Robotics Traveling Van

Andy Babcock, David Jimenez, Kaden Zaremba, Kyle Draper

Sponsor: Dr. Michael Shafer

Overview

- Goal 1: Build two robots for classroom demonstrations
- Goal 2: After the prototypes are finished, create a handful of the two robots
- Sub goals:
    - Make the robots safe
    - Provide educational value
    - Robots are easy to operate for both teachers and students

Introduction to Robots

- Robot 1: Self-Balancing Robot

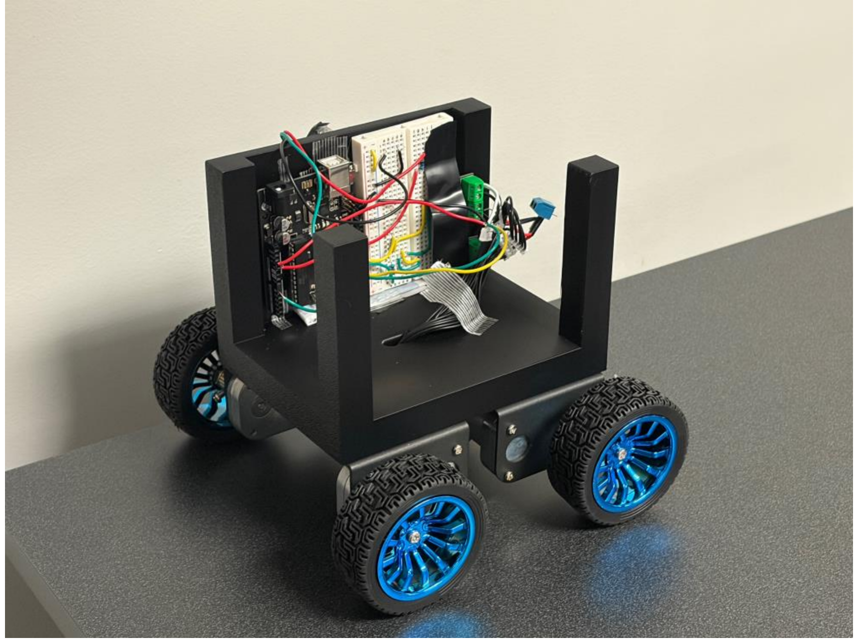


Figure 1: CAD Model of the self-balancing robot

- Robot 2: Ball & Beam Balance

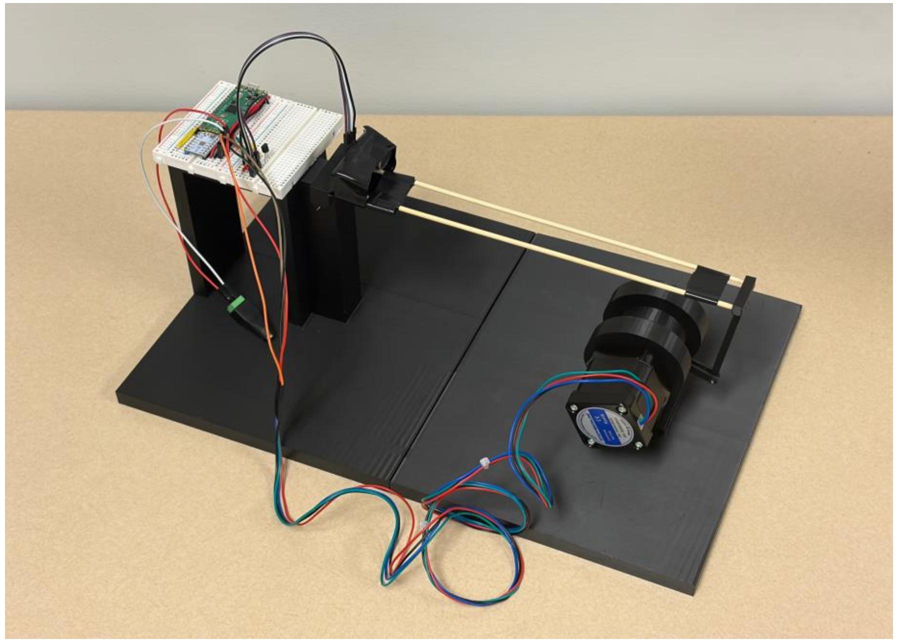


Figure 2: CAD model of the ball and beam balancing robot

Inverted Pendulum Robot (I.P.R.)

- The robot balances an inverted pendulum
  - Balances by rolling back and forth

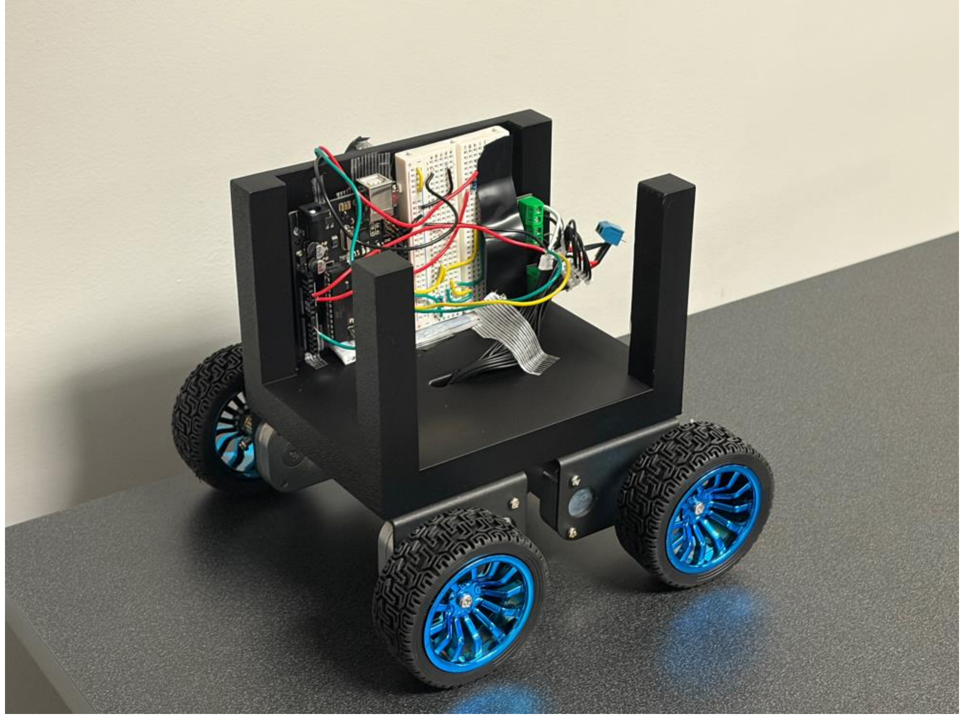


Figure 1: Self balancing robot

I.P.R. – Parts List

Figure 3: WH148 Potentiometer

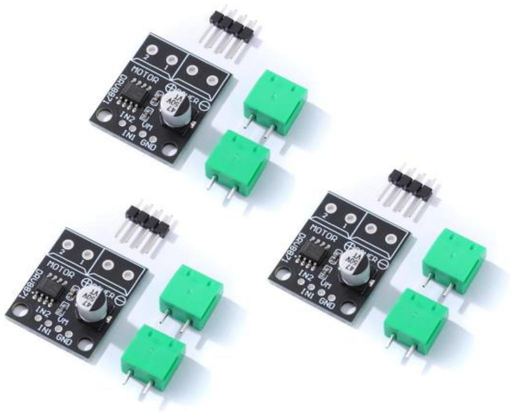


Figure 4: DRV8871 Motor Driver

Figure 5: GM3865-520 DC Motor with Encoder

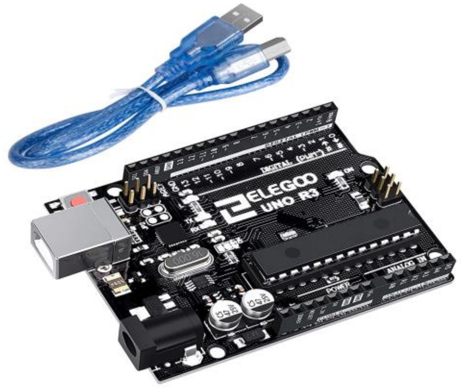


Figure 6: Elgoo Uno R3

I.P.R. – Prototype vs Final Design

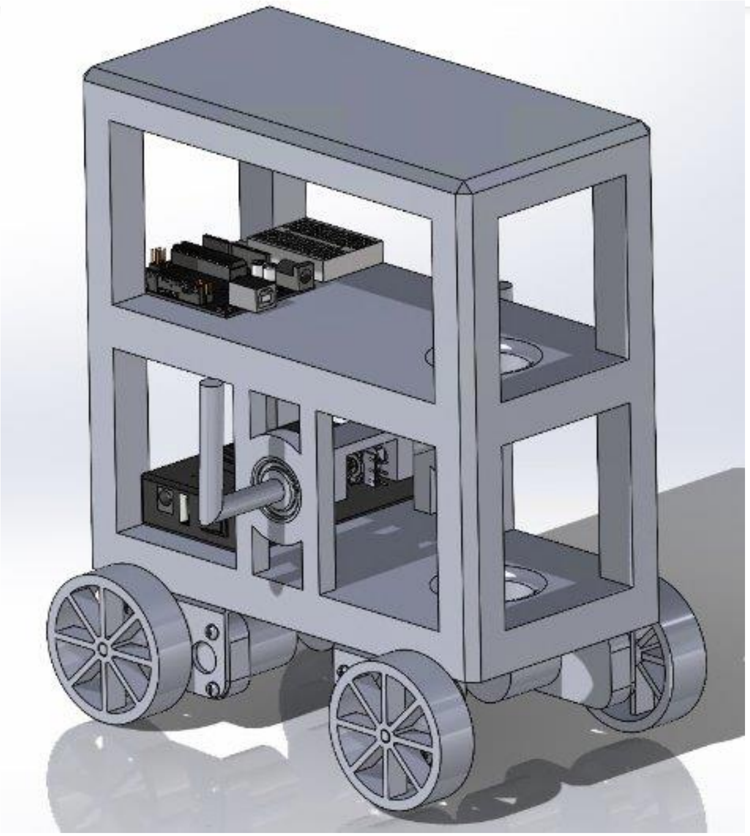


Figure 7: (Previous) CAD Model for the self-balancing robot

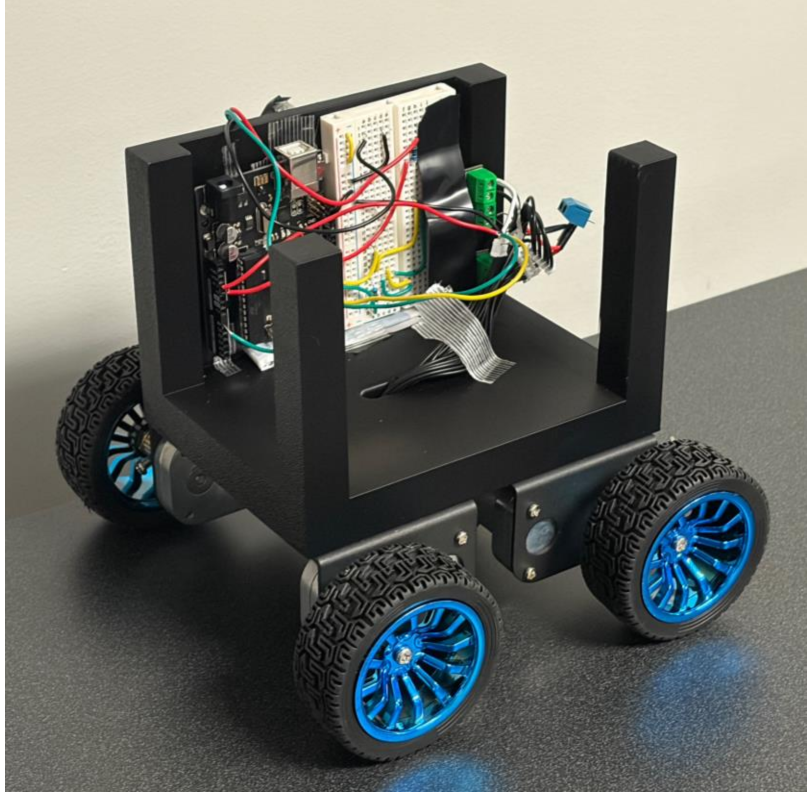


Figure 8: Electrical prototype of the CAD models circuits

I.P.R. - Circuit

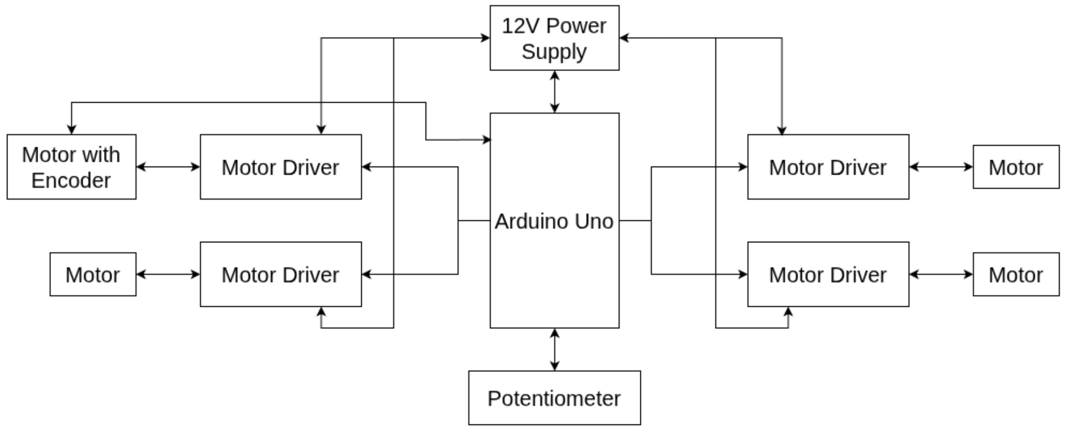


Figure 9: Simple Circuit Diagram for the Inverted Pendulum Robot

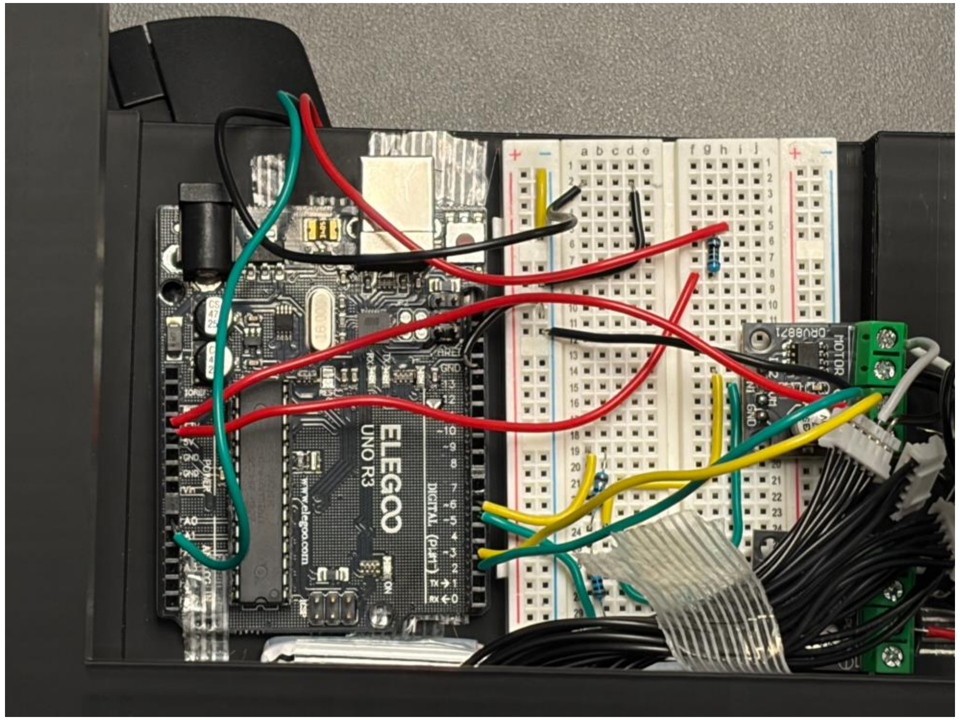


Figure 10: Code Flowchart for the Inverted Pendulum Robot

I.P.R. - Code

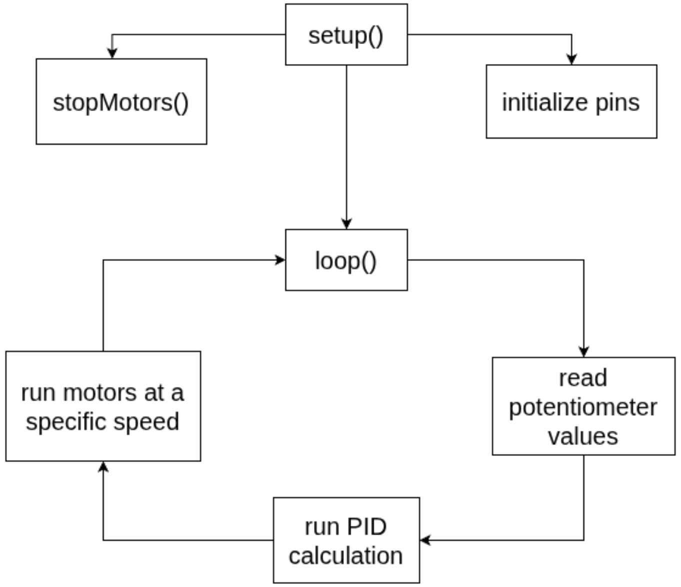
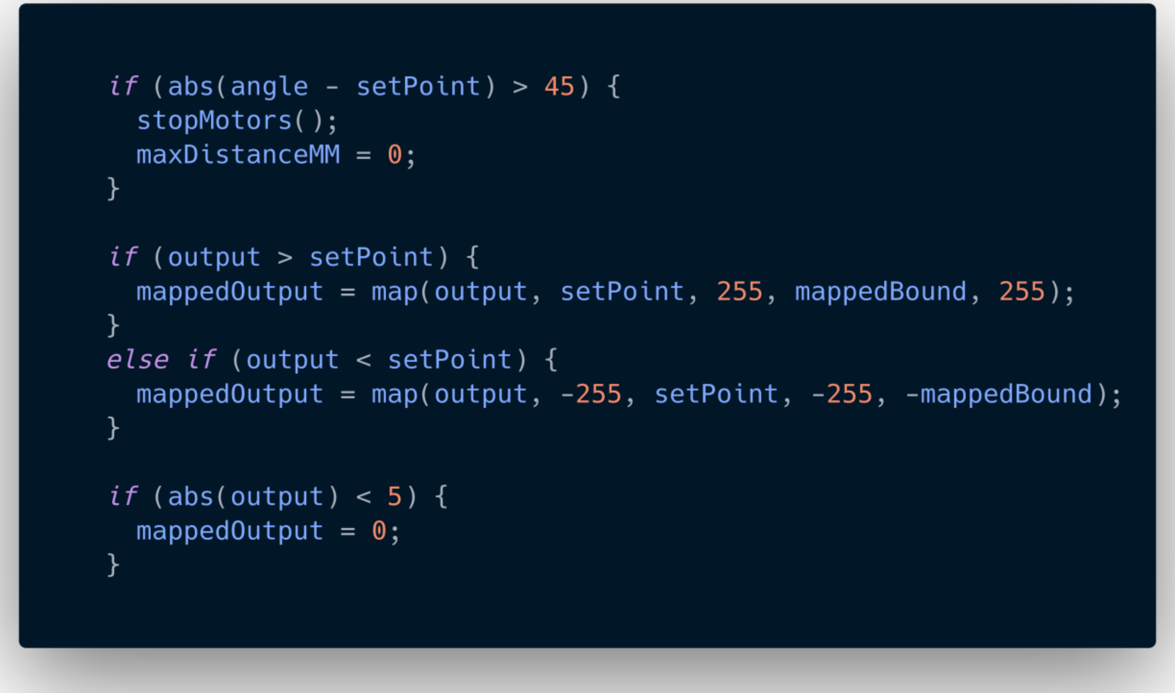


Figure 11: Code flowchart



```
if (abs(angle - setPoint) > 45) {
  stopMotors();
  maxDistanceMM = 0;
}

if (output > setPoint) {
  mappedOutput = map(output, setPoint, 255, mappedBound, 255);
}
else if (output < setPoint) {
  mappedOutput = map(output, -255, setPoint, -255, -mappedBound);
}

if (abs(output) < 5) {
  mappedOutput = 0;
}
```

Figure 12: Code snippet

I.P.R. - Future

- Potentiometer not accurate enough
  - Not a big deal, switch to Digital Encoders
- Arduino doesn't have enough compute to power a touchscreen
  - Switch to a Raspberry Pi Pico or similar
- Implement the touchscreen

Ball and Beam Balance Robot (B.B.B.)

- Actively balances a ball on a set of two beams
    - Tilts the beams based on balls position
    - Returns the ball to the center

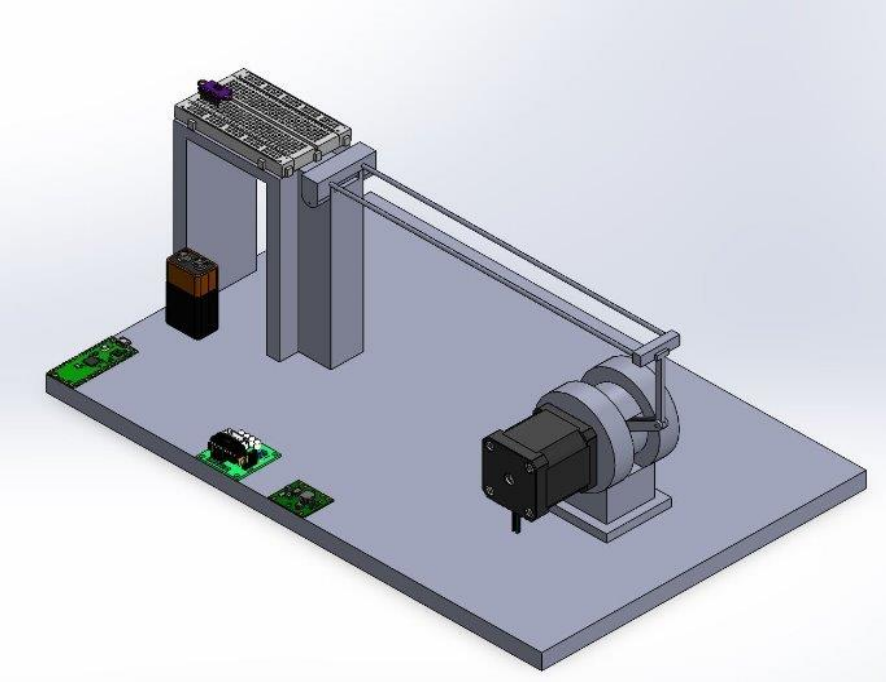


Figure 11: CAD Model of Ball and Beam system Drawn up by the Mechanical Engineering Team

B.B.B. - Parts List

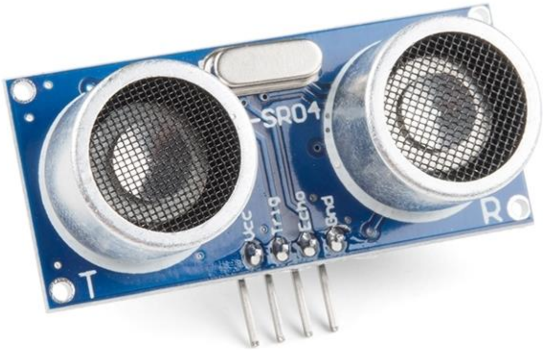


Figure 12: Ultrasonic Sensor

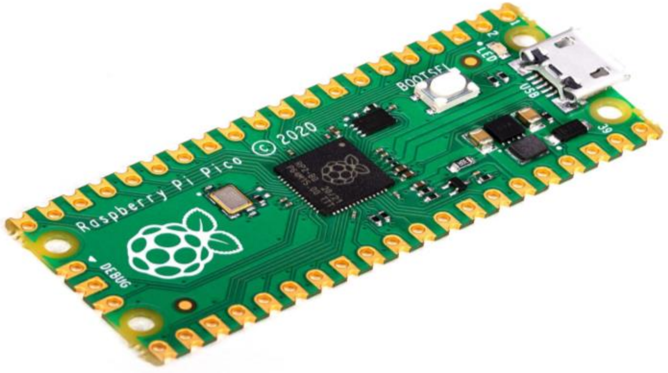


Figure 13: Raspberry Pi Pico

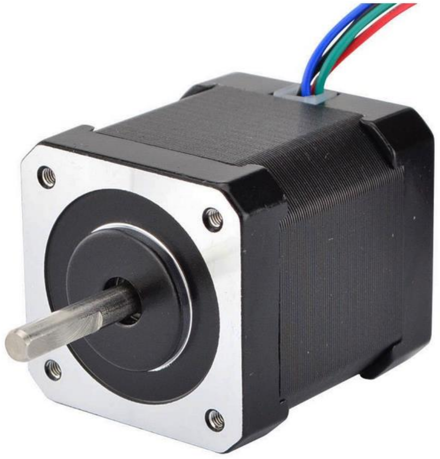


Figure 14: 12V stepper motor

B.B.B. – Working Prototype

- Ultrasonic sensor secured to one end of the beams
    - Reads position of ball
- Motor rotates arm
    - Tilts beam, moving ball

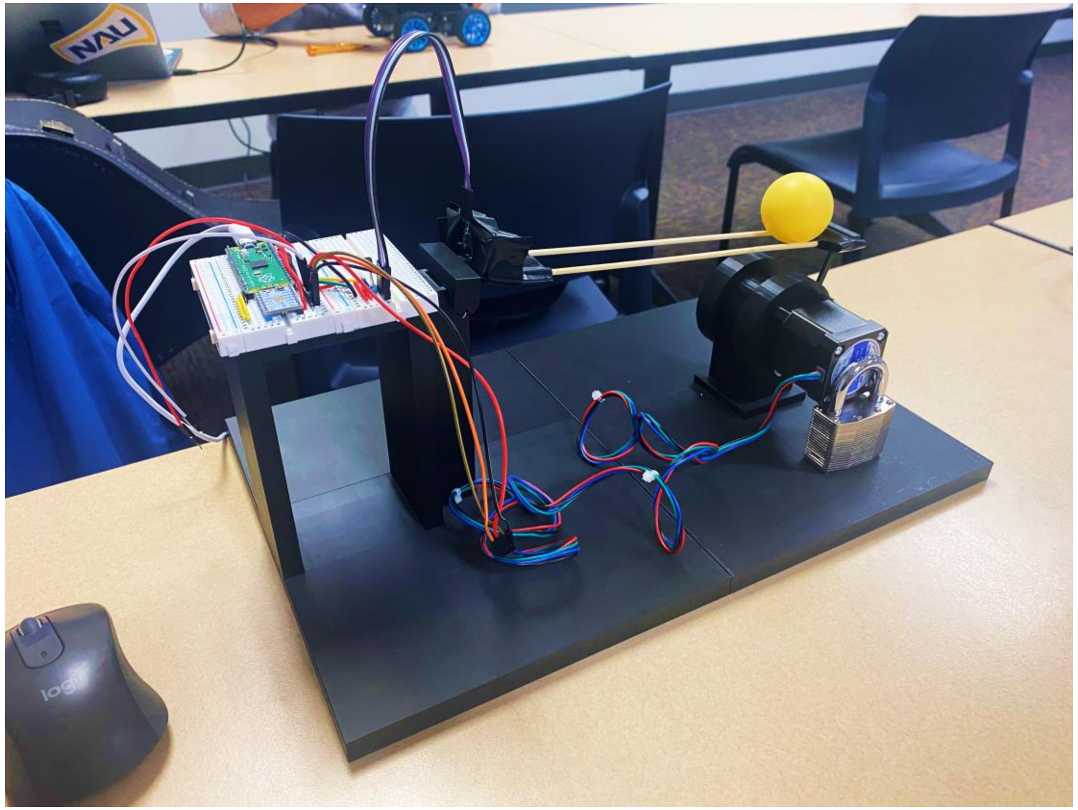


Figure 15: Working prototype Ball and Beam Robot

B.B.B. – Circuits

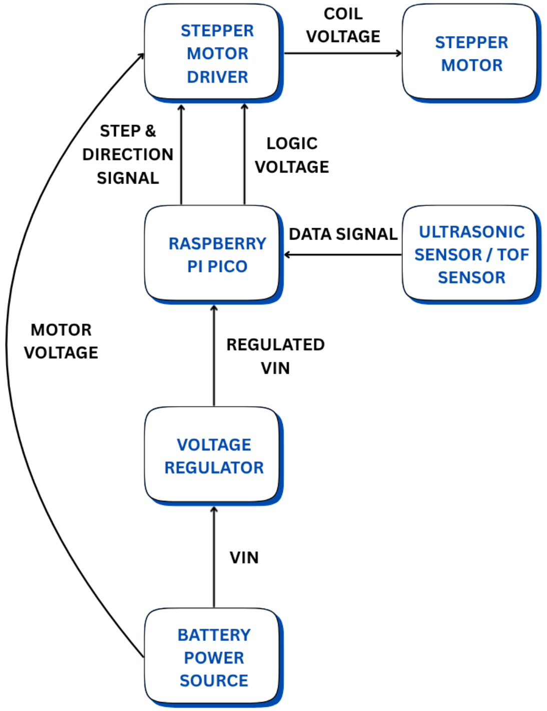


Figure 16: Circuit Diagram for Ball and Beam Robot

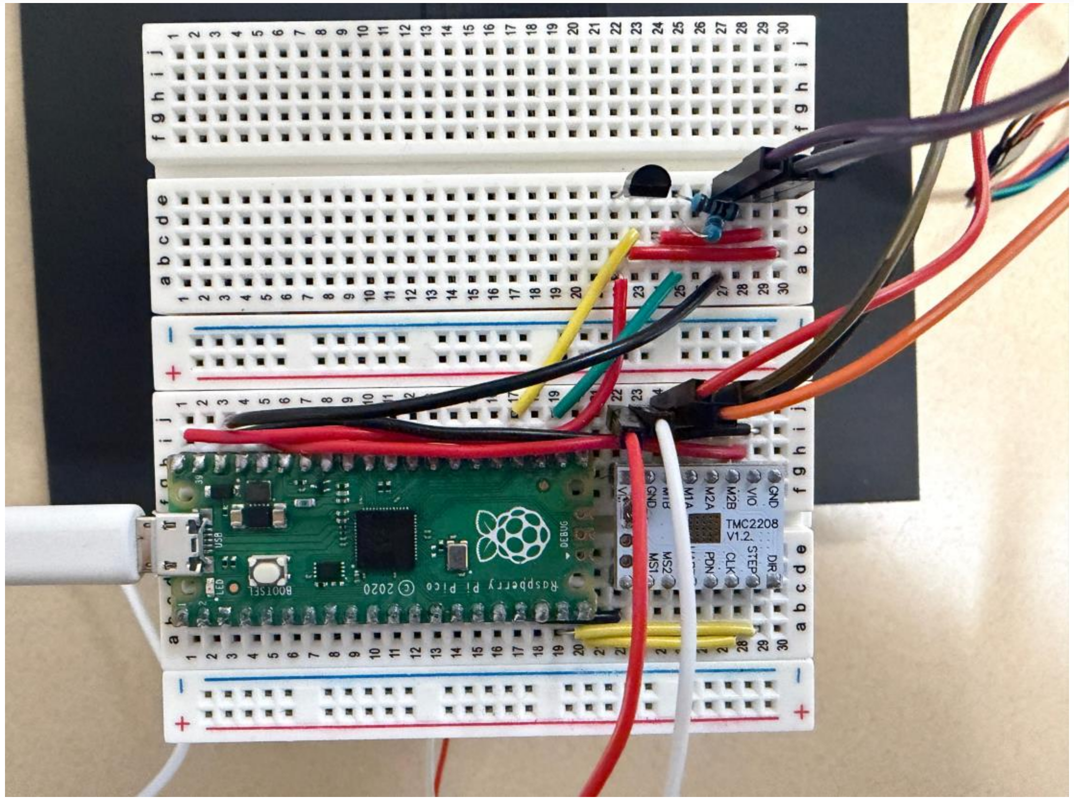
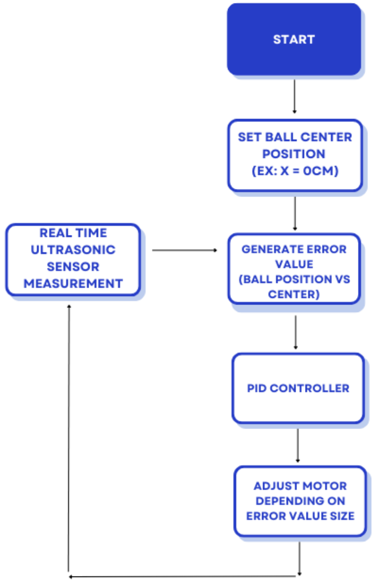


Figure 17: Physical circuit for Ball and Beam Robot

B.B.B. - Code



START

SET BALL CENTER POSITION (EX: X = 0CM)

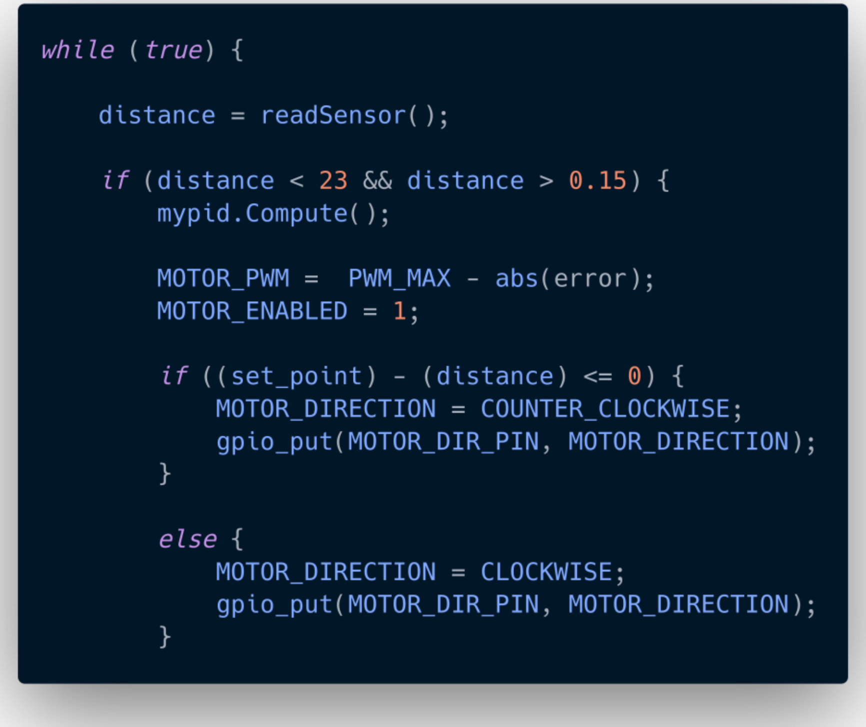
REAL TIME ULTRASONIC SENSOR MEASUREMENT

GENERATE ERROR VALUE (BALL POSITION VS CENTER)

PID CONTROLLER

ADJUST MOTOR DEPENDING ON ERROR VALUE SIZE

Figure18: Code flowchart



```
while (true) {

    distance = readSensor();

    if (distance < 23 && distance > 0.15) {
        mypid.Compute();

        MOTOR_PWM =  PWM_MAX - abs(error);
        MOTOR_ENABLED = 1;

        if ((set_point) - (distance) <= 0) {
            MOTOR_DIRECTION = COUNTER_CLOCKWISE;
            gpio_put(MOTOR_DIR_PIN, MOTOR_DIRECTION);
        }

        else {
            MOTOR_DIRECTION = CLOCKWISE;
            gpio_put(MOTOR_DIR_PIN, MOTOR_DIRECTION);
        }
```

Figure 19: A brief code snippet of the code running the P.I.D.

B.B.B. Future

- This robot serves as a predecessor to a more advanced robot: the Ball on Plate Robot.
  - B.B.B. is being used to iron out how everything interacts together.
  - Instead of one dimension, this will balance a ball in two.
  - Extra interactivity with a resistive touch screen instead of two beams.

Questions