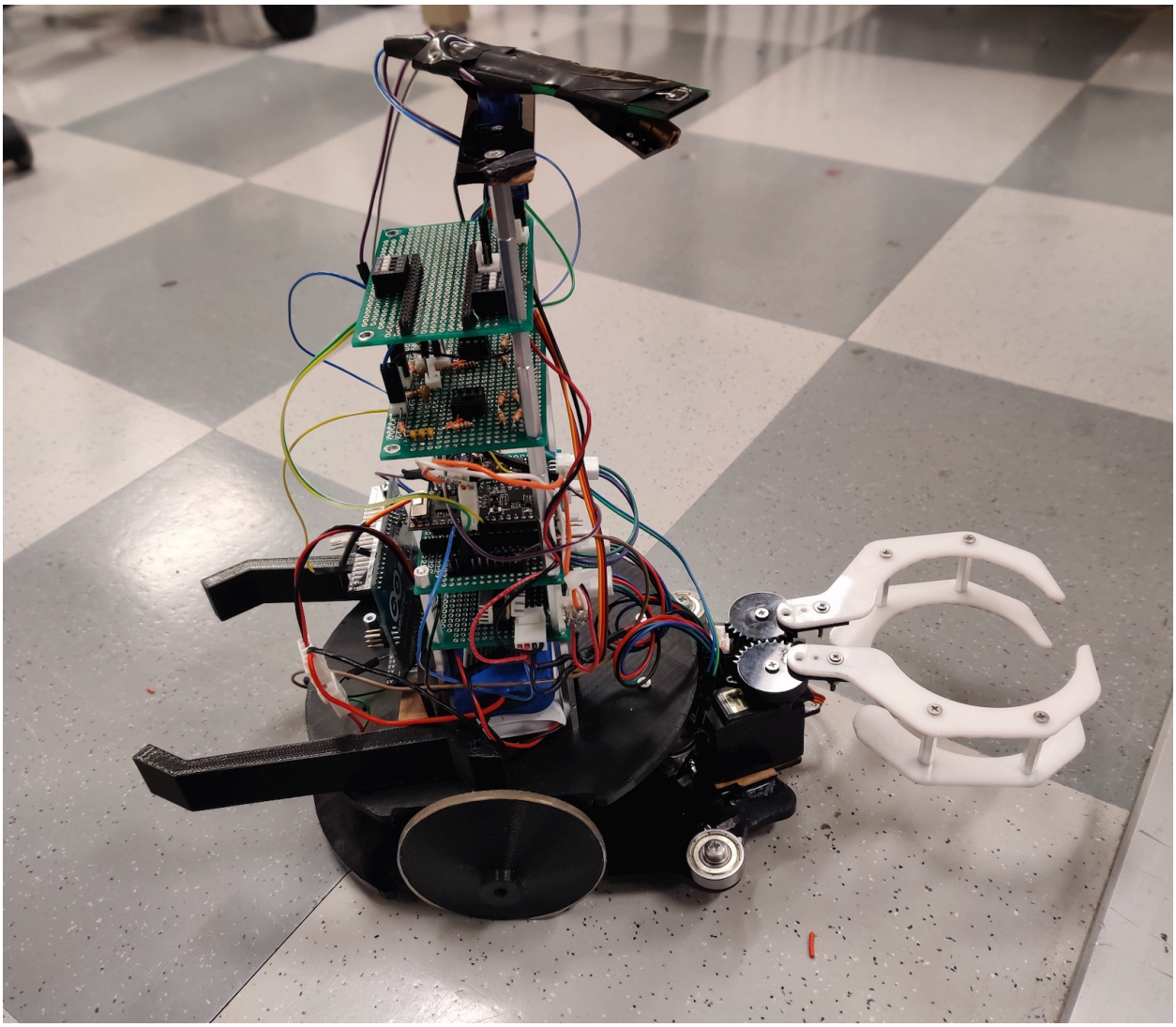
MEAM 510: Final Report

Team 3: Archit Hardikar, Rohit Bhikule, Saptarshi Sadhu



Drive: Differential Drive - 2 Wheels
Chassis: 3D Printed, laser cut jaw.
Battery: li-ion (18650) 7.4V system, boosted to 12v for motors. Stepped down to 5v for various circuits.
Motors: 2x Nema17 stepper motors

A. Functionality

Out approach for the game was to go 100% manual. We used UDP to control the bot as ESP NOW was unreliable.
Our strategy was to pick up the beacon first then head for the cans and when we are near the completion, we would direct the robot the doubling zone on the other side to remove enemy cans/beacon.

Initially, our aim was to use two vive circuits to determine location and orientation, but that system never worked due to noise and syncing issues. I changed to an alternate approach described below in detail.

Our robot used stepper motors and for motion and we used same subroutines for forward, left, right and backward within high functionality such as wall following beacon tracking etc.
Eg.

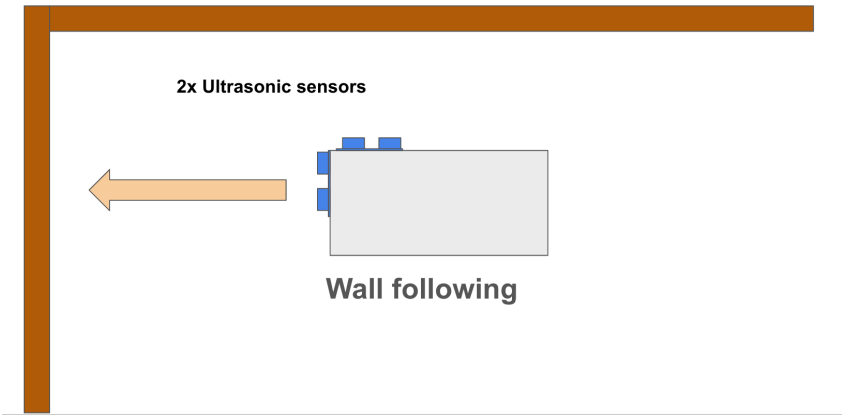
```
void forward(int s){
  int i = 0;
  digitalWrite(2,1);
  digitalWrite(9,0);
  for (i=0;i<s;i++){.
    digitalWrite(4,1);                  // single pin 4 controls steps for both stepper motors
    delayMicroseconds(2000);
    digitalWrite(4,0);
    delayMicroseconds(2000);        // subroutine to move bot forward
  }
}


void left(int angle){
  int i = 0;
  digitalWrite(2,0);
  digitalWrite(9,0);
  for (i=0;i<angle*1.16;i++){
    digitalWrite(4,1);
    delayMicroseconds(1000);
    digitalWrite(4,0);
    delayMicroseconds(1000);
  }
}                                       // subroutine to move bot left
  }
```

A.1 Wall Following:

We used two ultrasonic distance sensors for wall following. One in the front one at the right side. Here is the logic for it:

if(f>25){ forwardwall(250); ***>>> first goes straight if front sensor is greater than 25 cm then enters if loop.***
if(r<15) leftwall(30); ***>>> turns left on place if right side sensor reads below 15 cm***
if(r>20) rightwall(15); ***>>> turns right on place if right side sensor reads above 20 cm***
}
if(f<25){ leftwall(95);}. ***>>> turns left by 90 degrees if front sensor reads less than 25***
}



Video link: https://www.youtube.com/watch?v=WXn_kjBpCTk

A.2 Beacon Tracking

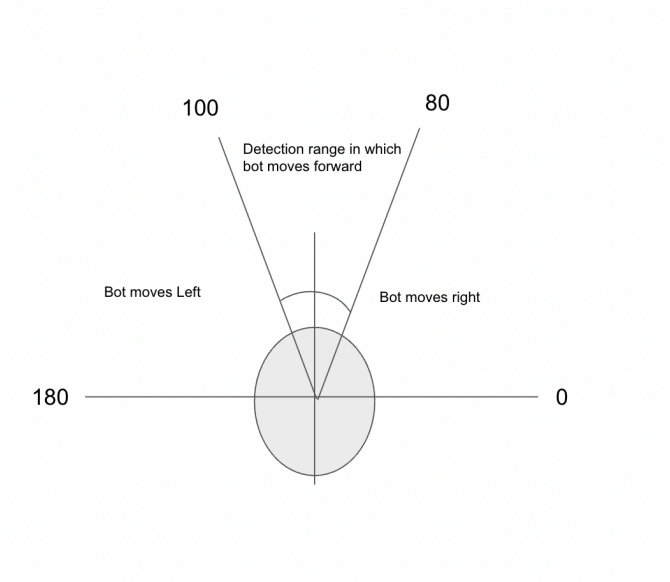
For beacon tracking we used one photodiode mounted on a servo. The servo continuously swept from 0 to 180 degrees and back. For each degree turn it sees if there is a signal from the beacon detection circuit. We used the main esp 32 board for the servo sweep and an Arduino UNO to detect a beacon signal and its pulse width and translate it to a digital high if the detected signal was in the frequency range we wanted.

Code excerpt on Main ESP board for moving at the detected angle:

```
if(digitalRead(inputfromarduino)==HIGH){
    if(i<80){right(90-i);}
    if(i>100){left(i-90);}
    if(i>=80&&i<=100){forward();}.
```

If frequency is detected within the range of 80-100 degrees, the bot moves forward. Refer to the diagram below for relative angle from the bot.

https://youtube.com/shorts/Iw9XT9fJOMw?feature=share



Code on the Arduino board which receives beacon signal and translates to digital high for positive detection for the ESP main board.

```
void setup() {
  pinMode(11, INPUT);
  pinMode(10, OUTPUT);
}

void loop() {
    digitalWrite(10,LOW);
    int t1 = pulseIn(11,HIGH);
    int t2 = pulseIn(11,LOW);
    int t=t1+t2;
    int freq = 1000000/t;
    if (freq>680 && freq<720){digitalWrite(10,HIGH); delay(5);}
}
```

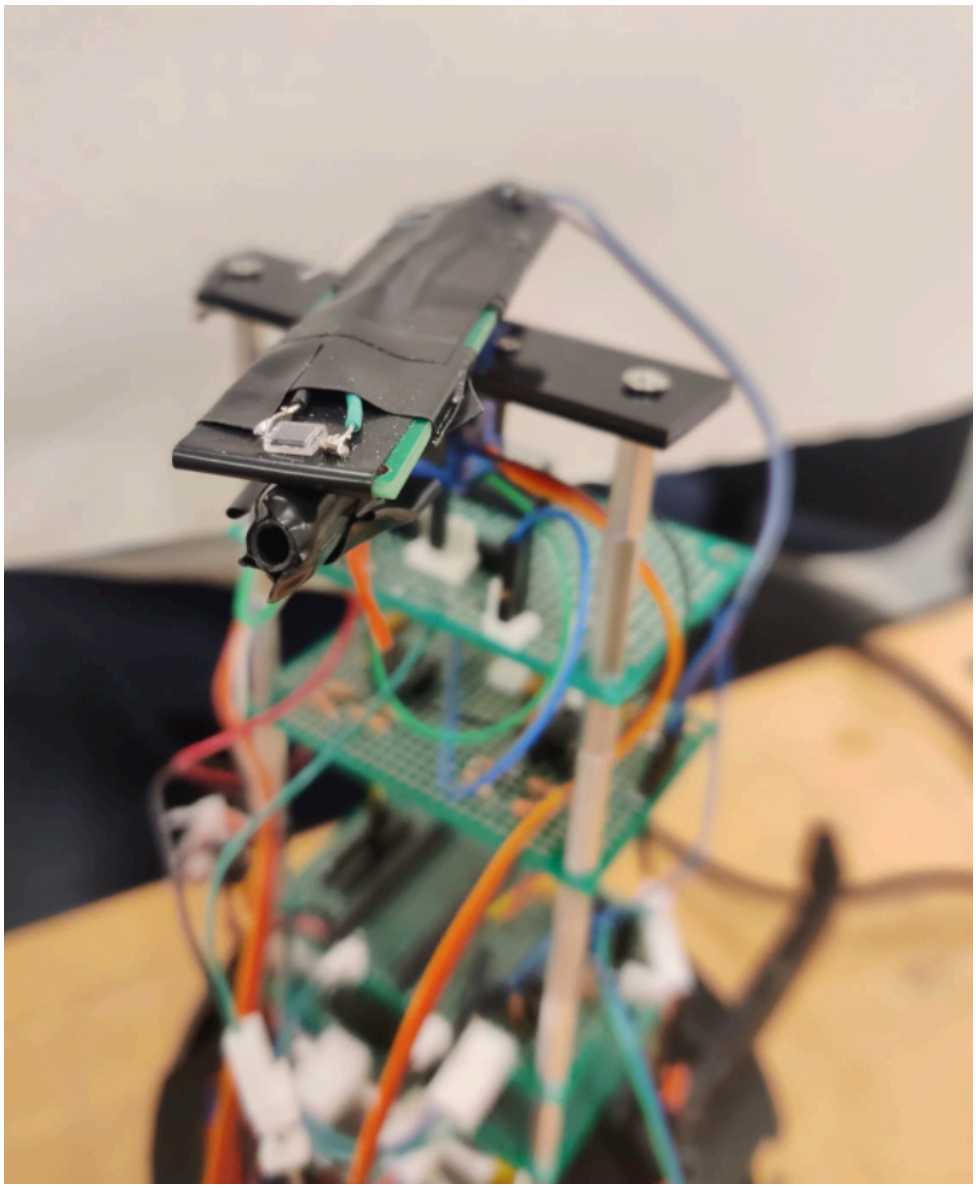
A.3 Vive localization

For localization using the vive we used a single PD-70 photodiode mounted on a servo arm. The arm swept continuously from 0 to 180 degrees and back. At each degree of change in angle the bot calculates the distance between the can coordinates with that of the sweeping photodiode.

Each incoming value is compared and the minimum is stored along with its corresponding angle of sweep.
After this we have the direction and exact angle at with the distance to the can is minimum. The bot then excecutes an exact rotaion and translation from the above excercise. Below is an excerpt from the code describing the above logic.

https://youtu.be/daiU30mP28w

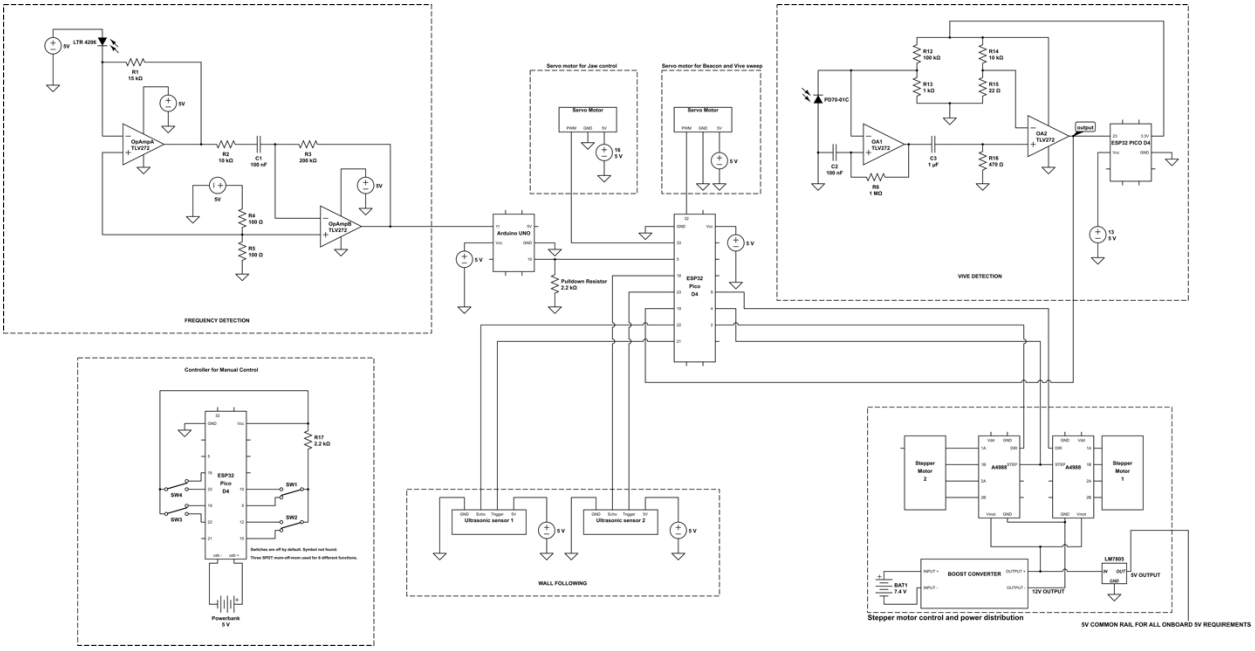
For vive data transmission we used a UDP and

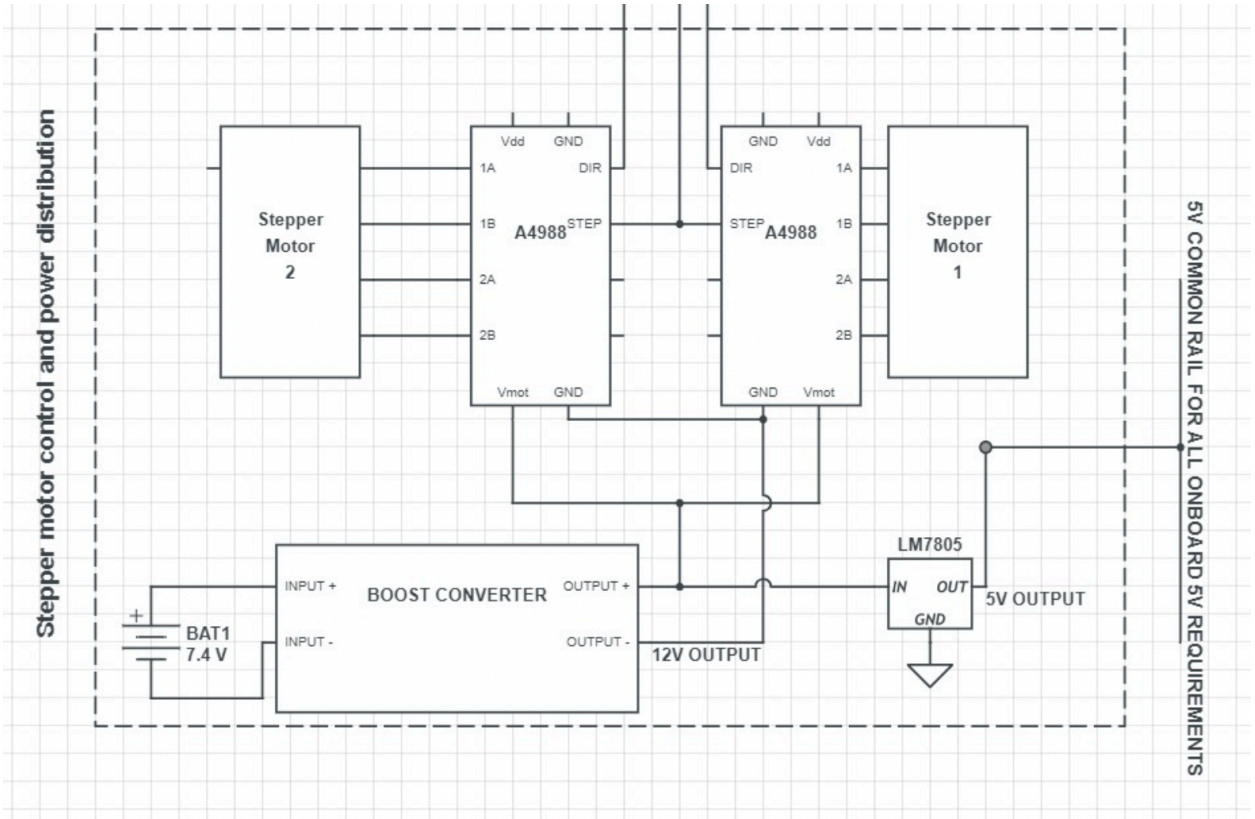


Vive signal detector and IR photodiode mounted on the same servo arm for sweep and detect angle strategy

B. Electrical Design

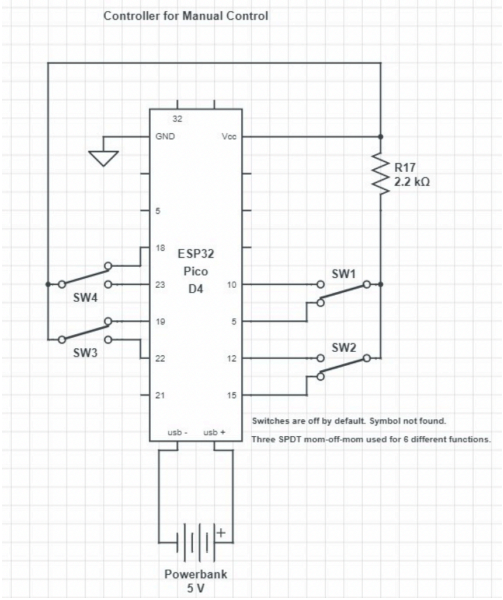
Master circuit diagram:



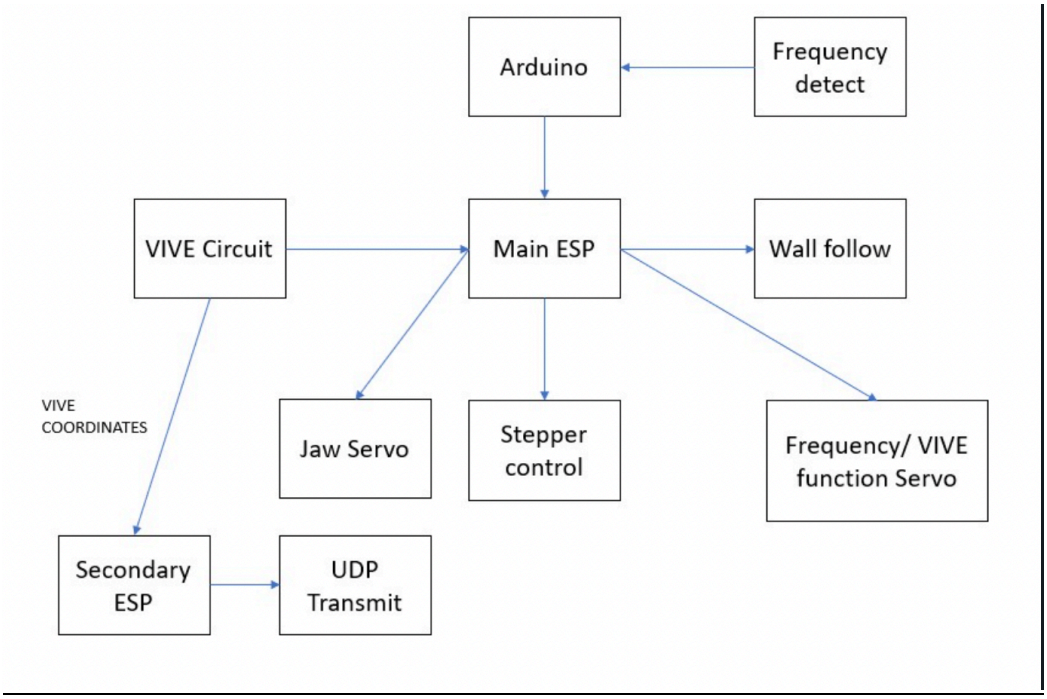


Onboard power distribution and stepper motor. Note: step and direction pins connect to main ESP32 board

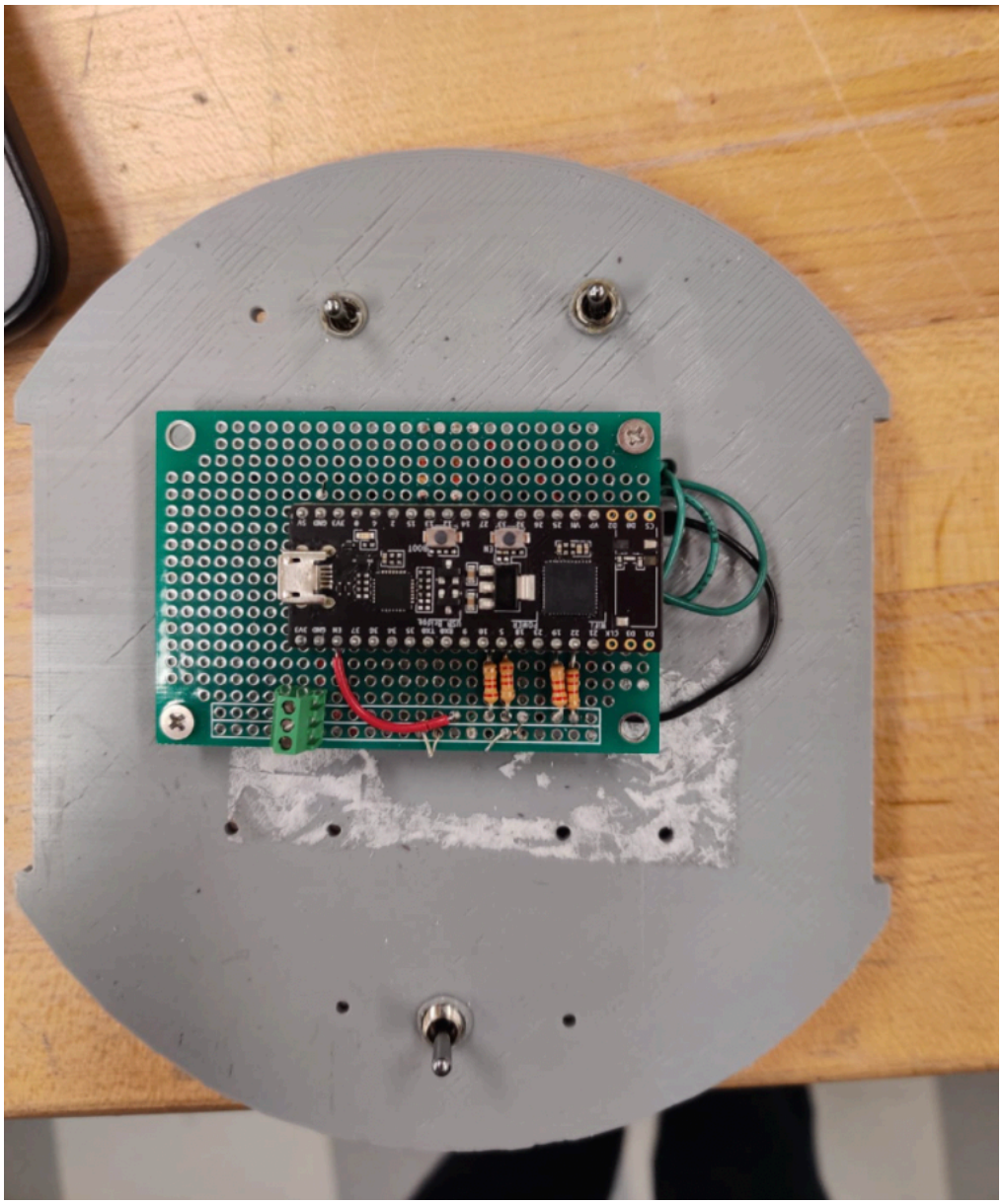
We used physical controller for controlling the bot and execute different functions.



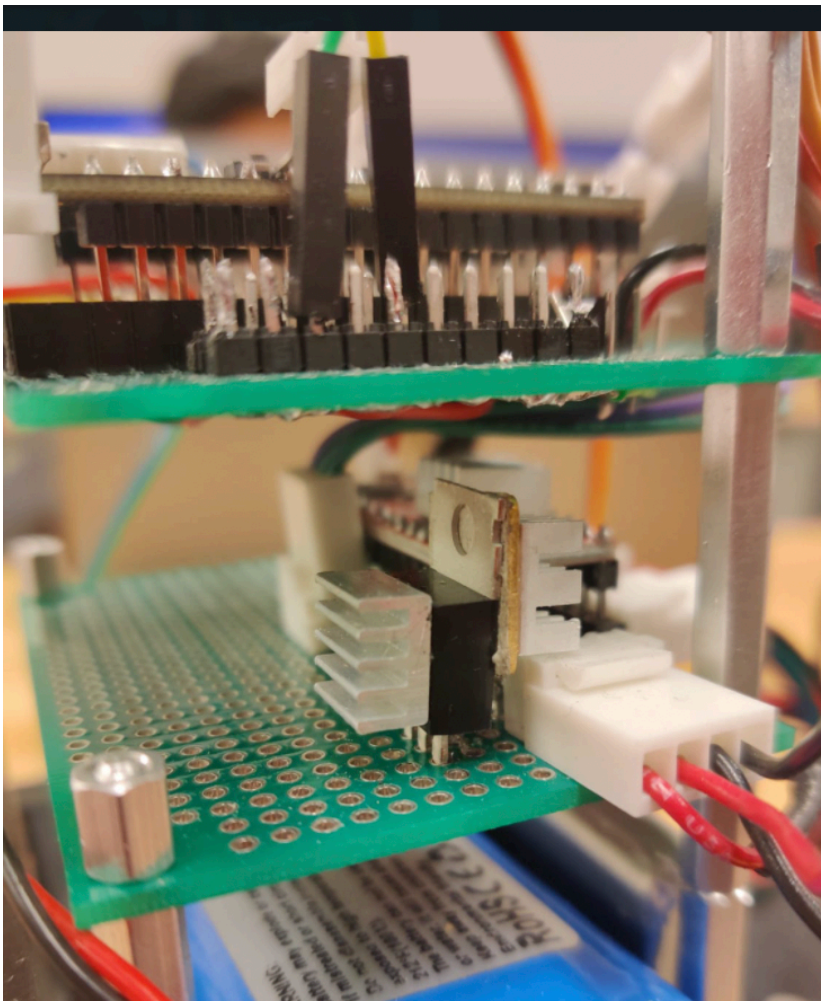
Circuit diagram for wireless controller. The controller used ESP NOW



Software control flow of various boards and sensors



Controller with SPDT mom-off-mom switches for control

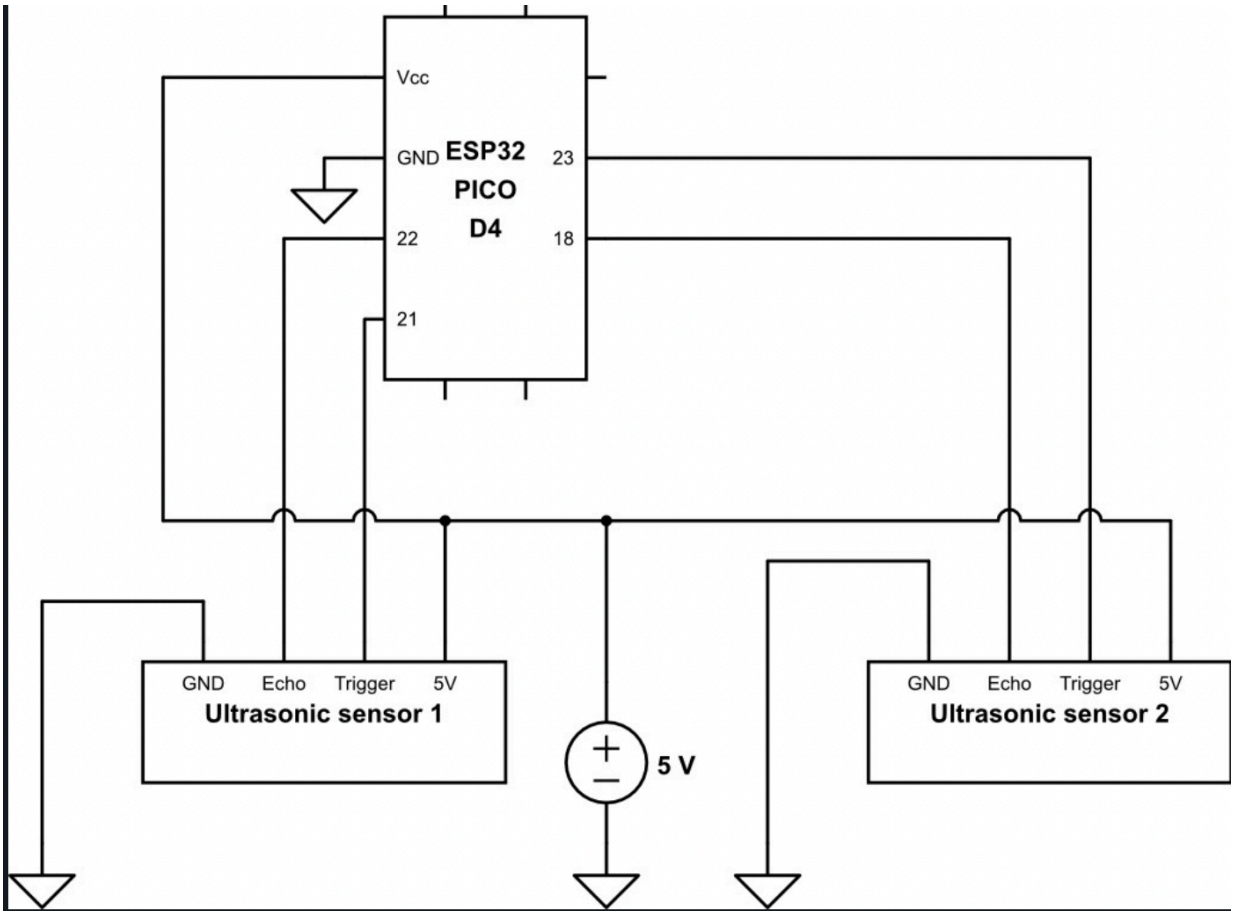


LM7805 voltage regulator takes in 12v and supplies 5v to all boards and sensor

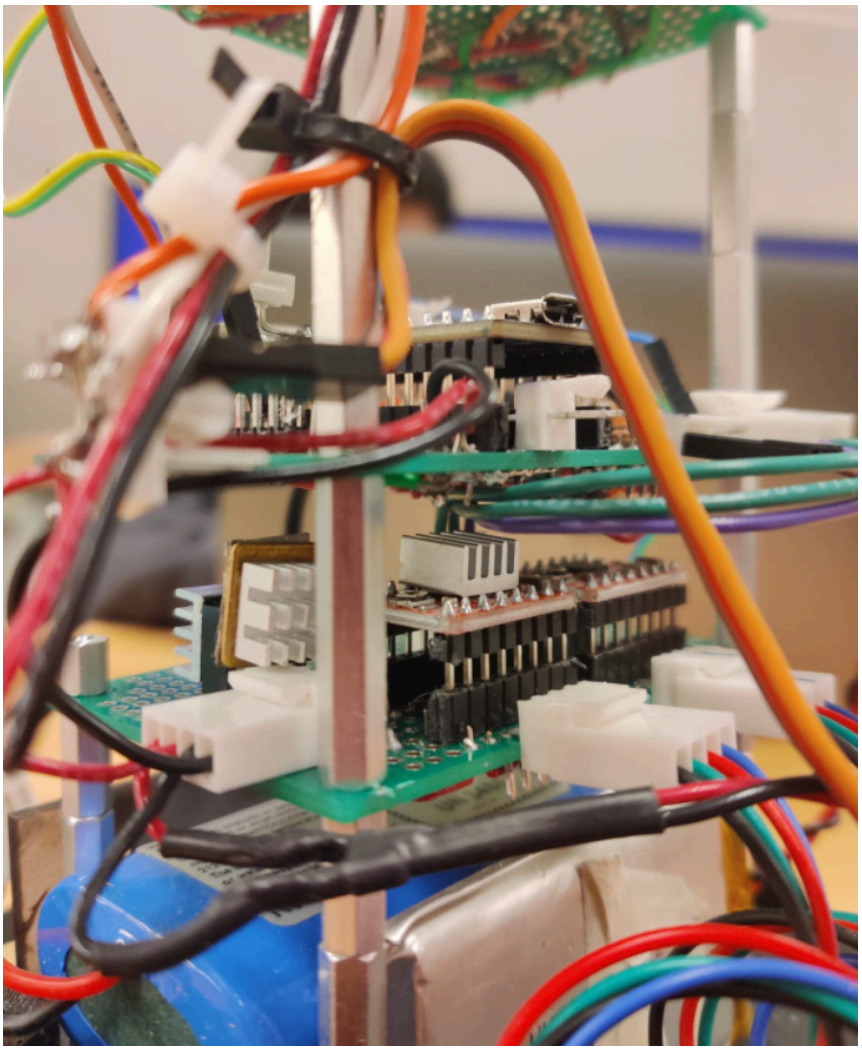
B.1 Wall Following

The circuit for wall following is shown as below. The basic circuit includes 2 ultrasound sensors connected to Arduino with the respective pins. Individual ultrasound sensor has 4 connections – GND, 5V input, Trigger and Echo. So 4 Input/Output pins on the ESP32 Pico D4 have been connected to a total of 2 triggers and 2 echoes.
Ultrasound sensor sends ultrasonic waves and detects them. The delay between the sending and receiving of the waves is noted. Processing this by using speed of sound = 340m/s yields us the distance between the sensor and the obstacle.

Problems – Sensor was giving a very noisy output. The data received would fluctuate a lot. The problem was resolved by taking an average of 5 values consecutively. This would reduce the variance in the data. At each point, the sensor would return the same observation 5 times and the board would process it. It would take decision accordingly.



Sensor circuit diagram for wall following



LM7805 and stepper motor drivers share same perf board as the drivers require both 12v and 5v

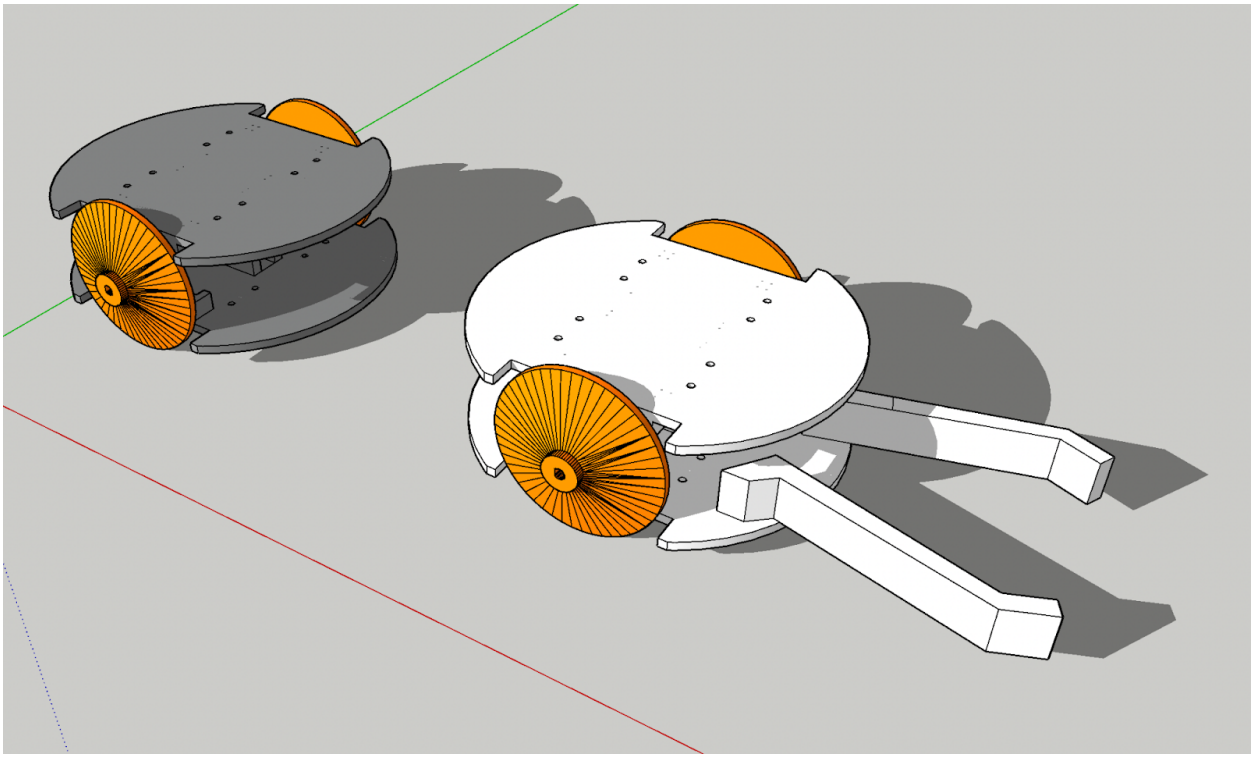
B.2 Beacon Tracking

We have used a single IR sensor to detect frequencies. The circuit used has been mentioned below. The code enables the received input to be processed and it allows the robot to detect 700Hz or 23 Hz frequencies only. On detection of the frequency, Arduino UNO receives the input on pin 11 which is turned high. Through a connection of pull down resistor of 2.2k ohms, a connection goes to the main ESP pin 5. The pin 5 gets turned on only upon receiving the frequency.
The IR sensor has been mechanically connected to the rotating shaft of the servo. The servo rotates 180 degrees and traces back its path. When pin 5 of ESP is turned HIGH, then the servo stops rotating. Angle gets calculated relative to robot orientation and stepper motor enables precise angular control which leads the robot to its required position.

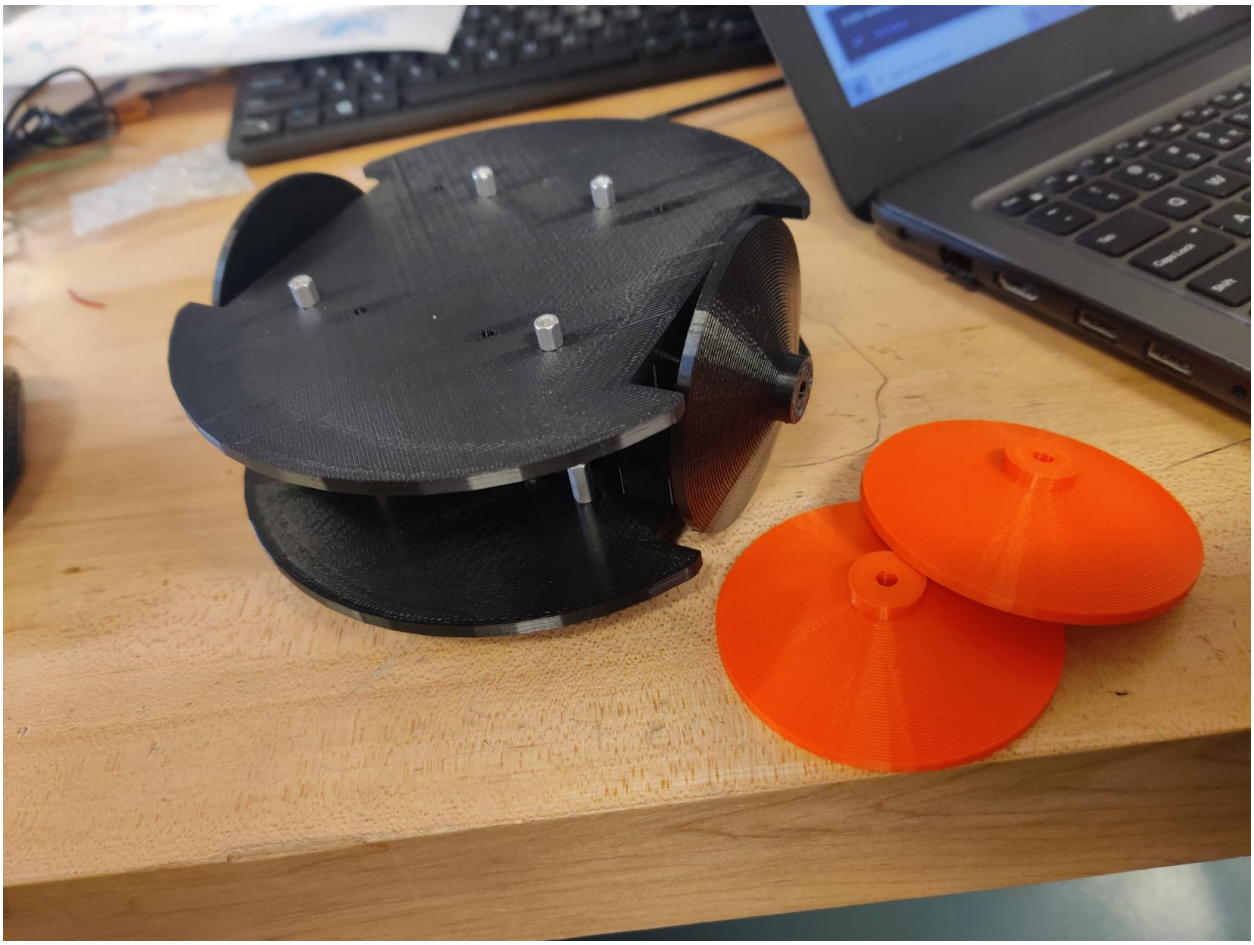
B.3 VIVE Localization

The VIVE circuit used in this robot is the one provided by the professor. The output of VIVE is given to the main vive which constantly rotates the SERVO. The servo is mechanically coupled with the PD-70 sensor. The sensor is rotated in 180 degrees and the distance between can coordinates and the robot location (instantaneous) is calculated at each point. The minimum distance is calculated at each point. The servo stops rotating at the minimum distance point. Another output of same VIVE goes to a second ESP32 Pico D4. This ESP is responsible for the UDP transmission of coordinates and the robot ID

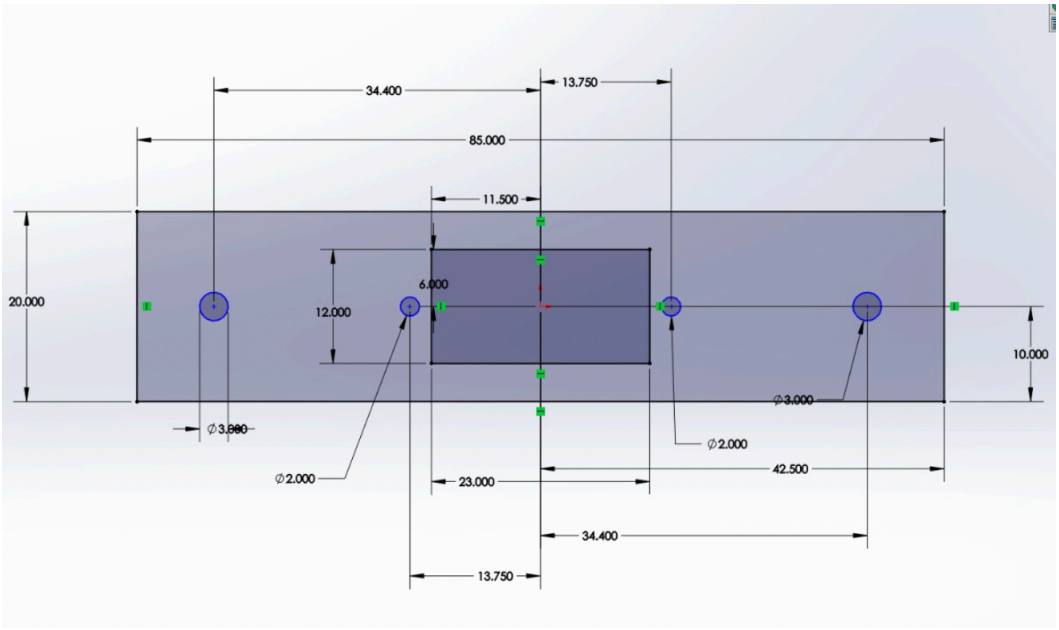
Appendix:



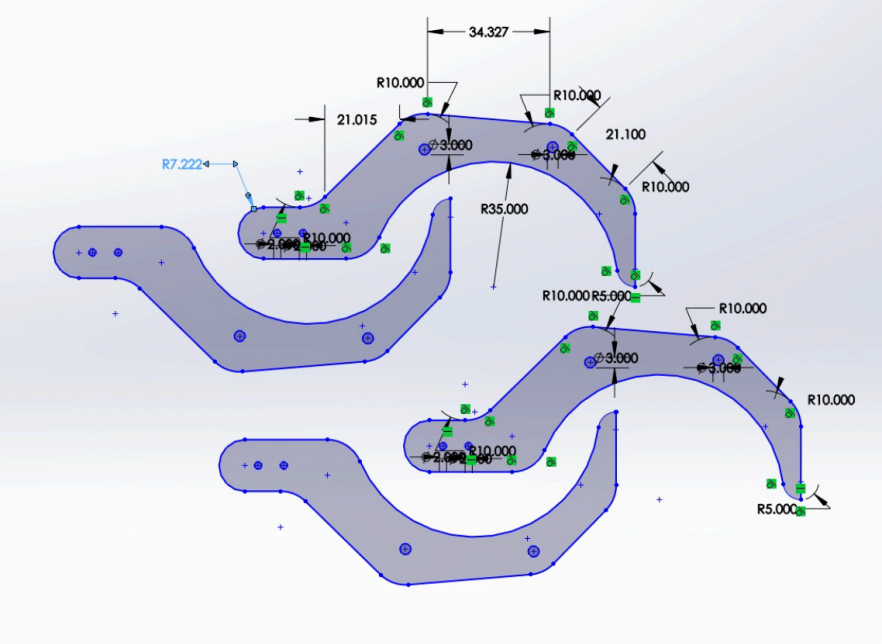
Evolution of our design



3d Printed Chassis and wheels. The thin width give precise movements



Servo mount


34.327
R10.000
R10.000
21.015
21.100
R7.222
R10.000
R35.000
R10.000
R5.000
R10.000
R10.000
R10.000
R5.000

Jaw clamps

D. Bill of Materials

Purchased Items

Product information					
Quant.	**Cat. or model no.**	**Vendor description and product selection notes**	**Link to product webpage**	**Unit cost**	**Total cost**
1	**Boost Converter Module DC - DC 3.0-30V to 5-**	**Boost Converter Module Output Voltage (5pcs)**	**https://www.amazon.com/XL6009-XL6009E1-Boost-Converter-Module/dp/B07XG323G8/ref=sr_1_3?keywords=boost+converter&qid=16384148**	**$ 8.51**	**$ 8.51**
1	3.7V 4400mAh-16.28wh Rechargeable Li-ion	li-ion battery x2	https://www.amazon.com/3-7V-4400mAh-16-28wh-Rechargeable-Li-ion-Battery/dp/B08PNSRM3Y/ref=sr_1_17?keywords=li+ion+7.4v+battery+2500mah&qi	$ 24.99	$ 24.99
1	QWinOut 2-4S 30A RC Brushless ESC Simonk	ESC FOR BLDC	https://www.amazon.com/QWinOut-Brushless-Controller-Multicopter-Quadcopter/dp/B07SFLJJQ5/ref=sr_1_1?keywords=ESC+2+PC&qid=1638419316&s	$ 36.98	$ 36.98
1	ARCELI 5PCS PS2 Joystick Game Controller	joystick	https://www.amazon.com/ARCELI-Joystick-Controller-Dual-axis-Breakout/dp/B077Z8QN3S/ref=sr_1_2?keywords=joystick+arduino&qid=1638417161	$ 8.69	$ 8.69
1	HiLetgo 5pcs A4988 Stepstick Stepper Motor	Stepper motor driver	https://www.amazon.com/HiLetgo-Stepstick-Stepper-Printer-Compatible/dp/B07BND65C8/ref=bmx_dp_204w57fn_4/147-6477652-	$ 9.29	$ 9.29
1	**Readytosky® 2212 920KV Brushless**	**BLDC**	**https://www.amazon.com/Readytosky-Brushless-Motors-Phantom-Quadcopter/dp/B075DD16LK/ref=sr_1_11?keywords=brushless+motor+diy+dro**	$ 39.99	$ 39.99
3	Ultrasonic Distance Sensor - 3V or 5V - HC-	ULTRASONIC DISTANCE SENSOR	https://www.adafruit.com/product/4007	$ 3.95	$ 11.85
					$ -
					$ -
					$ -
				Total cost	**$ 140.30**

Table of electronic components use	Column
Component name	Quantity
ESP32	3
Arduino	1
SERVO motor(SG-90)	1
SERVO motor(94102)	1
Ultrasonic sensors	2
Stepper motor driver	2
Li-on(3.7 V)	2
Boost Converter	2
Stepper motor	2
Voltage regulator	1
Switches(SPDT)	4
ON-OFF Button switch	1

Things that we tried-

1) Laser attached to top servo. This was for making sure that our robot could be visible if the VIVE failed in the opponent area. This drew too much current. So the design was used in match, but scrapped later.

2) A second robot made using H-bridges, 2 DC motors was used (the LAB 4 robot) and it was gripped in the front gripper. This fit perfectly in the model design.

3) 2 VIVES were used, but they provided unreliable readings, so it was scrapped.

Video Links:

1. Vive transmission : https://youtu.be/EpXJ2Dk6IiE
2. Beacon: https://youtube.com/shorts/Iw9XT9fJOMw?feature=share
3. Can: https://www.youtube.com/watch?v=daiU30mP28w&feature=youtu.be
4. Wall following: https://www.youtube.com/watch?v=WXn_kjBpCTk
5. Manual control: https://youtu.be/SPMsvdw2XyU