ECE 445

Senior Design Laboratory

Design Document

Navigation Vest Suite For People With Eye Disability

Team No. 5
Haoming Mei
(hmei7@illinois.edu)
Jiwoong Jung
(jiwoong3@illinois.edu)
Pump Vanichjakvong
(nv22@illinois.edu)

TA: Rishik Sathua

Table of Contents

ECE 445.....1
Senior Design Laboratory..... 1
Design Document.....1
Table of Contents.....2
Introduction.....3
1.1 Problem:.....3
1.2 Solution:.....3
1.3 Visual Aid:..... 4
1.3.1 Updates:.....4
1.4 High Level Requirements..... 6
2. Design.....7
2.1 Block Diagram..... 7
2.2 Physical Design.....9
2.3 Power Subsystem..... 11
2.4 Sensing Subsystem..... 35
2.5 Microcontroller Subsystem.....44
2.6 Computer Vision / Navigation Subsystem.....48
2.7 Sensory Output Subsystem..... 52
2.8 Software Design.....55
2.8.1 Raspberry PI:.....55
2.8.2 ST Microcontroller:.....61
2.9 Tolerance Analysis:.....76
2.9.1 Data Transmission:..... 76
2.9.2 Power Management.....76
2.9.3 PCB Heat Dissipation.....78
2.9.4 Sensing Subsystem (ToF + GPS + Magnetometer).....89
3. Cost & Schedule.....92
3.1 Bill of Materials (Cost Analysis)..... 92
3.2 Schedule.....95
4. Ethics and Safety.....97
User Data Privacy:..... 97
Campus and Lab Policies:.....97
User Safety:.....98
5. Sources and References.....99

Introduction

1.1 Problem:

People with Eye Disability often face significant challenges navigating around in their daily lives. Currently, most available solutions range from white canes and guide dogs to AI-powered smart glasses, many of which are difficult to use and can cost as much as $3,000. A participant emphasized the financial barrier, stating, “These solutions are too expensive, I don’t know anyone who can afford these without support from PFRON.”[1]

Multiple resources show the urgent needs for a navigation system suite for people with blindness. According to a study published in the National Library of Medicine, at least 2.2 billion individuals worldwide suffer from vision impairment, and there is an urgent need for efficient assistance due to the rapid rise in their population [2]. Additionally, research highlights that visual impairment (VI) significantly reduces quality of life … due to poor product design and limited adaptation [3].

Thus, based on the above sources, we realized that there is an urgent need for us as an electrical and computer engineer to build an innovative and intuitive vest navigation suite for people with blind disability.

1.2 Solution:

Our solution is to create a Navigation Vest Suite that provides assistance for people with visual impairments. It provides step-by-step routing through GPS and magnetometer, and delivers guidance through haptic and audio cues. Users can interact with the system through voice commands, and find out where the vest is by querying the destination through speech. While in motion, onboard sensors and object detection models will identify obstacles and people, announcing their presence and the distance through a speaker. Users will also receive direction of obstacles along with the turning angle required to avoid/acquire them through vibration motors and the speaker.

Haptic Action	Motion
Zero Haptic Feedback	No Obstacles Around
Front Haptic	Obstacle Present Infront
Right Haptic	Obstacle Present Right Side
Left Haptic	Obstacle Present Left Side

Based on the above table, motors are able to work together in conjunction to represent multi direction feedback as well. For instance, if both the front haptic and right haptic are activated, that means that the obstacles are present in both the front and the right side of the user. Additionally based on the proximity of the haptic vibration, the user will know how close the objects are. Higher the frequency, the obstacle is closer in proximity, lower the frequency, the obstacle is further in proximity. For now, we are planning on to create equation in the form of $Motor\ Haptic\ Frequency\ =\ 1/Distance\ to\ the\ Object$ equation will be tweaked later to compensate for each variable domain where: $Distance\ to\ the\ object \in [2cm,\ 400cm]$, $Motor\ Haptic\ Frequency \in [141Hz, 200Hz]$. The final effect is that whenever the user is closer to the object, the motor vibration will exponentially increase, whereas when the object is faraway from the user, the vibration frequency will slowly decrease.

The main reason we chose to house our components within a vest over glasses comes down to comfort. When utilizing vibration motors, wearers might feel some discomfort due to long usage, given that they are using the device for over longer periods of time. By utilizing a vest, not only are we able to house more components onto our device, but we are also able to enable users to wear the device more comfortably.

1.3 Visual Aid:

1.3.1 Updates:

- Audio Output: Wireless Earbuds/Headphones
  - Play audio via bluetooth from Raspberry PI
- Audio Input: Microphone

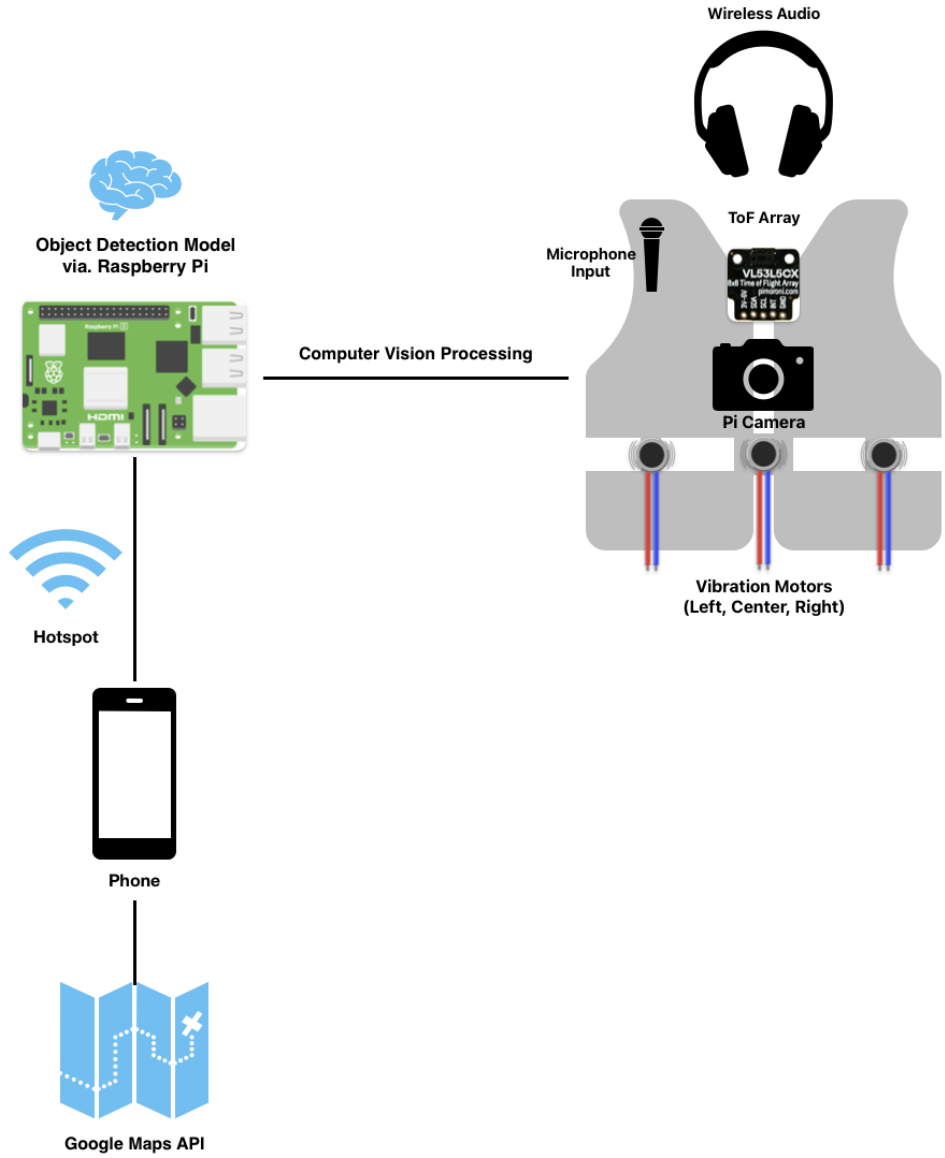


Figure 1.3.1 Visual Aid

We will be using Raspberry Pi that runs vision models (YOLO, MMDetection, MMTracking toolkit) to detect objects, object bounding boxes, and the relative direction to the obstacle. This information will be sent to the STM32 microcontroller, and it will fuse this input with sensor data from the GPS, magnetometer, and ultrasonic/ToF to generate auditory and haptic feedback to guide the user. Meanwhile, the Raspberry PI will also be processing the voice based activation for destination queries and system control (power on/off, settings).

1.4 High Level Requirements

One of the main issues with these types of projects are related to the computational performance of these devices , specifically its false detections. We will be using these metrics to compute the effectiveness of our software.

- Prevalence
- F1 Score
- FP/FN
- Recall/Precision

Testing the Sensor's Capability:

-> **True Positive Case means:** Object is truly present and motor vibrates accordingly

-> **True Negative Case means:** The object is truly not present and the motor vibrates accordingly.

False Positive Cases	False Negative Cases
1) The wall is detected as present, but isn't there. 2) The object detection model detects a street sign, but in reality a street sign does not exist. 3) After an object passes by, the camera ghosting or the device latency may detect the object to be still present, but in reality, the object is already passed. 4) The camera detected an object, but in reality it is a shadow of an object/person. 5) Pattern surfaces or walls may confuse the model to detect an object, but in reality it is just free space.	1) The wall is detected as not present, but it is there. 2) The object detection model fails to detect the street sign but in reality, the street sign exists. 3) The detection fails whenever a transparent object is present. 4) Environmental factors like weather, humidity, and terrain can obscure the camera and sensors, which can have false negative readings even though the object does exist. 5) Vest blind spots can be a potential false positive as sensors cannot detect any obstacles/objects in such areas.

1. The device must be able to detect obstacles at a minimum distance of 1 meter and generate haptic and auditory feedback with at least 70% precision, 65% recall. Similar

research employing YOLO models for outdoor obstacle detection reported ~80% precision and ~68% recall, providing a realistic benchmark for this requirement [5].

2. Vest must generate the accurate haptic and auditory feedback within 200 ms of obstacle detection, and the total detection to feedback delay must not exceed 320 ms threshold, which is the industry standard requirement for the acceptable limit for wearable obstacle-avoidance devices [4]
3. The device must provide at least 3–4 hours of continuous battery life without significant degradation in accuracy, precision, recall, or latency.

2. Design

2.1 Block Diagram

- Need to update Block Diagram
    - Sensory Output Subsystem
        - Audio Output: Remove Audio IC + Speaker Setup
            - Replace with: PI -> Bluetooth -> Wireless Headphones
    - Power Subsystem:
        - Battery is 6-8.4V
        - Two Power Lines 5V & 3.3V
            - Battery-> Buck -> 5V
            - Battery -> Buck # 2 -> 3.3V
        - User Input Subsystem merges will Sensing Subsystem
            - I2S Microphone

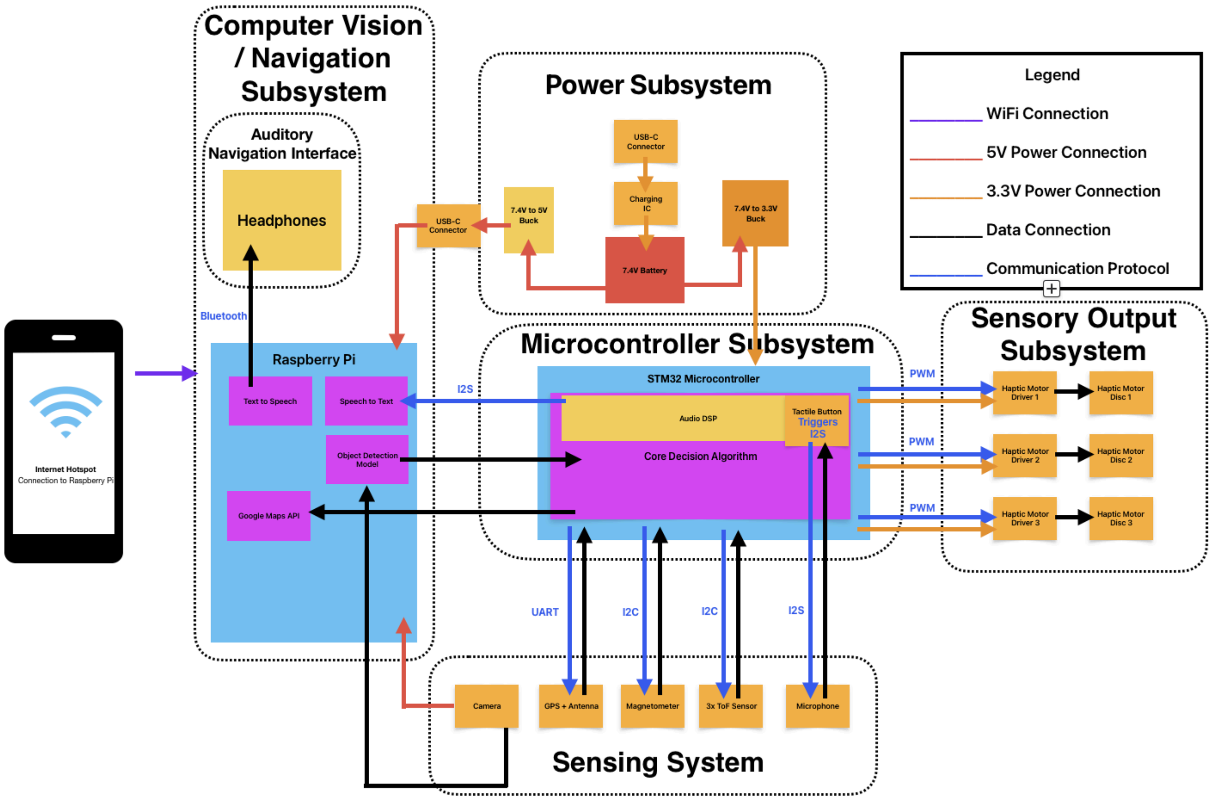

Computer Vision / Navigation Subsystem
Auditory Navigation Interface
Headphones
Bluetooth
Raspberry Pi
Text to Speech
Speech to Text
Object Detection Model
Google Maps API
Internet Hotspot
Connection to Raspberry Pi
Power Subsystem
USB-C Connector
Charging IC
7.4V to 5V Buck
7.4V to 3.3V Buck
7.4V Battery
USB-C Connector
Legend
WiFi Connection
5V Power Connection
3.3V Power Connection
Data Connection
Communication Protocol
Microcontroller Subsystem
STM32 Microcontroller
Audio DSP
Tactile Button Triggers
I2S
Core Decision Algorithm
I2S
Sensory Output Subsystem
PWM
PWM
PWM
Haptic Motor Driver 1
Haptic Motor Disc 1
Haptic Motor Driver 2
Haptic Motor Disc 2
Haptic Motor Driver 3
Haptic Motor Disc 3
UART
I2C
I2C
I2S
Camera
GPS + Antenna
Magnetometer
3x ToF Sensor
Microphone
Sensing System

Figure 2.2.1

2.2 Physical Design

Subsystem Overview & Requirements

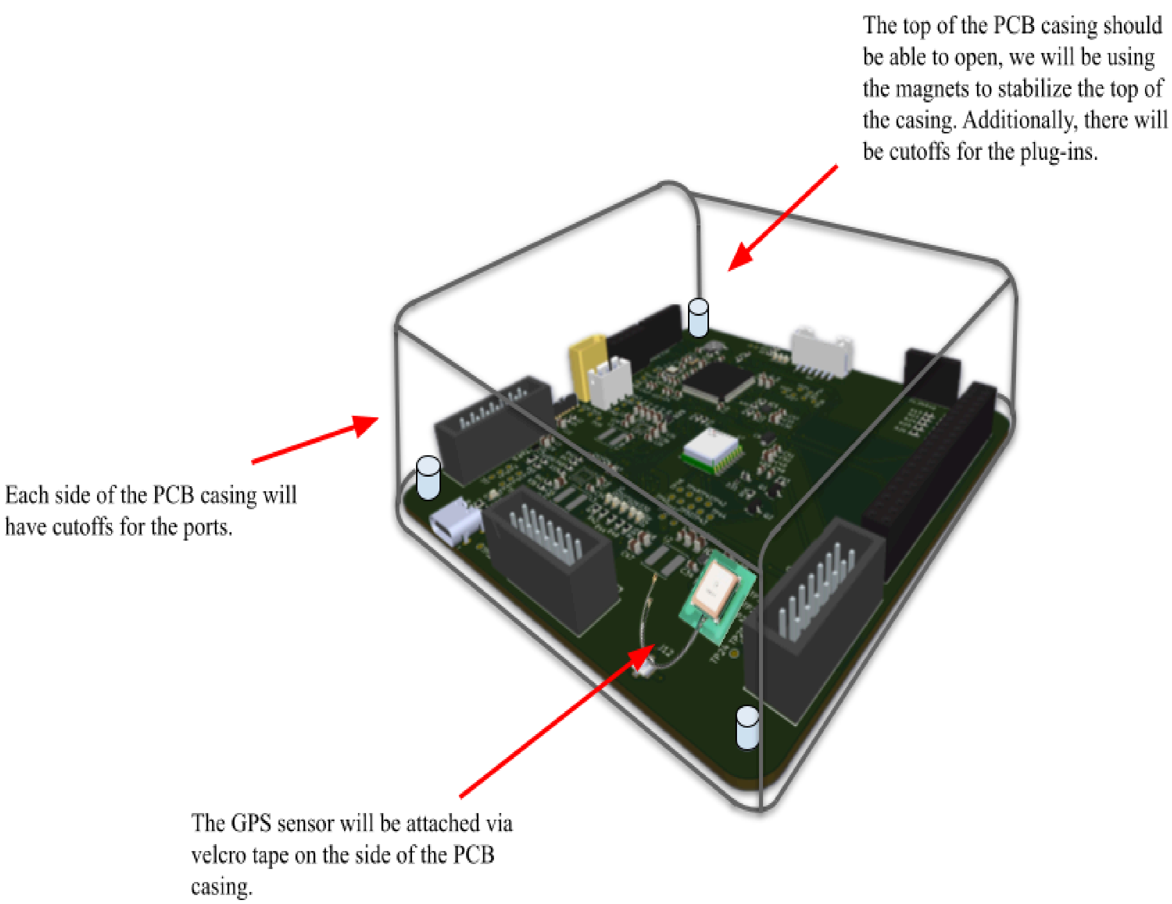


Figure 2.2.1 PCB with Casing with Annotations

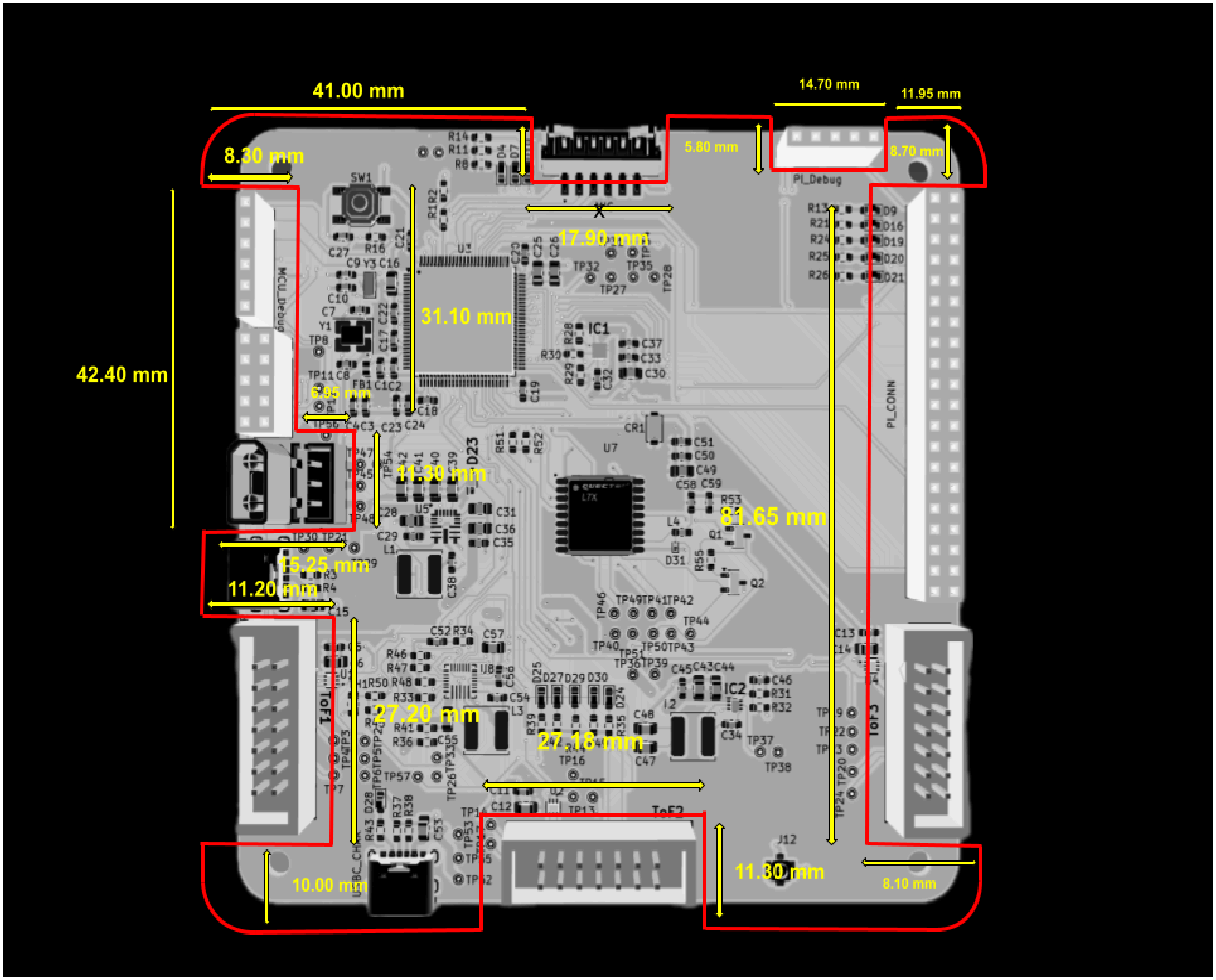


Figure 2.2.2 Top PCB Casing Dimension Information

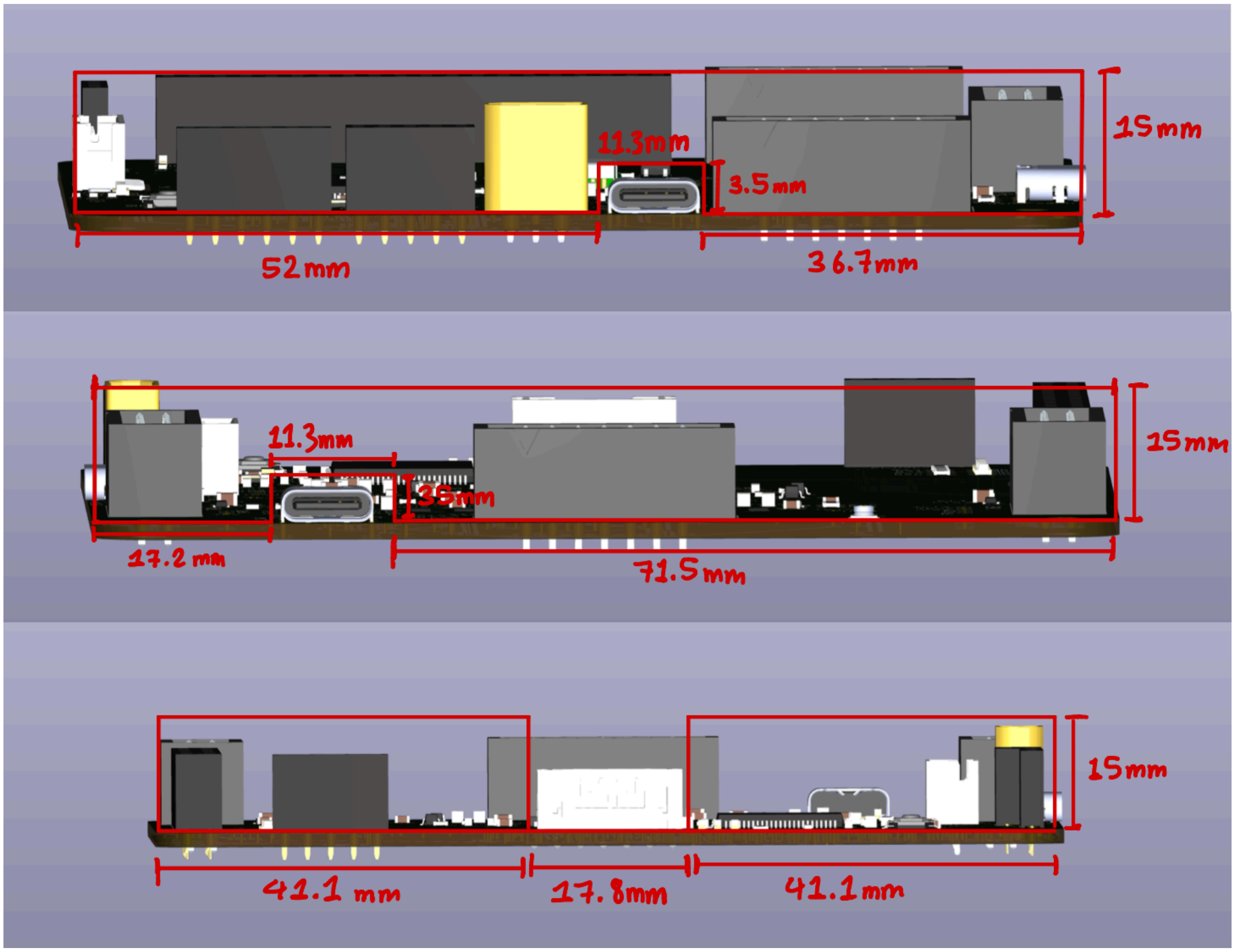


Figure 2.2.3 Side PCB Casing Dimension Information

2.3 Power Subsystem

<u>Description and Purpose:</u> The power subsystem is responsible for powering all of the other subsystems in our project. It consists of 2x3.7V Li-Ion batteries, supplying 7.4V at an estimated continuous discharge current of ~7-10A at 5000mAh. The design utilizes multiple Buck Converters to convert the battery voltage level to 5V, and also to 3.3V. It should be able to supply at least 25W for the 5V system and 7W for the 3.3V system.

<u>Power Analysis:</u>

- 5V: ~25W
  - Raspberry PI 5: 25W
- 3.3V: ~6W
  - GPS: 0.0825W
  - 3 x Motor Disc: 0.60W

- 3 x Motor Drivers: 0.012W
- 2 x ToF: 0.626W
- ST Microcontroller: 2.64W
- Magnetometer: 0.003W
- Speaker: 1W
- Audio Amplifier: 0.8W
- External DAC: 0.150W
- Microphone: 0.002W
- Others: ~1W

PCB Design & Implementation:

USB-C Battery Charging:

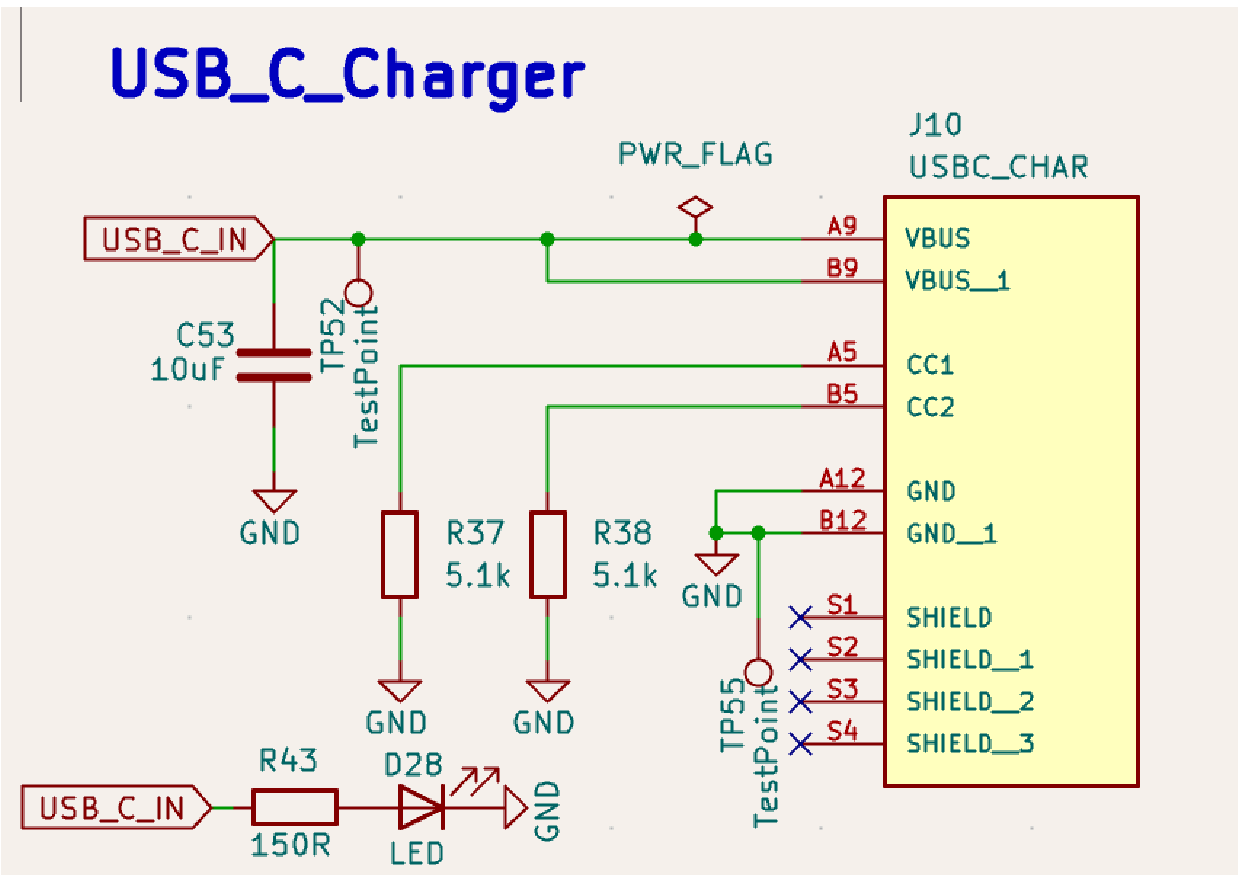
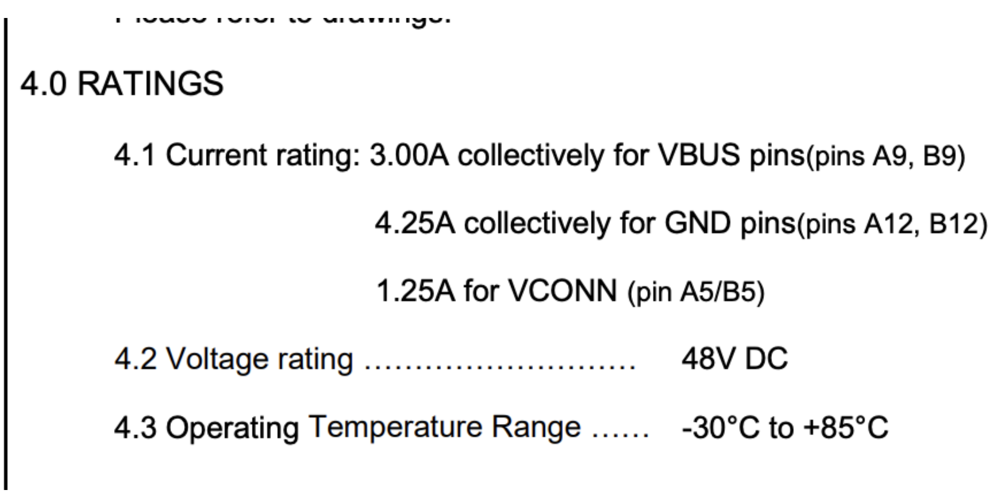


Figure 2.3.1: Kicad Schematics for USB-C Charger

Charging can be done via any type of USB-C cable, which should be connected to an adapter supplying 5V @ 3A peak. The current limit is programmable via an external resistor setup to the ILIM pin of the MP2639CGR-Z Charger IC, which will support up to ~2.98A of continuous charging. This USB-C connection goes into the VL connection of the Battery Charger IC, which allows charging from USB-C to the Battery.


4.0 RATINGS

4.1 Current rating: 3.00A collectively for VBUS pins(pins A9, B9)

4.25A collectively for GND pins(pins A12, B12)

1.25A for VCONN (pin A5/B5)

4.2 Voltage rating .... 48V DC

4.3 Operating Temperature Range ...... -30°C to +85°C

Figure 2.3.2: USB-C Charger Ratings [41]

This has an operating temperature range in between -30 degrees Celsius and 85 degrees Celsius, this will be indirectly regulated by the Battery Charger IC, as the MP2639C will decrease current or stop charging when the temperature is too cold or hot.

Battery Charging:

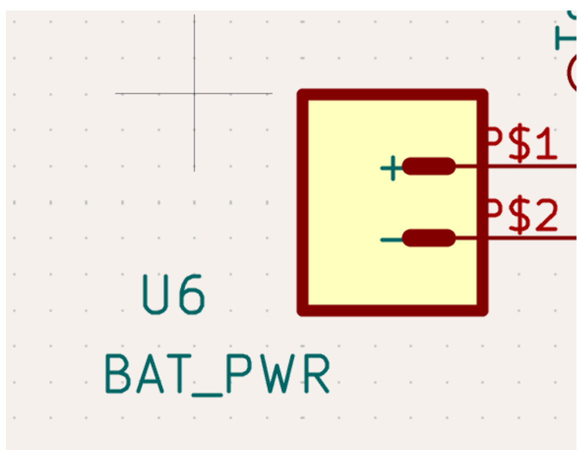


Figure 2.3.3

The battery power connection comes from the BAT_PWR XT-30 2 pin connector, and is connected to the VH of the Battery Charger IC in order to receive charge.

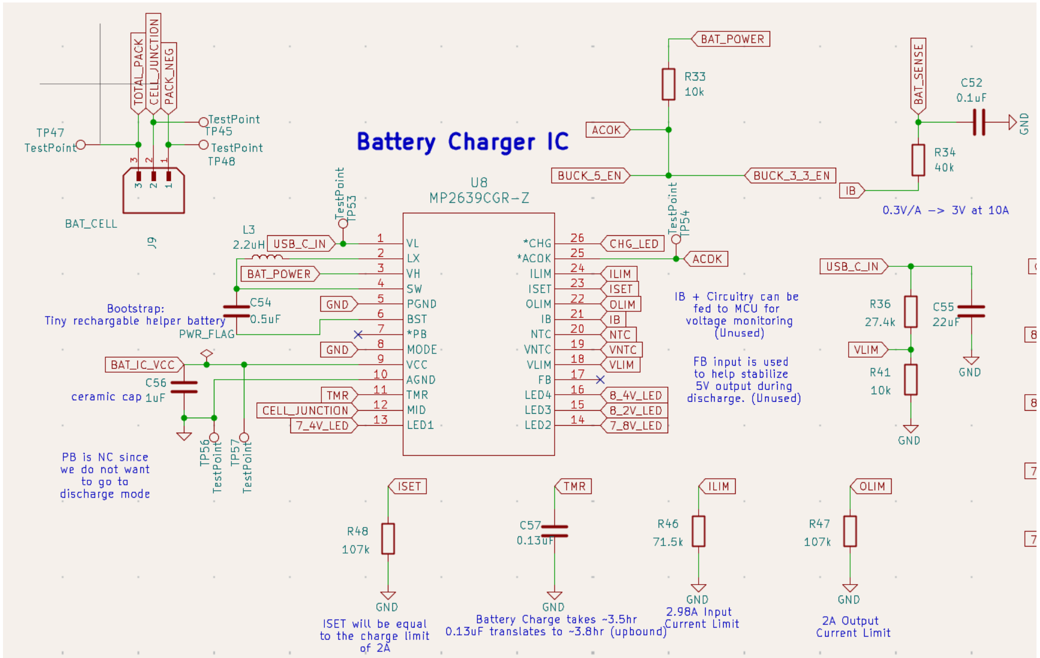


Figure 2.3.4: KiCad Schematics for Battery Charger IC

Battery Charger IC has the flexibility to accept voltage input ranging from 4V to 5.5V, the huge margin allows for ±5% tolerance from the USB Standard 5V. The MP2639C Charger IC takes CELL_JUNCTION input from the BAT_CELL JST 3 Pin connector, which is connected to MID, and this allows the Charger IC to balance the charge via internal balance MOSFET. The MP2639C also terminates charge when the battery voltage reaches within ±0.5% of 8.4V, this will prevent overheat caused by excessive charging. Charger IC supports both discharge from LH (Battery) to LV, and also LV (USB-C) to LH (Battery) via MODE. This is hard-wired to ground to always charge the Battery when the USB-C is detected. The Battery allows a maximum charge rate @ 2A, and the MP2639C Charger IC is programmed to ~1.99A of max current charge via ISET + a 107k external resistor, this is calculated using the equation from Figure 2.3.5.

$$I_{CHG} = \frac{640(k\Omega)}{3 \times R_{ISET}}(A)$$

Figure 2.3.5 MP2639C ISET Equation [38]

We also set the input current limit to 3A (~2.98A) to prevent overloading the USB-C charger, which is rated for 3A. This is calculated by the I(ILIM) equation from the datasheet as noted by Figure 2.3.6.

$$I_{ILIM} = \frac{640(k\Omega)}{3 \times R_{ILIM}}(A)$$

Figure 2.3.6 MP2639C ILIM Equation [38]

Additionally, the MP2639C Charger IC's ACOK output line, which pulls low when there is a valid USB-C power source, is wired to the enable pins for the buck converter, and this will disable 5V and 3.3V operating in charge mode. This is done intentionally because ~7-10A can be drawn during operation, which is higher than charging rate @ 2A. The user will not need to use the device during charge anyways, since the USB-C cable is wired.

In the case of battery degradation, which increases the overall charge duration, the TMR pin provides a built-in protection mechanism through an external timing capacitor. The capacitor capacitance determines the maximum allowed charging time, which helps in preventing overheating and overcharging after full charge. When the battery voltage is below V_{bat_low} at ~5.9V, or ~2.95V per cell, we enter trickle-charge mode, allowing 300mA of input charge. The trickle timer ensures that the battery cell reaches V_{bat_low} within the expected time, and if this threshold is not achieved, the charger will halt and we will know what the battery cell might be potentially damaged or is unsafe.

Specifically, battery charging speed varies depending on the capacity of said voltage range. For instance, 3.7V to 4.1V (per cell) contains 70-80% of usable capacity, while 3.3V to 3.7V contains 15%-20%, and ~10-15% from 4.1V to 4.2V.

According to the MP2639C datasheet, the efficiency for constant current and constant voltage range are as below.

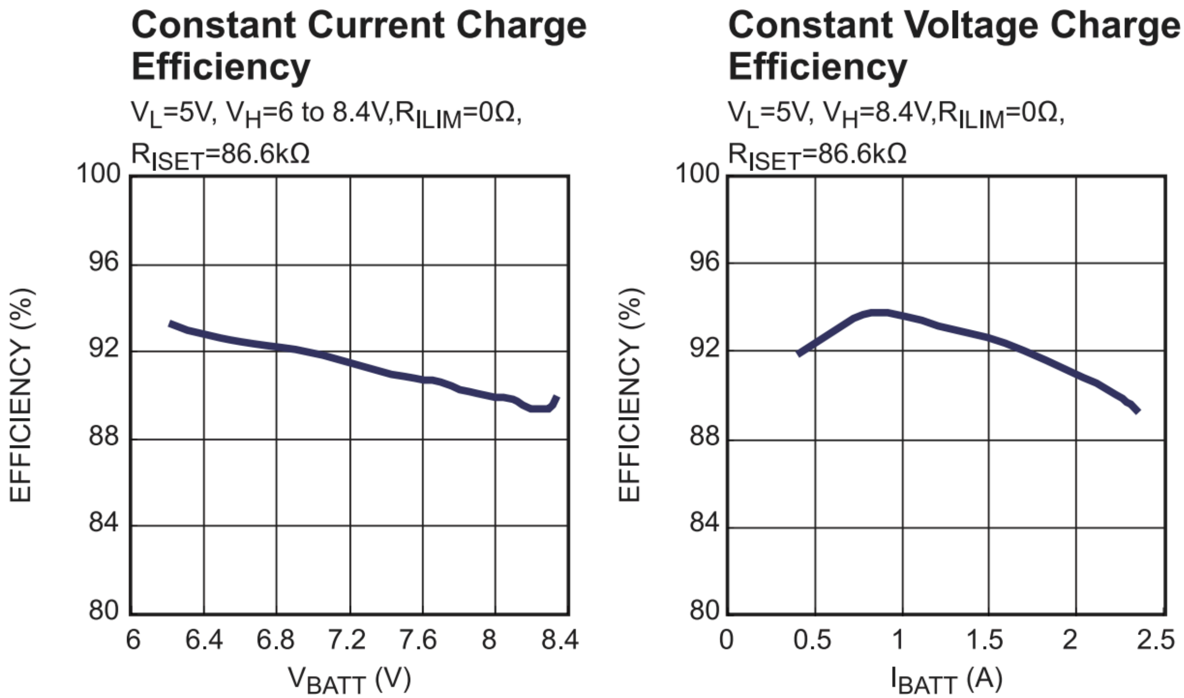


Figure 2.3.7 MP2639C Charging Efficiency [38]

Taking these into account we can calculate, the theoretical charge time for the 5000mAh battery:

- 3.3V to 3.7V (15%-20% Capacity): **Constant Current**
  - 0.1*5000mAh / (2000mA*0.89 efficiency) ~= 17 minutes
- 3.7V to 4.1V (70-80% Capacity): **Constant Current**
  - 0.8*5000mAh / (2000mA*0.89 efficiency) = 2.25 hour
- 4.1V to 4.2V (10-15% Capacity): **Constant Voltage**
  - Takes roughly 56 minutes according to 2.3.6

$$I(t) = I_0 e^{-t/\tau}$$

$$t_{CV} = \tau \ln\left(\frac{I_0}{I_{\text{term}}}\right)$$

$$I_0 = 2.0\ \text{A}$$
$$I_{\text{term}} = 0.25\ \text{A}$$
$$\tau = 0.4\ \text{h}$$

$$I_0 \text{ is the starting charge from constant current to constant voltage}$$

$$I_{\text{term}} = \text{termination current (13\% of constant current charge)}$$

$$\tau = \text{time constant, given by } \frac{2\ \text{A}}{5\ \text{Ah}} = 0.4,\ 0.4/0.89 \text{ efficiency } = 0.45\ \text{C}$$

$$t_{CV} = 0.45 \times \ln\left(\frac{2.0}{0.25}\right) = 0.45 \times 2.079 \approx 0.94 \text{ hours, which is roughly 56 minutes}$$

Figure 2.3.6 Constant Voltage Time Estimation

- Total Charge time: 3.46 Hours, or 3 hours and 28 minutes

We set an upper bound for the total charge time to at least 3.5 hours, and using the equation from Figure 2.3.7, we require a capacitance of 0.13uF, giving us 3 hours and 47 minutes of charge time.

$$\tau_{TOTAL\_TMR} = 6.05\text{Hours} \times \frac{C_{TMR}(\mu F)}{0.1\mu F} \times \frac{1A}{I_L(A) + 0.08} \qquad (4)$$

Figure 2.3.7 Total Charge Time Equation [38]

Respectively, the trickle time will be calculated using the equation from Figure 2.3.8, and it will be roughly 44 minutes.

$$\tau_{TRICKLE\_TMR} = 33.7\,\text{mins} \times \frac{C_{TMR}(\mu F)}{0.1\mu F}$$

Figure 2.3.8 Trickle Charge Time Equation [38]

Trickle Input Current: 300mA
Constant Current: 2A
Constant Voltage: 200mA (Typical: 10% of Constant Current Charge)

Trickle input current	I_{TC}			300		mA
Constant fast charge current	I_{CC}	R_{ISET} = 215kΩ	0.79	1.00	1.20	A
		R_{ISET} = 86.6kΩ	2.2	2.46	2.7	A
Termination charge current	I_{BF}	As the percentage of I_{CC}	2.5	10	17.5	%
		If 10% * I_{CC} < 167mA	38	150		mA

Figure 2.3.9 MP2639C Charge Currents per Phase [38]

Then, once the battery voltage exceeds V(BAT_LOW), we begin constant-current charge in fast charge mode, which is limited 2A as set by ISET. For the case where the battery was already above V(BAT_LOW) but does not finish charging within the expected length of time. We allow 3.5 hours of total charge time, which includes trickle, constant-current, and constant-voltage phases, and if the battery does not reach full capacity, we will stop the charge and the CHGOK LED will blink at a fixed 1Hz to indicate timer fault.

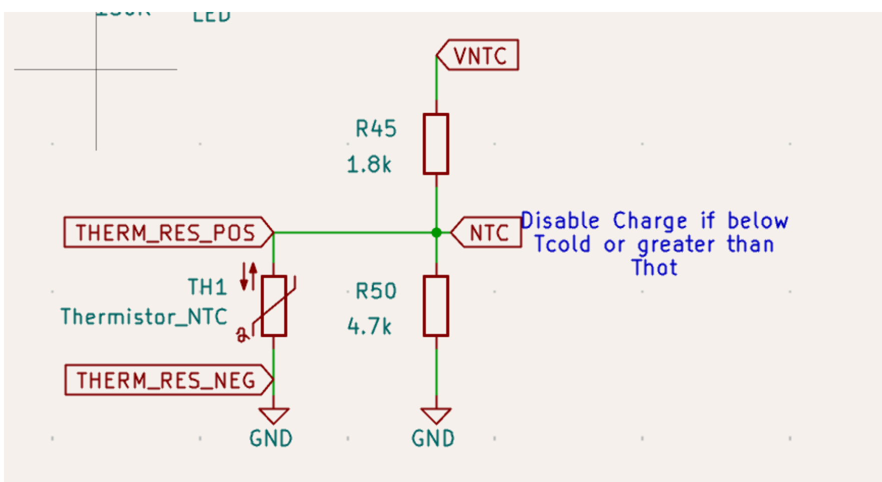


Figure 2.3.10: Temperature Sensing Schematics for Charger IC

Battery Charger IC operates from -40 degrees Celsius to 125 degrees Celsius, and along with NTC temperature sensing (Figure 2.3.10) it will stop charging when the battery is below 0 degrees Celsius or above 60 degrees Celsius, which not only manages the heat dissipation, but also lighten the power draw from the USB-C, which will help the USB-C in staying under its operating temperature max at 85 degrees Celsius. The MP2639C Battery Charger IC will also decrease the current draw when the battery temperature is beyond the nominal range.

$$V_{IB} = \frac{3 \cdot I_{BATT}}{400k} \cdot R_{IB}$$

Figure 2.3.11: IB Equation [38]

Current sensing is also employed using the IB pin of the MP2639C Charger IC. The IB pin produces a voltage proportional to the current draw of the battery, and this will help us in monitoring current for data analysis. Using the equation (Figure 2.3.11) and the recommended external resistor of 40k, we will get the equation as follows:

$$V_{IB} = 0.3 \times I_{BAT}$$

Figure 2.3.12: IB Current to IB Voltage Equation [38]

In other words, we will draw 3V for 10A of current draw from the battery, and this is the upper bound of the current draw according to the power analysis. On the other hand, the BAT_SENSE pin, goes to the microcontroller's ADC pin in order to interpret this via software, is perfect since STM32H743VIT6's ADC pins are safe to operate up to 3.3V max.

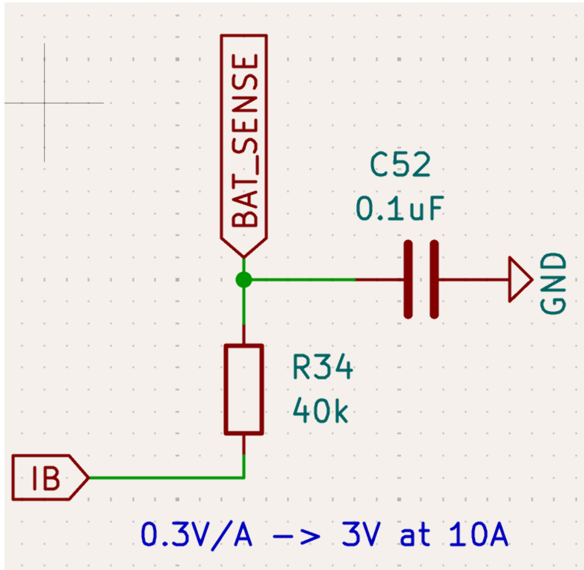


Figure 2.3.13: Current Sense

Additionally, a 0.1uF capacitor is added to form a low-pass RC filter to smoothen out the voltage at the IB pin before it is measured by the MCU's ADC (Figure 2.3.13). This is because ADCs are voltage sensitive, and the low-pass RC here prevents small switching spikes, ripple, and transient noises from the Charger IC.

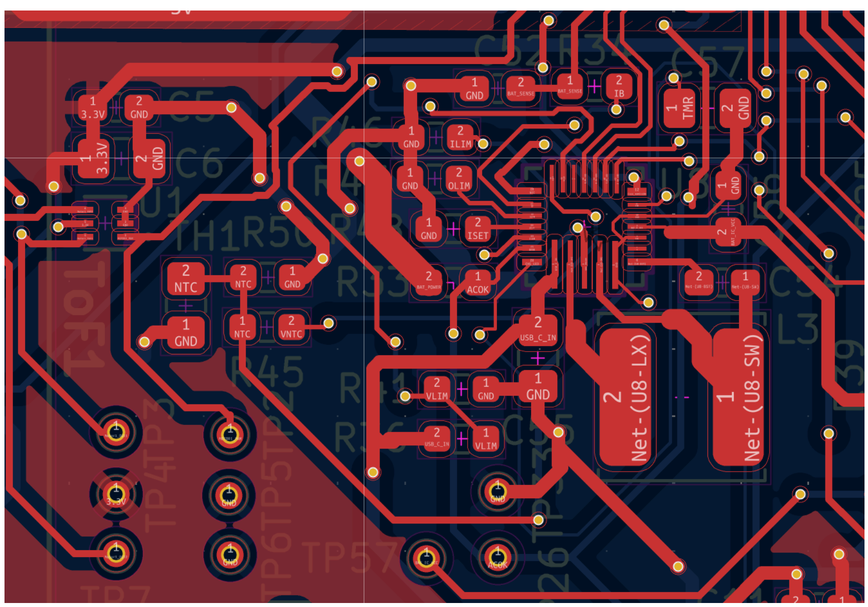


Figure 2.3.14: PCB Layout for the Battery Charger IC

Finally, in order to handle the large amounts of current going through the power lines, we utilize 30-40mil traces to route. Instead of the 1oz copper pour, we also used 3oz in order to dissipate heat better.

5V Buck Converter:

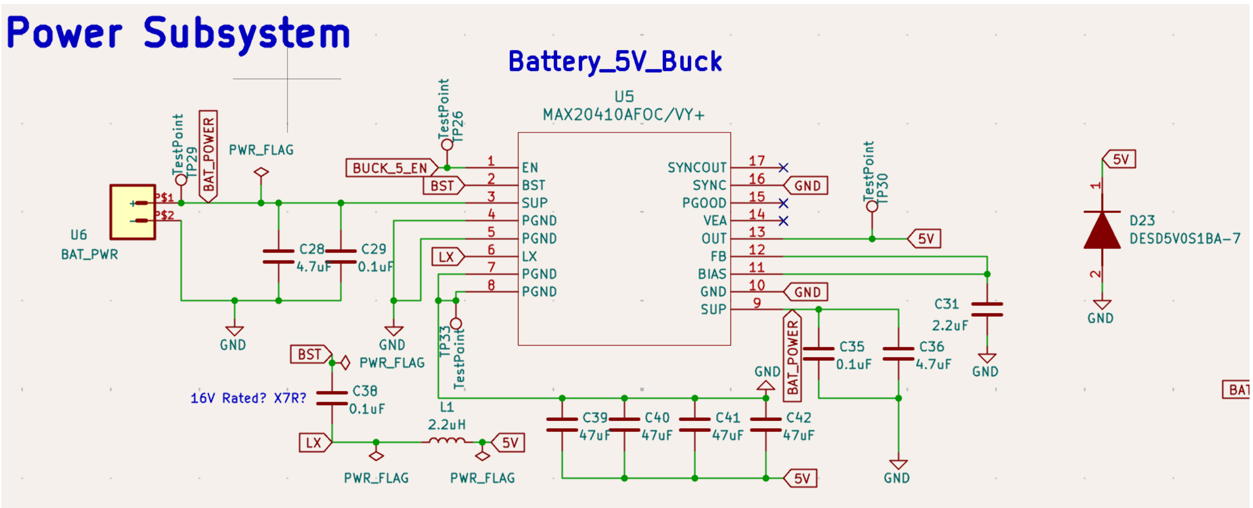


Figure 2.3.15: Kicad Schematics for MAX20410A Buck Converter

The 5V power line is used to power the Raspberry PI, and we chose the MAX20410A IC, which outputs a fixed voltage of 5V. The Buck Converter IC is highly accurate, as the output accuracy is precise such that the lowest output is 4.92V, while the max is 5.05, and both of these fit within the Raspberry PI's operating voltage (Figure 2.3.16).

$V_{OUT\_SKIP\_5V}$	V_{OUT} = 5.0V, skip mode, no load	MAX20408/ MAX20410	4.92	5	5.05
		MAX20408E/ MAX20410E	4.93	5	5.05

Figure 2.3.16: MAX IC Skip Mode Voltage Output Range [36]

The MAX20410A accepts input voltage ranging from 3V to 36V, which is perfect for our battery supplying voltage ranging from 6V to 8.4V. Additionally, the Raspberry PI 5 could draw up 25W during high CPU loads, and the MAX chip does provide up to 10A, which equates to 10*5 ~=50W of power. According to Figure 2.3.16, the efficiency of a similar model from the same line, provides up to 93% efficiency for current output around 5-10A, which perfectly fits our needs

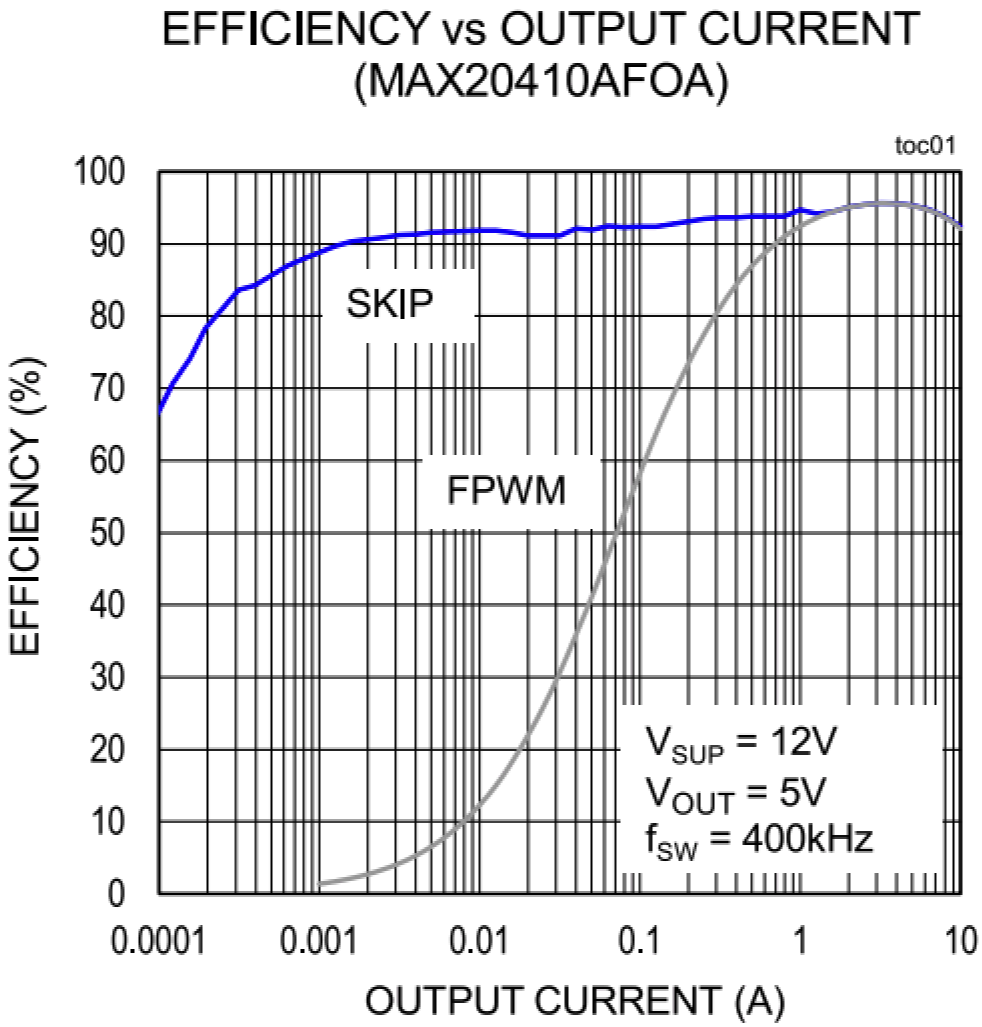


Figure 2.3.16 MAX20410A Efficiency Under Load [36]

Power loss will not be a huge issue since the chip can support up to 50W, we can effectively draw 27W that equates to 25W after power loss. In order to stabilize the current output, we employed a 2.2uH inductor on the LX and 5V output line. Using equation from Figure 2.3.17:

$$\Delta I_L = \frac{\left(V_{SUP} - V_{OUT}\right) \cdot V_{OUT}}{V_{SUP} \cdot f_{SW} \cdot L}$$

Figure 2.3.17: Inductor Ripple Current Equation [36]

Assuming the V_sup is 8.4V by worst case scenario, Vout is 5V, f_sw is 400kHz, and L is 2.2uH (as recommended by the datasheet), we will get roughly 2.3A peak to peak for the ripple current. This is roughly 23% of the buck converter at full load, and this is close to the 30% current ripple for inductor selection as recommended by analogDevice [42]. This range balances efficiency, transient response, and stability, and anything above 40% will cause more EMI than necessary, while ripple current ratio below than 15% will have a slower load transient response and the inductor is physically too large.

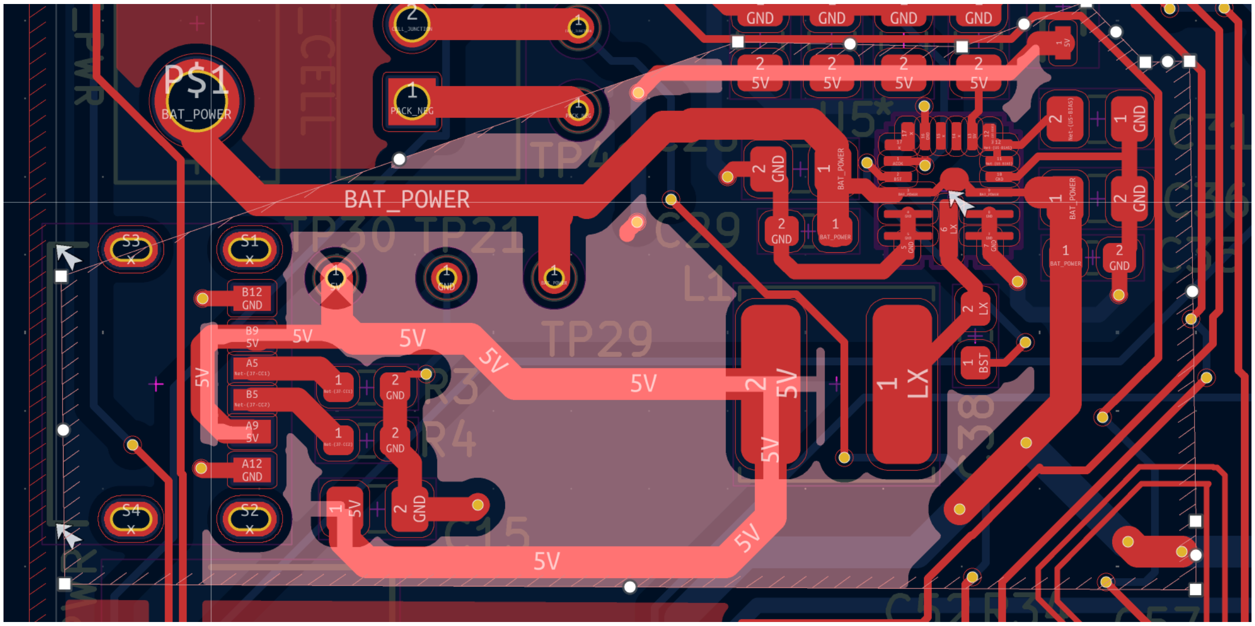


Figure 2.3.17 PCB Layout for MAX Chip

Additionally, we included a 3-oz copper area for the 5V to help with heat dissipation. Thicker traces around 30-40mil are used as well to help carry the current.

3.3V Buck Converter:

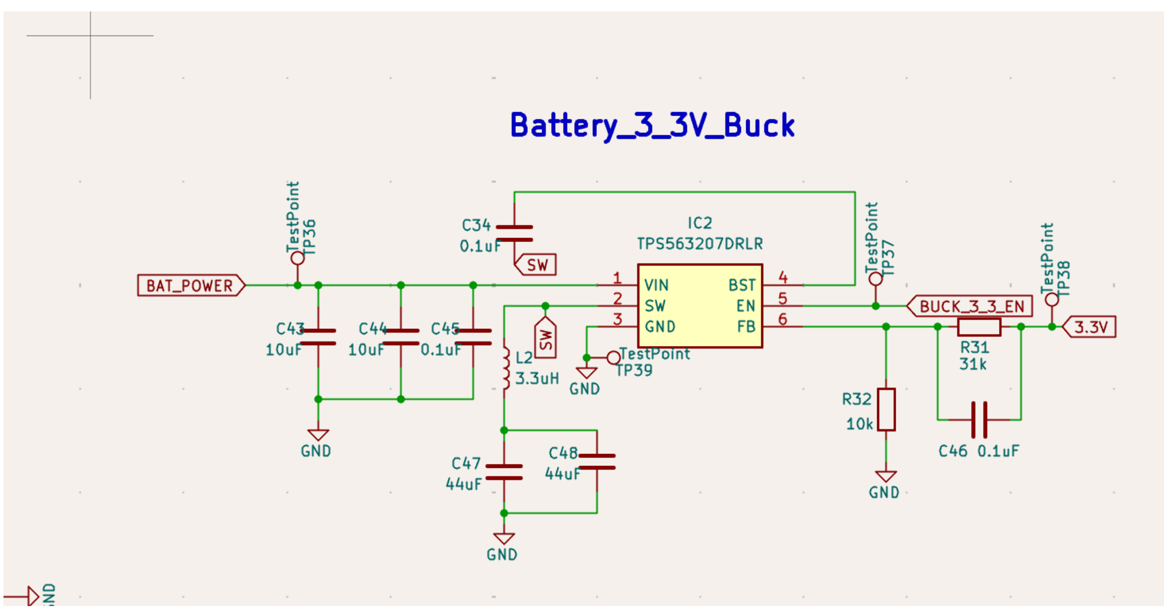


Figure 2.3.18: Kicad Schematics for TPS563207DRLR

The TPS563207DRLR supports input voltage range up to 17V, which fits our system's 6 - 8.4V battery range (Figure 2.3.19) Additionally, the power analysis above indicates that we require up to ~7-8W of power draw, and the TPS IC can provide up to 3A, which equates to 3A*3.3V = 10Watts, providing a safe upper bound for our 3.3V logic system. Note that the TPS chip has a 2% tolerance for the output voltage, which ranges from 3.234V to 3.336, which fits well within the recommended operating voltage for all of the 3.3V components as the power analysis shows.

(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

7.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
V_{IN}	Supply input voltage range	4.3		17	V

Figure 2.3.19: TPS Recommended Supply Input Voltage Range [30]

OUTPUT VOLTAGE (V)	R1 (kΩ)	R2 (kΩ)	TYP L1 (µH)	C8 + C9 (µF)			CFF(pF)
				MIN	TYP	MAX	
0.85	0.55	10.0	1.5	20	44	110	-
0.9	1.2	10.0	1.5	20	44	110	-
1	2.4	10.0	1.5	20	44	110	-
1.05	3	10.0	1.5	20	44	110	-
1.2	4.9	10.0	2.2	20	44	110	-
1.5	8.6	10.0	2.2	20	44	110	-
1.8	12.3	10.0	2.2	20	44	110	-
2.5	21	10.0	2.2	20	44	110	10-220
3.3	31	10.0	3.3	20	44	110	10-220

Figure 2.3.20: TPS Recommended Inductor and Resistors [30]

Now, because we wish to output 3.3V, we employed a voltage divider rule on the FB line as recommended by Figure 2.3.20 in the datasheet. Using the equation from Figure 2.3.21:

$$V_{out}=0.806 \times (1 + R_{FBT}/R_{FBB})$$

Figure 2.3.21: TPS Vout Voltage Divider Equation [30]

We can calculate the V_out by doing:

$$V_{OUT} = 0.806 \times \left(1 + \frac{R_{FBT}}{R_{FBB}}\right)$$

$$V_{OUT} = 0.806 \times \left(1 + \frac{31k\Omega}{10k\Omega}\right)$$

$$V_{OUT} = 0.806 \times (1 + 3.1) = 3.3046\,\mathrm{V} \approx 3.3\,\mathrm{V}$$

Figure 2.3.22: TPS V_out Calculation [30]

Efficiency:

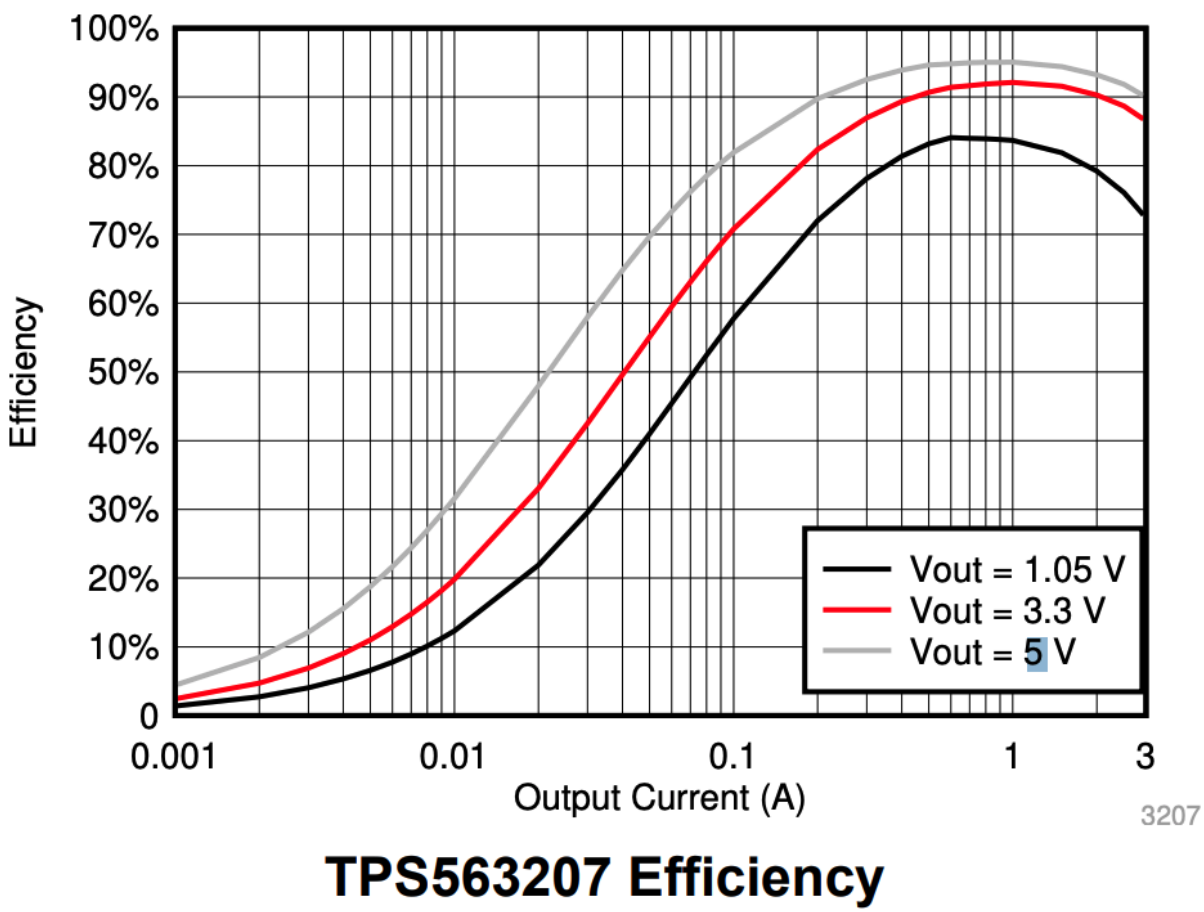


Figure 2.3.22: TPS Efficiency Graph [30]

We know that the 3.3V logic system utilizes 1-3A, and the efficiency of the TPS chip is shown in Figure 2.3.22, where it averages around 89%. This means that the converter will technically draw 11.2W from the battery and 1.2W of that will be dissipated as heat. Since we require only ~7-8W at max, the converter will only require 9W of power and dissipate 1W of that to supply 8W to the 3.3V logic system. 9W of power is 1W under the limit of the converter and this fits our needs well.

Additionally, we also require an inductor to help the TPS chip in converting from higher voltage + low current to lower voltage + higher current. As mentioned prior, analogDevice recommends ripple current that is ~30% of the max [42]. Using the Figure 2.3.23 equation and the fact that the typical switching frequency is 580kHz

$$\mathrm{II}_{P-P} = \frac{V_{OUT}}{V_{IN(MAX)}} \times \frac{V_{IN(MAX)} - V_{OUT}}{L_O \times f_{SW}}$$

Figure 2.3.23: Ripple Current Equation for TPS [30]

We calculate:

$$\Delta I_{L(P\text{-}P)} = \frac{V_{OUT}}{V_{IN(MAX)}} \times \frac{V_{IN(MAX)} - V_{OUT}}{L_O \times f_{SW}}$$

$$\Delta I_{L(P\text{-}P)} = \frac{\mathbf{3.3}}{\mathbf{8.4}} \times \frac{\mathbf{8.4 - 3.3}}{\mathbf{3.3 \times 10^{-6} \times 5.8 \times 10^{5}}} = \mathbf{1.05\ A\ (p\text{-}p)}$$

Figure 2.3.24: Ripple Current Calculation for TPS [30]

This equates to 1.05/3, or 35% of the max current output. This is roughly around the 30% ripple current recommended by analogDevice.

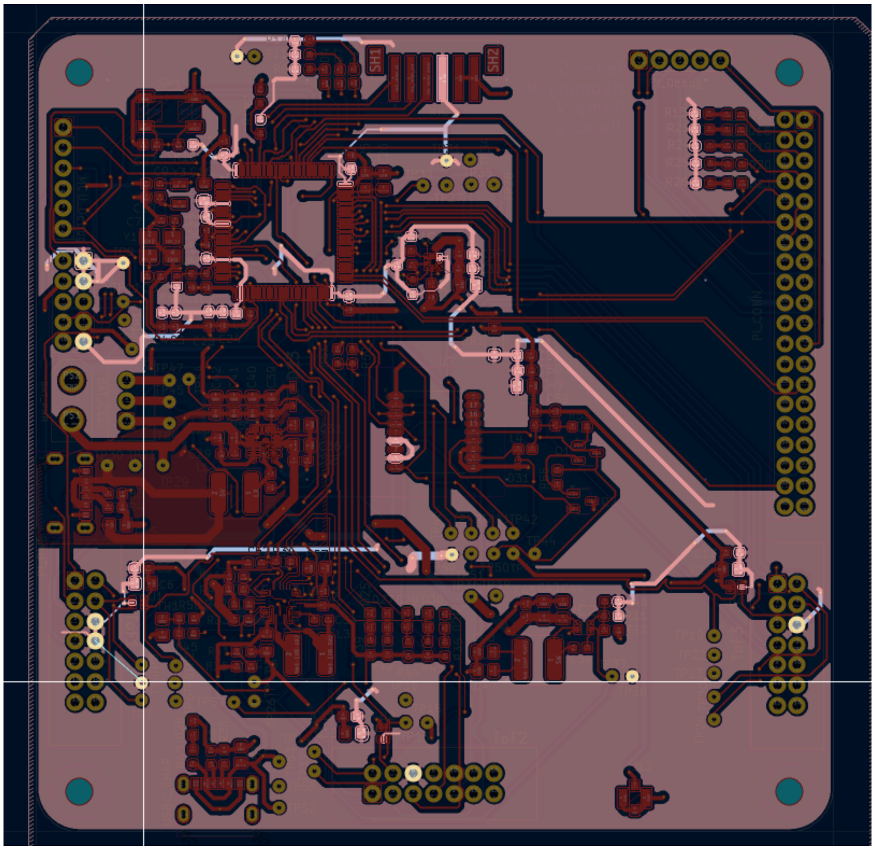


Figure 2.3.25: PCB Layout for the 3.3V Rail

For the PCB layout, we use 20-30mil traces to help carry the 3.3V current and also utilize a 3.3V copper zone to decrease routing complexity as well as helping with heat dissipation.

Overall, the converter will efficiently step down the 6-8.4V battery rail to a stable 3.3V logic supply within a reasonable tolerance along with low ripple.

- Battery: 7.4 V Battery
  - Over 2 Hour Battery Life
  - 37 Wh total / (~15W average load) ~= 2.5 hour of usage
  - 37 Wh total / (~30W peak load) ~= 1.25 hour of usage
  - Discharge termination Cutoff: ~3V per Cell x2 ~= 6V
  - Fully Charged Voltage: ~4.2V per Cell x 2 ~= 8.4V
  - 7.4V to 5V Buck Converter Input Voltage Range: ~3-36V

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage Range	V_{SUP}		3.0		36	V

Figure 2.3.26: Max20408 Voltage Range [27]

    - Aiming for at least 6V Battery Input
  - **Verification:**

        - Connect via the battery connector on PCB, and use the battery power test points on the board to verify the correct power output between 6V and 8.4V
- 7.4V-5V Buck Converter:
    - Minimum 25 Watts
    - Fixed 5V Output * 10A max ~= 50Watts
    - 85% Conversion Efficiency
    - Raspberry PI Voltage Cutoff: 4.63V(±5%)
    - Raspberry PI Voltage Upperbound: ~5.1V
        - Assumption: PI 5 official Power Supply is 5.1V
            - PI 5 can operate safety at that voltage

    - Buck Converter Output Range: (Mode Undecided)

$V_{OUT\_SKIP\_5V}$	V_{OUT} = 5.0V, skip mode, no load	MAX20408/ MAX20410	4.92	5	5.05
		MAX20408E/ MAX20410E	4.93	5	5.05
$V_{OUT\_PWM\_5V}$	V_{OUT} = 5.0V, PWM mode, no load	MAX20408/ MAX20410	4.93	5	5.06
		MAX20408E/ MAX20410E	4.95	5	5.05

Figure 2.3.27: Max20410E Chip Voltage Output Range [36]

        - Skip Mode: 4.93 ≤ 5 ≤ 5.05
        - PWM Mode: 4.95 ≤ 5 ≤ 5.05
        - Both Mode fits within the Raspberry PI's Cutoff and Upperbound
    - **Verification:**
        - Verify the Battery Output, solder the 7.4V to 5V Buck Converter
        - Use the 5V test points on the board to verify the correct power output to between 4.9 and 5.1V
- 7.4V-3.3V Buck Converter:
    - Minimum 7W
    - Converts 7.4V to 3.3V at 90% efficiency (~7W)
    - Max 3A * 3.3V ~= 10W fits our requirement
    - Microcontroller STM32H743VIT6 Operating Range: 3.0 to 3.6 (3.3V Typical)
        - Aiming to get within 3% of 3.3V (3.2-3.4V)

$V_{DD33USB}$	Standard operating voltage, USB domain	USB used	3.0	3.6
		USB not used	0	3.6

Figure 2.3.28: H750 Microcontroller Operating Range [37]

    - **Verification:**

        - Use the 3.3V test points on the board to verify the correct power output within 3% of 3.3V (3.20-3.40V)

Post-Assembly Verification:

- Idle/Active Current Draw Test:
    - Shunt-Resistor will be in series with the battery and ground
        - Measure the Voltage
        - Use Ohm's Law (V/R = I) to get the current draw

$$I = \frac{V_{shunt}}{R_{shunt}}$$

        - I * V = Power should be within 5% of the expected range

$$P = V_{battery} \times I$$

- Power Usage/Stability Over Time:
    - Triple-Channel Current/Voltage Sense (Undecided)
        - Check Voltage and Current Level
        - Monitor the values for at 15-30 minutes during typical operation
        - Save logs and feed to Python script for automated analysis/graph generation
        - Verify that all voltage bus remain stable within tolerance and that no excessive/transients current draw occur

BMS:

- Over-voltage Protection:
    - 5V Bus - Champ 7.4 to 5V Buck Converter output to ~5.25V
        - Buck Converter Absolute Max Voltage: 42V: N/A

Absolute Maximum Ratings

SUP, EN to PGND .... -0.3V to +42V
BST to LX .... -0.3V to +2.2V
BST to PGND .... -0.3V to +44V

Figure 2.3.29: Max20410E Absolute Max Voltage [27]

        - USB Standard for 5V Supplies allows for ±5% tolerance (~5.25V)
    - 3.3V Bus - Clamp the min(all components' Upper Limits): ~3.6V

- GPS: 4.3V

Power Management

Power Supply:

2.8–4.3 V

Figure 2.3.30: GPS Absolute Max Voltage [19]

- Motor Disc: 3.8V

	电机安装位置	任何位置都可以
2-5	Voltage Range for Use 工作电压范围	DC 2.5~3.8V

Figure 2.3.31 : Motor Disc Absolute Max Voltage [11]

- Motor Drivers: 5.75V

Over operating temperature range (unless otherwise noted)

		MIN	MAX	UNIT
Power supply pin voltage	VM	-0.5	12	V
Logic power supply pin voltage, DSG	VCC	-0.5	5.75	V

Figure 2.3.32: Motor Driver Absolute Max Voltages [12]

- ToF: 3.6V

Table 11. Absolute maximum ratings

Parameter	Min.	Typ.	Max.	Unit
AVDD, IOVDD	-0.5	—	3.6	V
SCL, SDA, LPn, INT, and I2C_RST	-0.5	—	3.6	

Figure 2.3.33: ToF Absolute Max Voltage [9]

- ST Microcontroller: 3.6V

$V_{DD33USB}$	Standard operating voltage, USB domain	USB used	3.0	3.6
		USB not used	0	3.6

Figure 2.3.34: H743 Microcontroller Operating Range [37]

- Magnetometer: 4.8V

Symbol	Ratings	Maximum value	Unit
Vdd	Supply voltage	-0.3 to 4.8	V
Vdd_IO	I/O pins supply voltage	-0.3 to 4.8	V

Figure 2.3.35: Magnetometer Absolute Max Voltage [21]

Figure 2.3.1.12: External DAC Absolute Max Voltage [22]

- Microphone: 3.63V

PARAMETER	RATING
Supply Voltage (V_{DD})	−0.3 V to +3.63 V
Digital Pin Input Voltage	−0.3 V to V_{DD} + 0.3 V or 3.63 V, whichever is less

Figure 2.3.36: Microphone Absolute Max Voltage [25]

- 5-3.3V Buck Converter: 19V

		MIN	MAX	UNIT
	VIN, EN	–0.3	19	V

Figure 2.3.37: TPS563207 Absolute Max Voltage [30]

- Under-voltage Protection:
  - Battery:
    - Discharge termination Cutoff: ~3V per Cell x2 ~= **6V**
  - 5V Line:
    - Raspberry PI Cutoff: **4.6V**
  - 3.3V Line:

- Charging IC: 2S Li-Ion Cells Charging
  - USB-C Charging
    - Input 3.9V to 6.2V operating
  - Battery Voltage Range: 3-4.2V
    - IC: 3.4-4.6V

PCB Design & Implementation:

To ensure that the system is not impacted by incoming transients from fast-switching and electrostatic discharge, transient voltage suppressor (TVS) diodes are placed on both the 5V and 3.3V line. It is important to highlight that Raspberry PI 5 already includes onboard ESD and surge protection from its USB-C power connection. But, it is important to put another layer of protection to avoid overloading Raspberry PI (costly) and the USB-C as well. The purpose of the TVS diode is not to continuously clamp or regulate voltage, as those are jobs of regulators and buck converters. The goal is to absorb and divert short voltage and transient spikes to reduce the impact or divert them from sensitive circuits, and keeping those spikes below damaging levels.

Definitions:

- Standoff Voltage: Maximum continuous voltage that the TVS diode can handle without conducting. When the voltage between the power lines is below this threshold, the TVS diode acts as an open circuit and it will not affect normal operation.
- Breakdown Voltages: The voltage ranges at which the diode begins to exponentially conduct to divert transient surge.
- Clamping Voltage: The maximum voltage that will be across the TVS diode as it conducts at its rated peak pulse current. For instance, a 12V spike across a 10V TVS diode will shunt some of the current to ground, and we will only see 10V across the power line.
- Peak Pulse Current: The maximum surge current that the TVS diode can handle for a short pulse of time (8/20us waveform) without damage
- Peak Pulse Power: The maximum power dissipation the diode can handle during a surge, or the amount of energy it can absorb.

5V Line Protection:

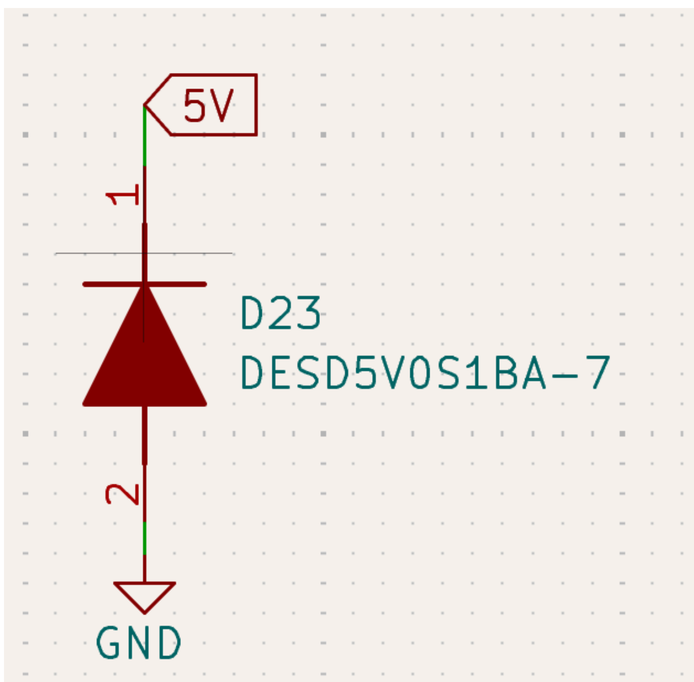


Figure 2.3.38: Kicad Schematic 5V TVS Protection

Electrical Characteristics (@T_A = +25°C, unless otherwise specified)

Characteristic	Symbol	Min	Typ	Max	Unit	Test Conditions
Reverse Standoff Voltage	V_{RWM}	—	—	5	V	—
Channel Leakage Current (Note 6)	I_{RM}	—	5	100	nA	V_{RWM} = 5V
Clamping Voltage	V_{CL}	— —	— —	10 14	V	I_{PP} = 1A, tp = 8/20µs I_{PP} = 12A, tp = 8/20µs
Breakdown Voltage	V_{BR}	5.5	—	9.5	V	I_R = 1mA
Differential Resistance	R_{DIF}	—	0.4	—	Ω	I_R = 10A, tp = 8/20µs
Channel Input Capacitance	C_T	—	35	45	pF	V_R = 0, f = 1MHz

Figure 2.3.39: DESD5V0S1BA TVS Electrical Characteristics 1 [43]

Characteristic	Symbol	Value	Unit	Conditions
Peak Pulse Power Dissipation	P_{PP}	130	W	8/20µs, per Figure 1
Peak Pulse Current	I_{PP}	12	A	8/20µs, per Figure 1

Figure 2.3.40: DESD5V0S1BA TVS Electrical Characteristics 2 [43]

- Standoff Voltage: 5V
- Breakdown Voltage: 5.5V-9.5V
- Clamping Voltage: 10V/14V
- Peak Pulse Current: 12A (8/20us)
- Peak Pulse Power: 130W

The nominal voltage for the 5V line can range from 4.92 to 5.05V, and at this point the TVS diode should be able to standoff voltage at this range. Thus, the TVS diode with 5V standoff voltage will not falsely conduct at the power supply input. If the buck converter outputs 5.05V, the TVS diode will begin to leak an insignificant amount of current, but will not conduct until 5.5V. This is chosen to provide a safe ~10% margin above the converter's maximum output of 5.05V. Since the 5V Buck Converter is rated for up to 10A, the 12A surge rating provides a safe net in case of transient events from the 5V Buck Converter. Additionally, we know that the 5V rail can deliver up to 50W (~27W in reality), and a 130W peak pulse power is well above that in order to absorb short, high-energy spikes with ease.

3.3V Input Protection:

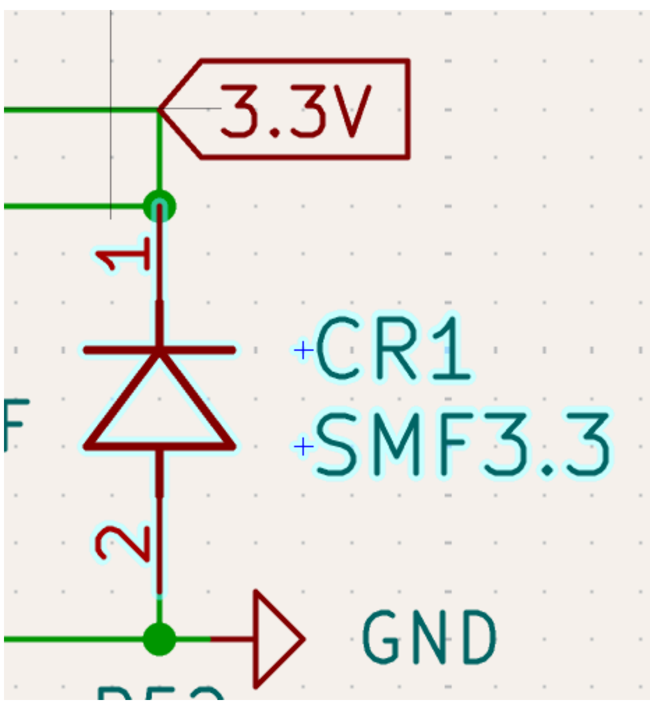


Figure 2.3.41 : Kicad Schematics 3.3V TVS Protection

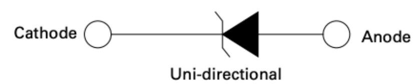
Parameter		Symbol	Value	Unit
Peak Pulse Power Dissipation at T_A=25°C (Note 1)	8/20µs (Note 2)	P_{PPM}	1200	W
	10/1000µs (Note 3)		200	W
Thermal Resistance Junction- to- Ambient		$R_{\theta JA}$	220	°C/W
Thermal Resistance Junction- to- Lead		$R_{\theta JL}$	100	°C/W
Operating Temperature Range		T_J	-55 to 150	°C
Storage Temperature Range		T_{STG}	-55 to 150	°C

Notes:
1. Non-repetitive current pulse, per Fig. 4 and derated above T_J (initial) =25°C per Fig. 3.

Applications

SMF3.3 series is ideal for the protection of portable electronics/ hard drives, notebooks, VCC busses, POS terminal, SSDs, power supplies, monitors, and vulnerable circuit used in other consumer applications.

Functional Diagram



Electrical Characteristics (T_A=25°C unless otherwise noted)

Part Number	Marking Code	Breakdown Voltage V_{BR} (Volts) @ I_T		Test Current I_T (mA)	Reverse Stand off Voltage V_R (V)	Maximum Reverse Leakage @ V_R I_R (µA)	Maximum Peak Pulse Current (10/1000µS) I_{PP} (A)	Maximum Clamping Voltage @I_{PP} (10/1000µS) V_C (V)	Maximum Peak Pulse Current (8/20µS) I_{PP} (A)	Maximum Clamping Voltage @I_{PP} (8/20µS) V_C (V)
		MIN	MAX							
SMF3.3	33	3.4	4.3	10	3.3	0.5	30.0	6.8	120.0	10.0

Notes:

Figure 2.3.42: SMF3.3 TVS Electrical Characteristics [44]

- Standoff Voltage: 3.3V
- Breakdown Voltage: 3.4-4.3V
- Clamp Voltage: 10V (8/20us)
- Peak Pulse Current: 120A (8/20usd)
- Peak Pulse Power: 1200W

For the 3.3V logic system, the buck converter can output anywhere from 3.234 to 3.336, and we required a TVS diode which can standoff around 3.3V so it does not falsely conduct. Thus we chose SMF3.3, its standoff of 3.3V will ensure that the TVS diode stays completely inactive during normal operation, with negligible leakage current with voltage output is 3.336 (max). Since all the 3.3V logic systems are only protected for larger components like buck converters, some sensors, and the microcontroller, this means that we need a breakdown voltage very close to standoff but not low enough to trigger upon normal operation. The SMF3.3 has a breakdown voltage standing at 3.4V, this helps the 3.3V logic system against surges and transients. We also know that the minimum of the maximum voltages of the 3.3V logic system is 3.6V, and this TVS diode conducts below that, which is great to prevent spikes beyond 3.6V. It is unusual that there would be 10V spikes, and the TVS diode does shunt some of the current to ground to reduce the damage in those cases. The peak pulse current is 120A, which is well beyond our requirement. Likewise, the 1200 W peak power far exceeds the expected transient energy, ensuring enough protection even during rare surge occurrence.

Overall, the TVS selection ensures that both the 5 V and 3.3 V rails remain immune to transient spikes without affecting performance.

Undervoltage Protection:

The current design does not include a dedicated external undervoltage protection circuitry. However, we plan to incorporate a resistor divider circuit, and utilize MCU's ADC to monitor the battery voltages. There will be software implementation/logic to alert the user when the battery is beyond 6V, which corresponds to the battery's termination discharge threshold. Operating below 6V can result in unstable converter operation and unreliable output. The alert will be generated and sent to the Raspberry PI, which will play a pre-recorded audio output to the headphone of the user, warning that the system needs charge, and maybe for how long.

On the other hand, the Battery Charger IC and the Buck Converters do provide some form of internal undervoltage protection.

<u>MP2639C Battery Charger:</u>
Charging is disabled when the USB-C input is below 3.6V, preventing unstable operation or reverse current issues. Charging will resume once the input voltage rises above 3.9V [38]. Additionally, MP2639C will also reduce the current draw in order to maintain the input voltage, which also helps with undervoltage.

<u>MAX20410AFOC/VY+:</u>
MAX20410A will stop operating if Vin is below 2.6V. While this protects the converter itself, these thresholds is well below the system's operating range and therefore provide no functional benefit at the battery level [36].

<u>TPS563207DRLR:</u>
TPS563207DRLR will shut down if Vin is below 3.6V. While this protects the converter itself, this threshold is well below the system's operating range and therefore provides no functional benefit at the battery level [30].

Reverse-Current Protection:
Power comes from the battery, and there is only one way the JST 3-pin and the XT30 connectors can be connected. Thus, reverse-connection risk is negligible and additional reverse-current protection is unnecessary.

2.4 Sensing Subsystem

<u>Description and Purpose:</u> The sensing subsystem is an integral part of the navigation vest system. The subsystem will gather location, direction information, audio from the GPS , magnetometer, and microphone. The raw information from the GPS and magnetometer will be passed to the STM32 microcontroller to be data processed, and finally passed into the Raspberry Pi that is running the Google Maps API. Similar to the above, the ToF sensor will monitor

distance and pass the draw distance output to the STM32 microcontroller. The ToF's input angles are as follows:

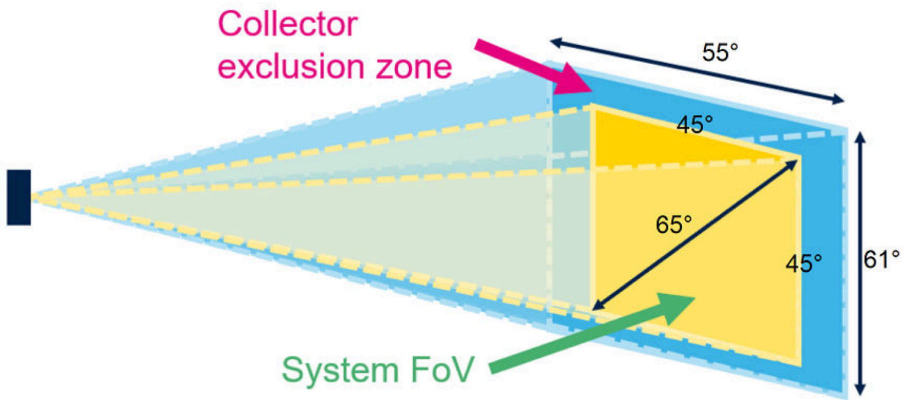


Figure 2.4.1: ToF Sensor Field of View [9]

On the other hand, the H743VIT6 microcontroller communicates to a I2S microphone, and performs digital signal processing before handing that information to Raspberry PI.

PCB Design & Implementation:

Time of Flight Sensors:

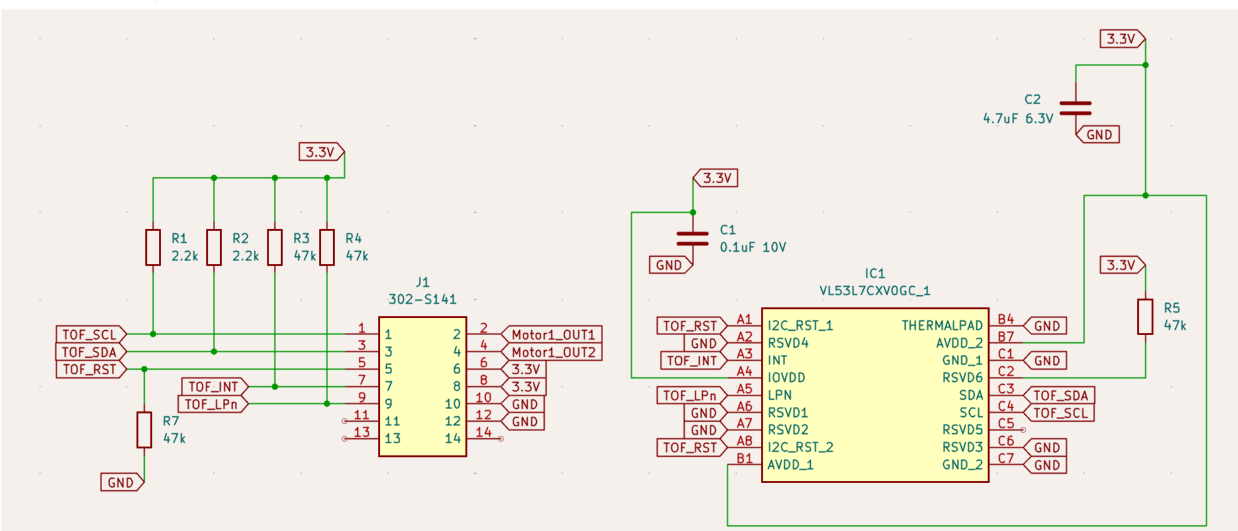


Figure 2.4.2: Kicad Schematics for the ToF Sensors

The Time of Flight sensors will be on 3 separate PCBs than the main, and they are positioned to gather a full view of objects in front. The VL53L7CX Tof sensor chip communicates with the microcontroller via the I2C interface and includes control pins for I2C reset (TOF_RST), low-power mode (LPn), and an interrupt output (TOF_INT). The setup involves routing a cable from the main PCB to each of the ToF PCBs. According to the datasheet [45], the VL53L7CX operates at 3.3V, and we supplied that power from the 3.3V Buck Converter to the PCB. Each

VL53L7CX also includes decoupling capacitors (0.1uF and 4.7 uF) across the power pins to filter switching noises and help stabilize the internal analog and digital domains.

According to the datasheet [45], the I2C protocol required pull-up resistors for the SCL and SDA lines. This is important as the I2C protocol registers signal based on falling and rising edges, and its default level is high. The TOF_RST and the TOF_LPn is driven by the microcontroller on the main PCB, and it allows us to control the ToF sensor in initialization, sleep/active mode, and etc. TOF_RST (active-high) is pulled down by default, while TOF_LPn is pulled up to enable I2C operation at power-on The interrupt pin is driven by the ToF sensors, and it connects to an external interrupt pin on the MCU, notifying the microcontroller when new distance data is ready.

Distance & Data Propagation Concerns:
Going from the front requires less cable length, and we can have lengths of < 20cm. However, going from the back requires at least 30cm on the side ones, and 50cm for the middle. To minimize wiring complexity and signal degradation, the main PCB was therefore positioned at the front, reducing overall cable length. The uxcell IDC 14-Pin Female-to-Female 30 cm ribbon cable was selected for this purpose, as it provides sufficient reach while maintaining good signal integrity.

t_{rCL}	rise time of SCLH signal	output rise time (current-source enabled) with an external pull-up current source of 3 mA			
		capacitive load from 10 pF to 100 pF	10	40	ns
		capacitive load of 400 pF[3]	20	80	ns
t_{fCL}	fall time of SCLH signal	output fall time (current-source enabled) with an external pull-up current source of 3 mA			
		capacitive load from 10 pF to 100 pF	10	40	ns
		capacitive load of 400 pF[3]	20	80	ns
t_{fDA}	fall time of SDAH signal	capacitive load from 10 pF to 100 pF	10	80	ns
		capacitive load of 400 pF[3]	20	160	ns

Figure 2.4.3: NPX I2C Guide [46]

However, longer I2C cables introduce bus capacitance, which slows the signal rise time (low level to high level) and affects communication, especially for I2C. According to NPX, I2C bus specification should remain below 400pF to ensure proper operation at 400kHz [46]. Higher bus capacitance slows down the rise and fall time, which makes the edge too slow and data can be misread. A 30 cm ribbon cable contributes approximately 30 pF / ft × 1 ft = 30 pF. PCB traces (~ 6 in) add 1–3 pF /in ~= 12 pF, while connectors and devices contribute ≈ 10 pF. Thus, total capacitance ≈ 52 pF would be far below 400 pF, ensuring reliable 400 kHz I2C operation between the main PCB and the ToF + Motor Driver PCB.

Connector of Choice:
The PCB houses both the motor driver and the ToF sensor, which requires 12 pins in total. An affordable option is the 302-S141, which are connectors with 14 positions (2.54mm). We would need at least 6 of these, and pricing was extremely important for this pick.

Magnetometer:

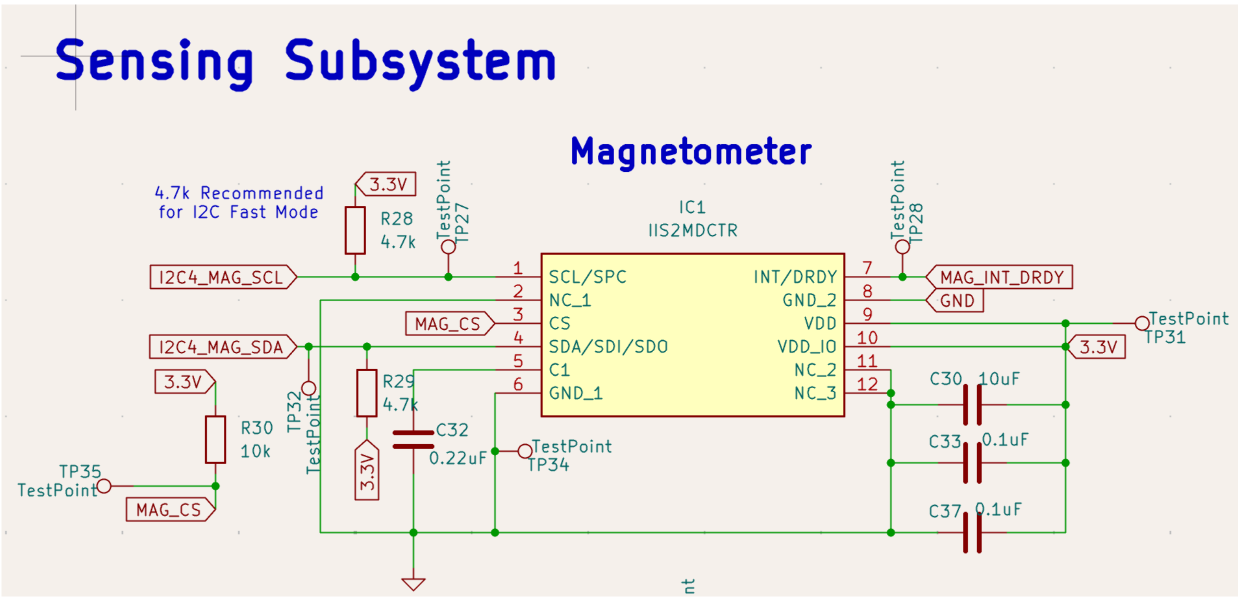


Figure 2.4.4: Kicad Schematics for the Magnetometer IIS2MDCTR

The magnetometer is positioned near the PCB's geometric center to ensure orientation is measured as close to the absolute reference point as possible. The magnetometer has two modes of communication, SPI or I2C. We chose I2C over SPI because communicating with the IIS2MDCTR magnetometer requires usage of half-duplex SPI master, which is more complex than I2C. As the datasheets recommended, we used a 4.7k pull-up resistor to set the I2C communication in Fast-Mode (400kHz operation), which will require the bus capacitance to be below 400pF like for the ToF. However, the magnetometer is within ~2-3 inches of the microcontroller, since it's on the main PCB. Thus, the signal propagation concern is negligible for the magnetometer design. Additionally, the MAG_CS line has to be pulled up to enable I2C for the magnetometer IIS2MDCTR, and a 10k pull up resistor is used for this purpose. The MAG_INT_DRDY pin is connected to one of the MCU's external interrupt pins, signaling when new magnetic field data is available for processing

Overall, the magnetometer setup ensures fast and reliable orientation sensing with minimal propagation delay and low bus capacitance.

GPS:

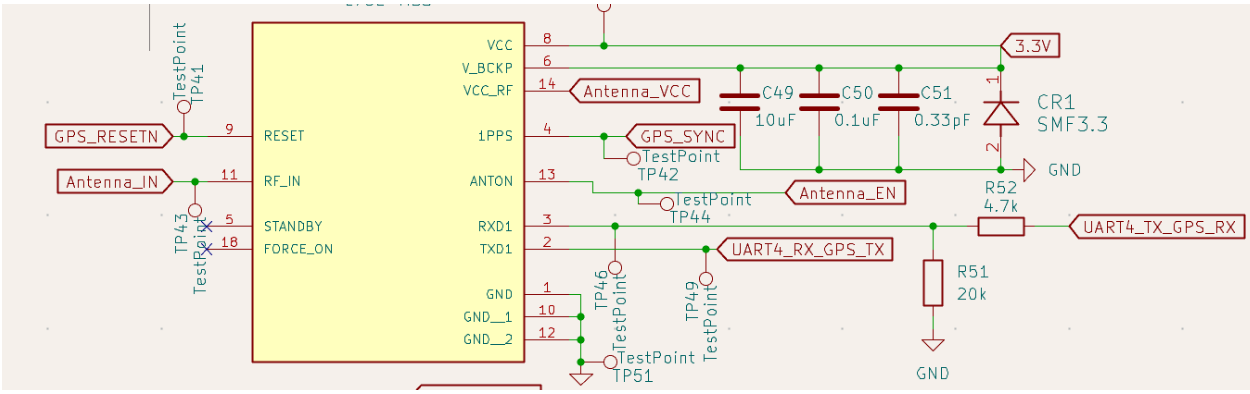


Figure 2.4.5: Kicad Schematics for the GPS (L76L - M33)

The GPS is placed in the exact center of the main PCB to ensure the most accurate position data. The operation voltage for the L76L - M33 GPS is 3.3V, however the I/O operating typical is 2.8V. We used a resistor divider system to drop the microcontroller input from 3.3V to roughly ~2.81V, ensuring the GPS I/O inputs are within the safe range. On the other hand, the microphone does recognize 2.8 as a high output, thus the GPS's Tx line does not require any resistor divider to transmit data to the microcontroller. Additionally the GPS chip also requires an external GPS antenna to acquire data from the satellite.

GPS Antenna:

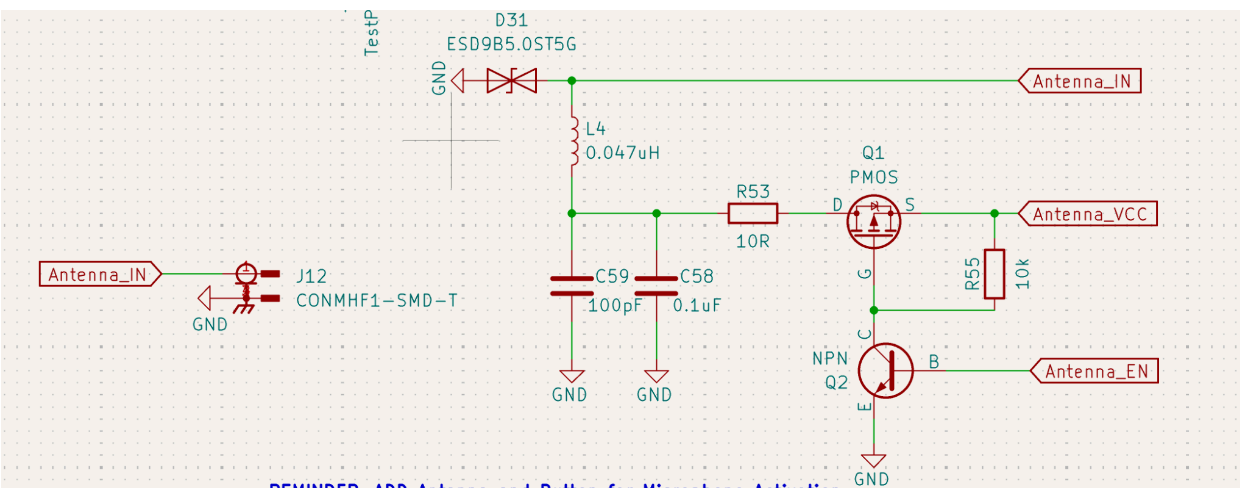


Figure 2.4.6: Kicad Schematics for the GPS Antenna

Table 5: Recommended Antenna Specifications

Antenna Type	Specifications
Passive Antenna	Frequency Range: 1559–1609 MHz Polarization: RHCP VSWR: < 2 (Typ.) Passive Antenna Gain: > 0 dBi
Active Antenna	Frequency Range: 1559–1609 MHz Polarization: RHCP VSWR: < 2 (Typ.) Passive Antenna Gain: > 0 dBi Active Antenna Noise Figure: < 1.5 dB Active Antenna Total Gain: < 18 dB

Figure 2.4.7: GPS L76L Antenna Requirements

We can choose between passive or active antennas. After looking through numerous options, a common pattern is identified for the passive antenna. The passive antenna tends to lack the wide range necessary for this GPS, while active antennas have been able to collect data from a much larger frequency range.

According to the datasheet and (Figure 2.4.7), we know that the antenna we found have to meet these requirements:

- Impedance matching: 50 Ohms
- Active Antenna Noise Figure: < 1.5dB
- Active Antenna Total Gain: < 18dB
- Polarization: RHCP
- VSWR < 2 @ Typical
- Frequency Range: 1.559 Ghz- 1.609Ghz

The specification of the PE51212 Active GPS antenna fits most of the requirements as indicated above. According to the datasheet:

		Port 1
Antenna	Frequencies (MHz)	1575.42-1602
	Polarization	RHCP
	Gain	5 dBic
	VSWR	<2.0:1
	Impedance	50 Ω
	Axial Ratio	
LNA	Gain	28±2 dB
	Noise Figure	<1.5
	Filter Insertion Loss	<3 dB
	Ex-band Attenuation	12dB@CF+50MHz/16dB@CF-50MHz
	Supply Voltage	2.2~5 VDC
	Current Consumption	5~15 mA

Figure 2.4.8: PE51212 Specification

- Frequency Match: 1575.42Ghz - 1602Ghz is a subset of 1.559 Ghz- 1.609Ghz
- Polarization: RHCP (same)
- Impedance Matching: 50 Ohms
- VSWR: Match
- Noise Figure: Match (<1.5dB)
- Total Gain: 28-1 dB (cable loss) -1dB (UMCX/U.FL connector)~= 26dB (PE51212 has 8dB more)

Microphone

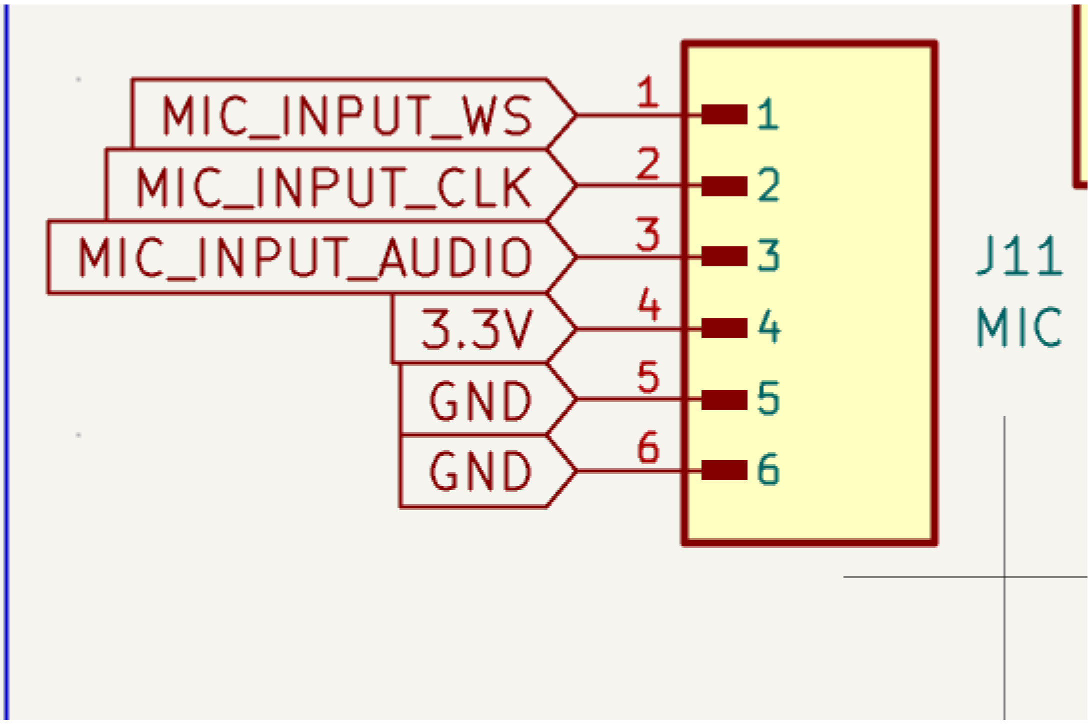


Figure 2.4.9: KiCad Schematics for JST Connector to Microphone

JST 6 PIN is used since the 5pin variation was hard to find and 6 pin does support all the I2S, power, and ground connection we need between the microphone and the main PCB.

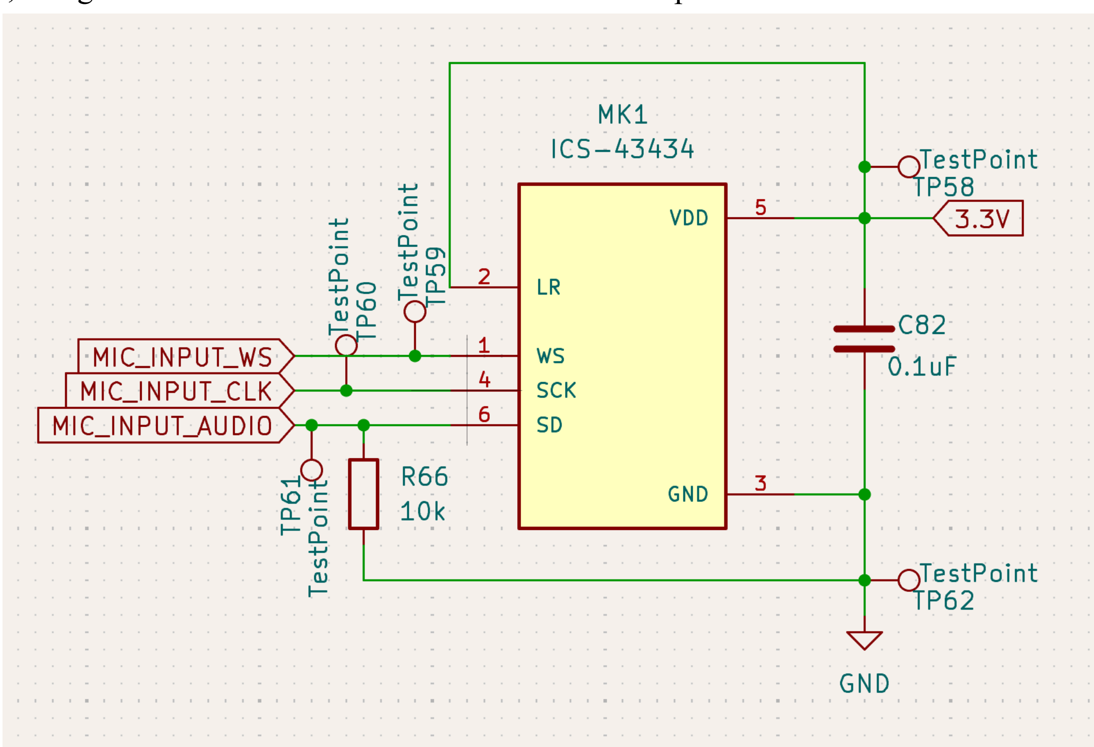


Figure 2.4.10: Kicad Schematics for Microphone ICS-43434

As shown in Figure 2.4.10, the microphone subsystem uses the ICS-43434 digital MEMS microphone, which communicates with the microcontroller via the I²S interface. The VDD pin is powered from the 3.3V rail and also decoupled with 0.1uF to do noise suppression. The I2S connection goes from the microcontroller to the microphone as the microcontroller is the one that initializes audio data transmission requests.

An I²S microphone was chosen instead of an analog microphone to eliminate the need for an external ADC and reduce MCU processing load. The MCU activates I²S streaming only when the push-button input is pressed, minimizing power consumption and computational overhead. This significantly reduces the load of the microcontroller as audio processing requires heavy load on the microcontroller. The LR pin determines whether the microphone outputs left or right channel data in a stereo configuration. Since we are not using a dual microphone, we can select it to be left or right by either grounding it, or making it high. I chose to connect LR to 3.3V so it's the right channel by default.

This configuration allows the system to have high-quality audio that will be filtered by the microphone's DSP, and decreases the need to process audio all the time, giving the microcontroller more time for the navigation/sensor fusion algorithm/data collection.

Requirements;

- ToF: Monitors Obstacle distance within its field of view and passes measurements to the ST Microcontroller
  - Monitor obstacle distance within its field of view from 0.1m to at least 4m
    - Accuracy ±5cm for distances ≦ 2m
    - Accuracy ±10cm for distances >2m
  - Data passes to ST microcontroller at minimum update rate of 10Hz
  - **Verification:**
    - Place/walk near objects at distances (0.1m, 0.5m, 1m, 2m, 4m) and record ToF readings
      - Confirm all readings fall within tolerance and updates occur at > 10Hz
      - Sweep object across sensor's FOV and confirm detection coverage to its specs
- GPS: Provide latitude and longitude for global positions and passes them to the ST Microcontroller
  - Must Provide latitude and longitude coordinate
    - Horizontal accuracy of ±5m outdoors
    - Update Rates of 1Hz at minimum
  - Cold Start must be ≦ 60 seconds

- **Verification:**
    - Compare GPS coordinate outdoors to Iphone location for at 5 minutes
        - Iphone: Use Compass Logging App
        - Log values and compare differences by passing it to Python script
            - Horizontal accuracy of ±5m outdoors
        - Update Rate of 1Hz can verified by doing: Total Time/Number of Samples
        - Repeat 2-3 times
    - Confirm that Cold Start is within 60 seconds using Iphone's timer/MCU Code
        - Repeat 2-3 times

- Magnetometer: Provides orientation in degrees and passes measurements to the ST Microcontroller
    - Provide orientation readings in degrees relative to magnetic north with resolution of ≦ 1 degrees and accuracy within 5 degrees after calibration
    - Data passed to STM32 microcontroller at 100 kHz minimum
    - **Verification:**
        - Compare magnetometer readings with iphone compass for multiple orientation
            - Iphone: Compass Logging App
            - Log Magnetometer reading via ST Code
            - Pass magnetometer and Iphone readings through Python script and check resolution and ensure accuracy within 5 degrees
            - Repeat 3-5 times

2.5 Microcontroller Subsystem

Description and Purpose: The microcontroller subsystem acts as the core of our navigation vest system. Most of our core decision algorithm is carried out inside of this subsystem. It essentially takes direct inputs from our sensing subsystem and sends commands to our sensory output system based on the sensing outputs.

PCB Design & Implementation:
STM32H743VIT6 Microcontroller:

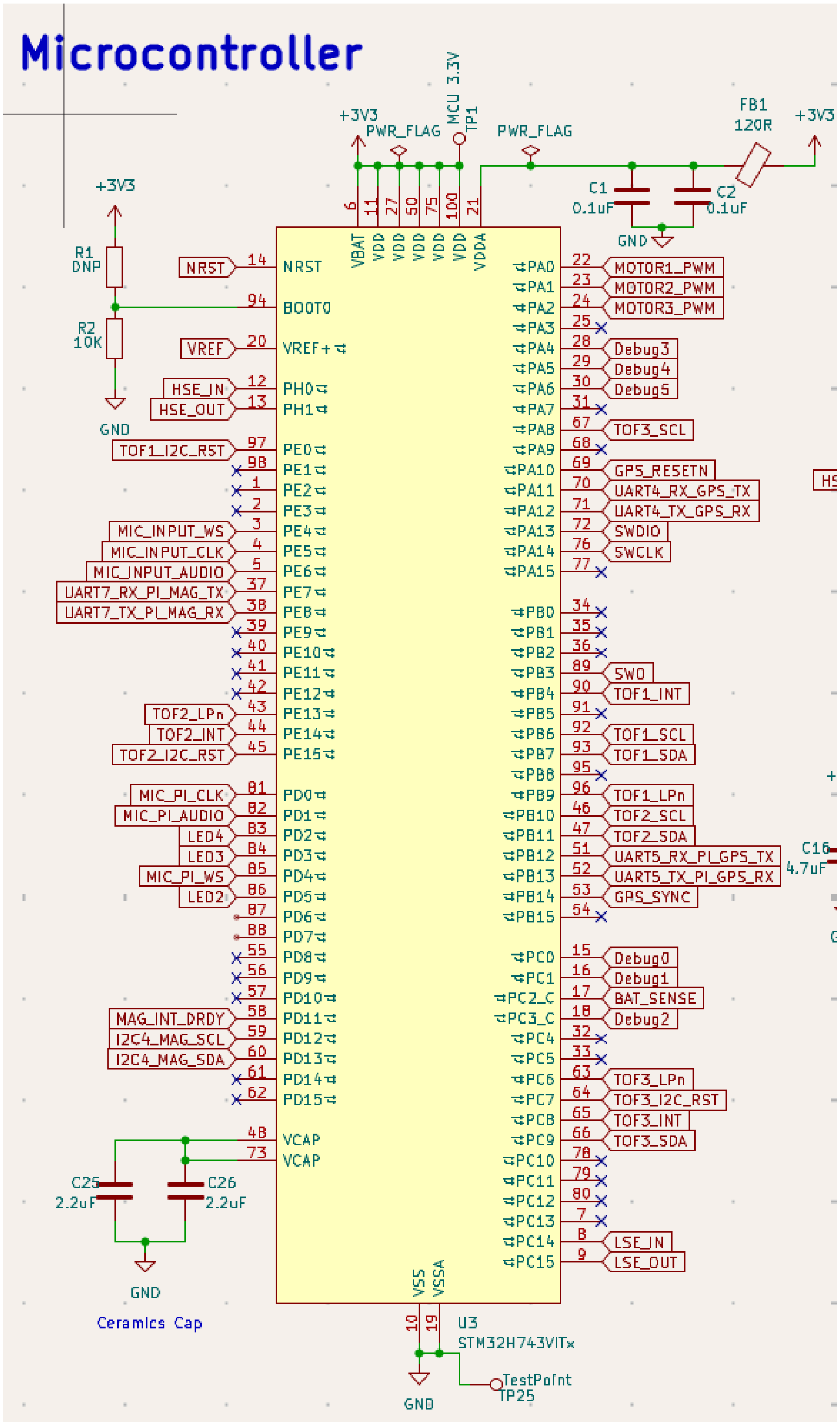


Figure 2.5.1: Kicad Schematics for the STM32H743VIT6 Chip

The STM32H743VIT6 is the main processing unit for the navigation system, and it manages all the communication between the sensors, including ToF sensors, GPS, magnetometer, and microphone. It is also responsible for processing the audio received from the microphone, and generating corresponding PWM signals to the vibration motor based on the distance to the obstacle. For these purposes, a 480Mhz high-speed, M7 core, and a large on-chip memory of roughly 864kB is chosen to perform real time sensor fusion and navigation tasks.

Each VDD pin is paired with a 0.1uF ceramic decoupling capacitor placed close to the pin to suppress high frequency noises. There are also bulk capacitors (2.2uF, 4.7uF, and 10uF) to stabilize transients and surges. On the other hand, the VDDA pin powers the microcontroller's internal analog peripherals like ADC and DAC. To reduce the switching noises generated from the digital 3.3V line, we employ 0.1uF decoupling and a ferrite bead to provide a cleaner, and stabilized VDDA line for the analog peripheral of the microcontroller.

The microcontroller subsystem originally used the 64-pin STMH7B0 package but it lacked sufficient pins for debugging and status LEDs, and thus a similar package, the H743VIT6, is chosen to accommodate these signals.

Programming and Debugging:

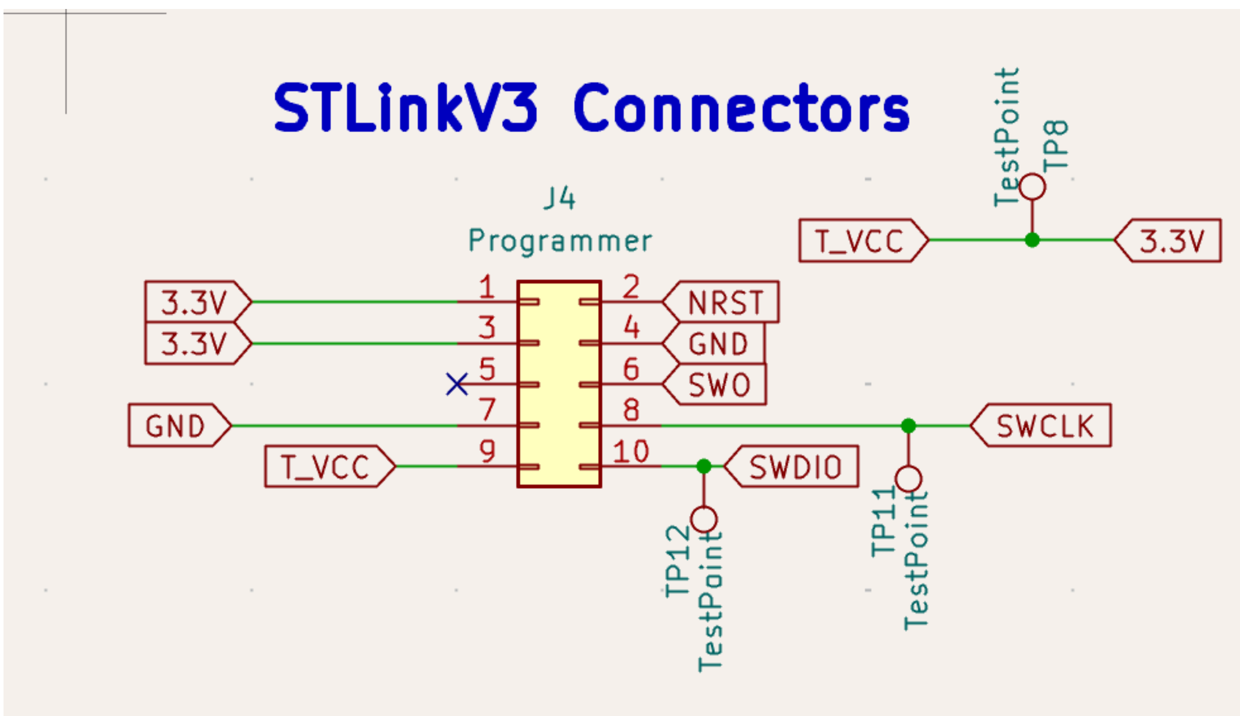


Figure 2.5.2: Kicad Schematics for STLINKV3 Programmer/Debugger

The microcontroller uses the SWDIO and SWCLK lines for programming and debugging via ST-LINKv3s. An SWO, or Serial Wire Output, pin is also added to allow printf serial output for a more convenient debugging/diagnostics.

Main Pseudocode for MCU

```
-----------------------------------MCU Code------------------------------------------

* all the output from the raspberry pi to the MCU will be treated as interrupt and there will be specific interrupt service routine for such.

main_function:

  set left motor <- 0
  set right motor <- 0
  set forward motor <- 0



  while(1): # while the user keeps walking in random direction

     depth_array_left = TOF_1.read() # depth Array is in size 8 by 8
     depth_array_right = TOF_2.read() # depth Array is in size 8 by 8

     depth_array_total = stitch_depth_array(depth_array_left, depth_array_right) # stitch the array to make full 8 by 16 map.

     right_most_map, mid_map, left_most_map = partition_map(depth_array_total) # will return three separate unique maps for right, forward, and left.

     # Prepare for the worst case scenario
     max_left = max(left_map)
     max_right = max(right_map)
     max_mid = max(mid_map)

     motor_left_output = output_pwm_intensity(max_left)
     motor_right_output = output_pwm_intensity(max_right)
     motor_mid_output = output_pwm_intensity(max_mid)


  return 0;
```

FIGURE 2.5.x MCU Pseudocode

Other than taking inputs from the sensing subsystem, we will also implement an interrupt-based communication with the Raspberry Pi, which will communicate instructions such as the path to the correct destination from the Google Maps API, as well as the output from the object detection model. Both of these responses will be translated into a speech output from the speakers connected to the microcontroller.

Requirements;

- Microcontroller should be able to support all communication protocols required by all of components from Sensing and Sensory Output simultaneous
  - **Verification:**
    - Unit Test Each Component from Sensing and Sensory Output

- Test Sensing Subsystem Altogether
- Test Sensory Output Subsystem Altogether
- Verify that Sensing and Sensory Output Subsystem can operate at the same time

2.6 Computer Vision / Navigation Subsystem

Description and Purpose: The computer vision and the navigation subsystem allows the user to be aware of surrounding important objects. For the object detection system, the camera will take an input around 30fps rate and pass the video input to the model running on Raspberry Pi. After the video is processed, it will output a list of objects or signs that it detected. We will then choose the top 10 detected objects or signs, then filter one last time down to the list of objects or signs that are in the important list. The important list includes lists of objects and signs both indoors and outdoors that matters the most when making navigation decisions. This way, we will have a method in which the speaker doesn't speak out the entire list of all surrounding objects that are not relevant to making important decisions during navigation. For the Google Maps API, once the button is pressed to speak into the microphone, the algorithm will fetch the destination and utilize real time GPS status from the MCU, allowing it to guide the user to the set destination. For the speaker & microphone systems, detailed layout is written under requirements below.

PCB Design & Implementation:

MCU to PI Connector:

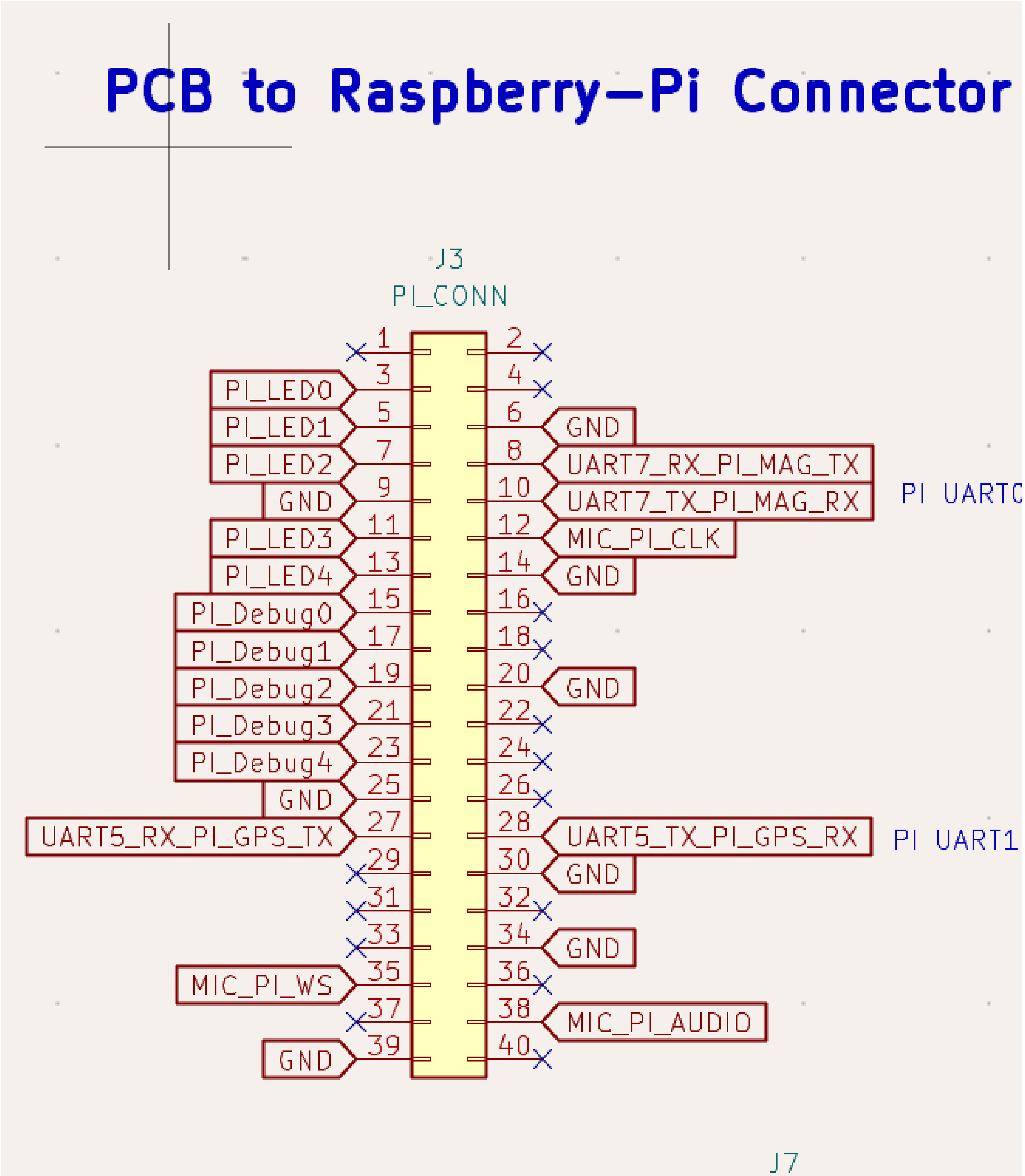


Figure 2.6.1: Kicad Schematics for MCU to PI Connector

Since the orientation and location data is fetched and processed using the microcontroller of the main PCB, we need to communicate this information to the Raspberry PI. Since the Raspberry PI has a 40-pin header, we will also implement a 40-pin header to connect all of the GPIO pins from PI to the main PCB. This connector routes UARTs, debug outputs, microphone audio, and also

LED outputs. The reason we require two separate UART is to use one for GPS's location data, and another for magnetometer's orientation data. This makes it easier to communicate each sensor's data instead of communicating everything through a single UART peripheral. Decoding and encoding for 1 UART for 2 sensors can be challenging.

In terms of debugging, we include debug lines that goes to female connectors on the main PCB so we can use logic analyzer probing on those pins as necessary. There are also 5 pins for LED visual status indicating when we debug from the Raspberry PI. This is useful because Raspberry PI itself does not provide any sort of debugging pins or LED.

USB-C PI Power:

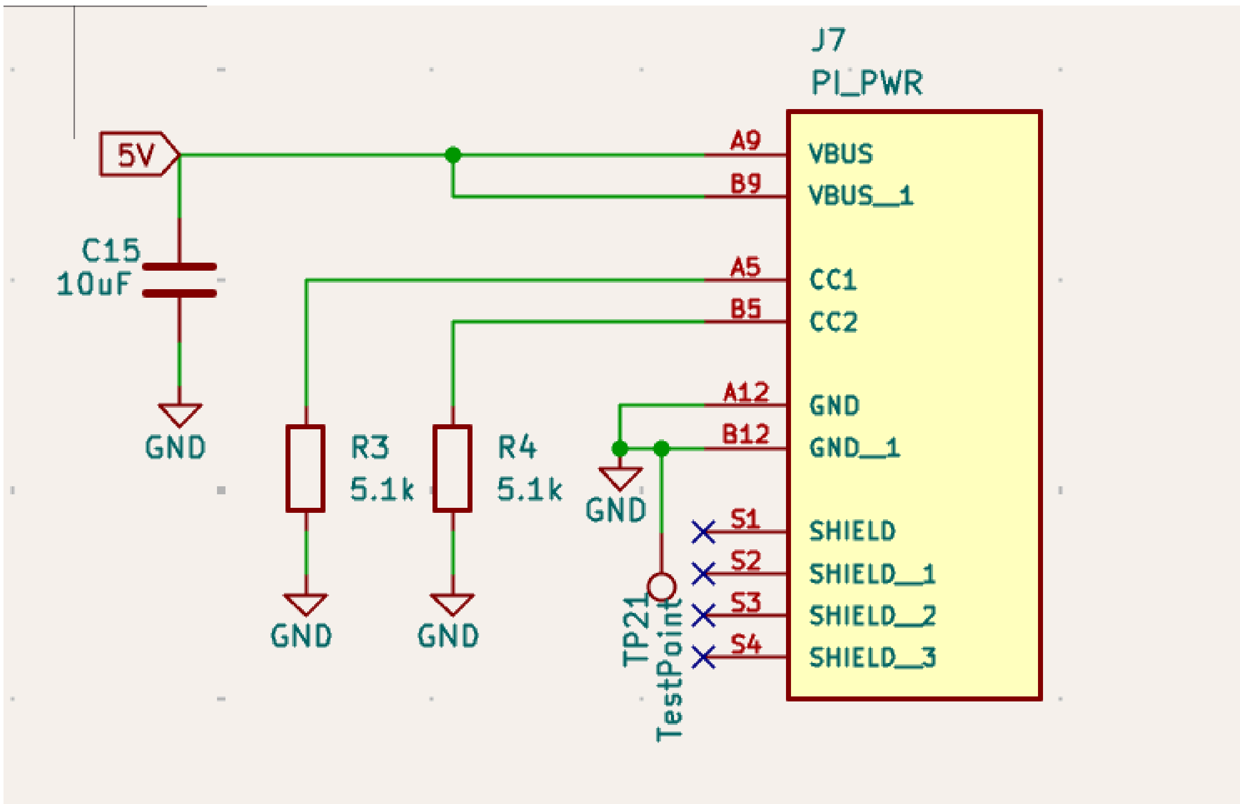


Figure 2.6.2: Kicad Schematics for the USB-C PI Power Source

The Raspberry PI 5 will be powered through this USB-C power connector. This connector takes the 5V output from the 5V Buck Converter of the main PCB, and delivers it to the Raspberry PI. Originally, power was intended to be delivered via GPIO pins, however this approach will not be sufficient since pins can't handle high current and it also bypasses Raspberry PI's onboard protection. By using the USB-C output from the microcontroller to the USB-C input of the Raspberry PI 5, we fully utilize Raspberry PI's over-current, surge, ESD, reverse-voltage, and more protections.

Requirements:

Parts:

- Raspberry Pi
    - Must provide sufficient memory to store and retrieve navigation data and audio prompts (at least 32GB)
    - RAM: (at least 4GB)
        - Capable of running Google Maps API and Computer Vision inference at a minimum rate of 20 FPS
    - Must support a camera input with resolution of at least 8 MP for reliable object detection
    - Has the necessary to support all communication, interrupts, and data transmissions

[32]

- Pi Camera Requirement
    - Compatible with Raspberry PI 4 and 5
    - Resolution greater than 8 megapixels
    - Must maintain minimum frame rate of 25FPS during operation

[33]

- Speaker & Microphone
    - The core layout for the audio system is utilizing I2S protocol to communicate between the audio modules and Raspberry Pi. Within the Raspberry Pi lies algorithm to convert from I2S readable format to text as well as conversion to one way another. This will allow a full experience of speech to text conversion and text to speech conversion.
    - The system will be tested with the following test suite:
        - I2S Signal Integrity and Clock Synchronization
        - Microphone Module Functionality
        - Speaker/Amplifier Module Functionality
        - STT Basic Accuracy with Clear Speech
        - STT Performance with Background Noise
        - STT Language with Vocabulary Coverage
        - TTS Basic Accuracy
        - TTS Performance with Microphone on
        - Full Loopback Latency (latency of TTS/STT)
        - Error Handling and Fall Back
- Google Map API (Alternative Map Apis will be also used to test) & GPS
    - The geocoding accuracy should be at least above 75% accuracy based on the equation: (Number of correctly geocoded address)/(Total Address) * 100. Our

approach is to create a test suite of 20 diverse addresses around the town of Champaign-Urbana.

    - The reverse geocoding accuracy should be at least above 75% accuracy based on the equation: (Number of correct & precise address)/(Total Tests) * 100. For the test suite, we will be taking 20+ coordinate pairs and checking if the returned address is correct and precise.
    - For both geocoding latency and the reverse geocoding latency should be less than or equal to 100ms response threshold. We will be testing 4 main categories when it comes to the latency measurement. We are planning to write a timeout latency test for re-routing, initial-routing, next maneuver updates, and geocoding with GPS. Specifically we are targeting map apis that consistently hits less than or equal to 100ms response time.
- Object Detection Model
    - Less than 5 billion parameters. The majority of the successful models such as YOLO V11 for Raspberry Pi have as little as 1.5 million parameters.
    - Object Detection Model should be able to detect the following: All the street signs and crosswalk lights that relate to pedestrians, Car, Bicycle, Motorcycle, Door, Chair, Elevator, Desk, and Stairs. 1
    - Ensure that the detection model fetches at least 70% mAP for critical labels mentioned in above bullet point.
    - Provide strict ML model evaluation based on *[35]* metrics.

2.7 Sensory Output Subsystem

Description: The sensory output is responsible for providing audio and vibration feedback to the user. Audio feedback includes three categories:

1. Direction prompts (e.g., turn left, adjust orientation)
2. Object detection alerts (e.g., warning of obstacles)
3. Destination Confirmation (arrival notification)

PCB Design & Implementation:

Motor Driver:

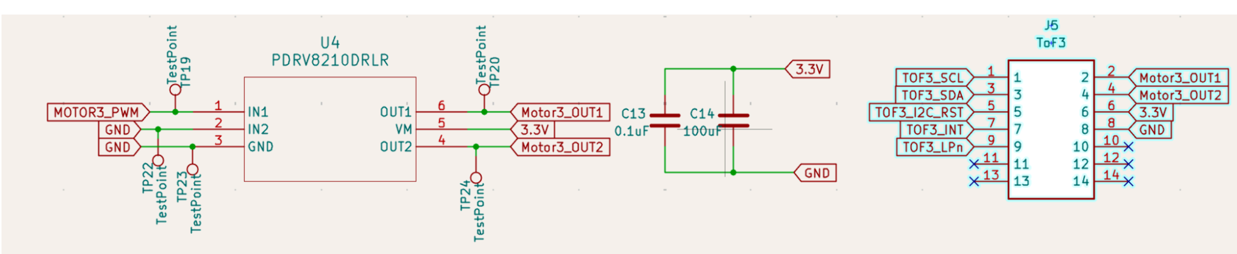


Figure 2.7.1 Kicad Schematics for the Motor Drivers

The motor driver, DRV82100DRLR, is a compact 11V H-bridge driver, which supports vibration motors that require 3.3V and 250mA peak current draw. Both of which are supported by the DV82100DRLR motor driver as it can take and supply 11V towards the motor at a peak of 1.76A. IN1 receives a PWM signal from the microcontroller, which controls the vibration strength through the duty cycle. IN2 is connected to ground, and this pin controls the direction of the motor rotation (forward or backward). For the vibration motor, direction does not matter so it is ok to connect IN2 to 3.3V or ground.

Table 8-3. PWM control mode with automatic sleep

IN1	IN2	OUT1	OUT2	DESCRIPTION
0	0	Hi-Z	Hi-Z	Coast (H-bridge Hi-Z)/ low-power automatic sleep mode
0	1	L	H	Reverse (OUT2 → OUT1)
1	0	H	L	Forward (OUT1 → OUT2)
1	1	L	L	Brake (low-side slow decay)

Figure 2.7.2 DRV82100DRLR PWM Control Table [12]

As the truth table (Figure 2.7.2) suggested, we will be switching from coast and forward since IN2 is low and IN1 is the only one that is changing (PWM). This makes sense because the motor should turn on when we are at the active part of the duty cycle (level-high at 1), and then it should coast off, and not stop immediately afterwards (level-low at 0). Decoupling of 0.1uF + 100uF bulk is placed in between the 3.3V and ground for each of the motor drivers to handle transient and suppress high-frequency noises. As mentioned above, the ToF sensor and Vibration Motor connector are on a separate PCB than the main one, thus we required a connector to transmit both of those signals to the external PCB. Refer to the ToF sensors section for a detailed explanation of the connector selection.

The audio feedback starts from Raspberry PI, converted to I2S through Text to Speech, and into I2S. The I2S is fed into an External Digital to Analog Converter, amplified, and played on a speaker.

At the same time, haptic signals are generated by the MCU as PWM, which works with the object detection alerts from Raspberry PI. The microcontroller decides how strong, and when to start/continue/stop vibrating each of the 3 haptic motor discs. The PWM signal is fed to the Motor Driver and it outputs to the speaker/haptic motor discs.
fetching the output from the MCU via I2S (for audio) and SPI (for control) protocols. This data is converted into analog audio via a DAC and amplified, while haptic data is translated into PWM and other motor drive signals.

Requirements;
Audio Feedback

- Must generate more than **30 distinct audio alert** (direction, object detection, destination)
- Must output at a minimum **70dB at 1 meter** for audibility in outdoor environment
- Latency from MCU trigger to audible playback must be **within 200ms**
- Parts:
    - External DAC [22]
    - Amplifier [23]
    - Speaker [24]
- **Verification:**
    - Measure Audio dB with Phone
    - Calculate and Log the Latency via Software from the
    - Configure MCU to output a GPIO trigger pulse when an audio alert is sent over I^2S
        - Connect oscilloscope channel 1 to this GPIO
        - Connect Channel 2 to the speaker's output (via amplifier or speaker
        - Measure the time difference between the GPIO trigger and the start of the audio waveform
        - Repeat for 5 representative audio alerts.

Haptic Feedback

- Must actuate the **3 vibration discs** individually via PWM control
- Must produce vibration force relative to the distance from obstacle detected
    - **At least 6 different vibration force**
- Response latency from **detection event** to haptic actuation must be **within 200ms**
- Parts:
    - Haptic Motor Driver [12]
    - Haptic Motor Discs [11]
- **Verification:**
    - Trigger each motor individually and simultaneous
    - Output 6 different PWM duty cycles from the microcontroller to each of the disc
    - Configure MCU to output a GPIO trigger pulse when an obstacle is detected
        - Connect oscilloscope channel 1 to this GPIO
        - Channel 2 to the motor driver's PWM input
        - Measure time difference between detection event and start of the first PWM signal

2.8 Software Design

2.8.1 Raspberry PI:

a. YOLO Object Detection

For the YOLO Object Detection model, there is multiple quantitative analysis that comes along to check how the model stands on performance. First thing in model quantitative analysis is to check the number of parameters:

"Number of Parameters"

Code Ran:

```
model = YOLO("yolo_post_train_outdoor.pt")
get_num_param = sum(p.numel() for p in model.parameters())
print(f'Number of Yolo Parameters: {get_num_param}')
```

Output:

```
Number of Yolo Parameters: 2624080
```

"Model Structure"

```
YOLO(
  (model): DetectionModel(
    (model): Sequential(
      (0): Conv(
        (conv): Conv2d(3, 16, kernel_size=(3, 3), stride=(2,
2), padding=(1, 1), bias=False)
        (bn): BatchNorm2d(16, eps=0.001, momentum=0.03,
affine=True, track_running_stats=True)
        (act): SiLU(inplace=True)
      )
      (1): Conv(
        (conv): Conv2d(16, 32, kernel_size=(3, 3), stride=(2,
2), padding=(1, 1), bias=False)
        (bn): BatchNorm2d(32, eps=0.001, momentum=0.03,
affine=True, track_running_stats=True)
        (act): SiLU(inplace=True)
      )
```

```
…… Truncated ………

              (1): Conv(
                (conv): Conv2d(80, 80, kernel_size=(1, 1),
stride=(1, 1), bias=False)
                (bn): BatchNorm2d(80, eps=0.001, momentum=0.03,
affine=True, track_running_stats=True)
                (act): SiLU(inplace=True)
              )
            )
            (2): Conv2d(80, 80, kernel_size=(1, 1), stride=(1,
1))
          )
        )
        (dfl): DFL(
          (conv): Conv2d(16, 1, kernel_size=(1, 1), stride=(1,
1), bias=False)
        )
      )
    )
  )
)
```

From the above code, we can observe that the Yolo model that we are using has around 2.6M numbers of parameters. That shows that the model is very small and efficient to be processed in raspberry pi. By calculation, 1M parameters equals to around 1MB of memory requirement with full 32FP. Since our Raspberry Pi has 16GB of memory, it is more than enough to fit the model with ease. The model structure is also simple, with sequence of convolutional layers as well as upsample layers, which can be serviced easier later if we need to upgrade the model for specific usage. More information about real time memory usage will be tested in the next portion of the design document.

There are multiple tests and analysis that we can consider when evaluating the model, especially on device models that will be constantly run on Raspberry Pi. We can consider the following model analysis question and tests:

(Model Analysis Criteria, Questions, and Steps):

Energy Consumption Per Minute

- What is the average battery consumption per minute during model forward pass?

a) Ensure that the battery is fully charged that powers the Raspberry Pi.
b) Before model initialization, ensure that the surrounding is clear with minimal object presence. Once the testing environment is chosen, make sure to maintain the testing environment for all the model energy consumption measurements.
c) Close all the running programs in Raspberry Pi. Record the Voltage of the battery before the model execution.
d) Execute the model for 1 minute.
e) Record the voltage of the battery after. Find the difference between the voltage readings before and after.

(Result):

Model List	Starting Average Voltage (V)	Ending Average Voltage after 1 minute (V)	Difference (V)
RPI YOLO11n Model	7.65	7.47	0.18

(Comment on the outcome):

We believe that the outcome of the test is positive. 0.18V/minute drop indicates that the model is efficient with power during the forward pass. Assuming standard average current usage of 1.8A and internal resistance of 0.1 ohms, with typical 7.2 Ah capacity, it can last around 4 hours of runtime.

Inference Speed + Reaction Time

- How does the accuracy vary based on the motion of the object? Objects with high mobility contribute to false positive / false negative cases. For instance, car is an object with high mobility, it is important that the model has high reaction time to update the user with current status of the environment with high accuracy

a) Ensure that the battery is fully charged that powers the Raspberry Pi.
b) Close all the running programs in Raspberry Pi. Execute the model.
c) Build a toy car setup (it can be anything that can move).
d) While the car is moving based on constant speed, record if the model detected the object of choice.

(Result):

Model List	Speed	Status	Accuracy
RPI YOLO11n Model	Slow Walk(0.8–1.8 m/s)	✅	80% mAP
RPI YOLO11n Model	Normal Walk(1.8–3.5 m/s)	✅	65% mAP
RPI YOLO11n Model	Fast Walk(3.5–4.5 m/s)	❌	0% mAP Failed

(Comment on the outcome): We believe that the outcome reflects a good indication about our model speed. The average human speed is around 1m/s ~ 2m/s, which the model can handle. More importantly, we also expect visually impaired individuals to walk slower, allowing the model to process with higher accuracy. Using quantization and the training loop, we hope to provide faster response time.

Real Time Memory Usage:

- What is the average memory consumption during model forward pass?

a) Ensure that the battery is fully charged that powers the Raspberry Pi.
b) Close all the running programs in Raspberry Pi. Execute the model.
c) Run the model for a minute. Type "htop" in the terminal. Monitor the usage of Ram and Swap.
   i) To do such, I need to execute the following command on the terminal: "vmstat 1 60 > /tmp/vmstat_output.txt". Such allows to pipe out the average usage to the output.txt file.
   ii) For all the models create a table based on the output.txt output statistics.

(Result):

Model List	Pre-Model Run Utilization (Average)	Post-Model Run Utilization (Average)
RPI YOLO11n Model	510MB/16GB	826MB/16GB

(Comment on the outcome): It is a perfect outcome. The model itself only used 5.04% of memory, even with constant forward passes.

Accuracy / mAP Score in Normal Lighting Conditions and Paused Motion

- Based on the import list of labels, what is the accuracy measurement under normal lighting conditions and Paused Motion?
- Important list includes: [Car, Bicycle, Motorcycle, Door, Chair, Elevator, Desk, Stairs, Pedestrian Street Signals]
- Specific List of Pedestrian Street Signals: [Walk Signal, Flashing Don’t Walk Signal, Steady Don’t Walk Signal, Pedestrian Countdown Timer, Accessible Pedestrian Signal, Pedestrian Hybrid Beacon]

a) Ensure that the battery is fully charged that powers the Raspberry Pi.
b) Close all the running programs in Raspberry Pi. Execute the model.
c) Scan all the objects listed above and check if it detects or not.
d) Calculate the corresponding accuracy calculation as: (# of object detected)/(#of total objects tested) * 100

(Result):

Model List	Label Name	Corresponding Accuracy	Detected?
RPI YOLO11n Model	Car	92% mAP	✅
RPI YOLO11n Model	Bicycle	93% mAP	✅
RPI YOLO11n Model	Motorcycle	88% mAP	✅
RPI YOLO11n Model	Door	0% mAP	❌
RPI YOLO11n Model	Chair	90% mAP	✅
RPI YOLO11n Model	Elevator	0% mAP	❌

RPI YOLO11n Model	Desk	0% mAP	❌
RPI YOLO11n Model	Stairs	0% mAP	❌
RPI YOLO11n Model	Pedestrian Stop Sign	95% mAP	✅
RPI YOLO11n Model	Pedetrain Traffic Light	90% mAP	✅
RPI YOLO11n Model	Zebra Crossing	0% mAP	❌

(Comment on the outcome): We believe that there are some more improvements that can be made. One good news is that for the objects that are trained showed more than 70% above mAP, but the objects that are not trained showed 0% mAP. During the course of the semester, we will be actively design the training loop to ensure that it meets our set criteria for mAP, as well as test more on the uncurated datasets to reflect the real world usage.

Cold Start Time:

- How long does it take for the model to start working when the subsystem is booted?

a) Ensure that the battery is fully charged that powers the Raspberry Pi.
b) Close all the running programs in Raspberry Pi. Execute the model.
c) Type model execution command. Set the clock to zero. Initiate the clock and the model at the same time.
d) As soon as the model is fully executed (terminal output shows or picamera2 software gets initiated), stop the timer and get the difference.

(Result):

Model List	**Average Cold Time Start**
RPI YOLO11n Model	3.76s

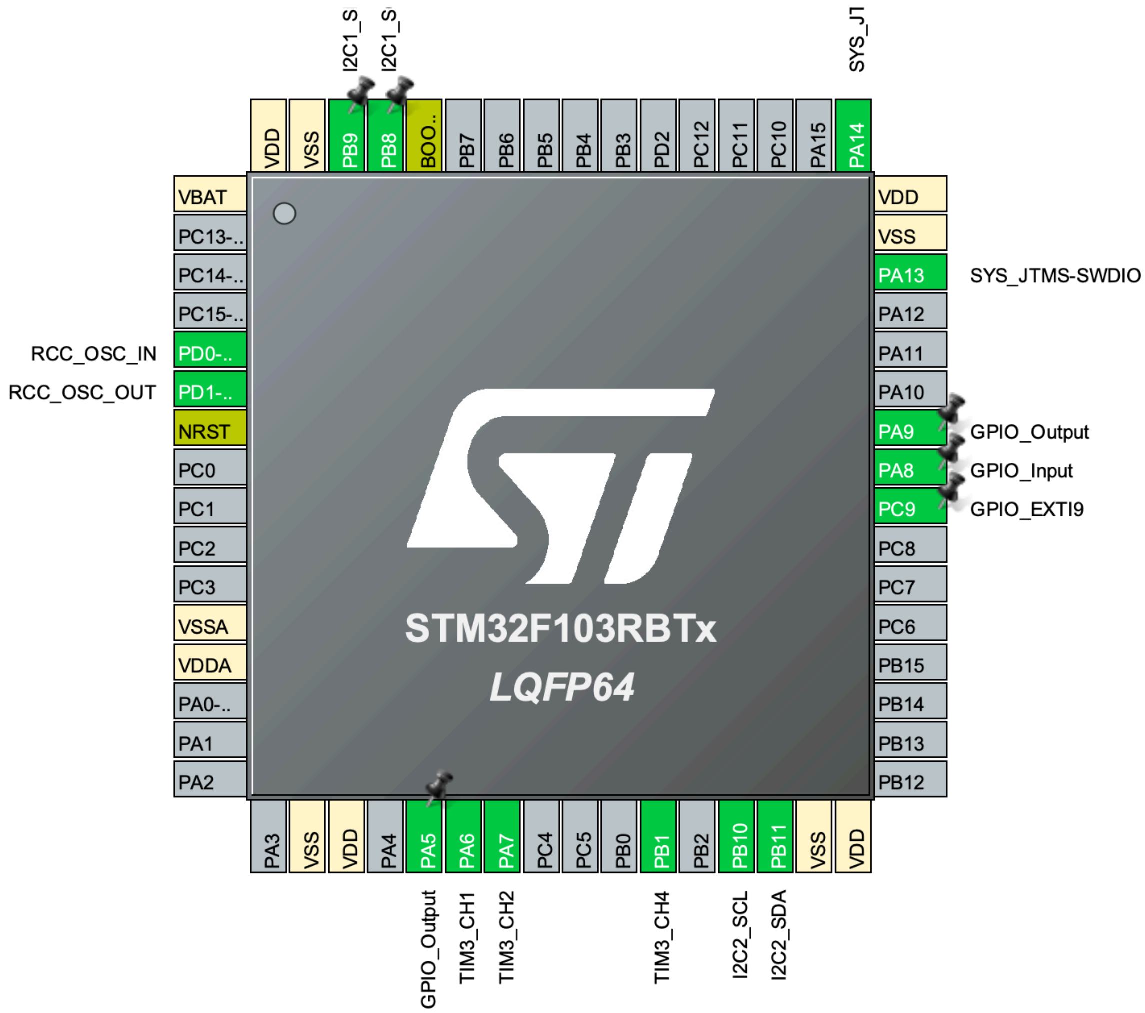
(Comment on the outcome): The average cold start time is less than 4s, which is mostly negligible. We believe that the model has a good cold start time, considering the time for the user to wear the vest. We will do more rigorous testing as we add more subsystems that can potentially delay the average cold start time of the system in general.

The last short comment on maintenance complexity is that it is fairly easy to service the model itself. The RPI Yolo model has model.train framework as well as online yaml files that allows it to easily obtain needed datasets.

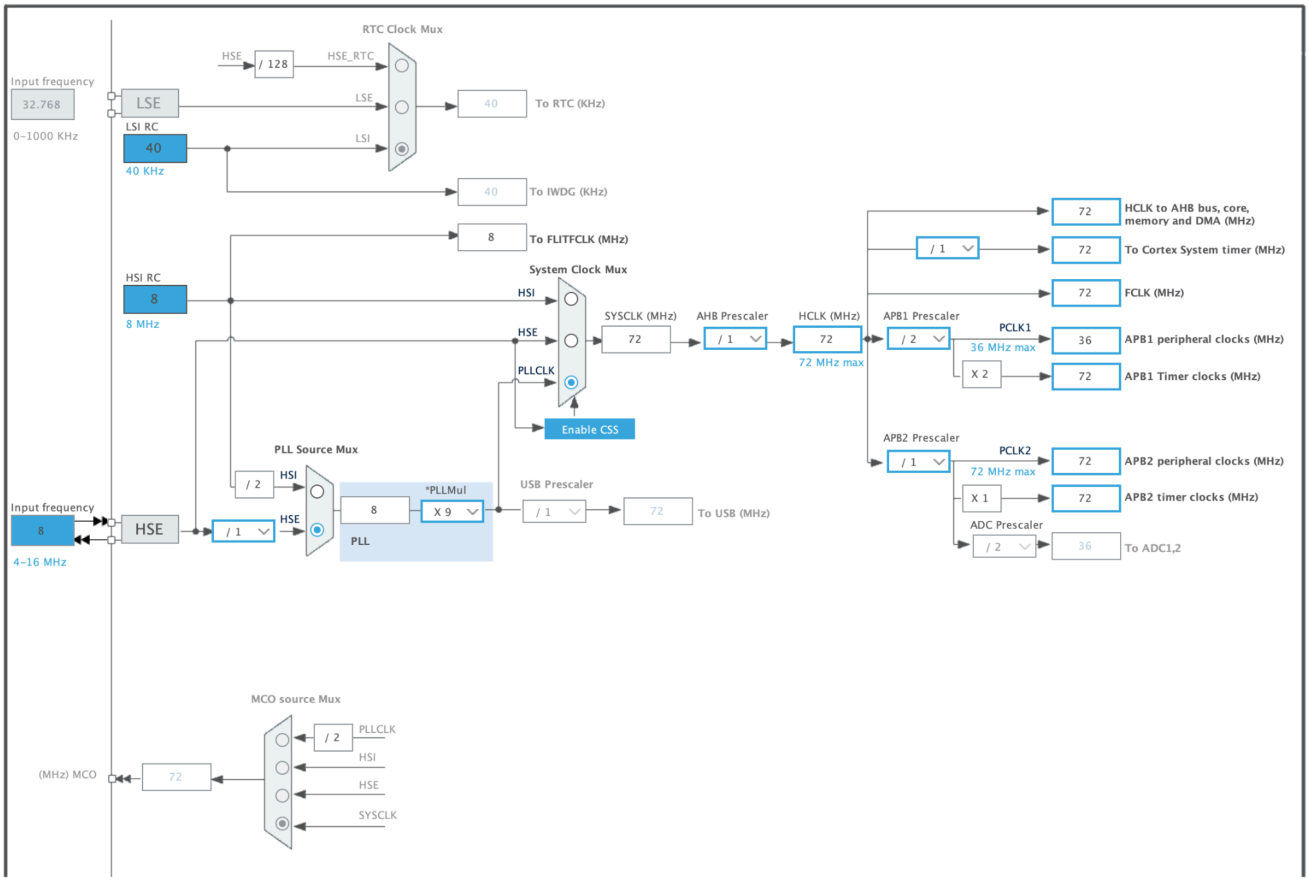
Additionally, quantization is also straight-forward. It can be done by exporting model.export to ONNX format and calling the ONNX api. Compared to other models in github, the RPI Yolo model not only works in both Raspberry Pi and mac with easy installment dependencies, but also provides suites for training and quantization for customizing the pretrained YOLO model.

2.8.2 ST Microcontroller:

For the ST Microcontroller, we have to set up the pins to enable the I2C protocol for the TOF sensor module and the TIMER to generate PWM signals for the motor.
(MCU Configuration)



(Timer Configuration)



One important factor to note is to set the timer HCLK to 72MHz for the I2C driving clock to configure 400KHz. We also have to ensure that we utilize the clock source to be an internal clock.

To ensure the functionality of each module, we have written unit testing functions and the execution function for the motor and the TOF sensor.

Motor Driving Code:

- The test function checks the functionality of the motor by increasing and decreasing the duty cycle from [0% ~ 100%]. The incrementation and decrementation should happen indefinitely.

(Algorithm):

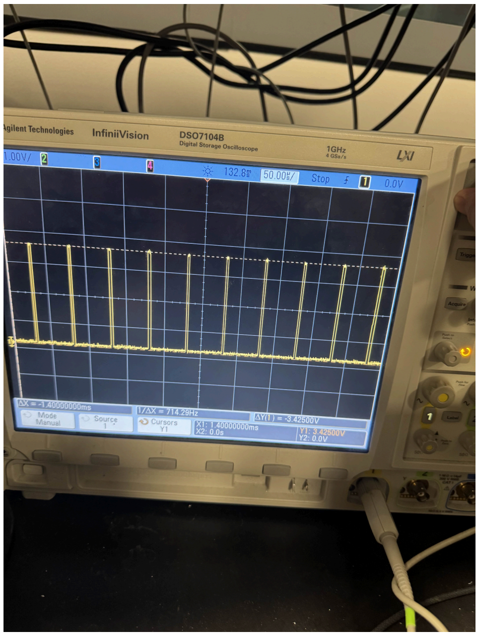
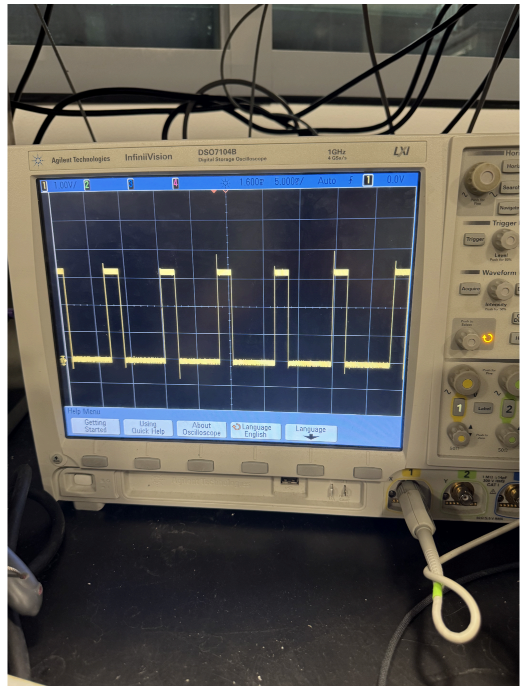
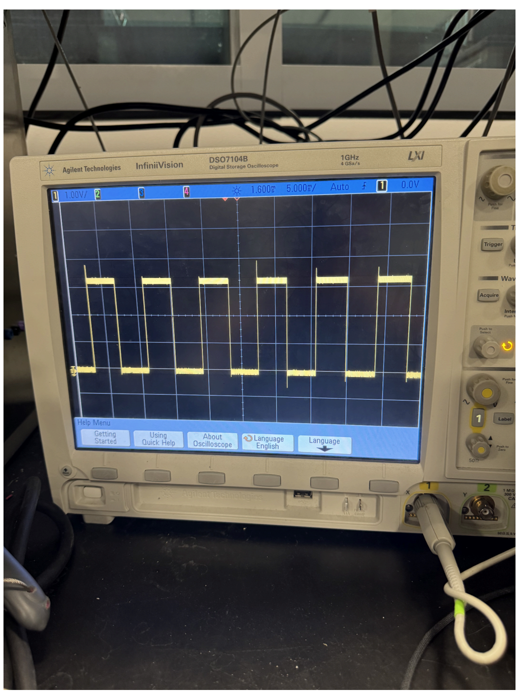
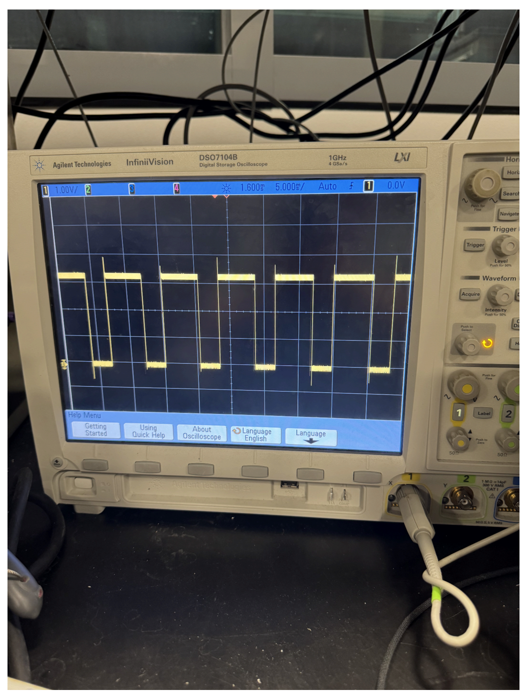
a) For debugging purposes, use a GPIO to toggle on and off the LED whenever the test function is run.
b) Start the PWM on the timer of choice using HAL_TIM_PWM_Start function.
c) The value of the capture register is directly proportional to the duty cycle, where the values of the capture register is [0 ~ pow(2,16)] Hence:

```
while(true):
     Placeholder = 0
     while(placeholder < (pow(2,16) - 1):
          Timer's capture register <- placeholder
          Increment the placeholder value by certain stride
          Place hardware delay to process
     while(placeholder > 0):
          Timer's capture register <- placeholder
          Decrement the placeholder value by certain stride
          Place hardware delay to process
```

(Implementation):

```
void test_motor() {

        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
        HAL_Delay(500);

        HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_2);
        int counter = 0; //* Just for debugging
        uint32_t test = 0;

         while (counter < 10000) {
//          htim3.Instance -> CCR2 = (uint32_t)(40000);
//          HAL_Delay(20);
             while(test < (pow(2,16) - 1)) {
               htim3.Instance -> CCR2 = test;
               test += 70;
               HAL_Delay(1);

             }

             while(test > 0) {
                  htim3.Instance -> CCR2 = test;
                  test -= 70;
                  HAL_Delay(1);

             }
             counter += 1;
         }

        HAL_TIM_PWM_Stop(&htim3, TIM_CHANNEL_2);

        return;
}
```

(Result):

Agilent Technologies InfiniiVision DSO7104B	Agilent Technologies InfiniiVision DSO7104B
Agilent Technologies InfiniiVision DSO7104B	Agilent Technologies InfiniiVision DSO7104B

Based on the readings, we can observe that the output is working as it follows the intention of the algorithm written in the motor testing code above.

Thus, extracting the motor driving algorithm, we can construct simpler implementation where we just have to make sure that the CCR2 register is set to the calculated distance (post function value) to generate the PWM signal:

```
void set_motor(uint32_t motor_val_test) {

      HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_2);
      htim3.Instance -> CCR2 = motor_val_test;
      HAL_Delay(10);
      HAL_TIM_PWM_Stop(&htim3, TIM_CHANNEL_2);
}
```

TOF Sensor Code:

- The implementation will be using the polling method. If needed, we will be implementing a callback function, which utilizes the interrupt method.
- The TOF sensor will be using I2C protocol, which has already implemented lower levels of abstraction.

Based on the TOF sensor provided setup from STM32, the resulting output from the TOF sensor will be stored in the following struct:

```
typedef struct
{
  uint8_t NumberOfTargets;
  uint32_t Distance[VL53L5CX_NB_TARGET_PER_ZONE]; /*!< millimeters */
  uint32_t Status[VL53L5CX_NB_TARGET_PER_ZONE];    /*!< OK: 0, NOK: !0 */
  float_t Ambient[VL53L5CX_NB_TARGET_PER_ZONE];    /*!< kcps / spad */
  float_t Signal[VL53L5CX_NB_TARGET_PER_ZONE];     /*!< kcps / spad */
} VL53L5CX_ZoneResult_t;

typedef struct
{
  uint32_t NumberOfZones;
  VL53L5CX_ZoneResult_t ZoneResult[VL53L5CX_MAX_NB_ZONES];
} VL53L5CX_Result_t;
```

Direct intuition is that after a full query is done, the result will be distance information will be written under the “Distance” array member of VL53l5CX_ZoneResult_t. The full function definition is as follows:

```
typedef struct
{
  int32_t (*Init)(VL53L5CX_Object_t *);
  int32_t (*DeInit)(VL53L5CX_Object_t *);
  int32_t (*ReadID)(VL53L5CX_Object_t *, uint32_t *);
  int32_t (*GetCapabilities)(VL53L5CX_Object_t *, VL53L5CX_Capabilities_t *);
  int32_t (*ConfigProfile)(VL53L5CX_Object_t *, VL53L5CX_ProfileConfig_t *);
  int32_t (*ConfigROI)(VL53L5CX_Object_t *, VL53L5CX_ROIConfig_t *);
  int32_t (*ConfigIT)(VL53L5CX_Object_t *, VL53L5CX_ITConfig_t *);
  int32_t (*GetDistance)(VL53L5CX_Object_t *, VL53L5CX_Result_t *);
  int32_t (*Start)(VL53L5CX_Object_t *, uint32_t);
  int32_t (*Stop)(VL53L5CX_Object_t *);
  int32_t (*SetAddress)(VL53L5CX_Object_t *, uint32_t);
  int32_t (*GetAddress)(VL53L5CX_Object_t *, uint32_t *);
  int32_t (*SetPowerMode)(VL53L5CX_Object_t *, uint32_t);
  int32_t (*GetPowerMode)(VL53L5CX_Object_t *, uint32_t *);
} VL53L5CX_RANGING_SENSOR_Drv_t;
```

Based on the following documentation, the setup will be as follows:

1. There are levels of abstractions. The pure HW.HAL (hardware abstraction level) codebase will interface with the hardware, assign MMIO setups so that the higher abstraction level can interface with HAL to perform embedded commands such as read and write functions. Next the platform.h and the platform.c interact with the HAL codebase, in which the specific I2C read and write functions are called.
2. Here is the specific abstraction level [Read Case]:

```
int32_t VL53L5CX_ReadID(VL53L5CX_Object_t *pObj, uint32_t *pId)
{
  int32_t ret;
  uint8_t device_id = 0;
  uint8_t revision_id = 0;
  uint8_t status = VL53L5CX_STATUS_OK;

  if ((pObj == NULL) || (pId == NULL))
  {
    ret = VL53L5CX_INVALID_PARAM;
  }
  else
  {
    status |= WrByte(&pObj->Dev.platform, 0x7fff, 0x00); //* WrByte is from
platform.h
    status |= RdByte(&pObj->Dev.platform, 0, &device_id); //* RdByte is from
platform.h
```

```
    status |= RdByte(&pObj->Dev.platform, 1, &revision_id);
    status |= WrByte(&pObj->Dev.platform, 0x7fff, 0x02);

    if (status == 0U)
    {
      *pId = ((uint32_t)device_id << 8) + revision_id;
      ret = VL53L5CX_OK;
    }
    else
    {
      *pId = 0;
      ret = VL53L5CX_ERROR;
    }
  }

  return ret;
}
```

-> Below (Lower abstraction)

```
uint8_t RdByte(
            VL53L5CX_Platform *p_platform,
            uint16_t RegisterAdress,
            uint8_t *p_value)
{
  return p_platform->Read(p_platform->address, RegisterAdress, p_value, 1U);
}

uint8_t WrByte(
            VL53L5CX_Platform *p_platform,
            uint16_t RegisterAdress,
            uint8_t value)
{
  return p_platform->Write(p_platform->address, RegisterAdress, &value, 1U);
}
```

-> Below (lower abstraction):

```
(This are the polling function, not the interrupt)
HAL_StatusTypeDef HAL_I2C_Master_Transmit(I2C_HandleTypeDef *hi2c, uint16_t
DevAddress, uint8_t *pData, uint16_t Size, uint32_t Timeout);

HAL_StatusTypeDef HAL_I2C_Master_Receive(I2C_HandleTypeDef *hi2c, uint16_t
DevAddress, uint8_t *pData, uint16_t Size, uint32_t Timeout)
```

For the lower level STM32 HAL code, we are planning to create an init function that takes in the correct I2C read and writes depending on the MCU or BreadBoard setup.

Now that we have the TOF sensor setup, we have to make sure that we get it working before the next breadboard demo. The TOF sensor utilizes the I2C protocol, which utilizes SDA and SCA pins to communicate between the master and slave.

(Algorithm):

1) Set the platform address to provided default address: VL53L7CX_DEFAULT_I2C_ADDRESS.
2) Reset the TOF sensor before the usage.
3) Check if the sensor is in good state by calling VL53l7cx_is_alive function.
4) Now that the sensor is alive, TOF sensor chip has to be initialized.
5) Set the desired resolution of the TOF sensor: choose between 8x8 or 4x4.
6) Use the get resolution function to check if the resolution is set properly.
7) Initiate the TOF sensor measurement using vl53l7cx_start_ranging. -> Based on the use cases, we can put a while loop or not for realtime measurement.
8) Check if the TOF data is ready. Use the function vl53l7cx_check_data_ready.
9) Fetch the data using the function vl53l7cx_get_ranging_data. From such, it should have fetched the grid information that the TOF sensor was measuring.
10) Before the function exits, apply vl53l7cx_stop_ranging, to ensure that the sensor stops measuring.

(Implementation):

```
void tof_motor() {

  char message[100];
  uint16_t message_len;
  message_len = sprintf(message, "TOF MOTOR FUNCTION EXECUTED\n");
  HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);

  //* Set the chip address
  Tof_Dev.platform.address = VL53L7CX_DEFAULT_I2C_ADDRESS;

  //* Reset the sensor
  VL53L7CX_Reset_Sensor(&(Tof_Dev.platform));
```

```
//* First Check if the sensor is alive
  tof_status = vl53l7cx_is_alive(&Tof_Dev, &tof_isAlive);
  if(!tof_isAlive || tof_status) {
    message_len = sprintf(message, "VL53L7CX not detected at requested address\n");
    HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);
    return tof_status;
  } else {
    message_len = sprintf(message, "VL53L7CX IS ALIVE!!!!!\n");
    HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);
  }

  //* Now that the sensor is alive, we initialize teh chip

  tof_status = vl53l7cx_init(&Tof_Dev);
  if(tof_status)
            {
    message_len = sprintf(message, "VL53L7CX did not initialize\n");
    HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);
                        // printf("VL53L7CX ULD Loading failed\n");
                        return tof_status;
            } else {
     message_len = sprintf(message, "VL53L7CX DID initialize\n");
    HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);
  }

  message_len = sprintf(message, "VL53L7CX ULD ready ! (Version : %s)\n", VL53L7CX_API_REVISION);
  HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);


  //* Now we have to set the resolution:
  tof_status = vl53l7cx_set_resolution(&Tof_Dev, VL53L7CX_RESOLUTION_8X8);
  tof_status = vl53l7cx_get_resolution(&Tof_Dev, &tof_resolution);

  message_len = sprintf(message, "VL53L7CX resolution: %d \n", tof_resolution);
  HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);

  //* Now we let the sensor to start measurement
  tof_status = vl53l7cx_start_ranging(&Tof_Dev);

  // int counter = 0;

  while(1 == 1) {

    //* now we want to check if the data is ready:

    tof_status = vl53l7cx_check_data_ready(&Tof_Dev, &tof_isReady);

    if(tof_isReady) {
```

```
    vl53l7cx_get_ranging_data(&Tof_Dev, &Tof_Result);
    message_len = sprintf(message, "Print data no : %3u\n", Tof_Dev.streamcount);
    HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);

   }
   for (int i = 0; i < 8 * 8; i++) {
      message_len = sprintf(message, "Zone : %3d, Status : %3u, Distance : %4d mm\n",
                                          i,

Tof_Result.target_status[VL53L7CX_NB_TARGET_PER_ZONE*i],

Tof_Result.distance_mm[VL53L7CX_NB_TARGET_PER_ZONE*i]);
      HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);
   }
   VL53L7CX_WaitMs(&(Tof_Dev.platform), 5);
  }

  tof_status = vl53l7cx_stop_ranging(&Tof_Dev);
  message_len = sprintf(message, "End of ULD demo\n");
  HAL_UART_Transmit(&huart2,(uint8_t *)message, message_len, 0xFFFF);

  return;
}
```

Mapping Functions for Distance and Motor Duty Cycle

- The mapping function allows the motor to vibrate based on the distance provided. What we really want is exponential increase in vibration intensity based on how close the distance is. Here are some candidates:

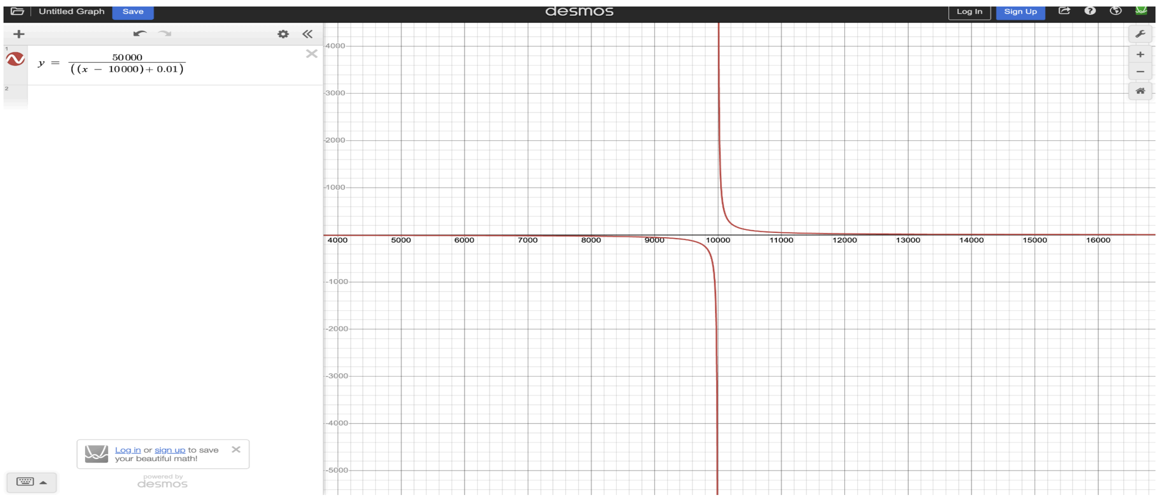
[Naive Rectangular Function] - Most Naive

- Sensors are assumed to be measured in cm for now.
- We can come up with a simple rectangular function, where if the distance is within the threshold, it will vibrate with a constant value, else it will not vibrate.
- The output of the function will be feeded into the set_motor function, in which will set the capture register.
- Downside is that the PWM will be consistent, so the user will have no sense of how close the objects are.

$$f(x) = \begin{cases} 40000 & \text{if } 7000 \geq x \geq 0 \\ 0 & \text{if } x < 0 \text{ } or \text{ } x > 5000 \end{cases}$$

[Naive Inverse Function]

- Now what we want is a function that allows exponential growth based on how near the object is.
- The naive way to go about this is to utilize the inverse function. When X is large, it converges to zero, else as the distance decreases, the vibration to the motor exponentially increases.
- The problem with this approach is that it is not easy to service because the function quickly shoots up when the distance is way too close. Inverse function is also not a good practice because if the number is too small, it can overflow.

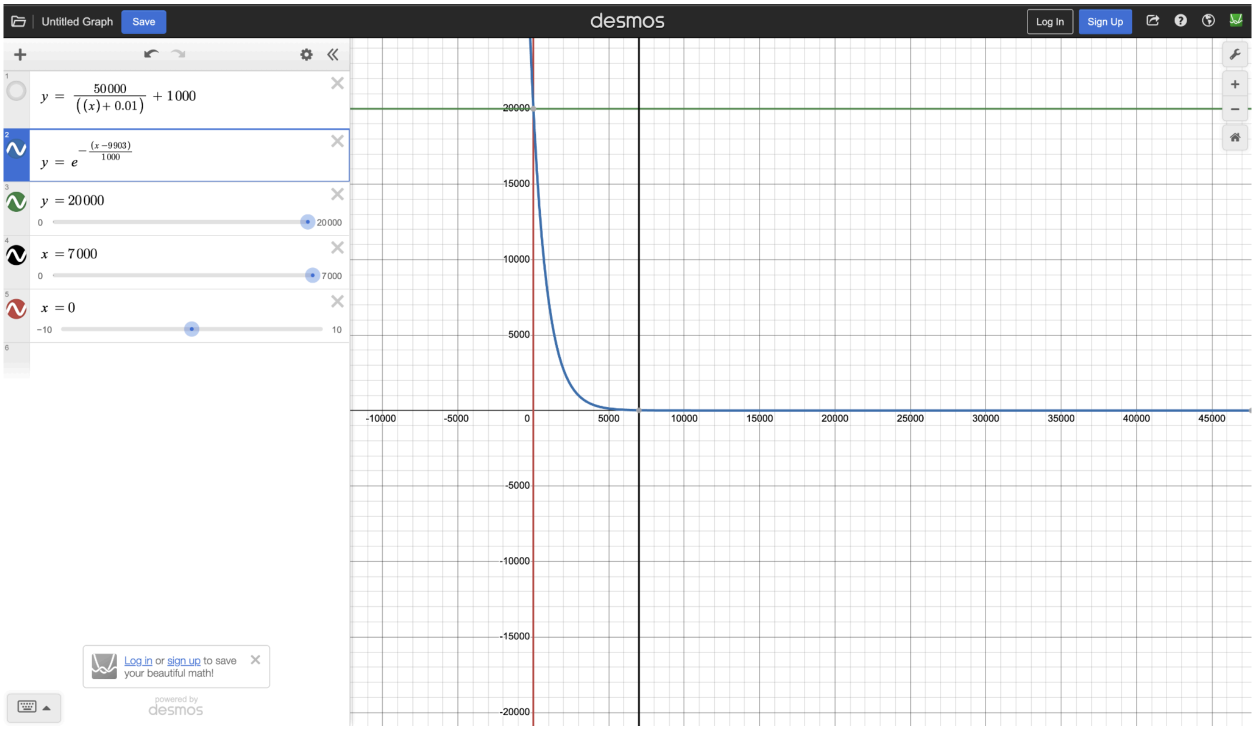


$$f(x) = \frac{50000}{(x - 10000) + 0.01}$$

An alternative better method is the exponential function.

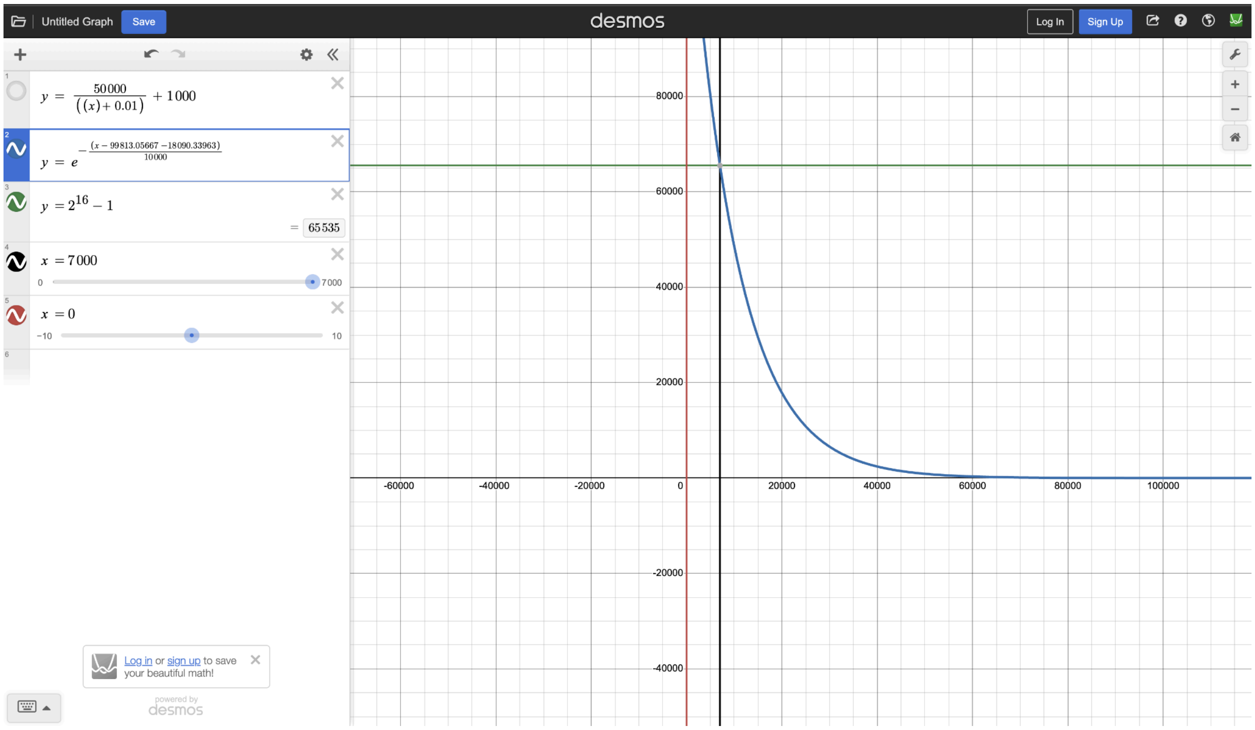
[Advanced Exponential Function] – Preferred, used in the breadboard demo

- Function is better to service.
- It has a smooth curve when it is within the distance threshold.
- Function is computationally expensive, so we will be using a lookup table for O(1) search for future implementation.



$$f(x) = \begin{cases} e^{\frac{-(x-9903)}{1000}} & \text{if } 7000 \geq x \geq 0 \\ 0 & \text{if } x < 0 \text{ } or \text{ } x > 5000 \end{cases}$$

Additionally for the future setup, we wish to utilize the rectangular function and the exponential function in tandem to allow smoother and more accurate position of the object. In this way, if it is within the close range, it will vibrate constantly for max within close ranges, for mid ranges, we would like to use the exponential function. Else, the vibration output will be zero.



$$f(x) = \begin{cases} 2^{16} - 1 & \text{if } SetDistance1 > x \geq 0 \\ e^{\frac{-(x - x_{offset})}{x_{stretch}}} & \text{if } SetDistance2 > x \geq SetDistance1 \\ 0 & \text{if } x < SetDistance1 \text{ } or \text{ } x > SetDistance2 \end{cases}$$

(Implementation in Codebase):

```
uint32_t map_motor_func(uint32_t input) {

        uint32_t final_dist = 0;

        if((input >= 0 && (input <= 7000))) {
                final_dist = exp(-(input - 9903 - 70)/1000); //* this is just the test func, will change later for tof
        } else {
                final_dist = 0;
        }

  return final_dist;
}

uint16_t map_pwm_func(uint16_t Distance) {
  uint32_t final_distance = Distance * 100;
```

```
 if(Distance > 1500) {
   final_distance = 1500; //* Capping the value due to duty cycle.
 }

  //* clipping if too far

  final_distance = map_motor_func(final_distance);

  return final_distance;
}
```

2.9 Tolerance Analysis:

2.9.1 Data Transmission:

GPS navigation typically consumes around 5-10 MB/hour and around 1-3MB/hour for live traffic updates [16]. By combining these totals, we can expect to require around 6-13MB/hour, which equates to 0.01-0.03 Mbps. To assume the worst case scenario, we could apply this formula:

- $Total\ Bandwidth\ (Mbps)\ =\ B\ +\ O$
    - $B\ =\ Bandwidth\ (Mbps)\ for\ usage$
    - $O\ =\ Overhead\ (Mbps)\ due\ to\ packet\ errors,\ retries$
        - $O\ =\ 0.25B$

In the worst case scenario, we could expect around 20-30% (25%) of our bandwidth to contain errors, which means we would need around 0.0125-0.0375 Mbps of bandwidth at minimum. If we estimated anything above the 20-30% threshold, we would have an error rate of around 40%, which is almost half of the usage bandwidth, making our transmission unreliable and not efficient. Although typical packet loss rates are at around < 1% in normal conditions, accounting for a 20-30 % tolerance will give us a larger bound to account for retries, and latency spikes [13].

When delivered over 4G LTE hotspot, those requirements are well below the limits of practicality. Since 4G LTE hotspots can provide around 30-40 Mbps of bandwidth on average, and about 20-30 ms of latency [18], we can safely predict that our GPS model should be able to work consistently, even with occasional drops in bandwidths.

2.9.2 Power Management

A. Battery Life

The system is powered by two 3.7 V Li-ion cells in series, producing a nominal pack voltage of 7.4 V and a total energy capacity of 37 Wh (5000 mAh × 7.4 V). The power draw varies between 15 W (typical) when only the microcontroller, sensors, and Raspberry Pi are active, and 30 W (peak) when all peripherals such as microphones, ToF sensors, and vibration motors are running simultaneously.

Nominal Runtime Calculation:

Battery life = Energy / Load power

- Typical load (15 W): 37 Wh / 15 W = 2.47 hours
- Moderate load (20 W): 37 Wh / 20 W = 1.85 hours
- Peak load (30 W): 37 Wh / 30 W = 1.23 hours

Including Tolerances:
Battery capacity can vary by ±5% due to temperature and aging. Converter efficiency varies ±3%, and total system power draw can change ±10% depending on operation mode. Applying worst-case conditions, the minimum expected runtime is approximately 1.25 hour under peak load and about 2.1 hours during typical use. This confirms that the design meets the 1-hour operational requirement for field testing even under tolerance limits.

a. Voltage Margins

The power subsystem converts the 7.4 V battery voltage to stable 5 V and 3.3 V rails. Each converter was analyzed for input range, output accuracy, and safety margin.

5 V Buck Converter (MAX20410AFOC)

- Input: 6.0 – 8.4 V
- Nominal output: 5.00 V ± 1.3% (range 4.93 V – 5.05 V)
- Efficiency: 93% at 5 A
- Raspberry Pi safe operating range: 4.63 – 5.10 V

At the lowest battery level (6.0 V), efficiency drops to 91%. With ±0.1 V ripple, the minimum output is about 4.83 V, which is still above the Pi's undervoltage limit of 4.63 V. The 5 V rail therefore has more than 200 mV of margin even in the worst case.

3.3 V Buck Converter (TPS563207DRLR)

- Input: 6.0 – 8.4 V
- Output tolerance: ±2% (range 3.23 – 3.37 V)
- Efficiency: 89% at 1–3 A

The STM32H7 microcontroller, ToF sensors, and GPS modules operate safely within 3.0 – 3.6 V. The 3.3 V rail thus has at least 200 mV of headroom under all load and voltage conditions.

Transient and Surge Margins:
Transient voltage suppressors protect each rail:

- 5 V line: DESD5V0S1BA-7 clamps @ 10 V, well above normal operation.
- 3.3 V line: SMF3.3 clamps @ 6.8 V

This ensures all rails stay below damaging levels during ESD or hot-plug transients.
Converter Efficiency and Power Loss:

- Power loss = Output Power × (1 / Efficiency – 1)
- 5 V buck converter: 25 W × (1 / 0.93 – 1) = 1.85 W loss
- 3.3 V buck converter: 8 W × (1 / 0.89 – 1) = 0.99 W loss
- MP2639C charger: 10 W × (1 / 0.89 – 1) = 1.23 W loss

Total estimated dissipation: 4.1 W across the PCB. These values are used later in the temperature analysis to determine board heating and safe operation.

Charge and Discharge Behavior

The MP2639C charger operates in two main phases:

- Constant-current mode (2 A, ~89% efficiency)
- Constant-voltage mode (~87% efficiency near full charge)
- Charge time = Battery capacity / (Charge current × Efficiency)
- 5000 mAh / (2 A × 0.89) = 3.5 hours

The NTC thermistor feedback network automatically reduces charging current when the battery exceeds 60°C, preventing overheating and ensuring safe charging at all ambient temperatures.

B. Temperature Regulation Tolerance Analysis

2.9.3 PCB Heat Dissipation

The PCB is housed in an enclosure case and is worn close to the user, thus maintaining thermal safety is very important. Most of the heat generation came from the 5V Buck Converter, 3.3V Buck Converter, and the Battery Charger IC. According to the datasheets [27], [30], and [38] we know that:

PCB Material Thermal Tolerance:

- FR-4 Continuous-Use Rating (~105 Degrees Celsius):
- All Components must be kept **below 105 degrees celsius**

Human Safety Margin:

According to electronics cooling, the safety temperature limit for users is around 48 degrees Celsius for external surfaces held, touched or worn against the body. This means that the enclosure temperature **must not exceed 50 Degrees Celsius** (Vest also limits some of the heat flow)

PART	METAL	VITREOUS MATERIAL	PLASTIC, RUBBER
Devices worn on the body (in direct contact) in normal use (>8 hr)	43–48°C	43–48°C	43–48°C
External surfaces held, touched or worn against the body in normal use (>1 min and <8 hr)	48°C	48°C	48°C

Figure 2.9.2.1 Safety Temperature Limits for Users 1 [51]

Touch temperature levels

Touch temperature limits

	Accessible parts [a]	Maximum temperature (T_{max}) °C			
		Metal	Glass, porcelain and vitreous material	Plastic and rubber [b]	Wood
TS1	Handles, knobs, grips, etc., and external surfaces held in normal use (>1 min) [c]	48	48	48	48
TS1	Handles, knobs, grips, etc., and external surfaces held for short periods of time or touched occasionally (>10 s and <1 min) [c]	51	56	60	60
TS1	Handle, knobs, grips etc., and external surfaces touched occasionally for very short periods (>1 s and <10 s) [c]	60	71	77	107
TS1	External surfaces that need not be touched to operate the equipment (<1 s) [c]	70 [d]	80 [d]	94 [d]	140
The limits for TS2 are 10 K higher than the TS1 limits.					

FIgure 2.9.2.2: Safety Temperature Limits for Users 2 [52]

Ambient Temperature:

We can assume that the room temperature is 25 degrees Celsius, but the PCB is housed within an enclosure and also the vest. This means that the ambient temperature within the enclosure is higher than 25 degrees Celsius.

Body Heat to Enclosure Calculations:

- Enclosure Size: 12cm x 12cm x 3cm thickness (most likely will use)
- Body to Enclosure Area: 0.12m x 0.12m
- Body Temperature: 34 Degrees Celsius
- Room Temperature: 25 Degrees Celsius
- Fabric Insulation from the Vest: ~3-5mm
- Typical Clothing Thermal Conductivity: 0.04 W/(m*k)
- Heat Transfer Coefficient: 5-10 W/(m^2 *K)

Surface:

$$q = h(T_{\text{skin}} - T_{\text{room}})$$

For $h = 5\text{–}10\, W/m^2 \cdot K$ and $\Delta T = (34 - 25) = 9\, K$:

$$q = (5\text{–}10) \times 9 = 45\text{–}90\, W/m^2$$

Then, total heat entering the enclosure's body-facing wall:

$$Q = q \times A = (45\text{–}90) \times 0.0144 = 0.65\text{–}1.3\, W$$

Internal Temperature Rise:

$$\Delta T_{\text{internal}} = Q \times R_{\text{th}} = (0.6\text{–}1.3)\, W \times (6\text{–}8\, {}^{\circ}C/W)$$

$$\Delta T_{\text{internal}} \approx 3.6^{\circ}C\text{–}10.4^{\circ}C$$

Figure 2.9.2.3: Heat Transfer Calculations

Thus, the ambient temperature within the enclosure is 25 degrees Celsius (Room Temperature) + ~10 degrees Celsius (body heat transfer) = **35 degrees Celsius.**

Maximum Temperature of PCB Traces:

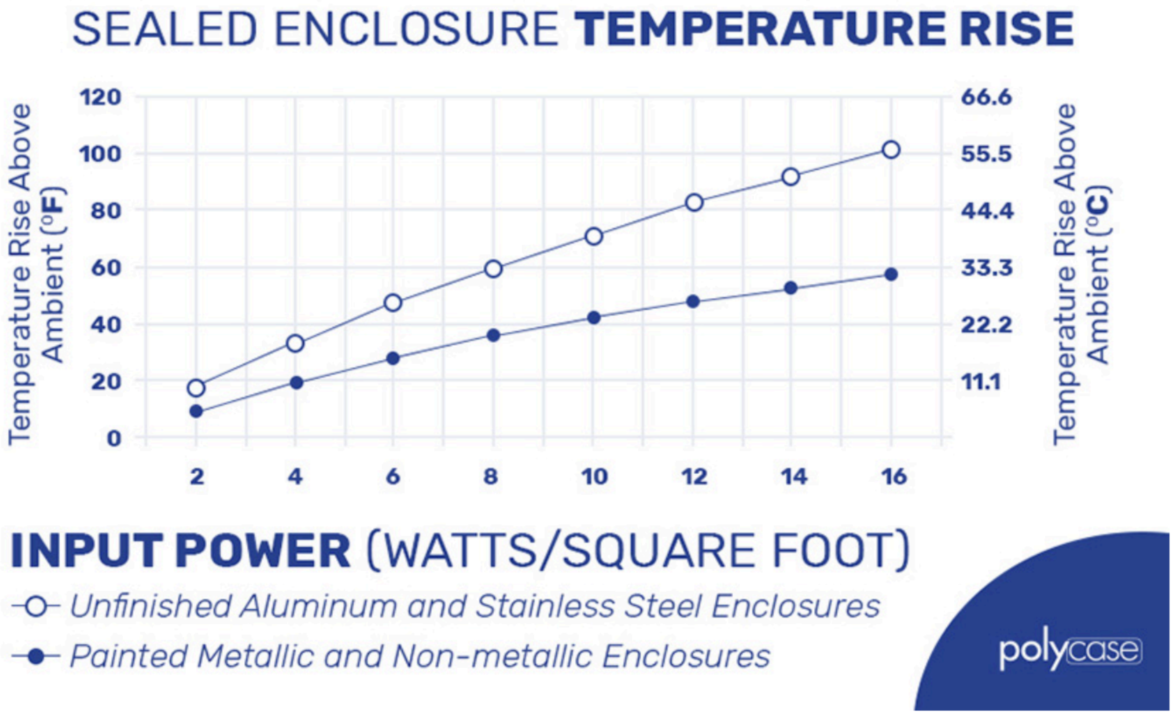


Figure 2.9.2.4: Polyfuse Input Power to Temperature Rise

Localized Temperature Estimates:

- 5V Buck Converter (MAX20410): 25 W output, 93% efficiency → about 1.85 W dissipated
- 3.3V Buck Converter (TPS563207): 8 W output, 89% efficiency → about 0.99 W dissipated
- MP2639C Charger: 10 W input, 89% efficiency → about 1.2 W dissipated

This means that there would be roughly 4.05W of heat dissipated by the components. We can upper bound that to 4.5W if we consider resistors, inductors, and capacitors. Theoretically, the enclosure should have a T_rise of roughly ~12 Degrees Celsius above the 35 degrees Celsius (ambient) as we calculated before. This means the enclosure will be **~47 degrees Celsius at peak load**. This value is **within the safety range** of acceptable temperature of wearable devices according to [51] and [52].

Battery Charger IC:

- Input Power: 5.5V(max recommended) *1.98 (max) ~= 10.89W
- Power Output: 10.89W * 89% Efficiency ~= 9.6921W
- Power Loss: 10.89W - 9.6921W ~= 1.1979W

- Thermal Resistance:
  - 42 Degrees Celsius/W (Junction to Ambient)
  - 9 Degrees Celsius/W (Junction to Case)
- Operating Conditions: -40 to 125 Degrees Celsius

$$T_J = T_A + (P_{\text{diss}} \times \theta_{JA})$$

$$T_J = 35^\circ C + (1.2\,W \times 42^\circ C/W) \approx 85^\circ C$$

Figure 2.9.2.5: Junction Temperature Calculation [49]

- 85 Degrees Celsius <= 125 Degrees Celsius (Recommended Operating Max)
- **Trace Width Requirement:**
  - Max Current: 2.98A
  - Copper: 1oz
  - Ambient: 35 Degrees Celsius
  - Temperature Rise: 50 Degrees Celsius
    - Tj is the max temperature due to power loss, thus we can set T_rise = Tj - Ambient Temperature
  - Max Temperature: 85 Degrees Celsius
    - Compliant with FR-4's rating of under 105 Degrees Celsius
  - Required Trace Width: 20.078 mil
    - We used: 30-40mil, which should be sufficient to help carry the current and tolerance the heat
- Heat Abosorbortion Implementation
  - Minimum: Thermal Vias, Thicker Traces above 20 mil, and connect USB_C Traces together to help dissipate heat
  - Additional: Heat Pad

First, calculate the Area:

$$A = \left(\frac{I}{\mathrm{k} \times T_{Rise}{}^{b}} \right)^{\frac{1}{c}}$$

Then, calculate the Width:

$$W = \frac{A}{t \times 1.378}$$

Figure 2.9.2.6: Trace Width Calculation [50]

5V Buck Converter (MAX20410A):

- Input Power: 8.4V (Battery Max)
- Output Power Required: 5.05V (MAX 5V output upper bound) * 5A ~=25W
- Efficiency Rating: ~93%
- Power Output (Actual): 27W * 93% efficiency ~= 25.11W
- Power Loss: 27W - 25.11W ~= 1.89W
- Thermal Resistance: [27]
    - Worse Case: 38.6 Degrees Celsius/W (Junction to Ambient Thermal Resistance)
- Operating Conditions: -40 Degrees Celsius to 150 Degrees Celsius

$$T_J = 35^{\circ}C + (1.89\,W \times 38.6^{\circ}C/W)$$

$$T_J = 35^{\circ}C + 72.9^{\circ}C \approx 108^{\circ}C$$

Figure 2.9.2.7: Junction Temperature Calculation for MAX

- 108 Degrees Celsius <= 150 Degrees Celsius (Recommended Operating Max)

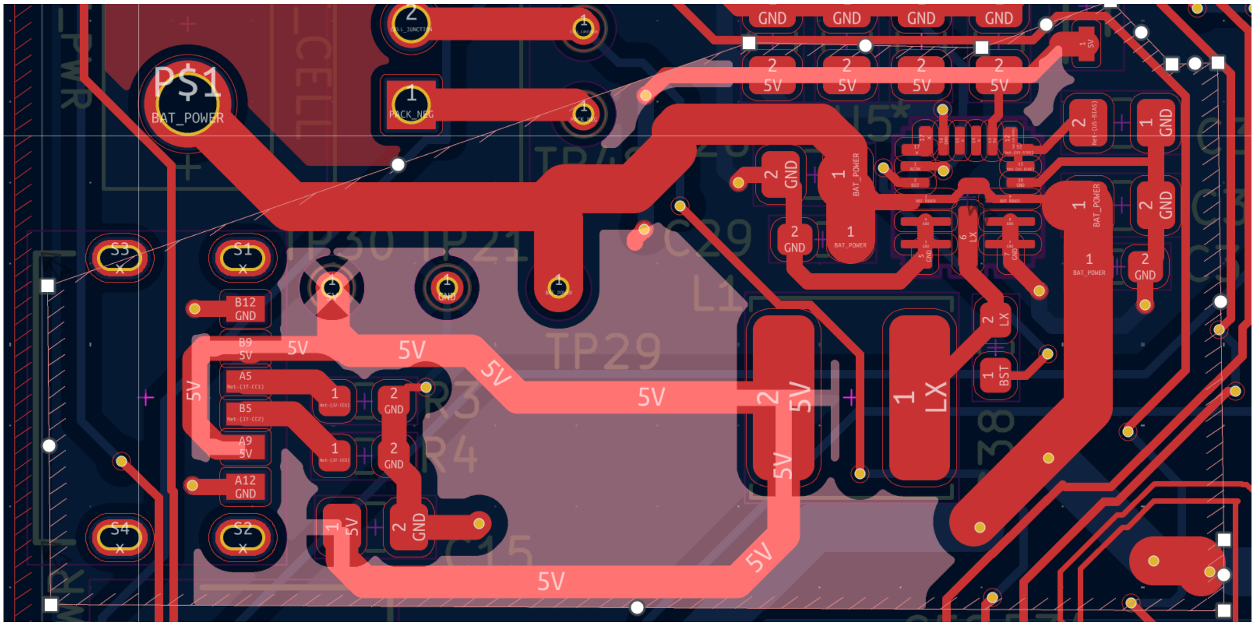


Figure 2.9.2.8: PCB Layout Front 5V Copper Zone

Figure 2.9.2.9: PCB 5V Copper Zone Filled Area

$$A = 159.3984\ \mathrm{mm}^2 = 1.59 \times 10^{-4}\ \mathrm{m}^2$$

$$t = 3.5 \times 10^{-5}\ \mathrm{m}, \quad k_{Cu} = 385\ \mathrm{W/(m{*}K)}, \quad h = 8\ \mathrm{W/((m{*}{*}2){*}K)}, \quad P_{diss} = 1.89\ \mathrm{W}$$

$$R_{th,cond} = \frac{t}{k_{Cu} * A} = \frac{3.5 \times 10^{-5}}{385 * 3.73 \times 10^{-3}} = 2.37 \times 10^{-5}\ {}^\circ\mathrm{C/W}$$

$$R_{th,conv} = \frac{1}{h * A} = \frac{1}{8 * 3.73 \times 10^{-3}} = 33.5\ {}^\circ\mathrm{C/W}$$

$$R_{th,total} \approx R_{th,conv} = 33.5\ {}^\circ\mathrm{C/W}$$

$$\Delta T = P_{diss} * R_{th,total} = 1.89 * 33.5 = 63.4\ {}^\circ\mathrm{C}$$

$$T_J = T_A + \Delta T = 35 + 63.4 = 98.4\ {}^\circ\mathrm{C}$$

$$T_J \approx 98\ {}^\circ\mathrm{C}$$

Figure 2.9.2.10: PCB 5V Actual Junction Temperature for MAX Chip

- 98 Degree Celsius is below 150 Degrees Celsius (Recommended Operating Max)

- **Trace Width Requirement:**
    - Max Current: ~5.35A
    - Copper: 1oz
    - Ambient: 35 Degrees Celsius
    - Temperature Rise: 63 Degrees Celsius
        - Tj is the max temperature due to power loss, thus we can set T_rise = Tj - Ambient Temperature
    - Max Temperature: 98 Degrees
        - The MAX Chip has 5V copper zone, thermal pads/heatsink, and thermal vias to help dissipate heat
        - Temperature should fall well below FR-4's rating of under 105 Degrees Celsius after thermal dissipation techniques as listed above
    - Required Trace Width: 39.116
        - We used: 40mil, which should be sufficient to help carry the current and tolerance the heat
- Heat Abosorbortion Implementation
    - Minimum: Thermal Vias, Thicker Traces than 35 mil, 5V copper zone
    - Required: Heat Pad / Heat Sink

3.3V Buck Converter (TPS563207DRLR):

- Input Voltage: 8.4V (Battery Max)
- Power Required: 3.3V * 2.72A ~= 8.98W
- Efficiency Rating: ~89%
- Power Output (Actual): 8.98W * 89% ~= 8W is required for the 3.3V logic system
- Power Loss: 8.98W - 8W ~= 0.98W
- Thermal Resistance: [30]
    - Junction to Ambient Thermal Resistance: 137 Degrees Celsius/W
- Operating Condition: -40 Degrees Celsius to 150 Degrees Celsius

$$T_J = 35^\circ C + (0.98\,W \times 137^\circ C/W)$$

$$T_J = 35^\circ C + 134.26^\circ C \approx 169^\circ C$$

Figure 2.9.2.11: Junction Temperature Calculation for the TPS

- 169 Degrees Celsius is above the operating condition of 150 Degrees Celsius
- We used a 3.3V Copper Layer Across the 100mm by 100mm front of the PCB

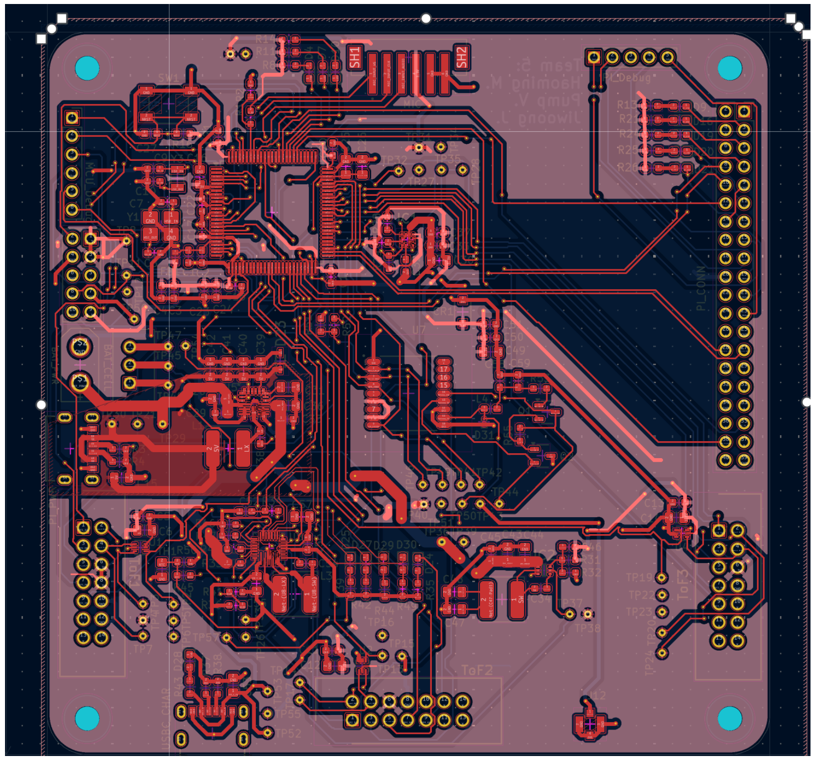


Figure 2.9.2.12: PCB Layout Front 3.3V Copper Zone

Figure 2.9.2.13: PCB 3.3V Copper Filled Area

$$A = 3730.1123\ \text{mm}^2 = 3.73 \times 10^{-3}\ \text{m}^2$$

$$t = 3.5 \times 10^{-5}\ \text{m}, \quad k_{Cu} = 385\ \text{W/(m*K)}, \quad h = 8\ \text{W/((m**2)*K)}$$

$$P_{diss} = 0.98\ \text{W}, \quad T_A = 35°C$$

$$R_{th,cond} = \frac{t}{k_{Cu} \times A} = \frac{3.5 \times 10^{-5}}{385 \times 3.73 \times 10^{-3}} = 2.37 \times 10^{-5}\ °\text{C/W}$$

$$R_{th,conv} = \frac{1}{h \times A} = \frac{1}{8 \times 3.73 \times 10^{-3}} = 33.5\ °\text{C/W}$$

$$R_{th,total} \approx R_{th,conv} = 33.5\ °\text{C/W}$$

$$\Delta T = P_{diss} \times R_{th,total} = 0.98 \times 33.5 = 32.83°C$$

$$T_J = T_A + \Delta T = 35 + 32.83 = 67.83°C$$

$$T_J \approx 68°C$$

Figure 2.9.2.14: Actual Junction Temperature for TPS

- 68 Degrees Celsius is well below the 150 Degrees Celsius Recommended Operating Max
- **Trace Width Requirement:**
  - Max Current: ~3A
  - Copper: 1oz
  - Ambient: 35 Degrees Celsius
  - Temperature Rise: 33 Degrees Celsius
    - Tj is the max temperature due to power loss, thus we can set T_rise = Tj - Ambient Temperature
  - Max Temperature: 68 Degrees
    - The MAX Chip has 5V copper zone, thermal pads/heatsink, and thermal vias to help dissipate heat
    - Temperature will fall below FR-4's rating of under 105 Degrees Celsius after thermal dissipation techniques as listed above
  - Required Trace Width: 26.077 mil
    - We used: 40mil, which should be sufficient to help carry the current and tolerance the heat
- Heat Abosorbortion Implementation
  - Minimum: Thermal Vias, Thicker Traces than 26 mil, 3.3V copper zone
  - Additional: Heat Pad / Heat Sink

Battery Power Trace Widths:

- MAX IC Draws 27W (Considered Power Loss)
- TPS IC Draws 9W (Considered Power Loss)
- Power Requirement: 36W
- Worse Case Scenario:
  - 36W / 6V (lowest Battery voltage)
  - ~= 6A

Thus, the battery power traces might be able to handle up to 6.5A (upper bounded)

Let ambient temperature be: 35 Degrees Celsius

Let Temperature Rise be: ~60 Degrees Celsius (As hot as the Bucks and Charger IC)

- **Required Trace Width:**
  - 52 mil
  - We used: 60mil, which should be enough

Originally, we were going to use 3oz copper to deal with the heat from power dissipation, however, it was too expensive and we decided to use less expensive but effective methods. In order to minimize heat, we employ strategies such as thermal vias, thermal pads/heat sinks, neck-downs, and thicker traces for high current carrying connections.

Thermal Vias & Heat Sink Effectiveness:

Thermal vias are critical in helping the PCB reduce heat, especially from high-power ICs and components. Each thermal via is a low-resistance thermal path that could lower the local junction temperature and help spread heat uniformly over the whole PCB. According to EETimes [53], 0.3mm outer outer-diameterer copper filled with vias in FR-4 (the one we used) found that proper thermal vias placement + heat sink helped reduce junction-to-ambient thermal resistance from 16 Celius/W to 5.73 C/W. This is super effectiveness as T_J = T_A + (power dissipated*T_JA).

We know that even without a heat sink, proper thermal via placements will help reduce the heat by some amount. For these reasons, our design adds thermal pads/heat sink for the MAX (5V Buck Converter) and also the TPS(3.3V Buck Converter) since they are the main heat-generating components. This ensures that heat is effectively transferred to copper planes and dissipated.

Conclusion: Battery Life & Temperature Regulation Tolerance

The power management system meets the design's runtime and safety goals. With a 37 Wh, 7.4 V Li-ion pack, the system achieves 1.2 – 2.5 hours of operation under typical and peak loads, satisfying the 1.25 hour minimum runtime requirement even under full load limits. Both buck converters maintain stable voltage margins across the full battery range, ensuring reliable operation of the Raspberry Pi, STM32 MCU, and sensors.

By using a combination of large copper planes, thermal vias, thicker traces, and localized heatsinks/thermal pads, we provided effective thermal regulation across the PCB. These techniques and layout placements help distribute heat evenly from high-power components, particularly the 5 V and 3.3 V buck converters, into the internal copper layers and surrounding air. As verified in simulation and analysis, this keeps the board surface and enclosure temperatures below 50 °C, meeting the human-contact safety threshold while maintaining each IC well under its rated junction limit. Together, the electrical and thermal design choices ensure both safe operation and reliable long-term performance of the power system.

Although analysis shows that heatsinks and thermal pads will be super helpful in dissipating heat, we don't necessarily need it, but we will at least implement thermal pads.

2.9.4 Sensing Subsystem (ToF + GPS + Magnetometer)

Being able to accurately detect the presence of objects reliably in our system is crucial.
When using a ToF sensor, we want to be able to sense the obstacles in front of us reliably within a 2m radius.
For our tolerance tests we propose the following:

a. **Measuring Distance from Various Surfaces & Glass**
   Since the ToF sensor heavily relies on light as a medium for sensing distance, objects such as glass or any reflective material could potentially cause some false-negative detection.
   The test is as follows:
   - Use a measuring tape to help determine the ground truth (actual distance)
   - Record and measure the readings at 20 different intervals for each material
   - Plot and compare the results

   To compute the error, we will apply a Mean Absolute Percentage Error (MAPE) over N samples.

$$\mu_e = \frac{1}{N}\sum_{i=1}^{N}(D_{\mathrm{meas},i} - D_{\mathrm{true},i})$$

$$\sigma_e = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(D_{\mathrm{meas},i} - \mu_e)^2}$$

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{|D_{\mathrm{meas},i} - D_{\mathrm{true},i}|}{D_{\mathrm{true},i}}$$

By using the equation above, we can evaluate the performance of the ToF sensor when measuring the distance from different types of objects.

b. ToF sensor + Vibration motor responsiveness

For this test, we are aiming to test the time difference between how long it takes for the motor to be signaled after an object (or person) appears within the 2m radius.

- Use a measuring tape to help determine where 2m is located
- Run a program which starts the timer when a key is pressed on the computer, and the program will stop the timer once the sensor detects you at the 2m distance
- Repeat for different speeds of approach (running, fast walking, casual walking, slow walking)

For each trial:

$$t_{latency} = t_{motor} - t_{objectdetected}$$

Where:

t_{motor} = timestamp when motor begins vibrating

$t_{objectdetected}$ = timestamp when ToF detects objects < 2m

Repeat measurements for the same speed for over 10-20 intervals

and Compute:

$$\bar{t}_{\text{latency}} = \frac{1}{N} \sum_{i=1}^{N} t_{\text{latency},i}$$

Anything within 150 ms for this measurement will feel instantaneous

Measure the system's consistency using the standard deviation:

$$\sigma_t = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (t_{\text{latency},i} - \bar{t}_{\text{latency}})^2}$$

Repeat across different speeds and compare.
We can also measure the response rate using:

$f_{response} = 1/\bar{t}_{latency}$

3. Cost & Schedule

3.1 Bill of Materials (Cost Analysis)

DigiKey:

Module	Model	Cost	Quantity	Subtota.
Raspberry Pi 5GB	5	N/A	1	Owned
ToF Sensor	VL53L7CX	$19.34	1	
STM32 Microcontroller	STM32H743VIT6	$10.09	1	
GPS Chip WisBlock GNSS GPS Module	L76L-M33 RAK12501 (GPS: Quectel L76K)	$10.57 (2) $8.90	1	*[19]* *[20]*
Magneometer	IIS2MDC	$1.71	1	*[21]*
Headphones				
Microphone	ICS-43434	$2.98	1	*[25]*
Button	Momentary	$1.25	1	*[26]*

	Button - Panel Mount (Red)			
RC (Debounce)	TBH	~$1-2/$0	1-2	ECE 445 Lab!
Battery	21700 5000mAh Battery for TX16S and TX12 MKII	$31.24 (total after shipping and taxes)	1 ~ 2	*[28]*
Buck Converter (7.4V to 5V)	MAX20410AFOC/VY+	$6.42	1	https://www.digikey.com/en/products/detail/analog-devices-inc-maxim-integrated/MAX20410AFOC-VY/17885186
Buck Converter (7.4V to 3.3V)	TPS563207	$1.33	1	*[30]*
Charging IC	MP2639C	$3.84 + others = $11.53	1	https://www.monolithicpower.com/en/mp2639c.html?srsltid=AfmBOorCyJdQaJCv-pw7-YBy9y2y6PIDhBRbbt6zK9ROoywcJMUZV3LR
PCB Shipping and Fabrication	N/A	$10 ~ $15	1	N/A
Head Connector (40 Position Right Angle)	PPPC202LJBN-RC	$2.43	1	*[31]*
Main to ToF + Vibration Motor Ribbon Cable (Side, Short)	uxcell IDC 14 Pins Connector Flat Ribbon Cable Female Connector	$8.99	1	https://www.amazon.com/gp/product/B07FKKP2TK/ref=sw_img_1?smid=A1THAZDOWP300U&th=1

	Length 30cm 2.54mm Pitch,5pcs (Amazon)			
Main to ToF + Vibration Motor Ribbon Cable (Middle, Long)	uxcell IDC 14-Pin Female to Female Connector Hard Driver Flat Ribbon Cable (Amazon)	$7.19	1	https://www.amazon.com/gp/product/B00Y20XVZU/ref=ox_sc_act_image_1?smid=A1THAZDOWP300U&psc=1
Main to ToF + Vibration Motor Headers	302-S141 (CONN HEADER VERT 14POS 2.54MM)	$0.30	3-6	*[39]*
Surface Mount Components (Capacitor, Resistor, Diodes, Etc)	TBH	$5 <	TBH	N/A
Capacitor: YAGEO CAP CER 0.1UF 16V X7R 0603	CC0603KRX7R7BB104	$0.006	27	$0.16
Capacitor CAP CER 22UF 16V X5R 0603 KYOCERA	0603YD226MAT2A	$0.47	4	$1.88

AVX				
CAP CER 18PF 16V X8R 0603 KEMET	C0603C180K4HACTU	$0.10	4	$0.4
CAP CER 10UF 16V X5R 0603	GRM188R61C106MAALD	$0.22	6	$1.32
CAP CER 4.7UF 16V X5R 0603	CL10A475KO8NNNC	$0.08	3	$0.24
CAP CER 2.2UF 16V X5R 0603	CL10A225KO8NNNC	$0.08	3	$0.24
CAP CER 0.22UF 25V X7R 0603	GRM188R71E224KA88D	$0.10	1	$0.10
CAP CER 47UF 16V X5R 1206 TDK Corporation	C3216X5R1C476M160AB	$0.81	6	$4.86
CAP CER 33PF 25V C0G/NP0 0603 KYOCERA AVX	KGM15ACG1E330JT	$0.10	1	$0.10
KA-TR CAP CER 0.47UF 25V X7R 0603 Taiyo Yuden	TMK107B7474	$0.12	1	$0.12
CAP CER 1.5UF 16V X7S 0603 TDK	CGA3E1X7S1C155K080AC	$0.27	1	$0.27

Corporation				
CAP CER 100PF 25V C0G/NP0 0603 KYOCERA AVX	KGM15ACG1E 101JT	$0.10	1	$0.10
CAP CER 0.15UF 25V X7R 0805 KYOCERA AVX	KGM21ER71E1 54KU	$0.13	1	$0.13
LED YELLOW CLEAR 0603 SMD Würth Elektronik	150060YS75000	$0.15	8	$1.20
LED GREEN CLEAR 0603 SMD Würth Elektronik	150060GS75000	$0.15	6	$0.90
ESD9B5.0ST5G	TVS DIODE 5VWM SOD923 onsemi	$0.10	1	$0.10
SMF3.3 Littelfuse Inc. TVS DIODE 3.3VWM 6.8VC SOD123F	F7701CT-ND	$0.49	1	$0.49
Diodes Incorporated TVS DIODE 5VWM 14VC SOD323	DESD5V0S1BA -7	$0.10	1	$0.10

Murata Electronics FERRITE BEAD 120 OHM 0603 1LN	BLM18SG121TN1D	$0.13	1	$0.13
SENSOR MR I2C/SPI 12LGA STMicroelectronics	IIS2MDCTR	$2.56	1	$2.56
Texas Instruments IC REG BUCK ADJ 3A SOT563	TPS563207DRLR	$0.32	1	$0.32
SENSOR OPTICAL I2C STMicroelectronics	VL53L7CXV0GC/1	$8.59	3	$25.77
CONN HDR 40POS 0.1 GOLD PCB R/A Sullins Connector Solutions	PPPC202LJBN-RC	$2.43	11	$2.43
CONN HDR 10POS 0.1 TIN PCB Sullins Connector Solutions	PPTC052LFBN-RC	$0.60	1	$0.60
CONN HDR 6POS 0.1 TIN PCB Sullins	PPTC061LFBN-RC	$0.42	1	$0.42

Connector Solutions				
CONN UMC JACK STR SMD TE Connectivity Linx	CONMHF1-SMD-T	$0.20	1	$0.20
FIXED IND 47NH 400MA 0.6OHM SMD Murata Electronics	LQG18HN47NJ00D	$0.16	1	$0.16
MOSFET P-CH 30V 4.3A SOT23-3L Alpha & Omega Semiconductor Inc.	AO3407A	$0.46	1	$0.46
TRANS NPN 40V 0.6A SOT23-3 Diotec Semiconductor	MMBT2222A	$0.13	1	$0.13
RES 10K OHM 1% 1/10W 0603 YAGEO	RC0603FR-0710KL	$0.10	7	$0.70
RES 5.1K OHM 1% 1/10W 0603 YAGEO	RC0603FR-075K1L	$0.10	1	$0.40
RES 300 OHM 1% 1/10W 0603 Vishay Dale	CRCW0603300RFKEAC	$0.10	1	$0.80

RES 4.7K OHM 1% 1/10W 0603 YAGEO	RC0603FR-074K7L	$0.10	4	$0.40
RES SMD 1K OHM 1% 1/8W 0603 Vishay Dale	CRCW06031K00FKEA	$0.10	1	$0.10
RES 32K OHM 1% 1/10W 0603 KOA Speer Electronics, Inc.	RN73R1JTTD3202F50	$0.10	1	$0.10
RES SMD 40K OHM 0.1% 0.15W 0603 Vishay Dale Thin Film	PAT0603E4002BST1	$0.60	1	$0.60
Estimated Total: $160 ~ $200				

3.2 Schedule

Phase Iteration	**Expectation**	**Tentative Schedule**	**Expected Amount of Time**

0 (Pre-Setup)	Getting software and hardware components ready for future implementations	- Hardware:
		  - Components Picking
		  - Schematics
		  - Layout
		  - Routing
		
		- Software:
		  - Machine Learning Model Selection & Testing
		  - Core Algorithm Pseudocode Design & Draft
			1 week (9/22-9/27)

		Implementation for phase 1 - Select accurate map APIs & Testing - Install all the IDEs, modules, and softwares	

1 (Naive Implementation)	Obstacle Avoidance	- Testing Basic Functionality:
		    - Being able to detect obstacles
		    - Generate Depth map for 2 8x8 arrays
		- Basic Breadboard Circuit
		    - ToF Sensor(s)
		    - Vibration Motor(s)
		    - GPS (Basics)
		    - MCU Development Board
		- Hardware:
		    - PCB Schematics Adjustment
		    - PCB Layout Adjustment
		    - PCB Routing Adjustment
			1-2 weeks (9/28-10/6)

2 (Object Recognition)	Obstacle Avoidance + Object Detection	- ML: (Testing Basic Functionality)
		    - Being able to classify basic pedestrian street signs and crosswalk lights.
		    - Being able to classify basic outdoor objects: Car, Bicycle, Motorcycle.
		    - Being able to classify basic indoor objects: Door, Chair, Elevator, Desk, Stairs.
		- Embedded:
		    - The core algorithm should take the output from the object recognition module
			2-3 weeks (~10/6 - 10/27)

		and provide output feedback to the audio module. - Magneometer - GPS (Advanced) - Audio Output - Audio Input - Final Hardware Adjustment - Possible PCB design updates (based on how well the first set of PCBs perform)	
3 (GPS Implementation)	Street Navigation	- Testing Basic Functionality: - The map API and GPS in tandem should lead the user to the set destination safely. - Should work within the campus domain via walking. Extreme cases include consideration of riding transportation to reach far destination. - Hardware Requirement: - MCU should handle all the system requirements without faults. - The battery should maintain at least 3 hours with all the systems working together.	4+ weeks (~10/27-End of Semester)

4. Ethics and Safety

User Data Privacy:

ACM code of ethics part 1.6 highlights that the responsibility of respecting privacy applies to computing professions, and computing professionals should take special care for privacy when merging data collections [7]. To satisfy and comply with ACM code standards, for any sensory input data, we will be asking consent to the user. Additionally, we won't be saving user's private locations by default, but such functionality can be enabled under the user's will to activate smart destination mode selection. Lastly, we will never be asking for a user's personal data such as name, phone number, and home address by default as it is not required for the core navigation functionality of the vest and ensures privacy.

Campus and Lab Policies:

As ECE students from University of Illinois Urbana Champaign, we will abide by the laboratory safety rules highlighted under Division of Research Safety. Based on Section 4.4, we will be following the minimum attire regulations for labs (PPE): *Closed-toe shoes, long pants (or equivalent), and safety glasses [8].* We will also ensure to follow the emergency procedures that deal with fire extinguishers, emergency power shut-offs, first aid kits, and eyewash stations.

User Safety:

The IEEE code of ethics states that we need to *"To uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities" and "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices" [6]*. In order to satisfy these standards, we must create our product to be as accurate as it can be, as inaccuracies may lead to injuries on those who are relied on.

Battery Safety: Our device should ensure the following configurations to abide by the IEEE 1725 $5.2.1 requirements [34]:

Feature	Requirement / Specification

Battery Chemistry	Lithium-Ion
Voltage	5v
Capacity Limit	[5000 mAh, 10000mAh]
Battery Label	Labels showing: Wh and Voltage
Operating Temperature	[0℃, 40 ℃], [32℉, 104℉]
Charging Temperature	[0℃, 35 ℃], [32℉, 95℉]
Physical Enclosure	Battery Ensure that can protect the main battery from physical impact

5. Sources and References

[1] A. Karpov, R. Yusupov, I. Mporas, et al., "Human–Computer Interaction for Users with Disabilities: A Systematic Review of Assistive Robotics," Journal of Signal Processing Systems, 2024. doi: https://doi.org/10.1007/s11265-023-01898-0

[2] N. M. Alotaibi, A. S. Alghamdi, M. S. Alzahrani, M. S. Alharbi, M. A. Almutairi, et al., "A smart wearable device for visually impaired persons based on deep and machine learning models," Journal of Imaging, vol. 10, no. 4, p. 82, 2024. doi: https://doi.org/10.3390/jimaging10040082

[3] M. A. Chavarria, L. M. Ortiz-Escobar, B. Bacca-Cortes, V. Romero-Cano, I. Villota, et al., "Challenges and opportunities of the human-centered design approach: Case study development of an assistive device for the navigation of persons with visual impairment," JMIR Rehabilitation and Assistive Technologies, vol. 12, p. e70694, 2025. doi: https://doi.org/10.2196/70694

[4] Y. Gao, D. Wu, J. Song, X. Zhang, B. Hou, H. Liu, J. Liao, and L. Zhou, “A wearable obstacle avoidance device for visually impaired individuals with cross-modal learning,” Frontiers in Neuroscience, vol. 19, p. 11933268, 2025. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11933268

[5] C. He and P. Saha, “Investigating YOLO models towards outdoor obstacle detection for visually impaired people,” arXiv preprint arXiv:2312.07571, 2023. doi: https://doi.org/10.48550/arXiv.2312.07571

[6] IEEE, “Code of ethics (Policies section 7.8),” [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8. [Accessed: Sep. 16, 2025].

[7] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," Section 1.6, "Respect privacy." ACM, adopted June 22, 2018. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: Sept. 18, 2025.]

[8] University of Illinois at Urbana-Champaign, Division of Research Safety, “Laboratory Safety Guide,” Section 4.4 “Personal Protective Equipment (PPE)” and Section 4.6 “Emergency Planning,” revised July 2025. [Online]. Available: https://drs.illinois.edu/site-documents/LaboratorySafetyGuide.pdf. [Accessed: Sept. 18, 2025.]

[9] STMicroelectronics, “VL53L5CX: Time-of-Flight 8x8 multizone ranging sensor with wide field-of-view,” Datasheet, Figure 1, pp. 10, 2022. [Online]. Available: https://www.st.com/resource/en/datasheet/vl53l5cx.pdf

[10] Micro Center, “Raspberry Pi 5 4GB,” Micro Center product page. [Online]. Available: https://www.microcenter.com/product/673712/raspberry-pi-5-4gb?src=raspberrypi. [Accessed: Sept. 18, 2025.]

[11] Adafruit Industries, “Vibrating Mini Motor Disc, Product ID 1201,” Adafruit product page. [Online]. Available: https://www.adafruit.com/product/1201. [Accessed: Sept. 18, 2025.]

[12] Texas Instruments, “DRV8210: 12-V, 1.76-A H-Bridge Motor Driver with auto sleep mode,” TI product page. [Online]. Available: https://www.ti.com/product/DRV8210. [Accessed: Sept. 18, 2025.]

[13] RF Wireless World, “4G LTE Packet Loss Rate: Formula and Calculation,” RF Wireless World, 2023. [Online]. Available:

https://www.rfwireless-world.com/terminology/4g-lte-packet-loss-rate.html. [Accessed: Sept. 18, 2025]

*[14] STMicroelectronics, "VL53L5CX-V0GC/1 - Time-of-Flight 8x8 multizone ranging sensor with wide field of view," *STMicroelectronics estore*. https://estore.st.com/en/vl53l5cxv0gc-1-cpn.html [Accessed: Sep. 18, 2025]*

*[15] STMicroelectronics, "VL53L5CX-SATEL/CPN - VL53L5CX Satellite Board," *STMicroelectronics estore*. https://estore.st.com/en/vl53l5cx-satel-cpn.html [Accessed: Sep. 18, 2025]*

[16] Voyeglobal, "Google Maps Data Usage: How Much Data Does Google Maps Use?," Voyeglobal, 2024. [Online]. Available: https://voyeglobal.com/google-maps-data-usage/#:~:text=Basic%20Navigation:%20If%20you're,updates%20or%20directions%20while%20offline. [Accessed: Sept. 18, 2025].

*[17] Digi-Key Electronics, "STM32H750VBT6 - STMicroelectronics," *Digi-Key Electronics*. https://www.digikey.com/en/products/detail/stmicroelectronics/STM32H750VBT6/9453371 [Accessed Sep. 18, 2025]*

[18] HighSpeedInternet.com, Mobile Hotspots: What Are Hotspots and How Do They Work?, HighSpeedInternet.com, 2025. [Online]. Available: https://www.highspeedinternet.com/resources/mobile-hotspots#:~:text=How%20fast%20are%20Wi%2DFi,or%205G%20will%20slow%20down. [Accessed: Sept. 18, 2025].

*[19] Digi-Key Electronics, "L76L-M33 - Quectel," *Digi-Key Electronics*. https://www.digikey.com/en/products/detail/quectel/L76L-M33/13923824 [Accessed Sep. 18, 2025]*

*[20] RAKwireless, "WisBlock RAK12501 GNSS Module," *RAKwireless Store*. https://store.rakwireless.com/products/wisblock-rak12501-gnss-module [Accessed Sep. 18, 2025]*

*[21] STMicroelectronics, "IIS2MDC - 3D magnetometer with 1.5 mA high-performance and low-power for industrial applications," *STMicroelectronics estore*. https://estore.st.com/en/products/mems-and-sensors/e-compasses/iis2mdc.html [Accessed Sep. 18, 2025]*

[22] Texas Instruments, PCM5100A: 2 VRMS DirectPath™, 100 dB Audio Stereo DAC with 32-bit, 384 kHz PCM Interface, Texas Instruments, 2025. [Online]. Available: https://www.ti.com/product/PCM5100A/part-details/PCM5100APWR. [Accessed: Sept. 18, 2025]

*[23] Digi-Key Electronics, "PAM8302AASCR - Diodes Incorporated," *Digi-Key Electronics*. https://www.digikey.com/en/products/detail/diodes-incorporated/PAM8302AASCR/4033281 [Accessed Sep. 18, 2025]*

[24] Tayda Electronics, Mylar Speaker 8 OHM 1 W (HY57MS0810-H8.4), Tayda Electronics, 2025. [Online]. Available: https://www.taydaelectronics.com/hardware-tools/speakers/mylar-speaker-8-ohm-1w.html. [Accessed: Sept. 18, 2025]

*[25] Digi-Key Electronics, "ICS-43434 - TDK InvenSense," *Digi-Key Electronics*. https://www.digikey.com/en/products/detail/tdk-invensense/ICS-43434/6140298 [Accessed Sep. 18, 2025]*

[26] SparkFun Electronics, Momentary Button - Panel Mount (Red) (COM-11992), SparkFun Electronics, 2025. [Online]. Available: https://www.sparkfun.com/momentary-button-panel-mount-red.html. [Accessed: Sept. 18, 2025]

[27] Analog Devices, MAX20408/MAX20410: Automotive 10 A, 3 V 36 V Input Synchronous Buck Regulator, Data Sheet, Rev. , Mouser Electronics. Accessed: Sep. 18, 2025. [Online]. Available: https://www.mouser.com/datasheet/3/1014/1/max20408.pdf

*[28] RadioMaster RC, "21700 5000mAh Battery for TX16S MKII," *RadioMaster RC*. https://radiomasterrc.com/products/21700-5000mah-battery-for-tx16s-mkii [Accessed Sep. 18, 2025]*

[29] Analog Devices / Maxim Integrated, "MAX20410EAFOC/VY+: 8 A Fully Synchronous Buck Converter, Input 3–36 V, Output 5 V," Analog Devices / Mouser, 2025. [Online]. Available: https://www.mouser.com/ProductDetail/Analog-Devices-Maxim-Integrated/MAX20410EAFOC-VY%2b?qs=sGAEpiMZZMsMIqGZiACxIRoU9mnTGarMTaXD529oJWRr2%2FFt6aC9yw%3D%3D. [Accessed: Sept. 18, 2025]

*[30] Texas Instruments, "TPS563207 3-V to 17-V Input, 3-A Synchronous Step-Down Voltage Converter datasheet," *Texas Instruments*. https://www.ti.com/lit/ds/symlink/tps563207.pdf [Accessed Sep. 18, 2025]*

[31] Sullins Connector Solutions, PPPC202LJBN-RC: 40-Position Right-Angle Gold Female Header Connector 0.100" Pitch, Sullins Connector Solutions / DigiKey, 2025. [Online]. Available: https://www.digikey.com/en/products/detail/sullins-connector-solutions/PPPC202LJBN-RC/776031. [Accessed: Sept. 18, 2025].

*[32] Raspberry Pi Ltd., "RP1 Peripherals Datasheet," *Raspberry Pi Datasheets*. https://datasheets.raspberrypi.com/rp1/rp1-peripherals.pdf [Accessed Sep. 18, 2025]*

*[33] Newark, "RPI 8MP Camera Board - Camera Board, 8MP, Raspberry Pi," *Newark Electronics*. https://www.newark.com/raspberry-pi/rpi-8mp-camera-board/camera-board-8mp-raspberry-pi/dp/32AJ8641 [Accessed Sep. 18, 2025]*

[34] CTIA Certification Program, Certification Requirements for Battery System Compliance to IEEE 1725 (Revision 2.5), Section 5.2.1, CTIA, Washington, DC, Apr. 2013. [Online]. Available: https://api.ctia.org/docs/default-source/default-document-library/ctia-certification-requirements-for-battery-system-compliance-to-nbsp-ieee-1725.pdf. [Accessed: Sept. 18, 2025].

*[35] J. Doe et al., "Title of Paper from arXiv," *arXiv preprint*. https://arxiv.org/html/2409.16808v1 [Accessed Sep. 18, 2025]*

[36] Analog Devices, MAX20408/MAX20410 Automotive 10 A, 3 V-36 V Input Synchronous Buck Regulator, Data Sheet, Rev. , Mouser Electronics. Available: https://www.mouser.com/datasheet/3/1014/1/max20408.pdf. Accessed: Sep. 18, 2025.

[37] STMicroelectronics, STM32H750IB Data Sheet, Rev. , STM32H750IB, STMicroelectronics. Available: https://www.st.com/resource/en/datasheet/stm32h750ib.pdf. Accessed: Sep. 18, 2025.

[38] Monolithic Power Systems, “MP2639C 2-Cell Li-Ion or Li-Polymer Switching Charger,” [Online]. Available: https://www.monolithicpower.com/en/mp2639c.html. [Accessed: 28 Sept. 2025].

[39] On Shore Technology Inc., "302-S141 Connector Header Vertical 14-Position 2.54 mm," ***DigiKey****, ED10522-ND, [Online]. Available:* *https://www.digikey.in/en/products/detail/on-shore-technology-inc/302-S141/2794233*

[40] Samtec, "IDSD-07-D-15.00-T-G IDC Socket Assembly," ***Mouser****, [Online]. Available:* *https://www.mouser.com/ProductDetail/Samtec/IDSD-07-D-15.00-T-G?qs=%252BZP6%2F%252BtExtBAGTjpj8Xr%252BA%3D%3D*

[41] GCT, USB4125, USB4130, USB4135: Product Specification (Rev. A3, Dec. 20, 2024), GCT Co., [Online]. Available: https://gct.co/files/specs/usb4125-spec.pdf

[42] F. Dostal, Selecting the Right Inductor Current Ripple, Analog Devices, Mar. 6, 2023. [Online]. Available: *https://www.analog.com/en/resources/technical-articles/selecting-the-right-inductor-current-ripple.html*

[43] Diodes Incorporated, DESD5V0S1BA — 5 V Bidirectional TVS Diode Datasheet, Rev. 3, Nov. 2018. [Online]. Available: https://www.diodes.com/assets/Datasheets/DESD5V0S1BA.pdf

[44] onsemi, ESD9B Series — Low Capacitance TVS Diode Datasheet, Rev. 8, Jan. 2023. [Online]. Available: https://www.onsemi.cn/pdf/datasheet/esd9b-d.pdf

[45] STMicroelectronics, VL53L7CX — Long-Distance 8×8 Multizone Time-of-Flight Ranging Sensor Datasheet, Rev. 3, Apr. 2023. [Online]. Available: https://www.st.com/resource/en/datasheet/vl53l7cx.pdf

[46] NXP Semiconductors, I²C-bus Specification and User Manual, Rev. 7, Oct. 2021. [Online]. Available: https://www.nxp.com/docs/en/user-guide/UM10204.pdf

[47] R. Talarico, Printed Circuit Board Design Considerations, Gonzaga University, 2015–2016. [Online]. Available: *https://web02.gonzaga.edu/faculty/talarico/ee406/20152016/documents/pcb_xilinx.pdf*

[48] Pasternack Enterprises Inc., PE51212 Active GPS Antenna – 26 dB Gain, SMA Male Connector, 3.0 V DC to 5.0 V DC. Pasternack Enterprises, Inc., Irvine, CA, USA, Product Datasheet, 2023. [Online]. Available: https://www.pasternack.com/images/ProductPDF/PE51212.pdf

[49] Texas Instruments. (2013). AN-1520: A Guide to Board Layout for Best Thermal Resistance for Exposed Packages (Rev. B). Retrieved from https://www.ti.com/lit/an/snva183b/snva183b.pdf

[50] Digi-Key "PCB Trace Width Conversion Calculator," Digi-Key, [Online]. Available: *https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-pcb-trace-width*

[51] Soleymani, Azita. "Thermal Design for Externally Worn Wearable Electronics." Electronics Cooling, July 9, 2025. Available:
https://www.electronics-cooling.com/2025/07/thermal-design-for-externally-worn-wearable-electronics

[52] IEC 62368-1, Audio/video, information and communication technology equipment – Part 1: Safety requirements, Table 9.2.5 https://www.china-gauges.com/common/down/name/5710ecb863a33.pdf

[53] B. Blackmore, R. Bornoff, and J. Parry, "Thermal Analysis of PCB Mounted Small Outline Packages," ***EE Times****, Nov. 22, 2015. [Online]. Available:*
https://www.eetimes.com/thermal-analysis-of-pcb-mounted-small-outline-packages/2/