CSCE 336 Embedded Systems Robot Design Project 2 Report

Name of Project & Robot: Robot Driving and Sensing

Student's Name: Dylan Mostek School of Computing University of Nebraska – Lincoln

Due Date: 4/01/25

Table of Contents

Objectives or Purpose: The main goal of this project was to create a robot that can move forward and avoid obstacles in its path. Instead of moving in a straight line without awareness, the robot needed to use an ultrasonic distance sensor to detect objects. When it noticed something too close, it would stop, turn, and then continue moving in a different direction. This way, it wouldn't crash into obstacles.

Additionally, I wanted to make the code reusable for future projects. To do this, I split the motor and sensor logic into two separate libraries — one for movement and one for the sensor and servo "head." This modularity allows the same setup to be used for more advanced tasks, like navigating a maze.

Preliminary Design: I broke the project into two main parts:

1. Movement: Using motor pins and PWM to control speed.
2. Head Movement: Using a servo to turn the distance sensor and scan the area.

I created two basic libraries for simplicity:

- motors.h / motors.cpp for movement control
- head.h / head.cpp for controlling the head and sensor

Here's an example function from the motor library for moving forward:

```
void motorMoveForward() {

   digitalWrite(4, HIGH);

   digitalWrite(5, LOW);

   digitalWrite(6, HIGH);

   digitalWrite(7, LOW);

}
```

Software Flow Chart or Algorithms: The robot's logic is simple:

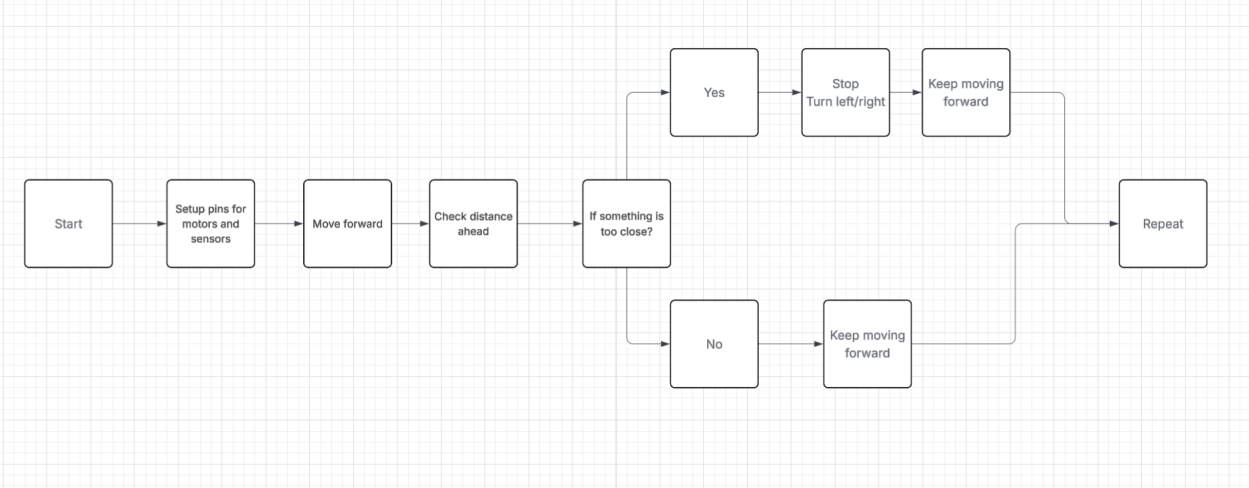
1. Move forward
2. Check distance using the ultrasonic sensor
3. If an object is detected close by, stop and turn
4. Repeat the process

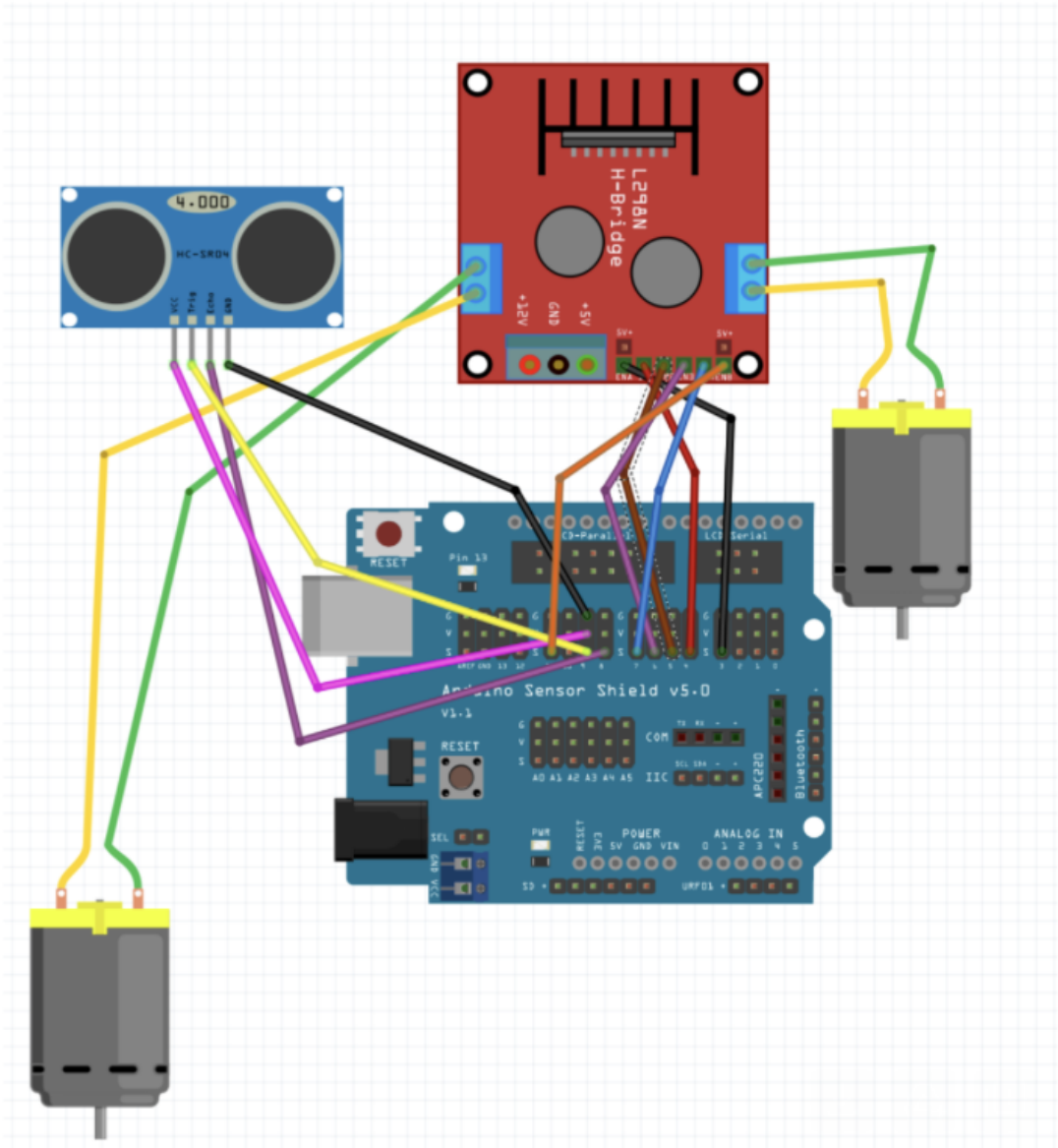
The follow_wall() function handles this decision-making process.

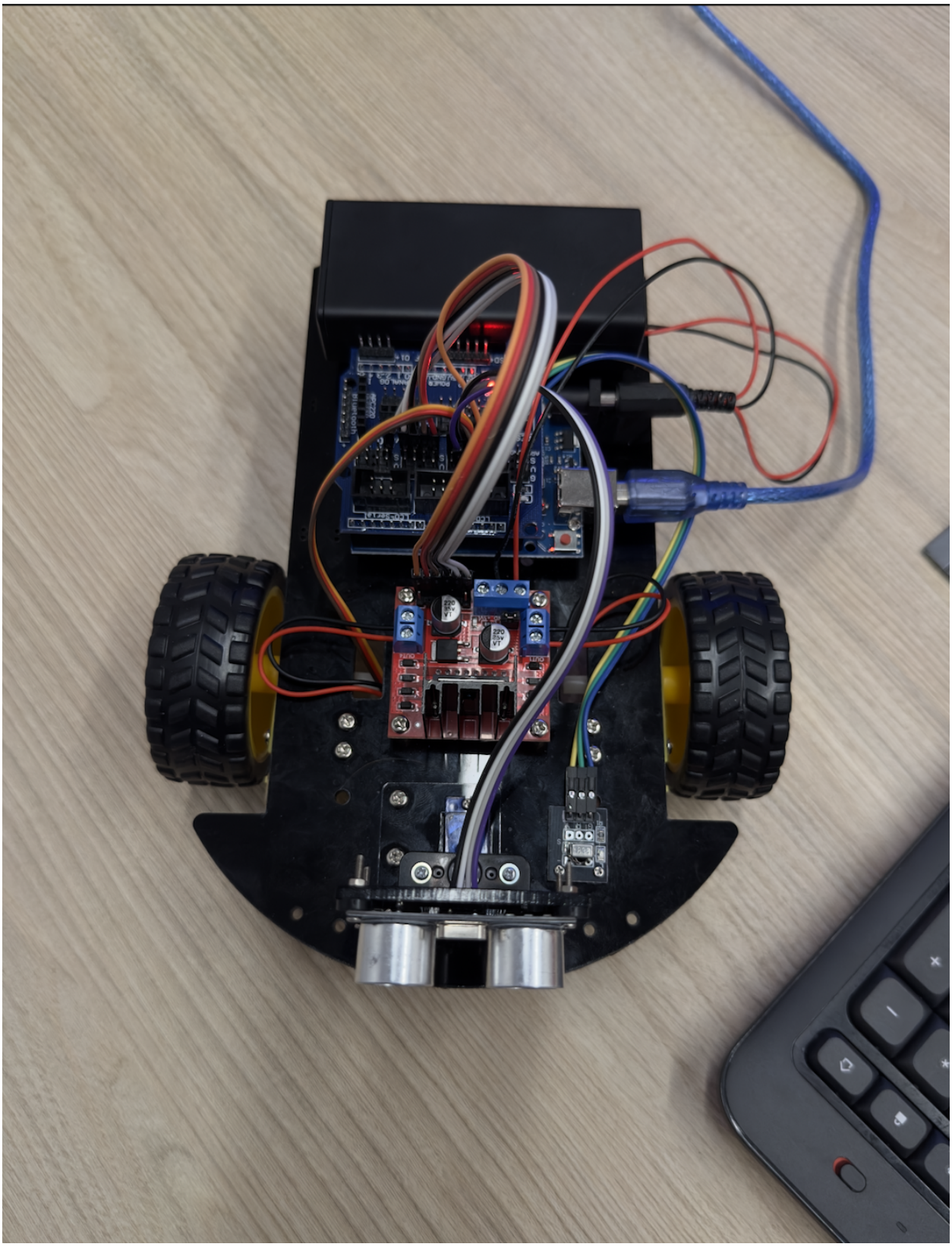
Hardware Schematic: The following components were used in the robot:

- Motor direction pins: 4, 5, 6, 7
- Motor speed (PWM): 3 and 11
- Servo motor: pin 10 (uses Timer1)
- Ultrasonic sensor:
    - Trig: pin 9
    - Echo: pin 8

Include the following pictures at the end of this section:


Start
Setup pins for motors and sensors
Move forward
Check distance ahead
If something is too close?
Yes
Stop
Turn left/right
Keep moving forward
No
Keep moving forward
Repeat


4.000
HC-SR04
L298N
H-Bridge
RESET
Pin 13
RESET
COM
IIC
POWER
ANALOG IN
Bluetooth



Debugging: Some problems arose during development:

- Servo not moving: Fixed by adjusting Timer1 settings.
- Motor direction issues: Corrected the HIGH/LOW values for direction pins.
- Unreliable sensor readings: Added a small delay after the trigger pulse to stabilize data.

Testing Methodology or Results: I tested each part separately:

1. Motors: Verified forward, backward, left, and right movement.
2. Servo: Confirmed it could move left, center, and right as expected.
3. Distance Sensor: Checked for correct distance measurements.
4. Full Robot Test: Ensured that the robot could stop and turn upon detecting an obstacle.

Answers to Lab Questions:

1. How fast does the signal from the sensor travel? - 340 m/s
2. Time for pulse to return (1 in): 148 μs
3. Time for pulse to return (1 cm): 58.8 μs
4. Sensor range and accuracy: 2 cm to 4 m, with 3 mm accuracy
5. Minimum recommended measurement cycle: 60 ms (working frequency: 40 Hz)

Observations and Conclusions: The project was successful in making a robot that could move forward and avoid obstacles. The modular design using libraries made the code easy to manage and reuse. I also learned how to control motor speed using PWM and how to get accurate distance readings with an ultrasonic sensor.

Documentation: I completed the project independently, but I referenced Arduino documentation and course slides for some technical details. All code versions were stored in my Bitbucket repo with relevant commit messages to keep track of changes and improvements.