AUTONOMOUS LIGHT SENSING ROBOT

Sharjeel Arif

200388331

Mayer Javed

200415906

Table of Contents

1.0 System Information

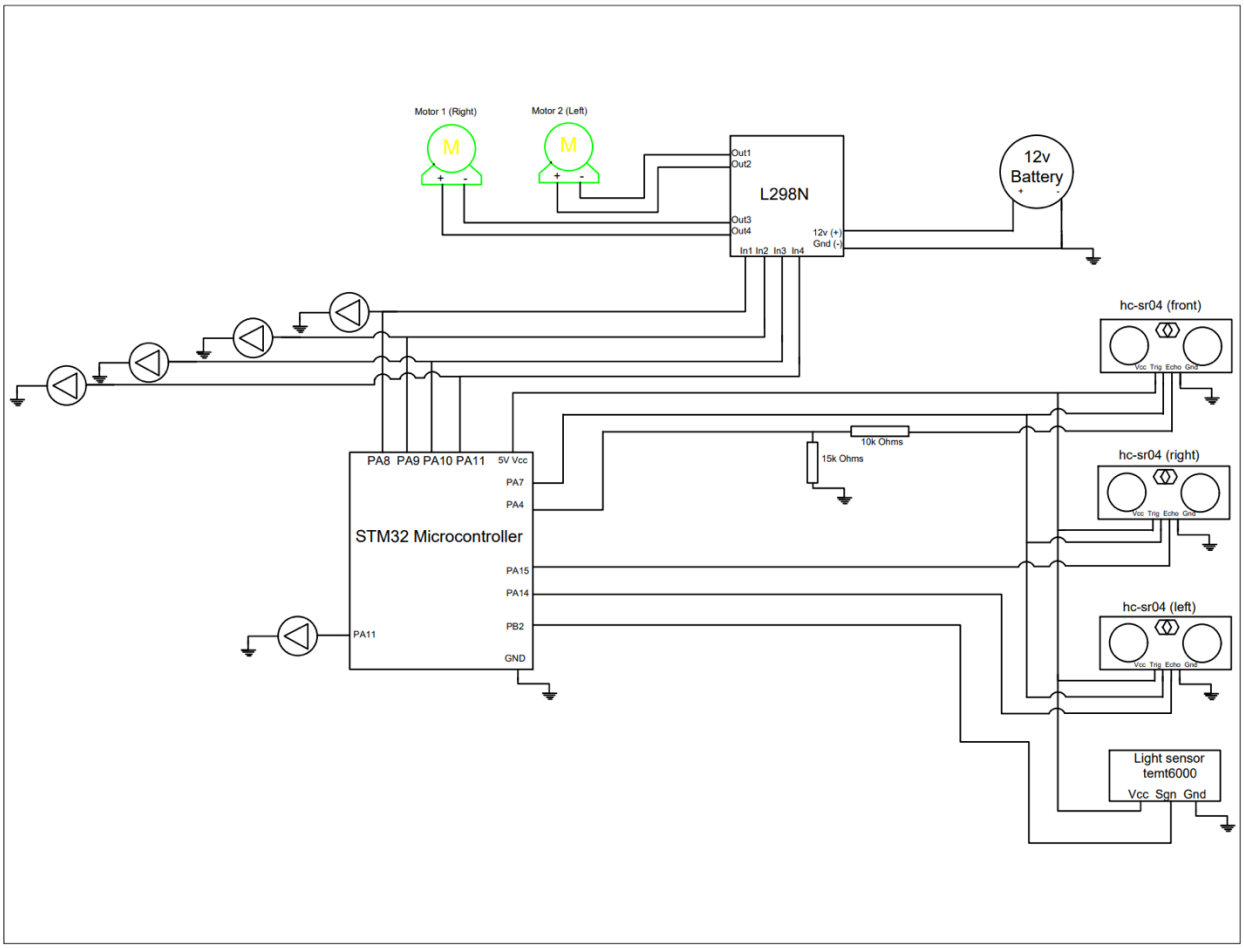
1.1 Description

The Autonomous Light Sensing Robot is capable of navigating through any terrain/maze it is introduced to. The goal of this robot is to mimic a hard-to-reach area that humans are not normally able to enter. The robot will be able to navigate through the maze and find a light source from a corner of the maze. The light source in this case mimics a radiation source or any other material that is harmful to humans.

1.2 Drive System

The robot will be front wheel drive, although for better traction, all-wheel drive is preferred, for the sake of simplicity and navigating tight corners we opted for front wheel drive. The robot will have one caster wheel in the back forming a triangle formation on the underside of the chassis. The caster wheel will allow the robot to make sharp turns on demand. For better stability two back wheels are preferred. However, again, for the sake of simplicity and dimension constraints due to the size of the maze only one caster is used.

1.3 System Diagram


Motor 1 (Right)
Motor 2 (Left)
M
M
+ -
+ -
L298N
Out1
Out2
Out3
Out4
In1 In2 In3 In4
12v (+)
Gnd (-)
12v
Battery
+ -
hc-sr04 (front)
Vcc Trig Echo Gnd
hc-sr04 (right)
Vcc Trig Echo Gnd
hc-sr04 (left)
Vcc Trig Echo Gnd
10k Ohms
15k Ohms
PA8 PA9 PA10 PA11
5V Vcc
PA7
PA4
STM32 Microcontroller
PA15
PA14
PB2
GND
PA11
Light sensor
temt6000
Vcc Sgn Gnd

1.4 User Interface

The user interface for this autonomous robot will not consist of much as we want most of this robot to be able to function autonomously. The user interface will therefore only consist of an LED to alert the user that the light source has been found. As well as including a start switch to start the robot's tasks.

1.5 Ratings and Voltage Specifications

Battery	12V
Drive Motor	5V
All Other Components will be regulated with a MP1584EN DC-DC Buck Converter.	

2.0 Operation

2.1 Startup

Once the start button is pressed on the breadboard (pushbutton switch) The robot will execute the flashed memory within its storage. The robot will have different modes it will enter when running. These three modes consist of USS Sensor Triggered, Limit Switch Triggered, Nominal Operation, and finally Light Source Found.

2.2 Nominal Operation

Nominal operation is a routine that will run when no other sensors are triggered. This routine will consist of the robot driving forward while no sensors are triggered. The next possible routine the robot can encounter is the USS Sensor Triggered or Limit Switch Triggered;

2.3 USS Sensor Triggered/Limit Switch Triggered

While driving the robot can encounter either one of the following routines, USS Sensor Triggered, Limit Switch Triggered. USS Sensor Triggered will be triggered if the robot gets too close to a wall. In which case the ultra-sonic sound sensor will raise a flag and the robot will adjust its motor speeds accordingly to avoid hitting the wall. The three ultra-sonic positioned at the front of the robot will allow it to detect walls from three directions, front, left, right. The pseudocode for this routine will resemble the following:

```
While(1) {
    If(left USS && right USS == 1) {
        Left motor = 1
        Right motor = 1
    }
    Else If(front USS && right USS == 1) {
        Left motor = 0
        Right motor  = 1
    }
    Else If(front USS && left USS == 1) {
        Left motor = 1
        Right motor = 0
    }
    Else {
         Uturn
    }
```

```
}
```

Note: The ultrasonic sensors will be set to detect at a low distance to avoid any unnecessary turns.

The robot does not have a sensor at the back side. This is because the robot's main function is to drive forward. However, we suspect that the robot's back side can hit a wall during turns. To overcome this issue, we decided to add limit switches to either side of the rear of the robot. These limit switches will trigger if the robot hits a wall while turning. In the case it does the robot will be set to reverse a few cm and try to make that turn again. The pseudocode for this routine will resemble the following:

```
If(LLS == 1) {
    Right motor = -1
    Left motor = -1
    Wait 500ns
}
If(RLS == 1) {
    Right motor = -1
    Left motor = -1
    Wait 500ns
}
```

The final routine that can be triggered is the Light Source Found routine. This routine will only be triggered if the robot has found the light source. As the robot is driving it will constantly be checking for a light source. The light source will be detected through

the photo sensor if the photosensor is ever triggered then the LED on the top of the robot will turned on to alert the user that the light source has been found.

3.0 Physical Construction

When constructing the robot a few things had to be considered in regards to the physical appearance. The main things that we had to keep in mind were the drive system, chassis, and the layout of the wheels to maintain balance when the robot would drive.

3.1 Drive system

For the drive system we decided to have a front wheel drive robot just for the ease of outputs on the microcontroller. Furthermore, we decided that having a front wheel drive motor would be easy to code for. As the main functions in driving would be easy to construct. Drive forward would be both motors turning forward. Driving in reverse would be both motors switching polarity and turning in reverse. Turning left would be the right motor operating with a positive polarity and the left motor operating in negative polarity creating a sharp left turn. Right turns would be the same as the left but the polarities would be swapped.

3.2 Chassis

The chassis we decided to 3D print. This is because the lack of availability to get one shipped in optimal time would cause significant delays in our project. Furthermore, we had easy access to a 3D printer and had previous experience working with 3D printers. Also, we had the freedom of designing based off our design and where we wanted the sensors and motors to be mounted. This is why we decided to 3D print our robot.

3.3 Layout

In regards to the layout of our design we decided to have two motors to drive the robot and a single caster wheel in the rear to offer stability and sharp turns in tight corners of the maze. For the actual placement of the robot, we decided to have 3 Ultrasonic sound sensors. One looking front, right, and left. The light sensor was situated at the front of the robot with no obstructions.

3.4 Table of figures

The final build of the robot can be seen in the following images:

Figure 1:

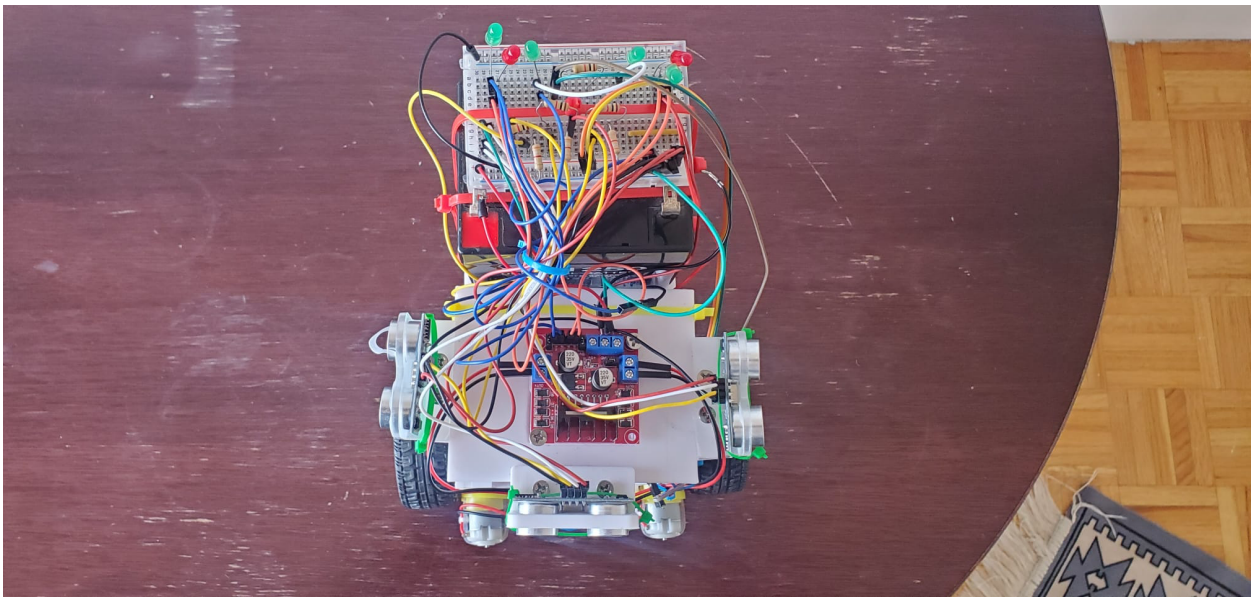


Figure 2:

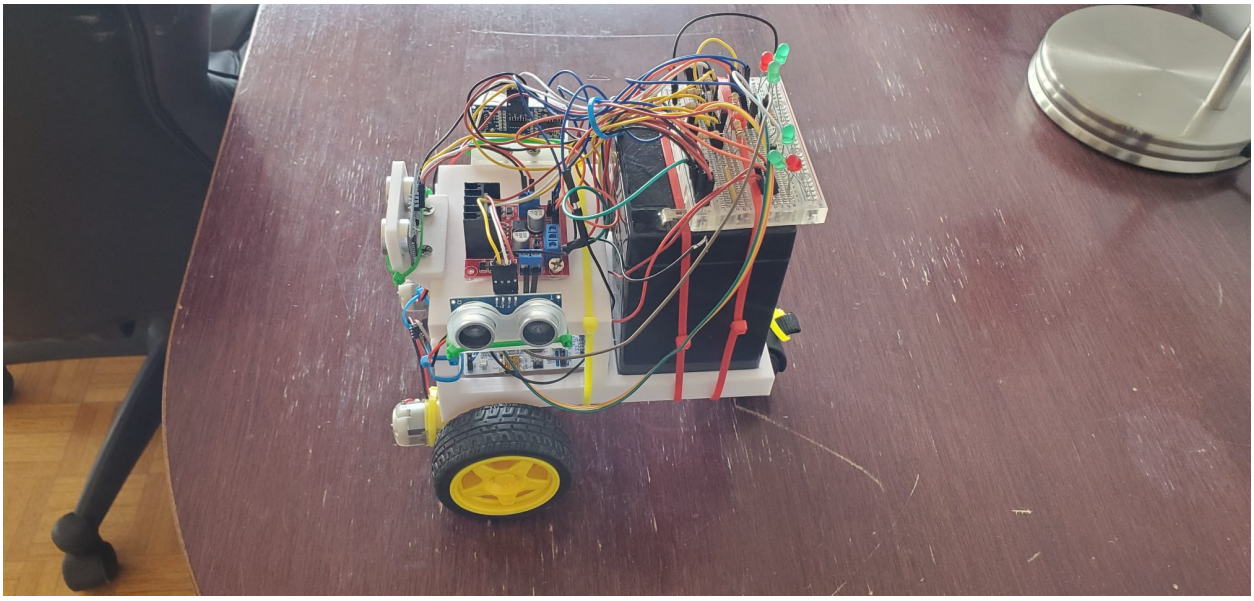
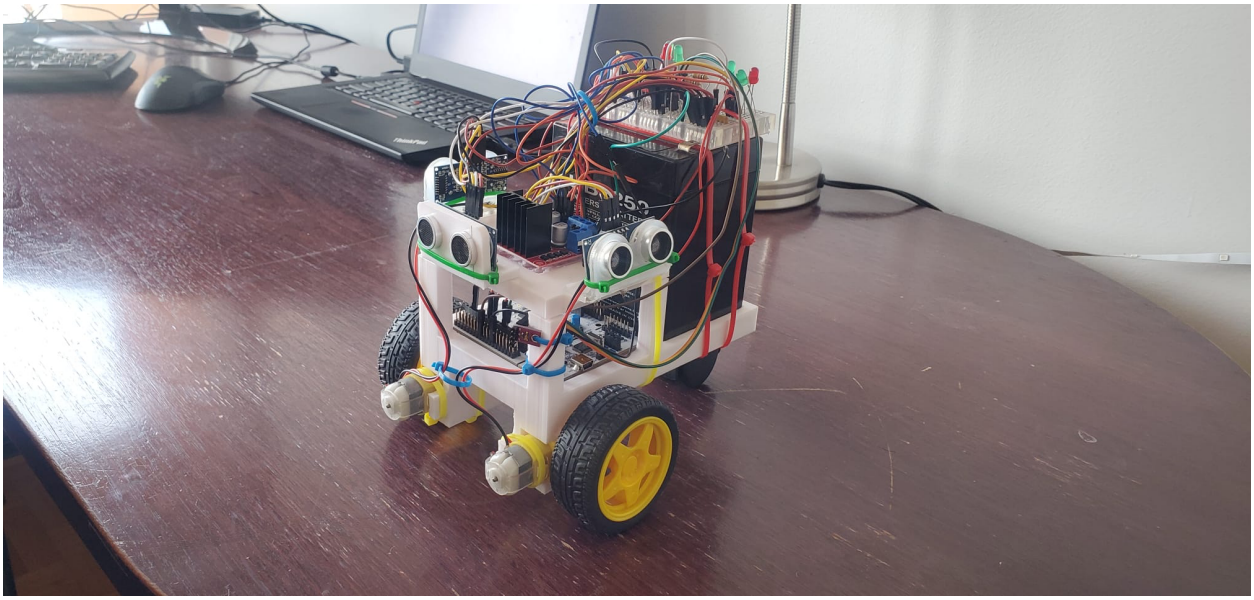


Figure 3:



4.0 Connections Inputs and Outputs on the STM32F103RB

4.1 Sensors

4.1.1 Ultra-Sonic Sound Sensor

Each ultrasonic sound sensor had 4 pins that needed to be connected. The Vs and GND were connected to 5v and GND on the STM32. The trigger pins on all the ultrasonic sound sensors were connected to the same PA7 pin on the microcontroller as they were outputting the same PWM signal. The echo pins however were connected to separate pins as each input from different sensors would have a different routine to follow. The right ultrasonic sound sensor was connected to PA15. The left ultrasonic sound sensor was connected to PA14. Finally, the front ultrasonic sound sensor was connected to PA4. The front ultrasonic sensor was connected on a separate 3.3V line as connecting it to a 5V line cause all the sensors to bug out and not work properly. We suspected the reason for this was because the sensors caused interference with each other with noise on the echo lines.

The right and left ultrasonic sound sensor's inputs were being read and their inputs were being converted into distance in mm. Whereas the front ultrasonic sound sensor we set it to a 3.3V ADC pin on the microcontroller and used if statements to control the output of the ultrasonic sound sensor.

4.1.2 Drive Motors

The drive motors were split into 4 different pins on the microcontroller to for ease of control on polarity with the drive controller we had. As a result, the positive polarity for turning the right and left motors forward were on pins PA8 and PA10 respectively. The negative

polarity for turning the right and left motors in reverse were on PA10 and PA11

4.1.3 Light Sensor

The light sensor's input was on PB2 and the output to the LED used PA11

5.0 Troubleshooting

5.1 Troubleshooting Sensors

There was a set routine for us when troubleshooting sensors. At first if something was not working as intended. For example, say that the left and front ultrasonic sensor were detecting obstacles but the robot failed to turn left to avoid the obstacles we would first use the ADALM to see if the outputs and inputs were being interpreted properly by the sensor. If all the readings were correct, we would go to check the code. We would check the code using the debugger to see if there were any bugs and hiccups preventing the expected output to be different.

5.2 Testing Stages

5.2.1 Stage 1:

The tests on the robot were also done in multiple different stages. The first test was conducted when the drive motors were attached. This would ensure that our drive motors were not faulty and that they worked prior to constructing the entire robot. The drive motors were tested by implementing code that made each motor go forward and reverse. In different sequences

5.2.2 Stage 2:

The second stage of testing was done when the ultrasonic sound sensors were hooked up and wired in properly. The test for this was first to check if each ultrasonic sound sensor was operating properly. We made sure they were by connecting the oscilloscope to the echo pin on the ultrasonic sound sensor while passing a PWM signal into the trigger pins. We would use the oscilloscope feature on the ADALM to see if we could read the echo properly. Once verifying that all ultrasonic sound sensors connected were functioning properly. We then had to make sure that they would interact with the drive motors correctly. To do this we connected the drive motor outputs to the ultrasonic sound sensor and we would manually introduce an object and see if the drive motors would interact properly

5.2.3 Stage 3:

Stage 3 of testing involved testing the light sensor to see if it was functioning properly. For this stage of testing. We hooked up a volt meter to the output of the light sensor. To see if the voltage output would change depending on the light input it received.

5.2.4 Stage 4:

The final stage of testing and troubleshooting included testing the entire robot as a whole. This testing was done by placing the robot in the scenario it would need to navigate. This test would ensure that all the robot's system worked in accordance with each other and did not malfunction.

6.0 Design Improvements/Recommendations

While our design did indeed do the job that it was assigned to do at first. We came across many problems that prevented it from functioning again. One of the main issues we faced was the interaction between the drive motors and the ultrasonic sound sensor. The ultrasonic sound sensor coded to halt and then reverse one motor when encountering an object. The issue we had with this that without a significant enough delay, the drive controller would have two different polarities going through it at the same time. This would cause it to short. If we had extra time, we would improve this design flaw by having a dual drive controller system to handle the opposite polarity. Or we would have the motor only top or go forward. We realized that we can make the robot turn by having one motor halted and the other rotating forward. This issue caused our robot functionality to stop as the robot would not drive anymore.

7.0 Software Development:

The code was developed using a state-based functionality. Where the robot would always be in one state. The code files for the project can be found in the finalProjectCode folder.

8.0 Conclusion

This report outlines the construction and performance of the Autonomous Light Sensing Robot. This project followed and applied multiple skills covered in the lab section of ENEL 351. Such skills include ADC, PWM, GPIO, SET and RESET, etc. The overall design and coding aspect of the robot was relatively straightforward. The only main issue we ran into was the drive controller burning out. This report also outlines the step we could have taken to fix up those mistakes to have a fully functioning robot.