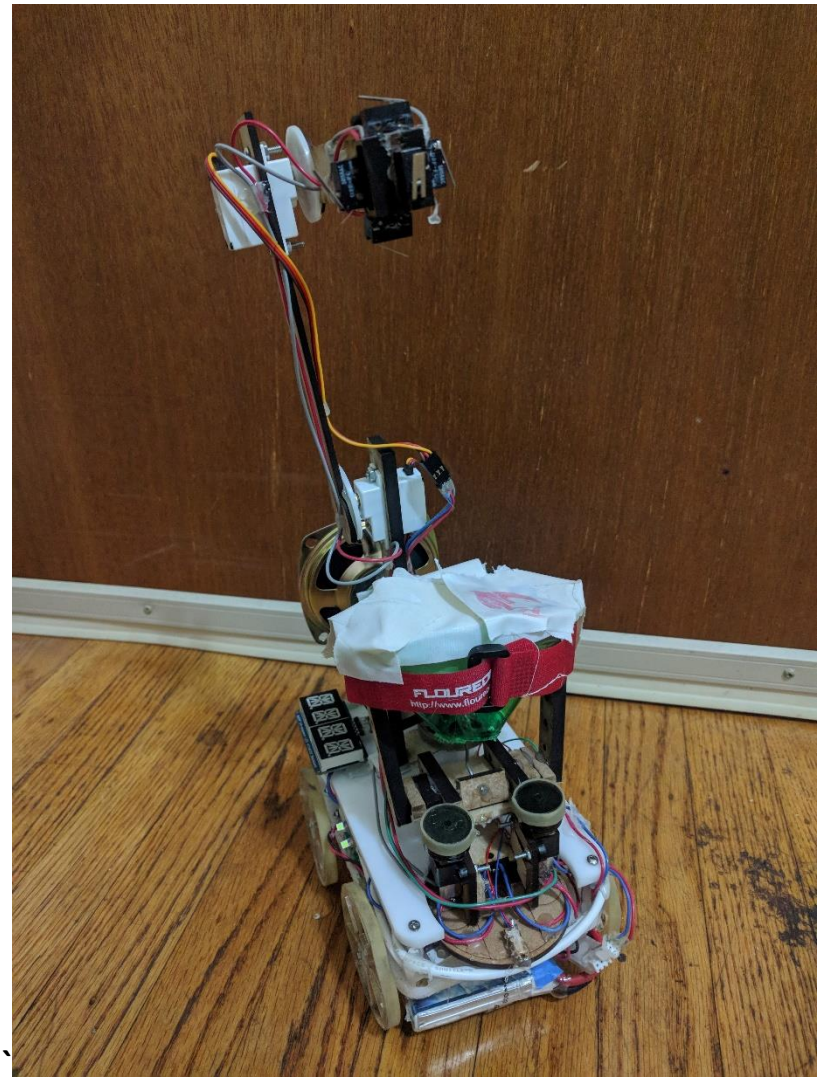
MECHATRONICS

Robot Design



``

-

Akarsh Varre

Game Video:

https://drive.google.com/open?id=1Cr0B_aYES04C4pmSxDz8JpXFFPYxBGxP

Video demonstrating the audio functionality:

https://drive.google.com/open?id=1h_OLEOPi5OCy_DZ8ppUJpx_ryrWS7O-_

Major Abilities

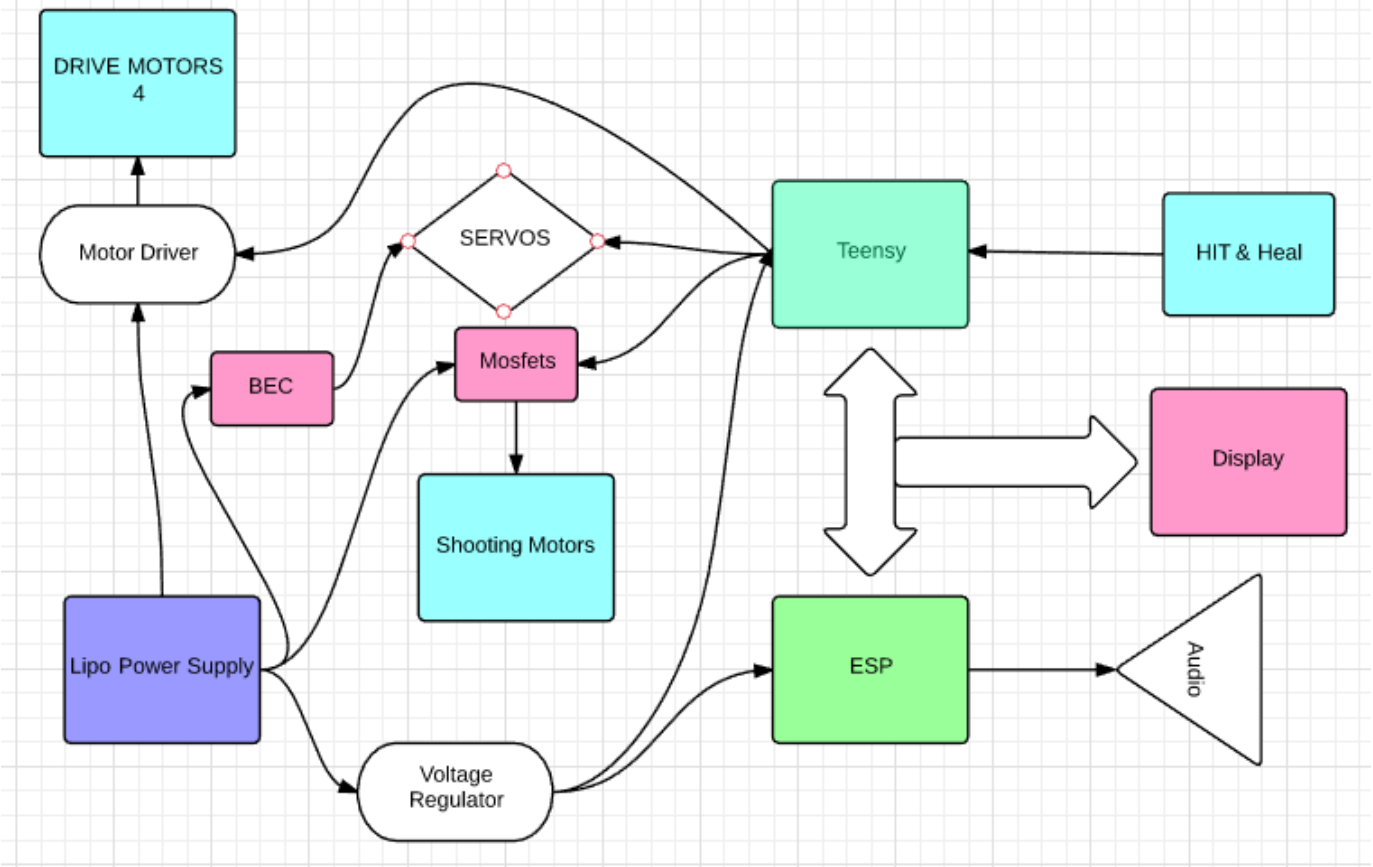
- 4-wheel drive
- Melee weapon
  3 Degree of Freedom Waldo arm
- Ranged weapon
  Ball shooting mechanism using 2 wheels
- Audio Player

Important Components

- Motors for wheels and the shooting mechanism
- Servos for the waldo and shooting direction control
- Audio - SD/MMC Socket, Audio I2S Breakout, speaker
- Power - LIPO battery, BEC for servos, Motor driver
- Control -Teensy, ESP
- Display – Alphanumeric display, leds

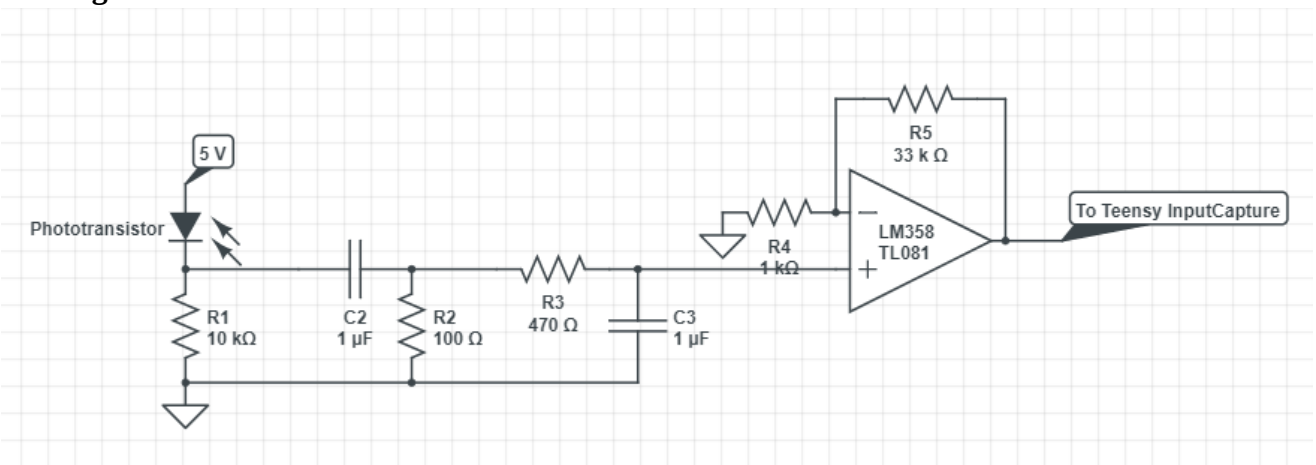
Electrical & Electronic design:

Elementary Block Diagram:



The entire circuit is difficult to illustrate in a single diagram so will be described in parts.

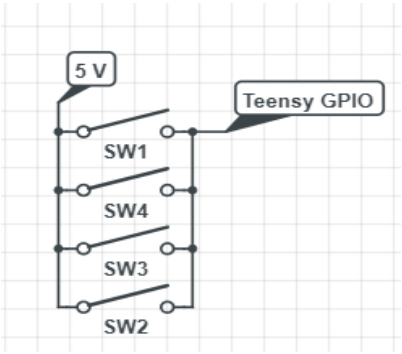
Healing Circuit:



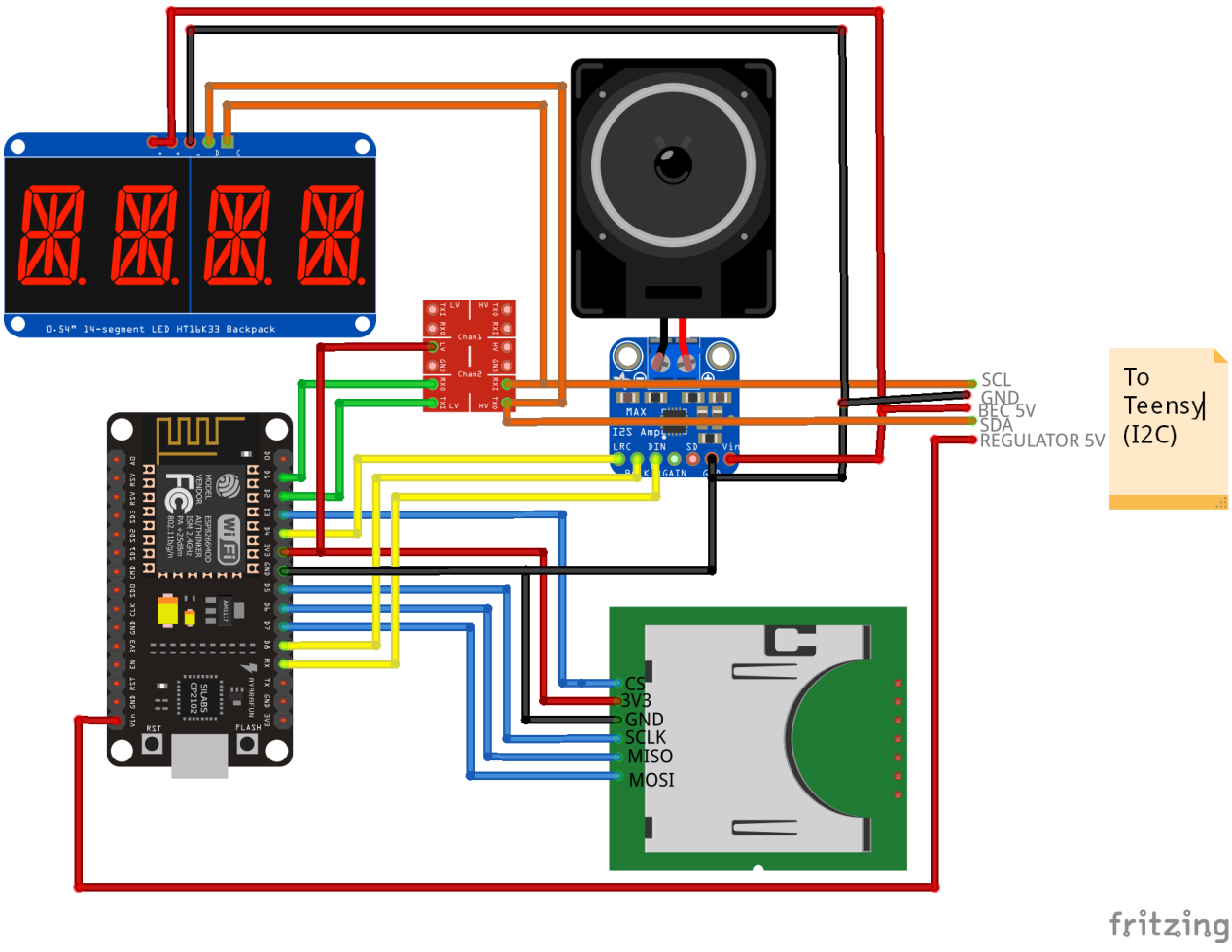
The output from the phototransistor circuit is passed through a banpass filter low pass at 330 hz and high pass centered at 150 hz, the signal is amplified with a high gain and send to Teensy Input capture, Timer 3

Hit:

Hit sensing is done via switches



ESP

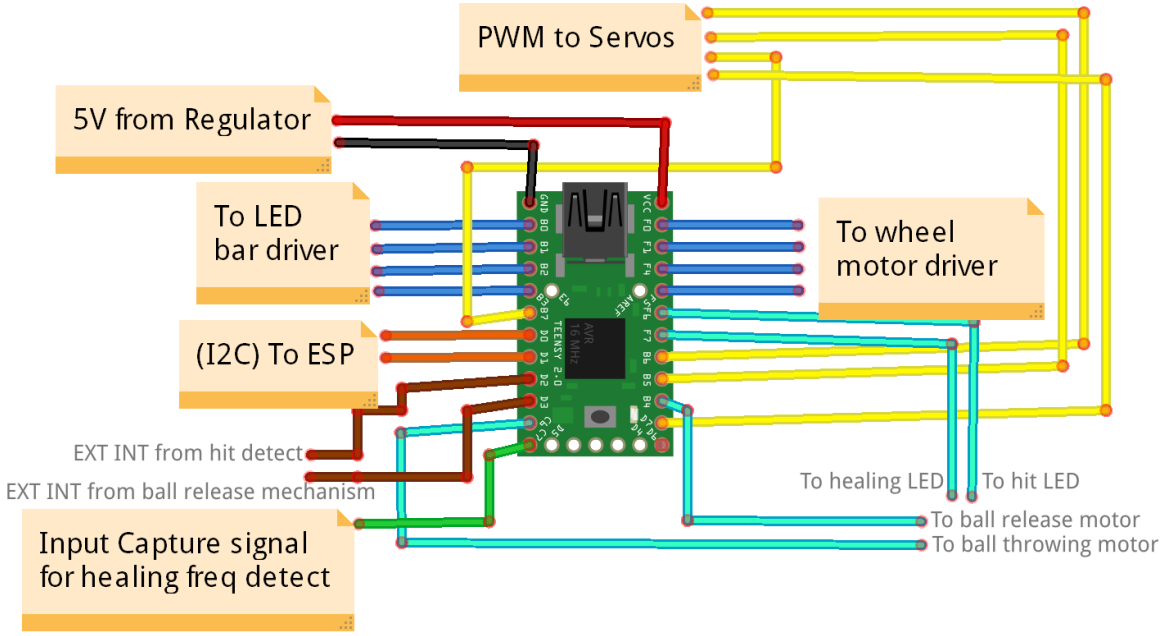


The ESP does the following

- Receives commands via Wi-Fi
- Audio
  - Interfaces with SD card Reader.
  - Drives the I2S audio amplifier

- Display
- Communicates with the Teensy I2C

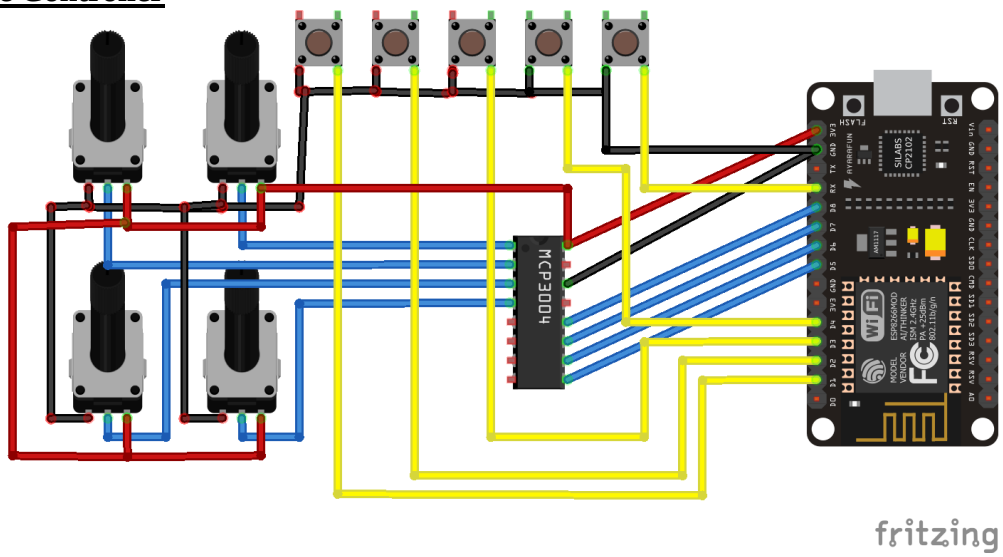
Teensy Circuit

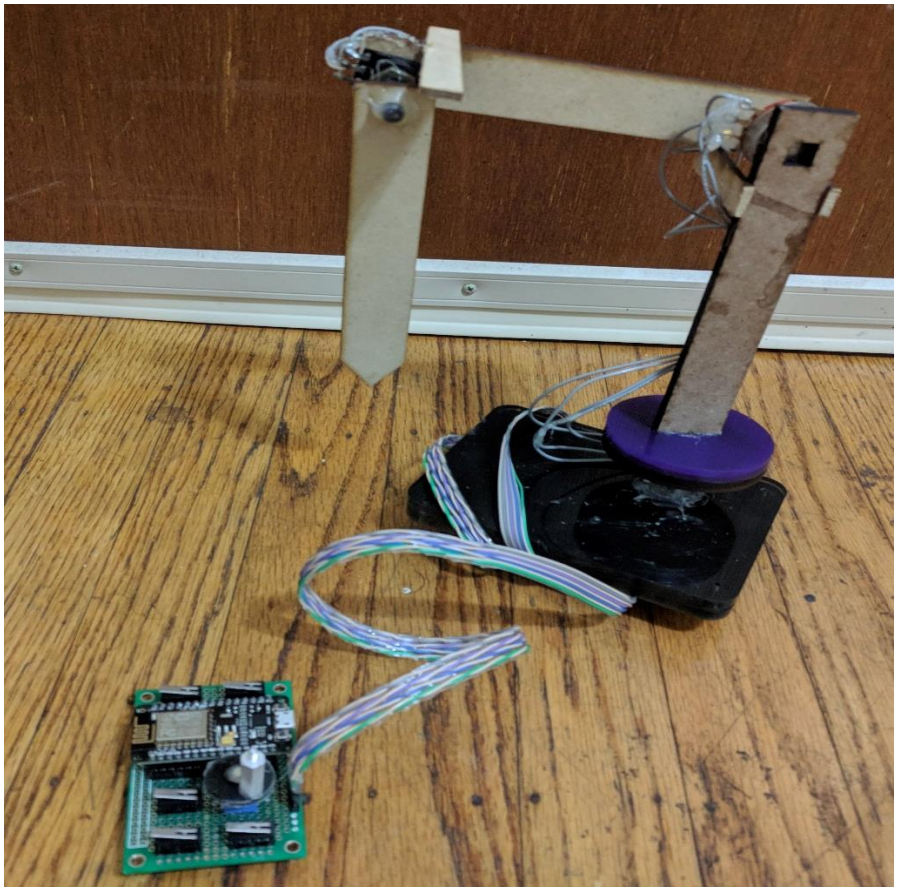


Teensy does the following

- The wheel motors through sn754410 motor driver
- The shooting motors through IRF630 MOSFETS
- Healing detection using Input capture
- Receives commands From ESP - I2C
- PWM for the servos
- Hit detection switch state reads, and signaling LEDs

Remote Controller





5 buttons

- 4 buttons are used to drive the wheels
  2 buttons, left set of wheels, forward and backward
  2 buttons, right set of wheels forward and backward
- 1 button to shoot balls

4 potentiometers
3 potentiometers for controlling the melle arm like a waldo
1 potentiometer to control shooting direction servo.
MCP3004 is a serial ADC chip which measures the pot signals and relays it to the ESP

Circuit Description and Challenges:

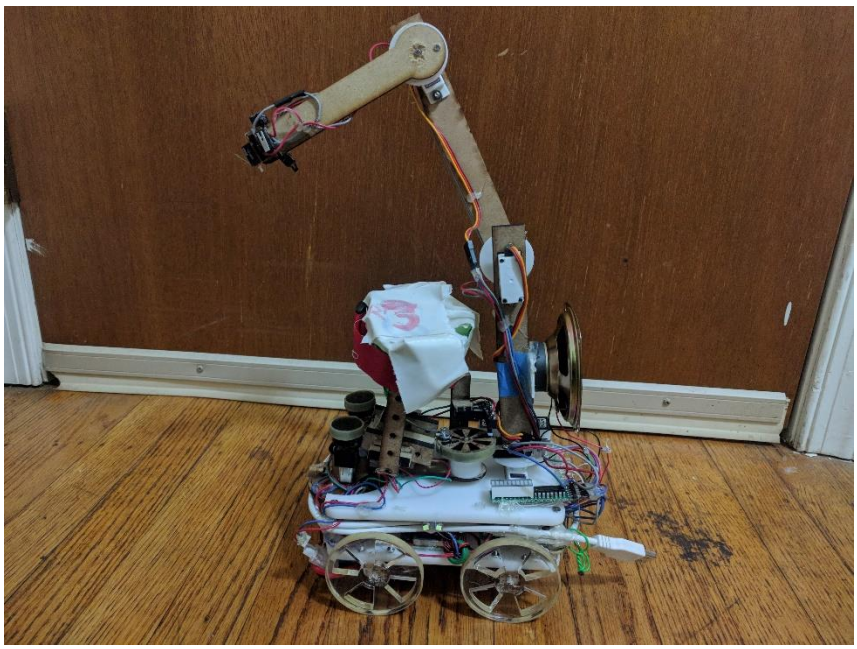
The challenge was to use minimum number of processing units to achieve all the above discussed capabilities which will reduce complexity as well as space requirements. Except D0 and TX in ESP and D6 in Teensy, all the available pins have been utilized efficiently. The D0 pin in the ESP (for both transmitter and receiver) and D6 pin in Teensy have been kept unconnected to access the on-board LED for heartbeat blinking. The TX pin in ESP is not used to keep Serial prints to PC functional.

1. Teensy as an I2C slave:
   To make Teensy work as an I2C slave which can both send and receive data by implementing the full ISR and the status register checks was a real challenge. The Teensy receives all the motors and servo commands coming from ESP and sends back the hit or healing status to the ESP.
2. Servos:
   For driving 4 servos the Teensy has been used. 3 output compare pins of timer-1 and one output compare pin of timer-4 are used to generate PWM.
   It was a challenge to configure 8 bit timer-4 to generate a 50Hz PWM signal with desired pulse widths.
   The Servos are powered from the 5V 3A BEC.
3. Wheel motors:
   The wheel motors are driven by SN754410 H-bridge drive with logic signals from the Teensy.
4. Ball throwing and releasing motors:
   These 2 motors are driven by 2 IRF630 MOSFETS which are driven by the Teensy.
5. LEDs.:
   A set of white LEDs for signaling a healing detection and 4 super bright green LEDs for signaling a hit detection are driven by 2 IRF630 MOSFETS which are driven by the Teensy.
6. LED bar:
   Challenge: We had only 4 pins left on the Teensy to control the 10 digit LED bar. Driving the LED bar using comparators would eat up significant amount of space and would increase complexity.

Solution: We used a 10-bit decoder 74LS42 which can light up single LEDs at a time using 4 control lines. We are cycling through all the LEDs very fast such that a persistence of vision is achieved so that it looks as if all the 10 LEDs are glowing.

7. Healing frequency detection:
   An analog bandpass filter is implemented whose output is amplified by a very high gain using LM358 op amp. The output signal of the op amp is a square wave which can be used as a digital signal. The Input capture for timer-3 in Teensy is then used to measure the frequency which is very precise.
8. 4-digit alphanumeric display:
   The adafruit 4 digit alphanumeric display is connected to the I2C bus of the ESP. In the same bus the Teensy is connected as an I2C slave.
9. Audio:
   Audio is being played by the adafruit I2S audio codec module which is connected to the ESP. The audio files in .wav format are saved in an SD card. The SD card is connected to the ESP on the SPI bus.
   Challenge: The hardware chip select of the SPI bus is overlapping with the I2S pins.
   Solution: The SD APIs of Arduino allows any free GPIO pin to be used as a chip select. So, we moved the chip select pin to D3.
   Challenge: The UART RX pin overlaps with the I2S pins.
   Solution: Calling i2s.begin() after Serial.begin() will only modify the pinmux settings of the RX pin to use it for I2S. So, the result will be a TX-only UART where we can print data to serial monitor on PC, but not send data from the PC.

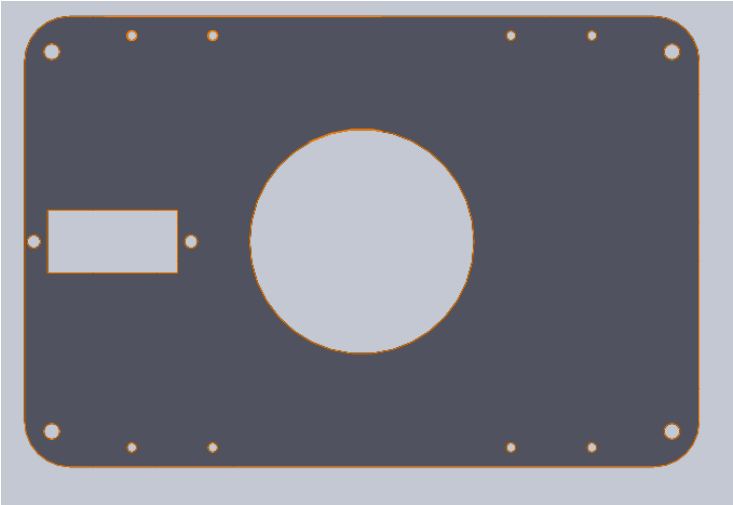
Mechanical Design



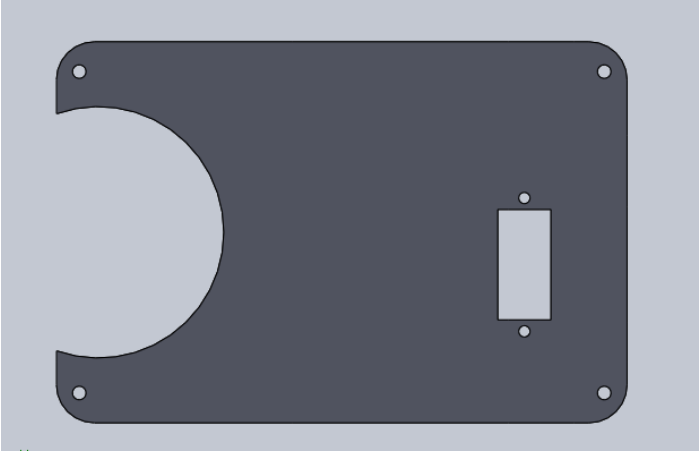
Base

- The base is rectangular with its dimensions being **13cm (L) x 10cm (W).** It is fabricated twice to have a standoff. These two bases are connected with the help of **4-40 standoffs**. The space in between the bases is used for placing the circuits, which is described above.
- The lower base is designed for the placement of motors and the reflective doom for the photo transistor circuit.
- The upper base is designed especially for the WALDO placement and the shooting mechanism. A hole with a diameter of **6cm** is cut in the upper base so as to account for the rotating mechanism for the shooting
- Below pictures of the lower and upper bases indicate the above described features
- Healing light sensing dome (**3cm dia hemi hollow sphere**) is fixed to the lower base at the center of it.

Lower Base



Upper Base

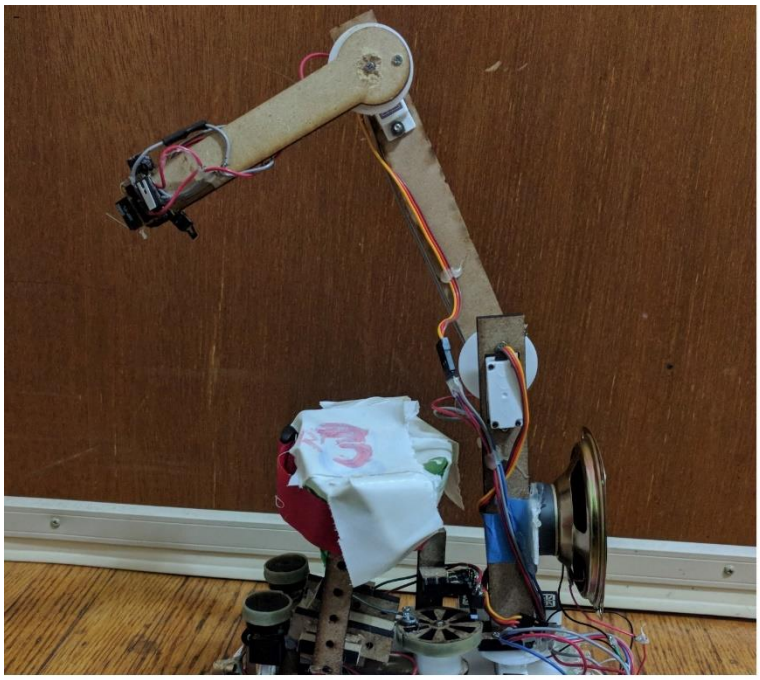


Wheels:

- Wheels are fabricated out **of 1/4th inch Acrylic**.
- The robot is designed for a four-wheel drive and so, **four Pololu Micro Metal Gear Motors of 150:1 ratio** are chosen to account for it.
- Two switches are integrated for each of the two sets of motors. Actuating one set of switches triggers two motors (both front and back on one of the side) to rotate in clockwise/ anticlockwise respectively, making the bot move forward and backward
- The radius of the wheels are **3cm**.
- To provide enough traction to the wheels rubber bands are wrapped around

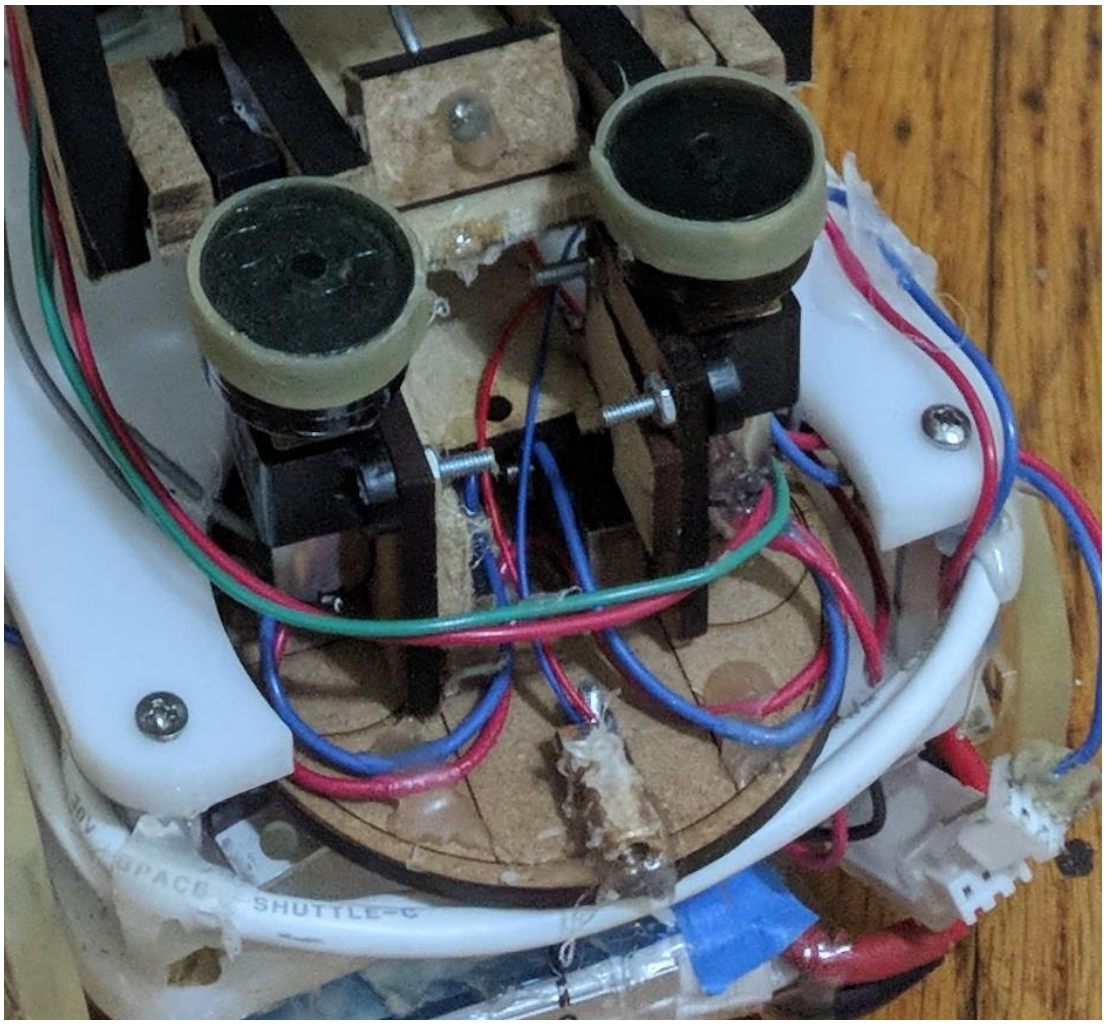
WALDO aka MEELE

- For the Meele weapon, we chose a 3DOF WALDO mechanism using servos.
- 5 switches are attached to the final link of the WALDO to sense the hit
- All the three links of the WALDO on the bot are actuated using Servos
- The links are fabricated out of 1/8th inch MDF
- The Link1 is of 13cm in length
- The Link2 and Link3 are of 10cm in length.
- Sensing side of the WALDO is mimicked as the actuating side.
- Potentiometers are used for sensing part of the WALDO
- The switch action i.e. sensing the hit is integrated with the speaker(attaches to the LINK 1 of the WALDO) so as to produce sound when hit

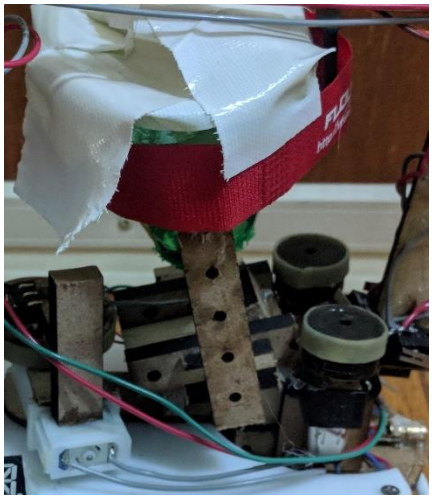


SHOOTING MECHANISM

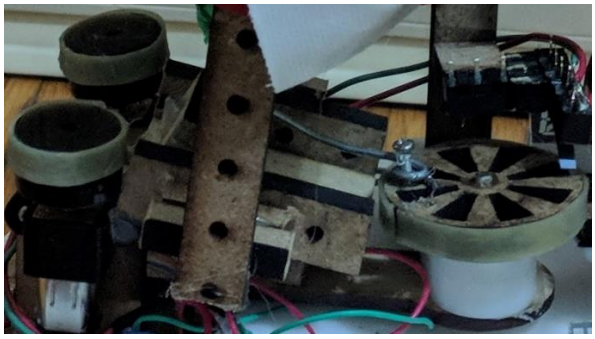
- We had implemented shooting part which can also aim (rotate) and snipe the target
- For the shooting mechanism we had used **two Pololu Micro Metal Gear Motors** which are placed on **5cm MDF Strips** using its brackets.
- These two strips along with the motors are erected on a circular base at right measurements.
- The two motors are coupled with **2cm dia 1/4th inch Acrylic wheels.**
- Rubber bands are wrapped around them to provide enough traction to shoot the ball
- The wheels are adjusted in such a way so that the Nerf ball fits perfectly in between the two wheels



Balls are fed to the wheels with the help of a cone.



- An inclined platform is made so as to move the balls to the wheels for shooting
- This movement is performed a Prismatic Link.
- This link is actuated using a wheel for which the link is attached to the screw.
- The screw is on the wheel and has the link extension wrapped around it and is made in such a way that it freely rotated around it
- This wheel is then attached to the shaft of the motor. So, when the motor rotated the prismatic joint actuates
- **Rotational motion is converted to a translational motion**



- The circular base is then coupled to a Servo so as to account for the rotation shooting
- The whole set up is interlinked with each other and to the servo of the circular base so as to account for the rotating mechanism
- The wheel has a dia of **3cm**. the link is of **3cm** in length
- The inclined platform is about **2.5cm** in length(inclined)
- The cone that is loading the balls is made out of a COKE bottle. The coke bottle is cut form its mouth to a **height of 3cm** so as to account for **10 NERF balls**
- The three motors are coupled to one actuation input from the motor driver

Challenges faced

- The shooting part was a bit easy, but making it rotate is challenging.
- We faced difficulties in placing the MDF motor strips at an accurate position as, slight misplacement of those will result in misaligned shooting
- The axis of the motor shafts has to match with the center line of the circular base and also both these are to be placed at a distance of **1.8cm.**

Code:

The ESP is responsible for receiving packets over wifi, relaying them to the Teensy over I2C, display live data to the alphanumeric display and also play audio from SD card.

The Teensy is responsible for driving all the actuators and LEDs and acquiring data from switches and phototransistor. Other than the logic for checking the detected frequency from Input Capture and driving the LED bar, the whole code on Teensy is event driven. It responds to I2C events and external interrupt events and acts accordingly.

Link to code:

https://drive.google.com/open?id=1LSkFZ_EybHMHo4GXEUvi2w9u17LedzzP