Intelligent robotics I

VikingBot

Abdullah Barghouti
Travis Hermant
Jagir Laxmichand
Abdirahman Alasow

Overview

Components:

- 2 DC motors
- Raspberry pi 3B
- Raspberry pi camera
- Servo motors
- IR sensor

Robot Features:

- Movement (forward, backwards, right, left)
- Bobble head (up and down)
- Image and gesture detection
- Obstacle avoidance

3D Design

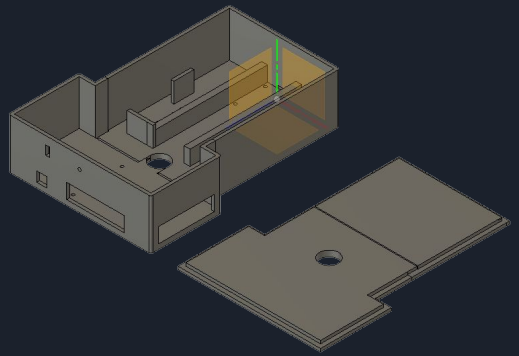
Achievements:

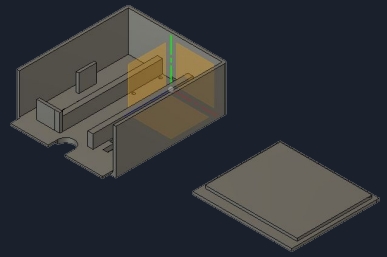
- Design that houses all the components
- Light and compact design
- Secure all components in place during operation and transportation

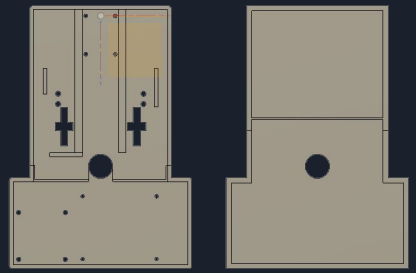
Issues:

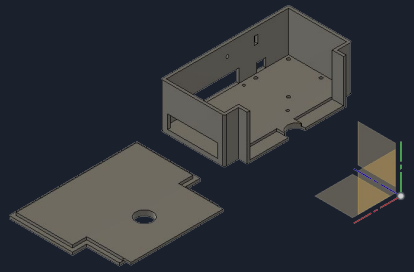
- Split into 4 pieces for printing
- Will require some drilling for sensors on front and back that would fail during printing

3D Design pictures and schematics









Hardware

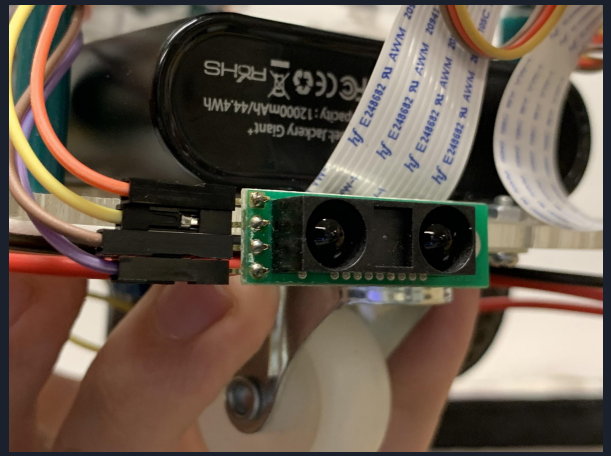
Achievements:

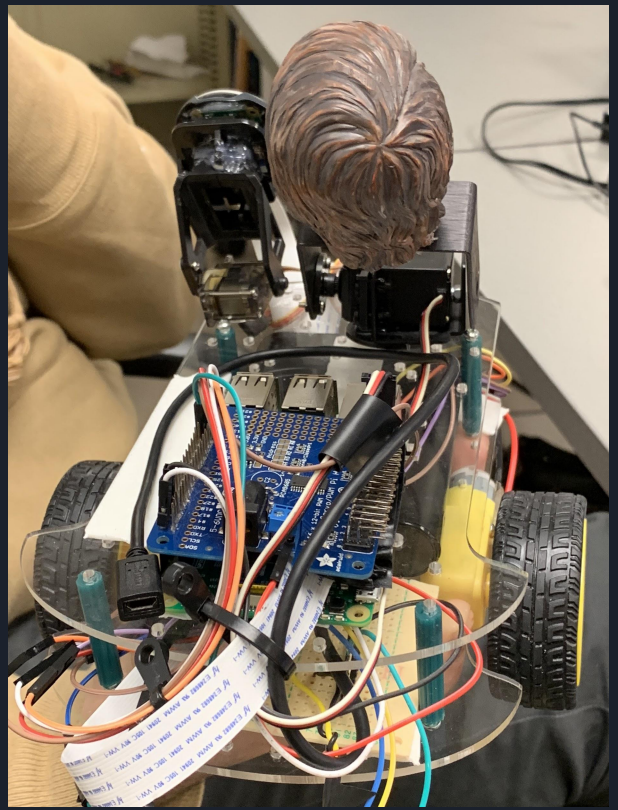
- Car is very fast
- Servos are strong and with full rotation
- Pi Camera is integrated with OpenCV
- H-Bridge controls motor movement

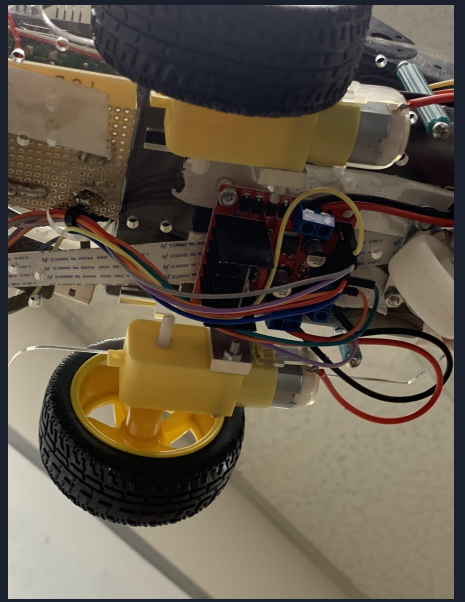
Issues:

- Car is sometimes too fast
- Cannot power two servos
- Raspberry Pi may require own PSU

Hardware Pictures and connections







Software

Tools used:

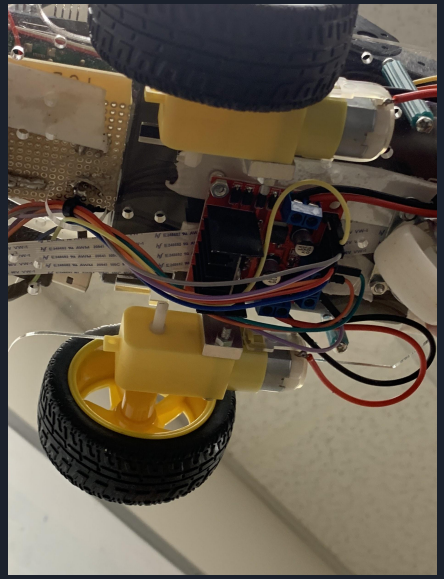
- Python
- OpenCV
- NumPy

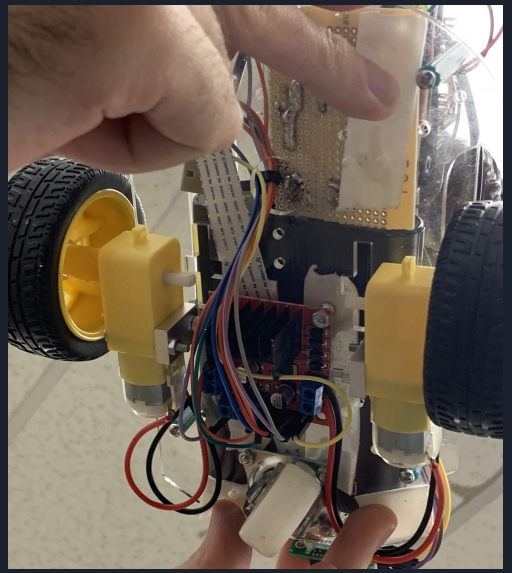
Issues:

- Finding the right libraries
- Difficulties connecting wirelessly

Motor Movement

- Create a driver that is referenced by other modules to move the robot
- L293E H-bridge



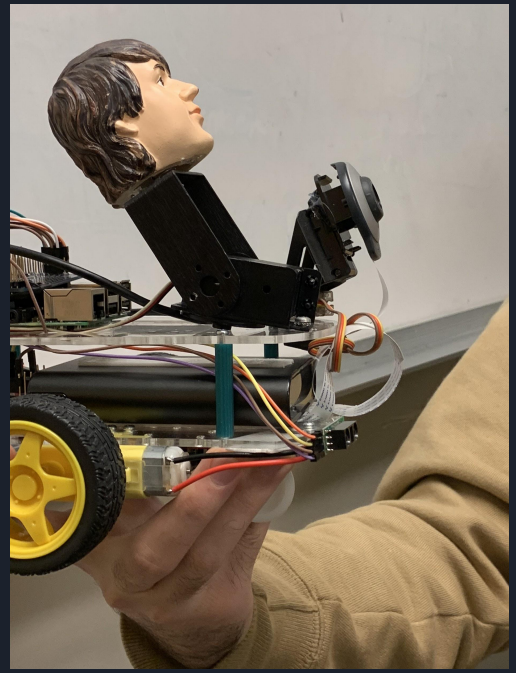


Servo Movement

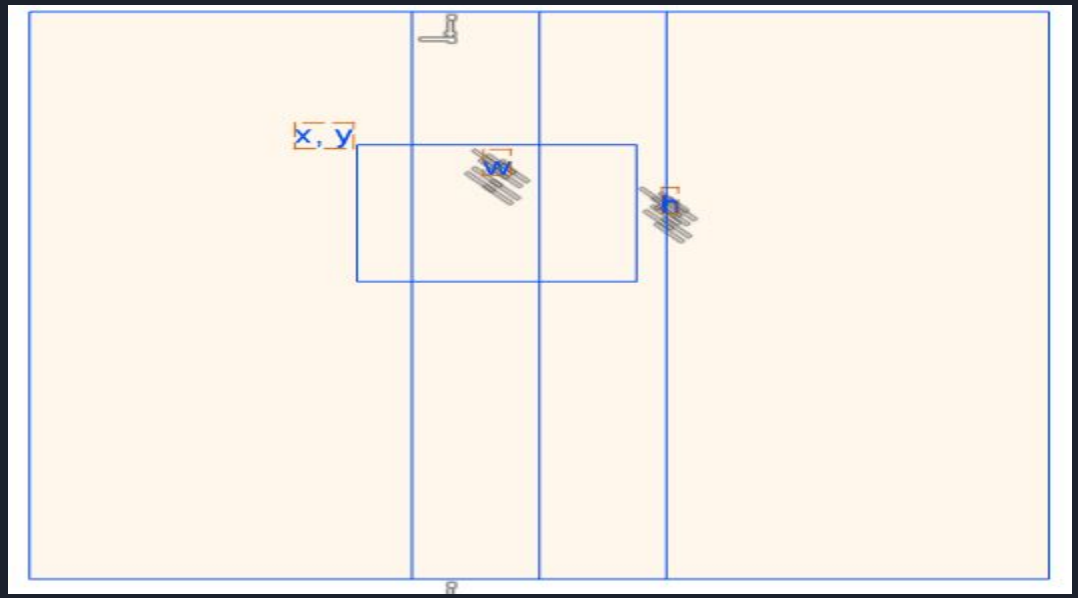
Due to singular servo, it is limited to one axis of movement

Since the primary function of the robot is as a car, the servo acts as a mechanism for gestures and is connected to the head

There is a secondary servo attached to the camera, it is unplugged due to power constraints, but if that is solved it can



Facial tracking



if (220 <= (x + (w/2)) <= 420): RobotSTOP()

elif ((x + (w/2)) < 220): RobotLEFT()

elif((x + (w/2)) > 420): RobotRIGHT()

Ball Detection and Tracking

- Segmentation based on colour
- Transforming image into HSV (i.e. Hue Saturation View)
- Masking an Image based on threshold value of color
- Hough transform and contour detection to get the region of interest
- Proportional algorithm to track a ball

Smile Detection

- Use of common haar like features
- Preprocessing for Haar cascade classifier
- Haar cascade as a kernel to detect a face
- Use Haar cascade smile kernel to identify the smile
- Movement of servo based on smile detection

Overall possible improvements

- Case needs to be 3D printed and assembled
- Possible 3D printed head
- Speaker for sound
- More robust implementation of each component
- May need small battery for pi itself, so if servos/motors draw too much power it won't shut down
- Might need to implement duty cycles for motor driver so it can operate at different speeds
- Integration of various OpenCV algorithms in single application

Conclusion

- All in all the project was a great learning experience. We got to explore 3D printing, OpenCV, and controlling the motors/ servos.
- Using openCV we implemented facial recognition and object tracking functionality
- The IR sensor allows to avoid obstacles while tracking an object/ face

Videos

OpenCV

Servo

VikingBot object detection