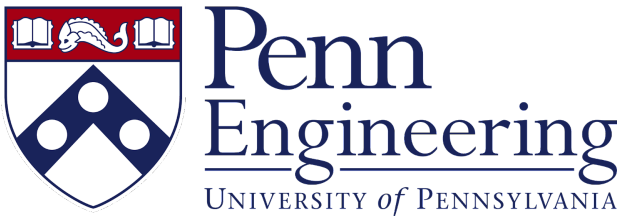

Autonomous Robot Performance Review

Prepared By: Alexander Koldy, Gene Liu, Darra Snow, and Charles Yoo

Submitted: December 21, 2023

Functionality

The purpose of the final lab was to design a semi-autonomous robot that could be controlled to complete various tasks. Our robot was designed to receive commands wirelessly over UDP and execute tasks autonomously. These tasks included: wall-following, police car tracking, real trophy tracking, fake trophy tracking, and stop. For the evaluation, little consideration was given to when certain commands should be sent to the robot. However, for the competition, this would have definitely been important in order to be successful. One way we thought of approaching the competition was to start by pushing the police car to a desired position (deep into the opposing side). Then we would command the robot to switch between real and fake trophy tracking. Since our robot could only push and not pull, we would likely command the robot to seek real trophies while on the opposing side (so that they could be pushed to our side) and to seek fake trophies while on our side (so that they could be pushed to the opposing side). We did not anticipate using the wall following functionality for the competition.

To complete the police car task, our team decided to use the vive coordinates of our robot and the police car. The robot is designed to use the vive coordinates to align itself with the police car and then proceed straight to push the car. For the evaluation, we designed the robot to push the police car indefinitely during this task, which was more than sufficient for the evaluation. However, for the competition we would likely push the police car until we reached a desired distance. This can be done relatively easily by comparing the starting position of the police car to its current position.

To complete the wall following task, our team set up the robot to move forward indefinitely until it comes in contact with a wall. When this happens, the limit switch is triggered and the robot reverses a little before turning 90 degrees to the left. It then repeats the process. This procedure was sufficient for the evaluation and would have likely carried over unchanged to the competition.

To complete the trophy tracking task, our team decided to use the phototransistors to detect the beacons. It then determines whether or not the beacon has a frequency matching the type of beacon it is looking for (which depends on whether the robot is searching for real or fake trophies). If the beacon is the correct one, the robot proceeds forward until it pushes the beacon. We were not able to fully implement this function for the robot due to time constraints and problems with the phototransistor circuit. However, if we did, we would have had to make

considerations regarding how to search for the beacon and which direction the robot should push the beacon.

Mechanical Design

We originally decided to try using our chassis from lab 4 to complete the tasks for the final lab. This design used a 2-wheel differential drive with a ball caster. This design featured two TT-motors, one SN754410 motor driver, and one Lipo battery. As seen in figure 1, the chassis for this design was relatively small and lightweight.

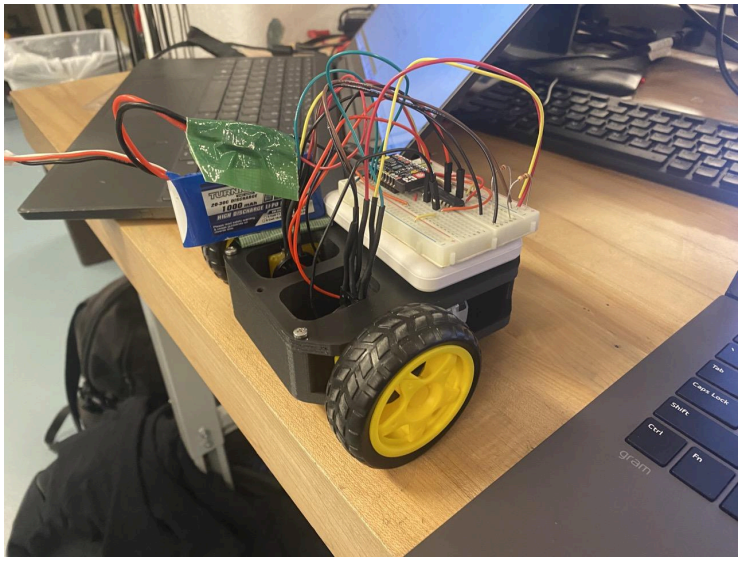

Figure 1. Original Chassis from Lab 4

One problem we found with this design is that the chassis was too small. It was difficult to find space to mount the new sensors and circuits that we would be using for the final lab. There was also no space to mount the vive sensors so that they would be visible to the lighthouse. Additionally, we found that two TT motors were not strong enough to push the police car. Therefore, we decided to switch to a 4-motor skid steering setup so that we would have enough torque to push the car. This also meant needing space to mount two more motors. To address these problems, we designed a new, larger chassis that had a design more tailored to the needs mentioned above. The final robot design and CAD model can be seen in figures 3 and 4 in the appendix. A dimensioned drawing of the final design can be seen in figure 5.

The reason we had used a differential drive in lab 4 is because of its simplicity. However, we found that the skid steering was not that much more difficult and performed just as well, if

not better than the differential drive. With the 4-wheel drive, our robot seemed to drive straighter than with the 2-wheel drive. We also considered using four mecanum wheels to allow for holonomic control. This would increase our maneuverability and make it easier to change directions without having to turn. However, we were concerned that the lower friction provided by mecanum wheels would make it difficult to push the police car. Therefore, we opted to continue using the wheels that we had used for lab 4.

Electrical Design

We initially were using an ESP-32 S2 Saola board to control our robot because it offered more ports than an ESP-32 C3. However, the board began experiencing overheating issues and had trouble connecting to the GM lab router. Because of this, we had to replace the S2 with a C3, which greatly reduced our flexibility with pin assignments. The robot was powered by two 7.4V LiPo batteries, one to power each motor driver, and one USB power bank used to power the C3.

In lab 4, we decided to drive the motors using an SN54410 motor driver. However, the voltage drop across this motor driver is significantly high, which reduces the overall efficiency of the design. For the final lab, we opted to use L298N motor drivers instead, which had smaller voltage drops and allowed us to run our motors at higher voltages. The enable signal for the motor drivers was sent using one pin and an inverter due to the limited number of pins available on our C3. The phototransistor signal conditioning circuit provided during lecture did not work for us so we made some modifications to it. This included sending the signal from the capacitor to ground and tuning the gains of the circuit. These changes can be seen in figure 6 in the appendix. We tried to tune the gains of this circuit to maximize the range of the beacon signals, however we were only able to get a range across the short distance of the track. This meant that we would not be able to see the beacon across the entire field and would need to adjust our beacon tracking software accordingly. The vive sensor circuits were the same as the ones provided during lecture and can be seen in figure 7 of the appendix. This circuit generally worked as expected but sometimes the x and y coordinates would be swapped. We did not find a great way of dealing with this in hardware so we decided to handle it in the software. The robot features three types of sensors to navigate around its surroundings. The first is a front-mounted beacon sensor, which consists of two phototransistor circuits separated by an opaque divider.

These sensors can be used to detect the direction of the beacons and code can be used to calculate the beacon frequencies and determine whether or not the trophy is real. The second is a front-mounted limit switch. This sensor can be used to determine when the robot has collided with a wall or trophy. The last one is a pair of Vive sensors, which are attached to the top of the robot. These sensors can be used to calculate the position of the robot within the field. Since we have 2 sensors, we can also calculate the heading of the robot. We originally intended to mount the ToF sensor on the front of our robot to be used to find the distance to the wall during wall following. However, when placed in the field, we found that the ToF sensor was not very reliable and would return false readings that messed with our turning mechanics. This appeared to occur the most when the robot approached the tape on the ground of the field. We were not able to resolve this issue so we decided to replace our ToF sensor with a limit switch. This limit switch would detect when the robot bumped into a wall and was used to trigger the turning procedure for our robot. This turning procedure was to reverse for a few seconds before turning 90 degrees to the left and continuing straight until the next wall. This ended up working a lot better than the ToF sensor because there was no noise to interfere with the limit switch.

During this lab, we encountered a lot of problems with noise that made the electrical design more difficult than originally anticipated. This was especially true for the phototransistor circuits as they were very difficult to get tuned properly and involved lots of testing and adjustments. It was also a noticeable issue when working with the ToF sensor. To deal with these issues we opted to go with a different hardware sensor that would not be as susceptible to noise. Another way of dealing with this is by using code to get rid of erroneous signals.

Processor and Code Architecture

A single ESP-32 C3 was used to control the entire system. 9 GPIO pins from the MCU were used: 2 for IR sensor input, 2 for vive sensor input, 1 for the wall following limit switch, 2 for motor PWM, and 2 for motor direction. The way these pins are assigned can be seen in figure 2. The logical motor inputs were wired together for each side of the robot, such that the wheels on one side would always turn in the same direction.

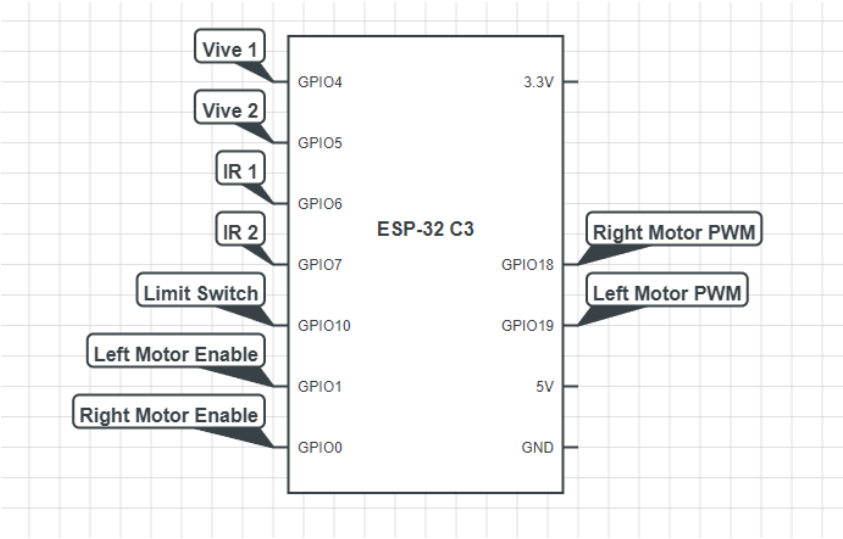

Figure 2. Block Diagram of ESP-32 C3 Pin Layout

A similar overall structure to the lab 4 software was used, where the microcontroller hosts a web server in which users can access on the same network to control the bot. However, instead of the low level directional control implemented previously, for this project there are instead several buttons that correspond to the evaluation tasks, including stop, wall following, fake/real beacon tracking, and police car pushing. Upon clicking one of these buttons, the mode of operation changes, causing the robot to perform the specified task within each loop. An ESPNOW packet is sent following any change in mode. In addition, a one hertz timer executes a function to handle all UDP communication, including receiving all other robot coordinates (including the police car), as well as sending our own vive coordinates.

Our wall following strategy includes driving forwards until hitting the wall with the limit switch, upon which the robot reverses for a short duration then turns 90 degrees. This process is repeated until the robot makes a full circuit around the field. The turning behavior is timing based and involves using delay functions. It is therefore affected by things like print statements and battery voltage and must be adjusted accordingly. In practice, this did not really need to be adjusted too much between runs.

To locate and push the police car, the UDP-received vive coordinates of the police car and the vive coordinates were used. The robot is programmed to turn in place until the heading between the two vive sensors on the robot align with the heading between the front vive sensor on the robot and the vive sensor on the police car. Once they are aligned, the robot goes forward until it makes contact with the police car and pushes it.

To track the beacons, the code utilizes 2 IR sensor inputs and bang bang tracking to turn towards the beacon and navigate towards it. The code also calculates the beacon frequency based on the period of the received signal. If the frequency is correct, then the robot travels to the beacon but if it isn't then the robot continues searching for the correct beacon. Since our phototransistors could only detect beacons along the short distance of the field, we would have had to add additional behavior for the case where the robot cannot detect any beacons. For this behavior, we would have likely had the robot rotate in place to scan a ~180^{o} cone in front of itself. If it doesn't detect the correct beacon, then it would move forward a distance equivalent to the short distance of the field and repeat the process until it finds the beacon.

Most of the code infrastructure and sensor inputs worked well. However, there was an error with UDP vive coordinate broadcasting that caused our controller to crash whenever a packet would be sent. This is likely due to a race condition or timing issue occurring in conjunction with the UDP function, which runs on a 1hz timer. In addition, the sensor readings were often noisy, which a combination of circuit maintenance and sliding window averaging helped to reduce.

Retrospective

The final project provided an excellent opportunity to synthesize the skills used in the previous four labs into one final, dynamic project capable of accomplishing different tasks. It was a refreshing experience to identify parts that would be necessary for functionality, purchase them, and assemble them according to our own design. The class was excellent about providing example code and component structures. Examples, when demonstrated, were also a significantly helpful resource to understanding and applying the lessons to practical application. The class was a fast-paced environment that required constant attention and time-commitment. This made it difficult to keep up with assignments and material at times.

Perhaps the best part of the class was the accessibility to help. There were frequent TA hours and advice on ED was provided quickly. That being said, resources for individual students were limited. It would have been a significant boon to be able to schedule one-on-one time to talk with a TA about lab assignments.

Appendix

Table 1. Bill of Materials

Bill of Materials			
Quantity	Item	Part ID	Cost Per
2	7.4 V Lipo Battery	B06ZYRCPS3	$17.63
2	Vive Photodiode	PD70-01C/TR7	N/A
1	VL53L4CX ToF Sensor	5425	N/A
2	LTR 4206 IR Phototransistor	160-1988-ND	N/A
1	ESP32-C3	K056	N/A
4	DC Gearbox Motor - "TT Motor"	AD479	$1.99
4	Wheel (65 mm diameter)	AD480	$0.50
2	L298N Motor Driver	AD624	$4.29
1	USB Charger	HYD009	$11.99
		Total Cost: $65.79	

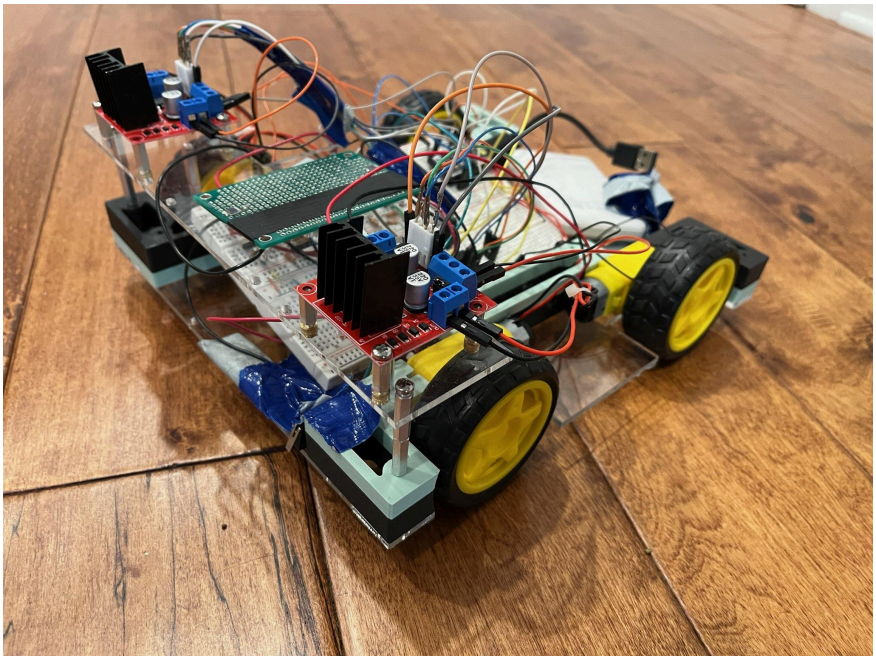

Figure 3. Final Design

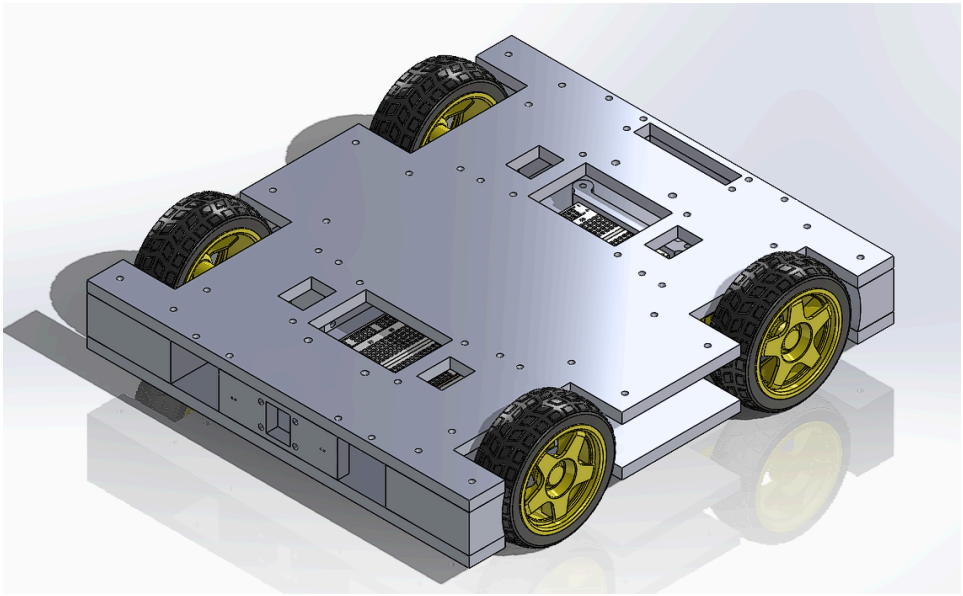

Figure 4. Chassis CAD Design

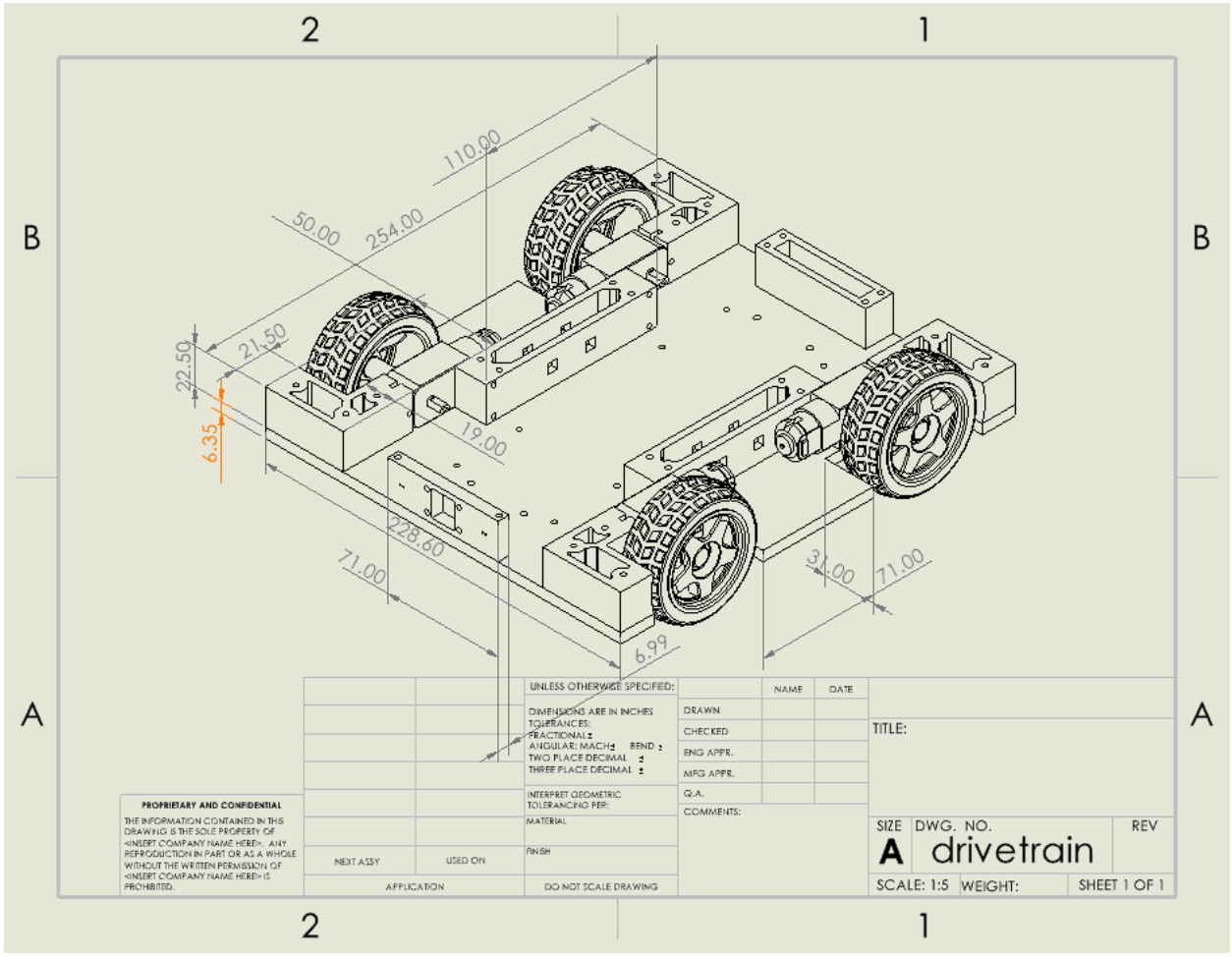

Figure 5. Dimensioned CAD Drawing

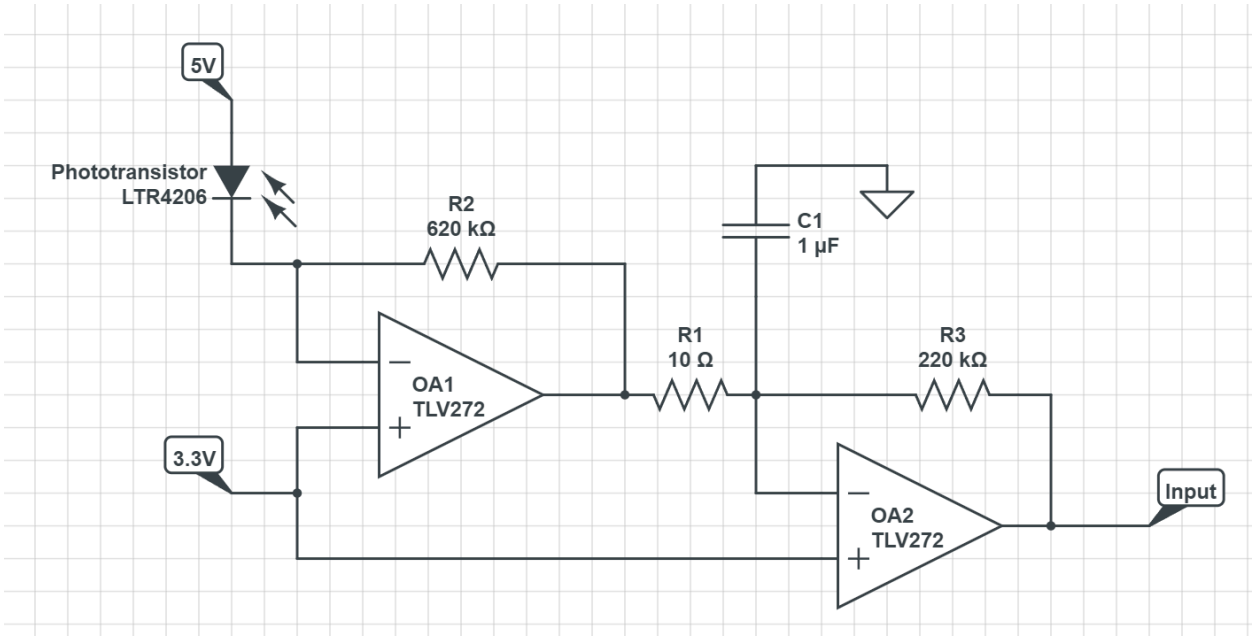

Figure 6. Phototransistor Sensing Circuit

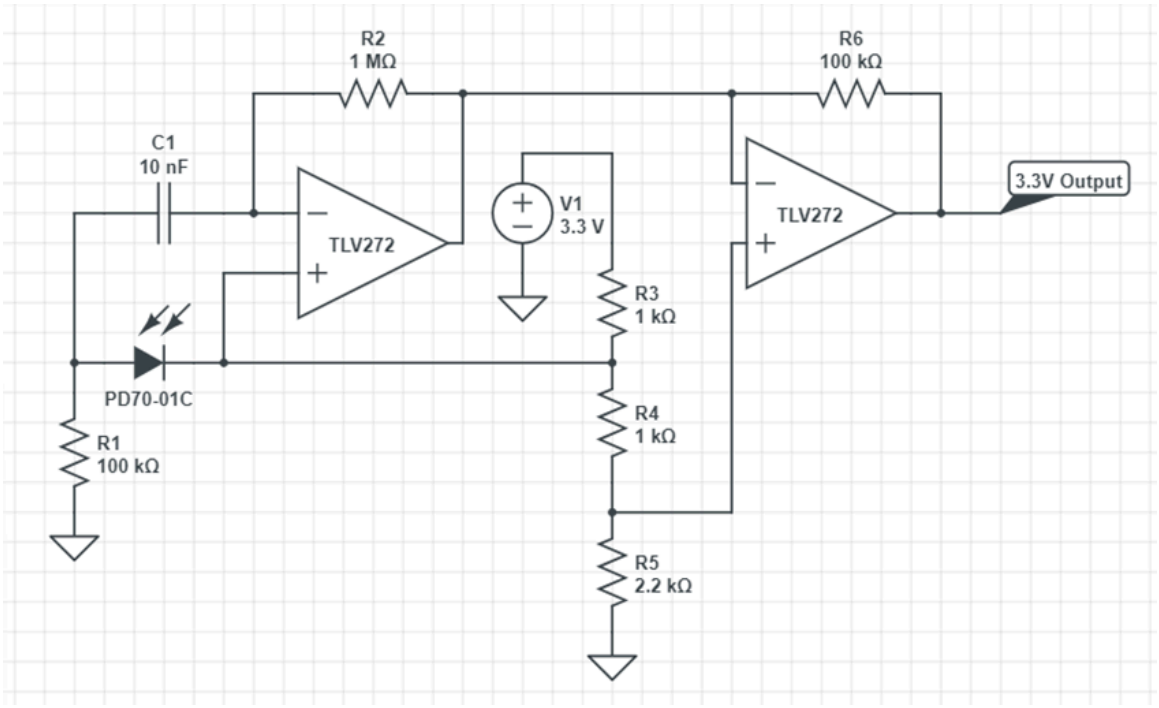

Figure 7. Vive Sensing Circuit

Table 2. Data Sheet Repository

Data Sheet Repository	
7.4 V Lipo Battery	https://www.batteryequivalents.com/lithium-polymer-batteries/zeee-sdl-9543125-2s-7-4v-50c-5200-mah-lipo-battery.html
USB Charger	https://miadybattery.com/miady-5000mah-mini-portable-charger/
L298 Motor Driver	https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf