Design Review 2

1. Major Subsystems

For each of the subsystems, we demonstrate its functionality and that the systems indicate that it meets the requirements we had set out in earlier design reviews.

1.1. Power Management

The power management subsystem currently consists of two 12 V batteries connected in series to provide 24 V to new stronger motors than we had earlier. Based on expected current from the motors and the ampere hour rating of the batteries, the batteries will provide power for the 45 minutes we require. With the connections between the motors and batteries, we are currently experimenting with different wire gauges, considering the current running through the wires and stability of the connections.

Figure 1 shows an image of the schematic, showing an LDO which will step the input from the battery to the 3.3 V for the ESP32 and other connections, including the ultrasonic sensors and GPS module. We will use a fixed voltage LDO rather than an adjustable one.

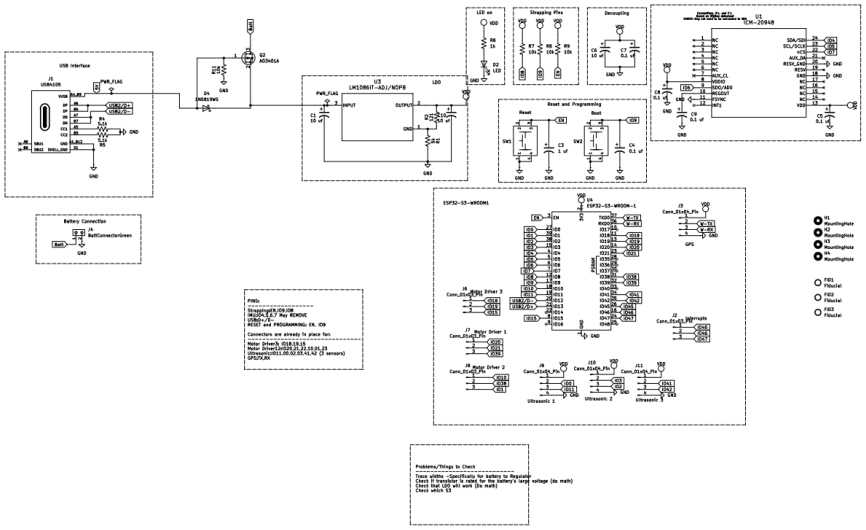

Figure 1. KiCAD Schematic with LDO

1.2. Physical

The current physical setup is shown below in figure 2. Due to the recent changes of motors, a more stable setup of the wheels was not possible for this design review. The physical setup does demonstrate the viability of keeping the mower a similar size to a standard push mower remaining within a reasonable weight that we expected. The motors are able to be secured to the

board and the motors can move with the wheels attached. Testing showed that the mower was able to drive forward with this setup, and will consistently work when stable connections are created.

Figure 2. Physical build

1.3. Navigation

Navigation is demonstrated with the working ultrasonic sensors, received data from the IMU, and a GPS program. The ultrasonic sensors, where there will be two on the front and one on the back, reliably demonstrate that they can detect objects from a meter away. The ultrasonic sensor is demonstrated with a program which stops the motors driving forward when it detects an object. For the distance the ultrasonic sensor can detect, the mower will have more than sufficient room to navigate away from the obstacle. The IMU receives data about the magnetic field which will be helpful in providing another reference for position.

The GPS setup is shown in figure 3. The current GPS program allows a user to select GPS points by pressing an interrupt button at that position. The program adds that point to an array of saved points. When the user is ready, the user presses another interrupt button to switch to searching mode. The program then gives the distance and direction to each point in the order that they were saved. The output to the serial monitor is shown in figure 4 while the program is searching for points. The program shows the distance to the next point, the direction to the next point and the distance the user is currently headed. When the point is found, the program gives directions to the next point. Later, the program will be developed further to have the points selected be the boundary corners and the program will use a style of interpolation to create points to find

covering the whole lawn. This demonstration shows the ability to collect and then search for points, essential for the system.

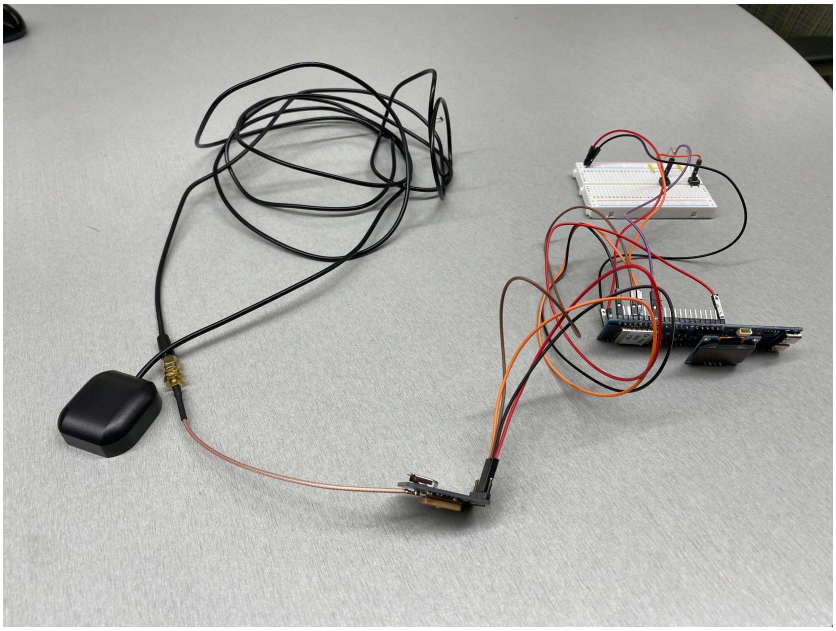

Figure 3. GPS setup

```
find dist: 0.000287    find azimuth: -163.01    cur azimuth: 8.13
find dist: 0.000274    find azimuth: -163.84    cur azimuth: 8.13
find dist: 0.000269    find azimuth: -161.82    cur azimuth: 8.13
find dist: 0.000257    find azimuth: -159.15    cur azimuth: 8.13
find dist: 0.000247    find azimuth: -158.20    cur azimuth: 8.13
find dist: 0.000242    find azimuth: -155.85    cur azimuth: 8.13
find dist: 0.000232    find azimuth: -154.70    cur azimuth: 8.13
find dist: 0.000225    find azimuth: -151.70    cur azimuth: 8.13
find dist: 0.000215    find azimuth: -150.26    cur azimuth: 8.13
find dist: 0.000209    find azimuth: -149.22    cur azimuth: 8.13
find dist: 0.000196    find azimuth: -146.93    cur azimuth: 8.13
find dist: 0.000189    find azimuth: -145.67    cur azimuth: 8.13
find dist: 0.000176    find azimuth: -145.62    cur azimuth: 8.13
find dist: 0.000169    find azimuth: -144.16    cur azimuth: 8.13
find dist: 0.000157    find azimuth: -140.91    cur azimuth: 8.13
find dist: 0.000154    find azimuth: -140.01    cur azimuth: 8.13
find dist: 0.000141    find azimuth: -139.40    cur azimuth: 8.13
Found Pointfind dist: 0.000458    find azimuth: -7.66    cur azimuth: 8.13
find dist: 0.000468    find azimuth: -6.55    cur azimuth: 8.13
find dist: 0.000475    find azimuth: -5.53    cur azimuth: 8.13
find dist: 0.000478    find azimuth: -4.57    cur azimuth: 8.13
find dist: 0.000482    find azimuth: -3.63    cur azimuth: 8.13
find dist: 0.000482    find azimuth: -3.63    cur azimuth: 8.13
find dist: 0.000478    find azimuth: -3.66    cur azimuth: 8.13
find dist: 0.000478    find azimuth: -3.66    cur azimuth: 8.13
find dist: 0.000478    find azimuth: -3.66    cur azimuth: 8.13
find dist: 0.000478    find azimuth: -3.66    cur azimuth: 8.13
find dist: 0.000478    find azimuth: -3.66    cur azimuth: 8.13
find dist: 0.000478    find azimuth: -3.66    cur azimuth: 8.13
find dist: 0.000478    find azimuth: -3.66    cur azimuth: 8.13
find dist: 0.000474    find azimuth: -3.69    cur azimuth: 8.13
find dist: 0.000474    find azimuth: -3.69    cur azimuth: 8.13
find dist: 0.000474    find azimuth: -3.69    cur azimuth: 8.13
```

Figure 4. GPS program serial monitor while searching

1.4. Remote Control via BLE and App

This subsystem is demonstrated with a basic BLE app. The ESP32 can easily connect to other devices over BLE. The simplest way of demonstrating the connection is using the nRF Connect which shows devices and their characteristic functions. A simple program was created on the ESP32 was created, in one characteristic taking a value which turns on or off the builtin led and for the next range of values received on the ESP32 prints from “hard left” to “hard right”, representing a remote control input. Another characteristic reads values from the ESP32 with a notification system rather than continuously monitoring. The program sends floats to the receiver, representing the float GPS points. For an interface, an app was created with MIT App Inventor. The UI when connected to the ESP32 is shown in figure 5. The app successfully sends values to the ESP32 and can print the notional turning direction with the slider, shown in figure 6. The App Inventor is tedious to use for large projects and is not intuitive, so we also created some MATLAB scripts which use the BLE program. The MATLAB script like the app can send values to the ESP32 and read the floats, as shown in the command window in figure 7. We are also looking to investigate the bleak library on Python for the creation of an app as an option. The notifications also trigger when they are expected, demonstrating a working BLE system. For the integration, values will be replaced with useful data to be sent to or received from the ESP32.

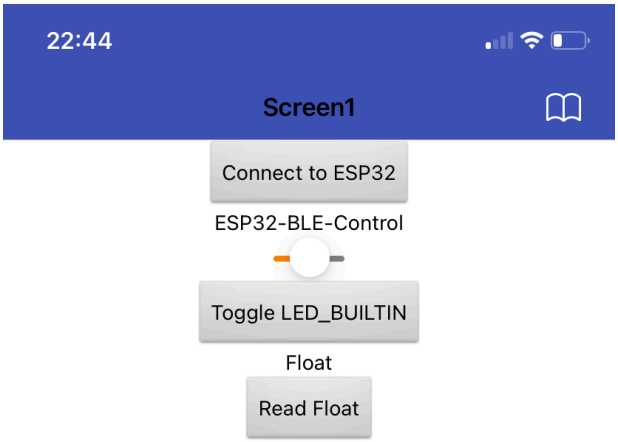

Figure 5. MIT App Inventor BLE app

```
Left
D NimBLEServer: Gatt Write event
4
Straight
D NimBLEServer: Gatt Write event
4
Straight
D NimBLEServer: Gatt Write event
5
Right
D NimBLEServer: Gatt Write event
5
Right
D NimBLEServer: Gatt Write event
6
Hard Right
D NimBLEServer: Gatt Write event
6
Hard Right
D NimBLEServer: Gatt Write event
D NimBLEServer: Gatt Write event

Hard Right
D NimBLEServer: Gatt Write event
6
D NimBLEServer: Gatt Write event
6
Hard Right
```

Figure 6. Writing direction to ESP32 over BLE

```
>> write(write_c, 1)
>> write(write_c, 0)
>> floatValue = typecast(uint8(read(read_c)), 'single')

floatValue =

  single

   19.9500

>> floatValue = typecast(uint8(read(read_c)), 'single')

floatValue =

  single

   19.0700

>> floatValue = typecast(uint8(read(read_c)), 'single')

floatValue =

  single

   21.7500

>> floatValue = typecast(uint8(read(read_c)), 'single')

floatValue =

  single

   27.6000
```

Figure 7. MATLAB reading and writing to ESP32 over BLE

1.5. Safety

The safety subsystem is demonstrated with the IMU detecting the 35º degree tilt to stop the mower. The IMU reliably and accurately detects the angle. This setup provides the required safety feature to prevent the motor running while it is tipped over. In the safety subsystem, a button or switch will be integrated into the physical circuit which opens the connection between the motors and the battery. This currently takes the form of removing the wire to quickly and surely stop the motors but will later be a physical switch.