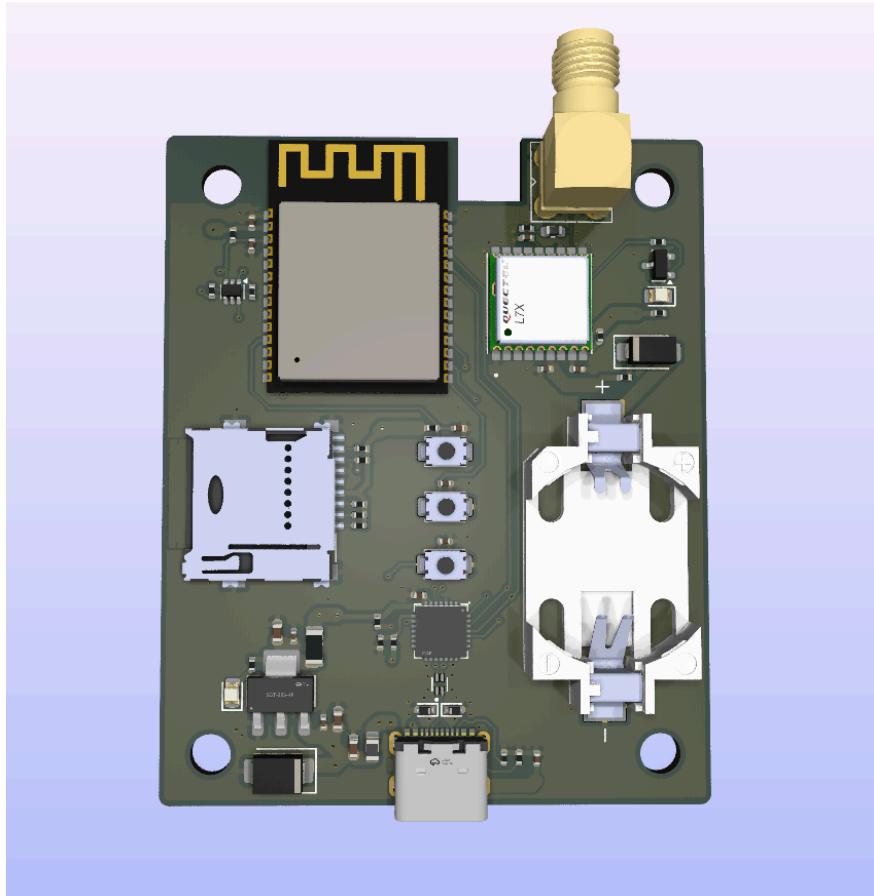
## Schematic



Jake Odgers

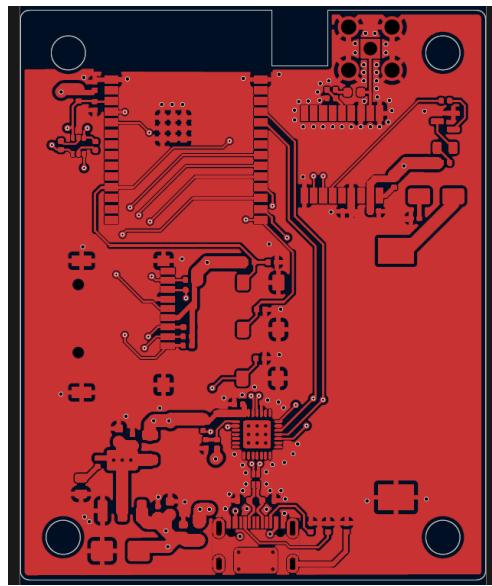
# Mileage Logger Project

## PCB Layout

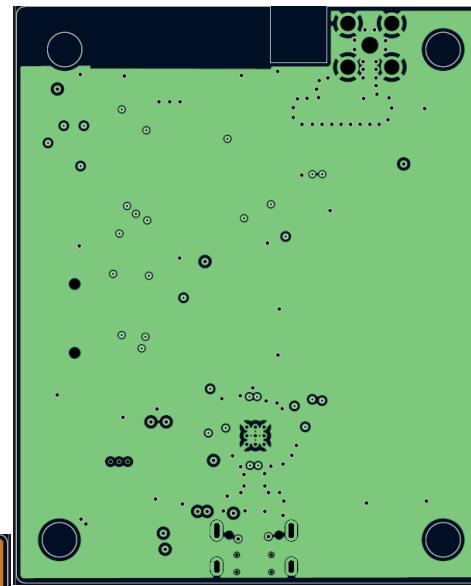


# Mileage Logger Project

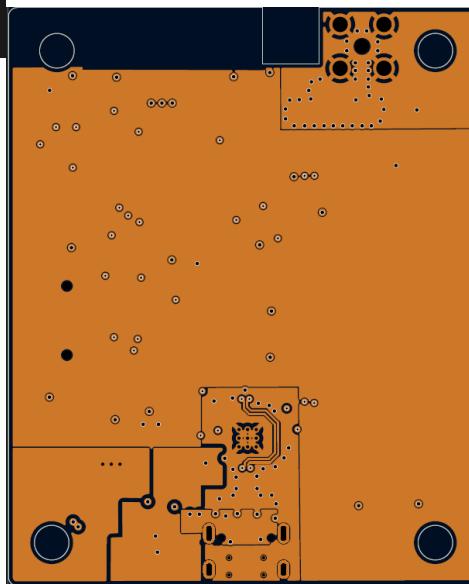
## PCB Layer View



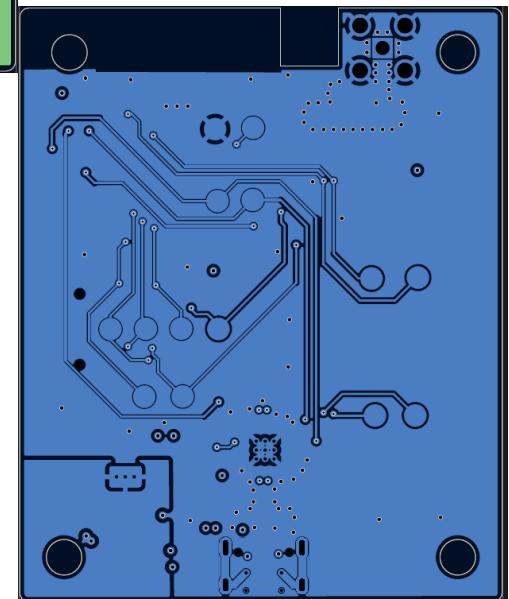
Layer 1 - Signal  
(with GND pour)



Layer 2 - GND



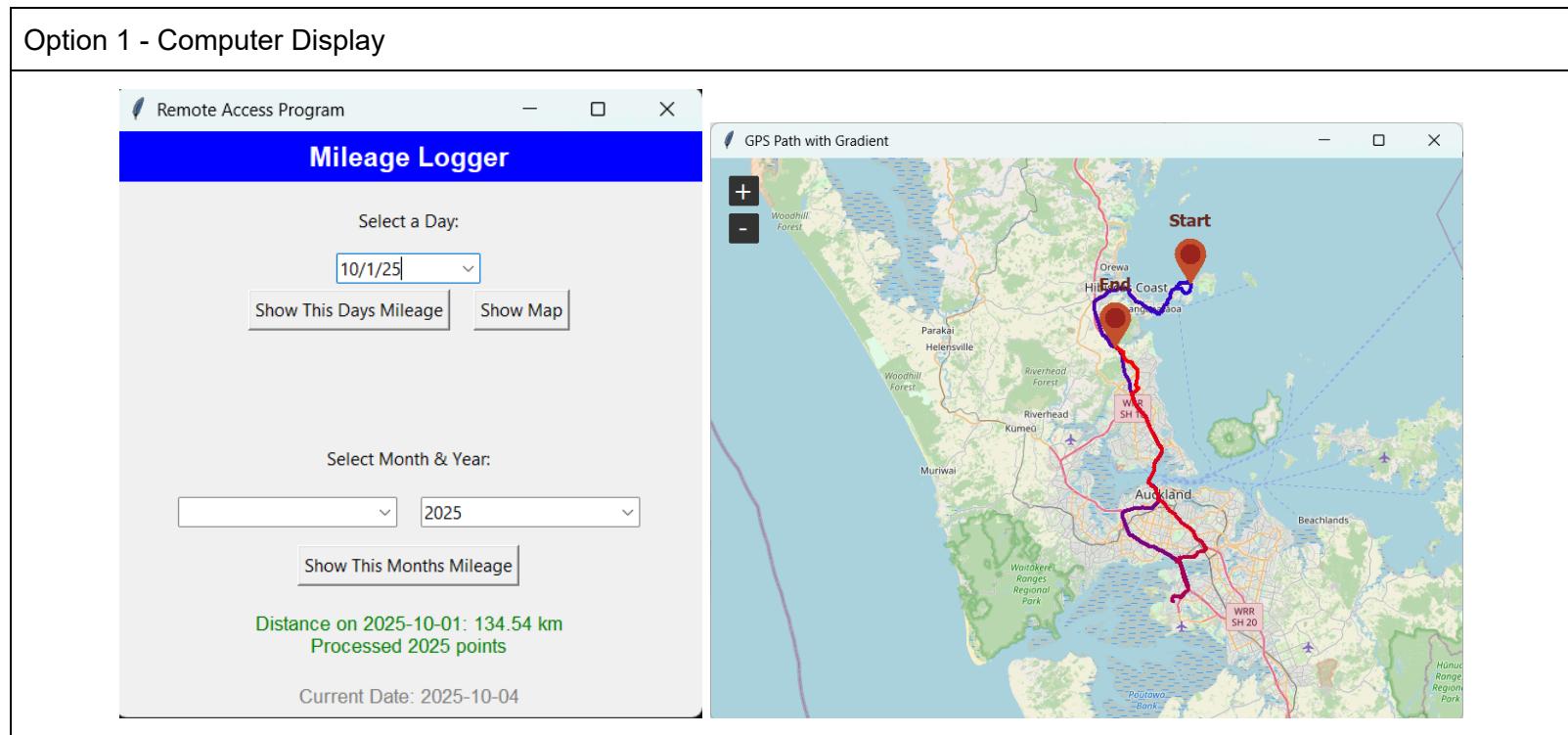
Layer 3 - Power  
(with GND pour  
around differential  
pair)



Layer 4 - Signal  
(with GND pour)

# Mileage Logger Project

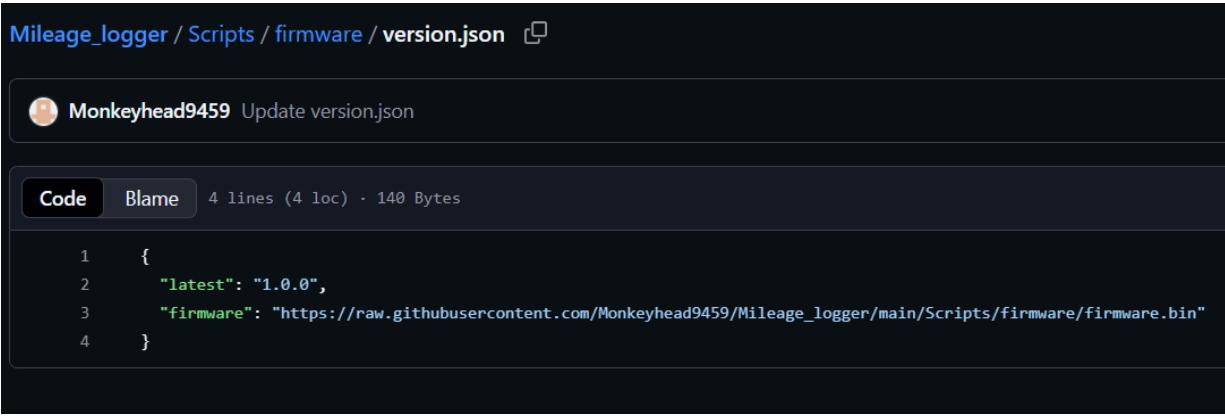
GUI



# Mileage Logger Project

## Back End

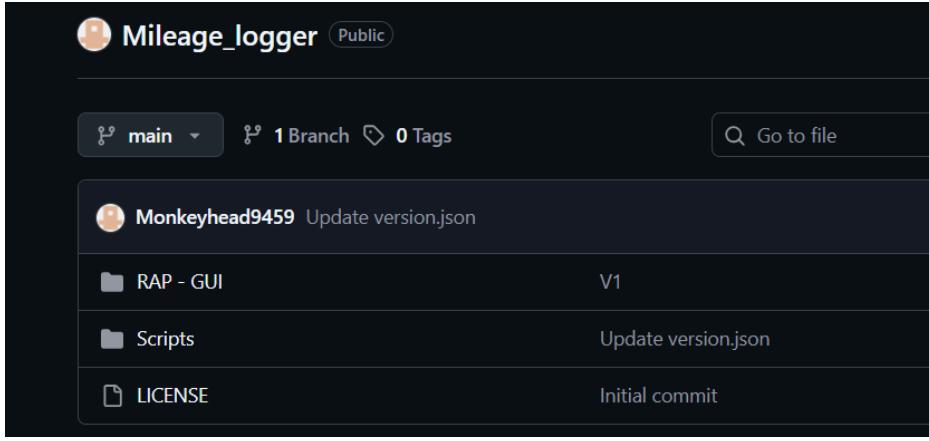
ESP32 REMOTE FIRMWARE UPGRADE THROUGH GITHUB



A screenshot of a GitHub commit page for the 'Mileage\_logger' repository. The commit is titled 'Update version.json' by 'Monkeyhead9459'. It shows the following JSON code:

```
1  {
2    "latest": "1.0.0",
3    "firmware": "https://raw.githubusercontent.com/Monkeyhead9459/Mileage_logger/main/Scripts/firmware/firmware.bin"
4 }
```

GITHUB VERSION CONTROL



A screenshot of the 'Mileage\_logger' GitHub repository page. The repository is public and has one branch ('main') and zero tags. The repository contains three files: 'RAP - GUI', 'Scripts', and 'LICENSE'. The 'RAP - GUI' folder is labeled 'V1'. The 'Scripts' folder has an 'Update version.json' commit. The 'LICENSE' file is an 'Initial commit'.

# Mileage Logger Project

## Back End

AWS DYNAMODB TABLE FOR CLOUD STORAGE

DynamoDB > Tables

Tables (2) [Info](#)

	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read cap
<input type="checkbox"/>	raw_data	Active	dev_esp32 (S)	-	0	0	Off		On-demand
<input type="checkbox"/>	raw_data_v2	Active	dev_esp32 (S)	timestamp (S)	0	0	Off		On-demand

API GATEWAY TO LAMDA FUNCTION FOR ESP32 DATA STORAGE

esp32WriteToDynamodb

▼ Function overview [Info](#)

[Diagram](#) | [Template](#)

esp32WriteToDynamodb

Layers (0)

API Gateway

[+ Add trigger](#)

# Mileage Logger Project

## Improvements:

- Incorporate Google Roads API for more accurate mileage logging by snapping to roads
- Make a portable option for battery connection.
  - SMPS (Increase efficiency)
  - Smaller profile
    - ESP32-C3-MINI-1U
    - Offboard wifi and bluetooth antenna
    - Local/Smaller GPS antenna
- Option 2 for display is making an app through Flutter to connect to the ESP32 through bluetooth and store and display results as well as change journey status from business to personal.