Obstacle Avoiding Robot

By Rodrigo Lamas

To view live demo click here: https://www.youtube.com/watch?v=4XGYt0cm734

Introduction

The obstacle avoiding robot is an autonomous robot whose whole purpose is to keep moving while avoiding any and all obstacles such as walls, boxes, or water bottles. Once the robot encounters an obstacle it will stop, move back 3 centimeters, check its surroundings, and readjust its route away from obstacles. This is accomplished using the HC-SR04 ultrasonic sensor, a motor shield with integrated L293D chips, and two DC motors.

Certain pitfalls of the robot has trouble is traversing mazes, although it can move through them, they need to be very ridged mazes because the robot has trouble moving straight due to its third wheel that keeps it standing up. This third wheel is very delicate to movement which makes him trail off from the straight trajectory given to the robot by its two DC motors. Another limitation of the robot are obstacles that are out of its peripheral vision. Obstacles such as soda cans or rocks are too short to be picked up by the sensor. In similar behavior, potholes or edges in a table or steps will also see the robot fall. Lastly obstacles at an angle are also an issue to the robot because trigger pin of the Ultrasonic sensor sends out an ultrasonic wavelength that is picked up but the echo pin. In the unfortunate event that obstacles are at an angle, the ultrasonic wavelength will bounce off the wall into the distance instead of going back to the echo pin. Thus the robot will crash into the obstacle. But once facing the obstacle the sensor will pick it up and it will proceed to readjust its trajectory.

Complexities

Successfully implemented complexities:

- HC-SR04 Ultrasonic Sensor (Used to detect obstacles)
- Motor Driver L293D (Create communication to DC motors without a breadboard)
- DC Motors (Used to move the robot)

Complexities not used/changed from project proposal

- Servo Motor
    - In the project proposal it was described that I would be using Servo motors to move the robot. However, I decided to switch to DC Motors because of the limited range of motion of the Servo Motors.

User Guide

For operating the obstacle avoiding robot is best if user deploys it on flat surfaces such as tile floor or kitchen counters rather than textured surfaces such as carpets. The robot still works on carpets, but it will just move slower. Surfaces like grass, gravel, sand or dirty will not work with the robot either. To deploy the robot, one must just set it on a surface and slide the switch on its right shoulder to on, once its on the robot will begin to move autonomously. To turn the robot off one must slide the switch back to off. Any obstacle with that is 15 centimeters or less away from the robot will cause it to stop, look around and readjust its path.

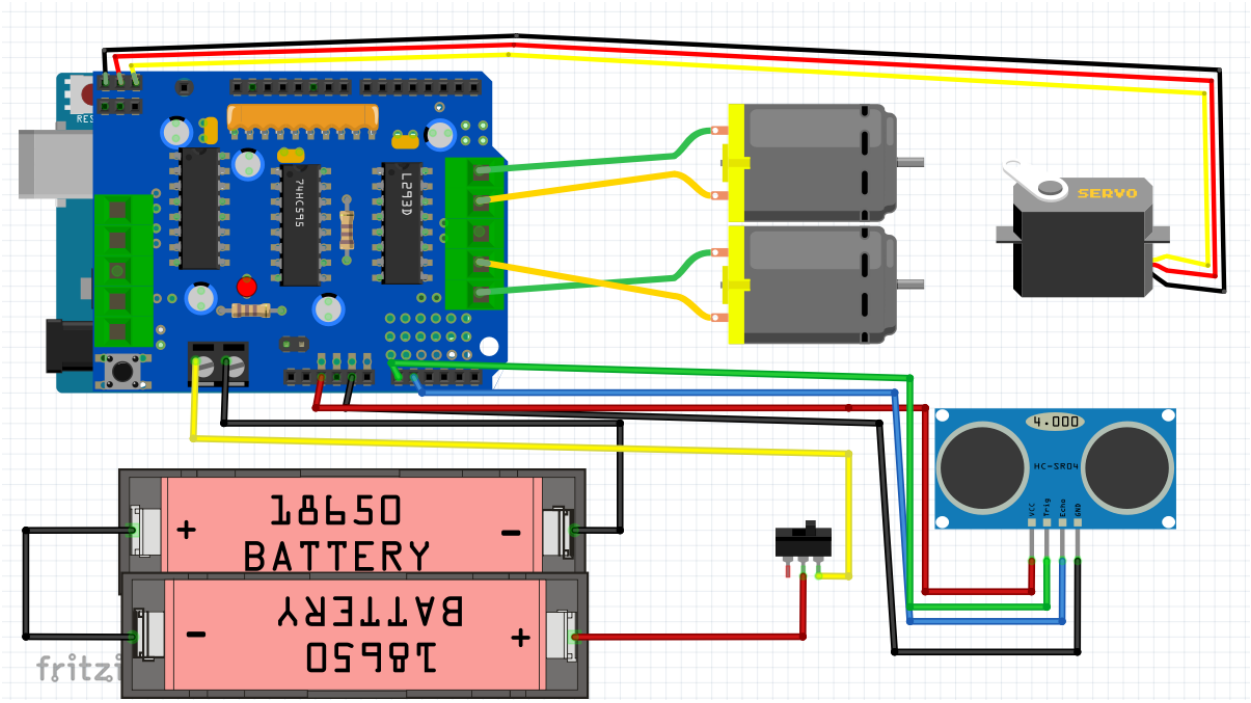
Hardware Components Used

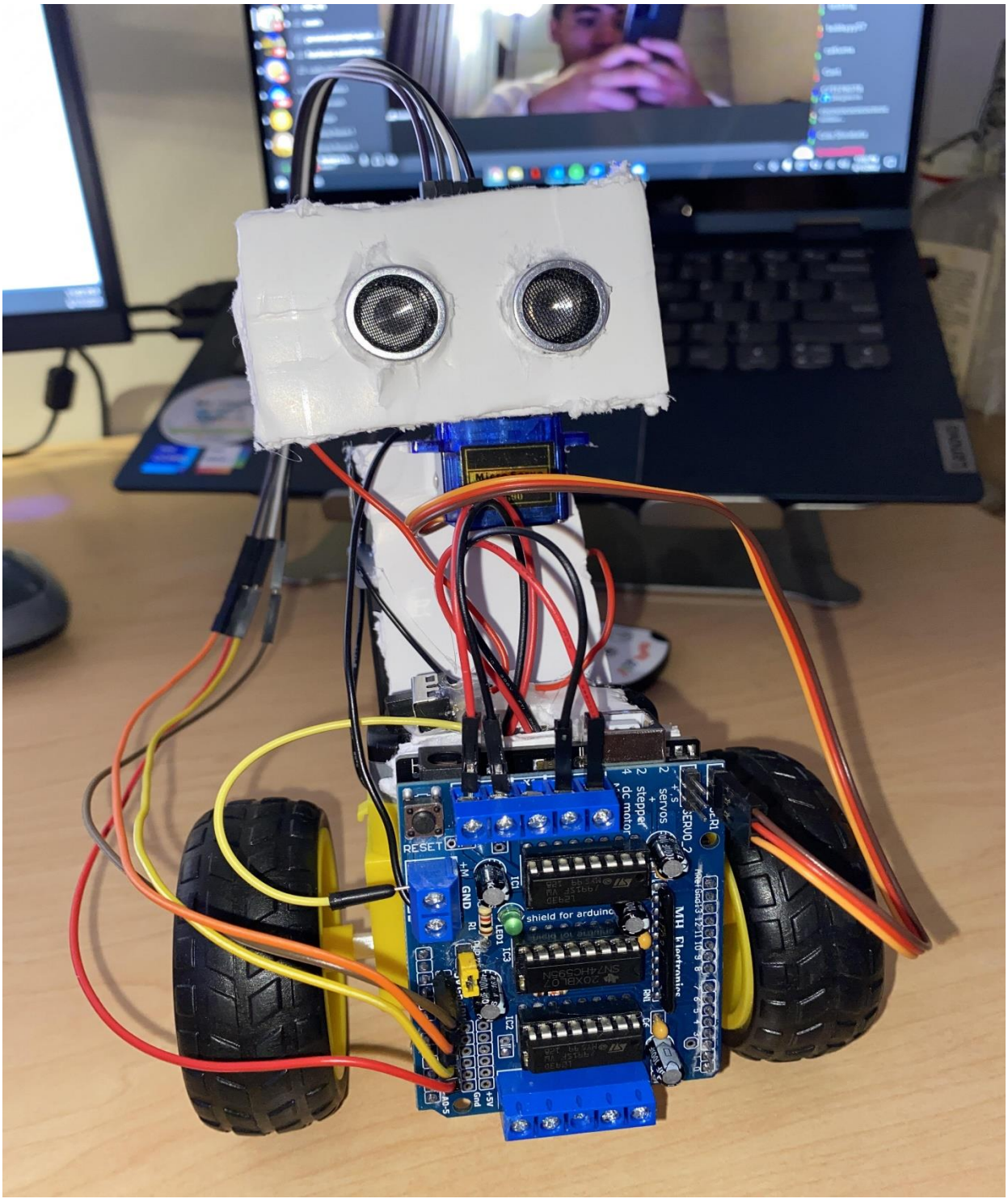
- HC-SR04 Ultrasonic Sensor (Used to detect obstacles)
- Motor Driver L293D (Create communication to DC motors without a breadboard)
- DC Motors (Used to move the robot)
- Arduino Uno
- Servo Motor
- Slide switch
- Two 18650 battery holders
- Two 18650 batteries

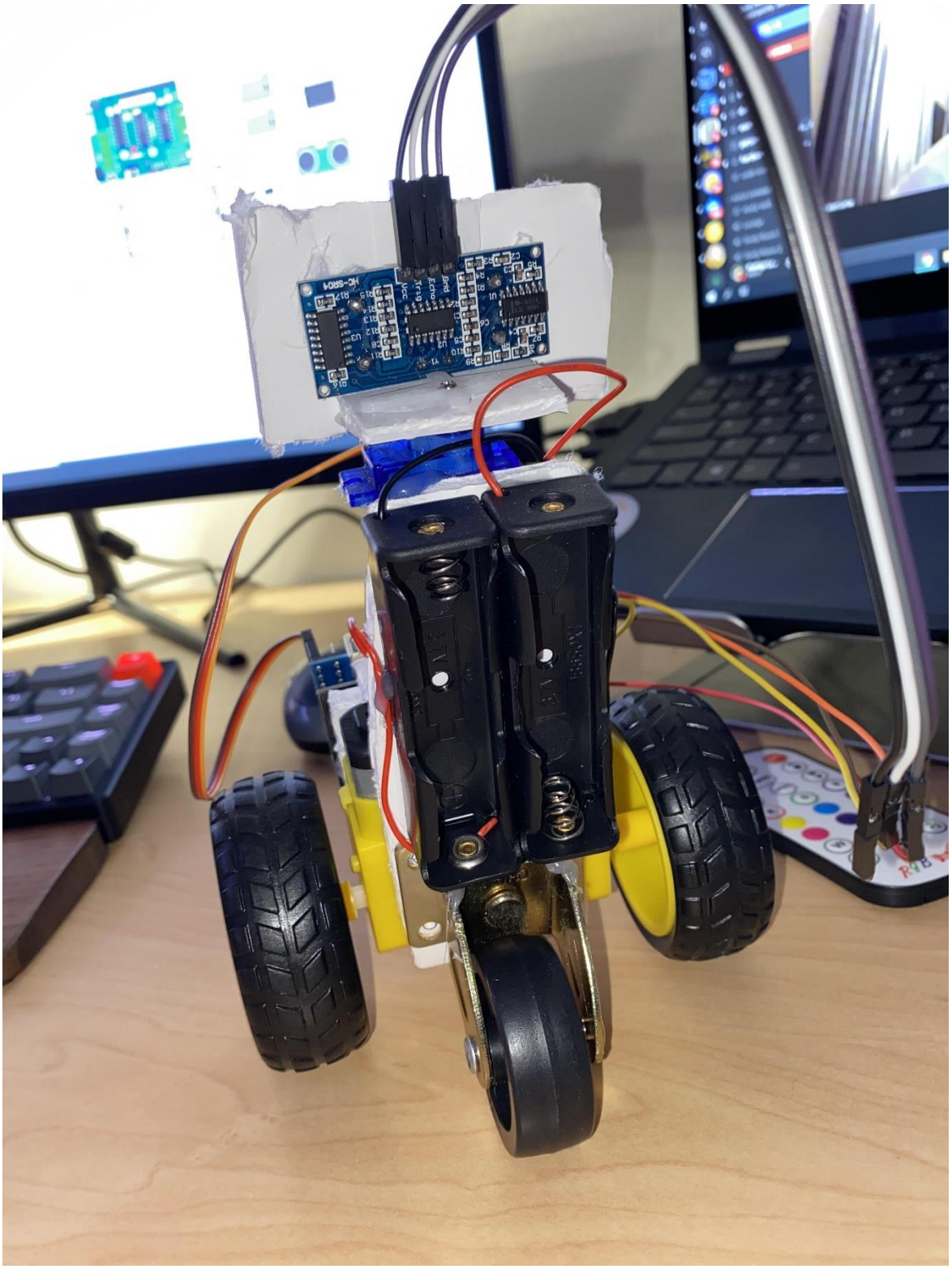
Software Components Used

- Used the Adafruit AFMotor.h library.
    - https://learn.adafruit.com/adafruit-motor-shield/library-install
    - This library provide support for speed and direction control of up to four DC Motors. It was used to declare and control the movement motors on the robot.
- Used the Arduino NewPing.h library
    - https://github.com/livetronic/Arduino-NewPing
    - This library makes working with the ultrasonic sensor easy. It allows the user to declare ultrasonic sensor objects within their code called sonar. It also has functions that read in the input from the sensor and automatically translate the readings to integers that represent the distance of objects in centimeters.
- Used the Arduino Servo.h library
    - https://github.com/arduino-libraries/Servo
    - This library provided support for using servo motors. It allowed access to the integrated gears and shaft to precisely control the rotation of the motor angles ranging from 0 to 180 degrees.

Wiring Diagram

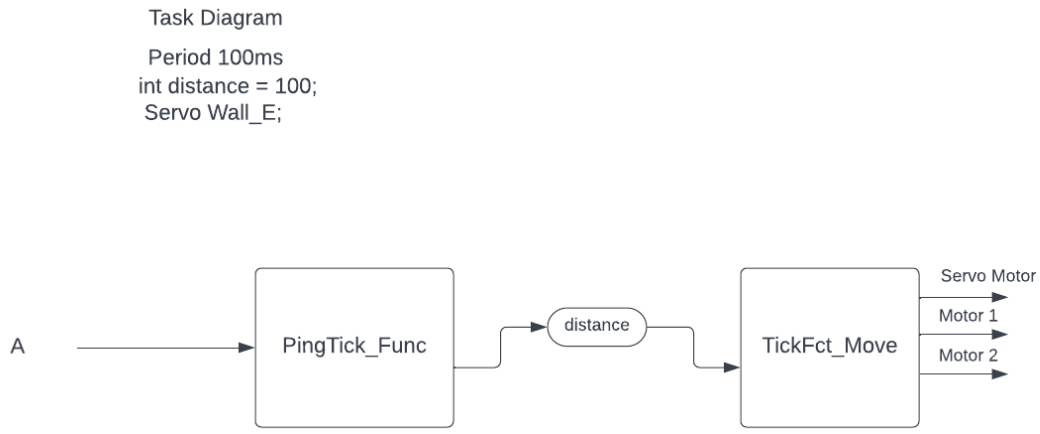

SERVO
4.000
HC-SR04
VCC
Trig
Echo
GND
18650
BATTERY
+
-
18650
BATTERY
-
+


RESET
+ M
GND
shield for arduino
LED1
servos
stepper
dc motor
SERVO_2
MH Electronics


HC-SR04
Trig
Echo
Vcc
Gnd

Task Diagram

The motor shield used to control the DC motor sits on top of the Arduino connecting to all analog and digital pins. The shield has dedicated ports to which one connects up to four DC motors and two dedicated ports for connecting Servo Motors. Thus the shield takes care of delegating all the different pin connections without the user having to worry about it. Therefore, my task diagram just outputs to the two DC and servo motors as is unclear which digital pins are being used.



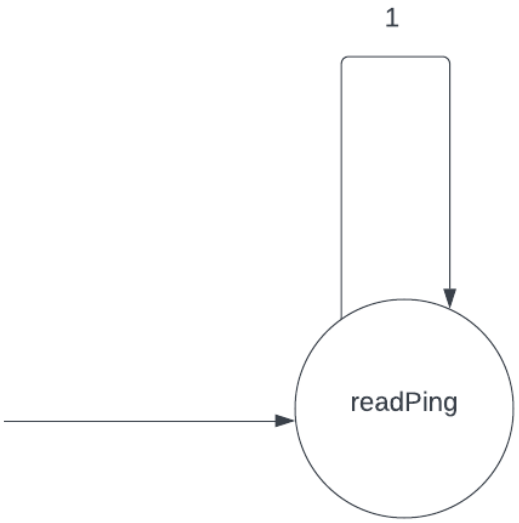
Next page has the SynchSm diagrams

SynchSM Diagram

Inside PingTick_Func:

Period 100ms

```
int trig_pin = A0;
int echo_pin = A1;
int max_cm_distance = 200;
NewPing sonar(trig_pin, echo_pin, max_cm_distance);
```




```
distance = sonar.ping_cm();

if(distance == 0){
      distance = 250;
}
```

Inside TickFct_Move