THE LINE FOLLOWER ROBOT

ROBOTIC APPLICATION DEVELOPMENT

Course work

Group Members:

MA-HDSE-211F-005 A S I Viraj

MA-HDSE-211F-019 S A H D S Arachchige

MA-HDSE-211F-012 D A C Imalka

MA-HDSE-211F-016 P H G N Chamod

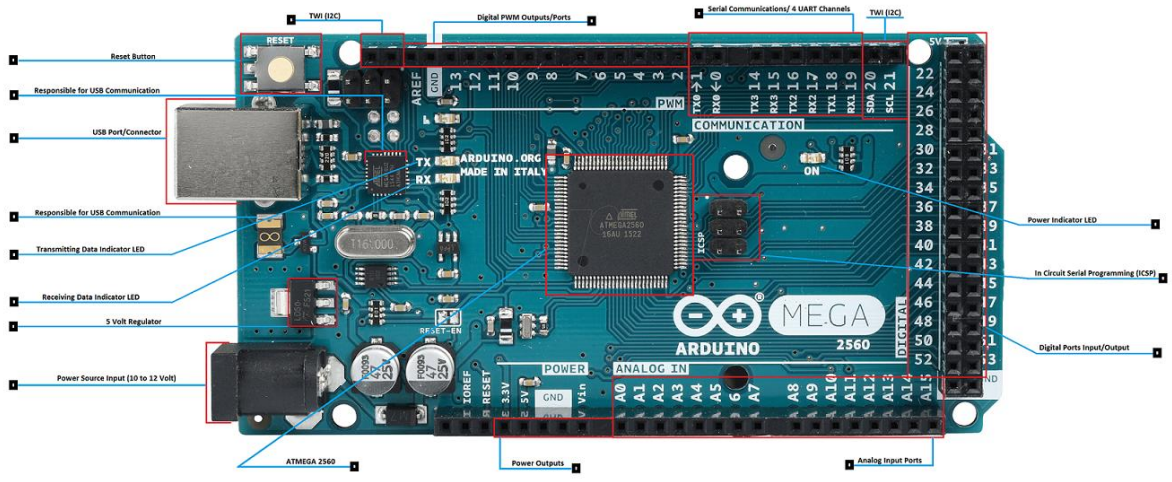
Table of Contents

Introduction

Robot is a machine that is usually designed to reduce the amount of human work where it is applicable. It is usually developed for reducing risk factor for human work and increase comfort of any worker. High performance, high accuracy, lower labor cost and the ability to work in hazardous places have put robotics in an advantageous position over many other such technologies. In this documentation a line follower has been presented which will trace a black line on a white surface. We have make use of sensors to achieve this objective. The main component behind this robot is Arduino mega2560 board which is a brain of this robot. The idea proposed in this document is by using machine vision to guide the robot We have made a robot that has several works to perform besides following a line. This robot follows a line without going to other direction.

Arduino Mega Board

ArduinoMEGA2560: The Arduino Mega 2560 is a type of microcontroller board. It has 54 digital I/O pins. Of which 15 can be used as PWM outputs, 16 are analog inputs, 4 UARTs (Hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, and a reset button. It contains everything needed to support the microcontroller. Simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.



Equipment Used

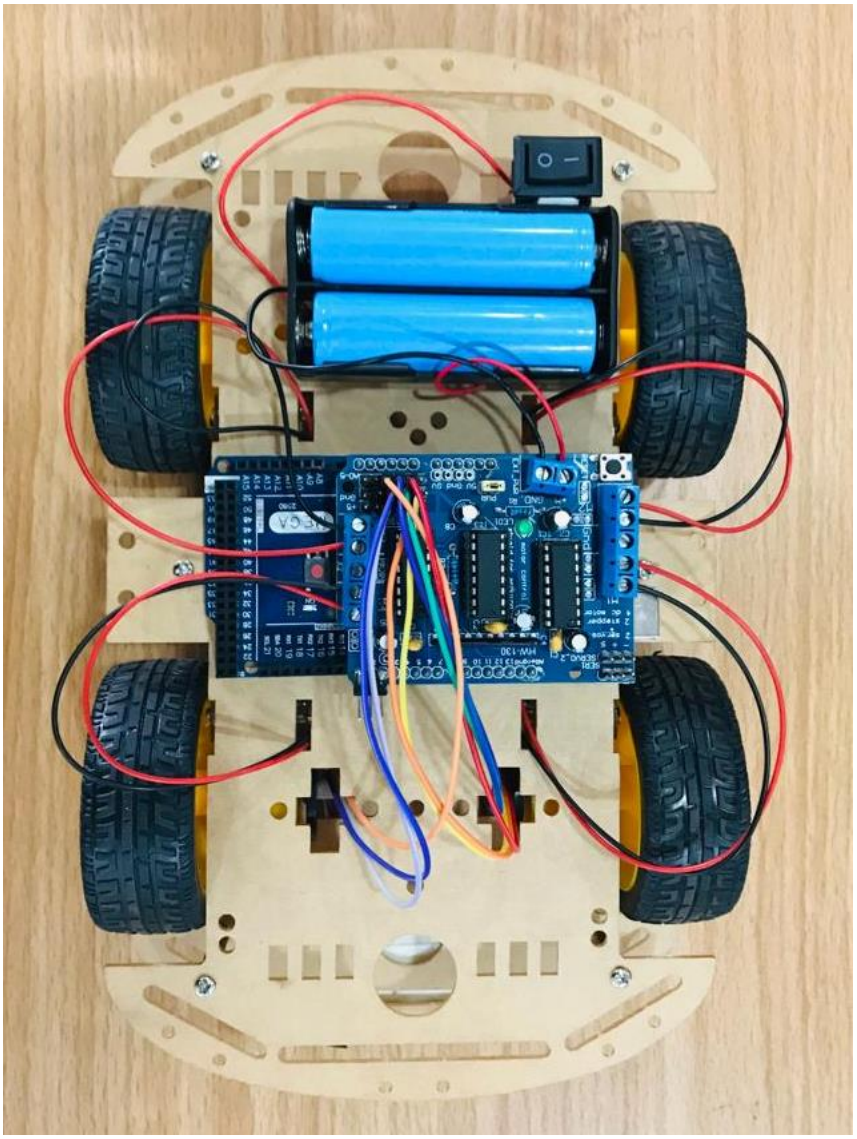
• Arduino Mega	01
• Motor shield	01
• Smart Robot Car chassis and the body	01
• IR sensor module	03
• DC Gear motors	04
• Wheels	12
• Rechargeable battery (18650) 3000mah 3.7v	
• & holder.	02
• Nut& bolts	-
• Switch	01
• Jumper wires	-

Tools required: -

- Soldering Iron
- Hot Glue Gun
- Screw Driver

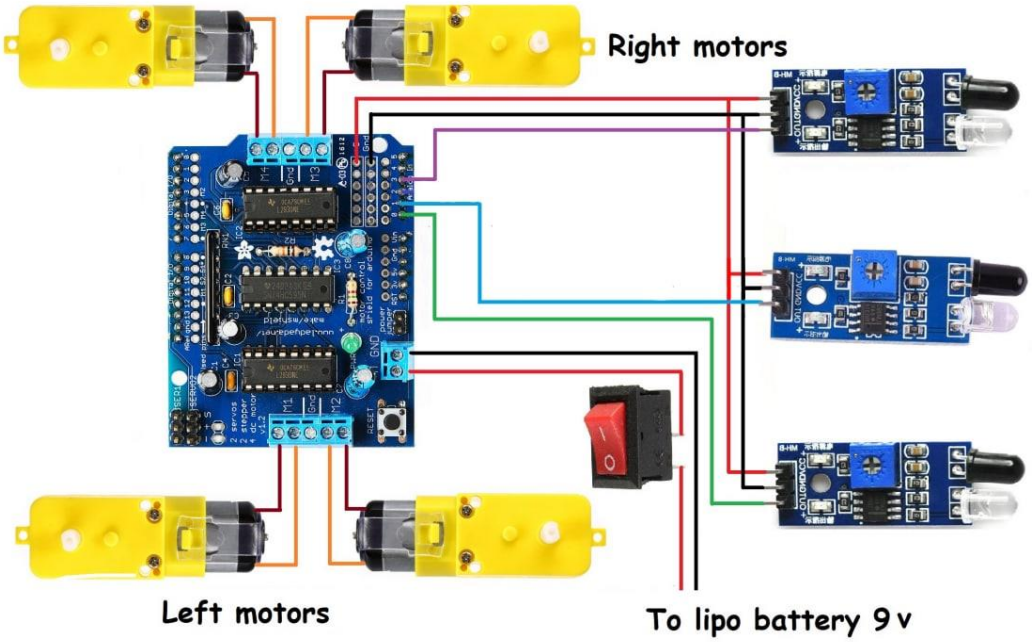
Block Diagram

Here firstly, we chose a configuration to develop a line follower only using two infrared sensors with connection of Arduino Mega through motor driver IC. We followed a block diagram on the regard. The block diagram illustrates the connection for the development of the line follower which follows a black line on white surface.



After that, we have used the following block diagram for connecting 3 IR sensor modules with our line follower for obstacle detection purpose for our line follower.

Circuit Design



IR Proximity Sensors

Circuit connections for IR Proximity Sensors:

- The IR sensor is a 3 three wired sensors, in which the brown and the black wire are used to connect the sensor to the power supply, whereas, the red wire is connected to the load, or Arduino, in our circuit.
- The load wires of the IR sensors are connected to pin 8, 9 and 10.

The objectives:

- The robot must be capable of following a line.
- It should be capable of taking various degrees of turns.
- The robot must be insensitive to environmental factors such as lighting and noise.
- It must allow calibration of the line's darkness threshold.
- Scalability must be a primary concern in the design.
- The robot should be identifying various changes (like Y, +) in the following line

Methodology

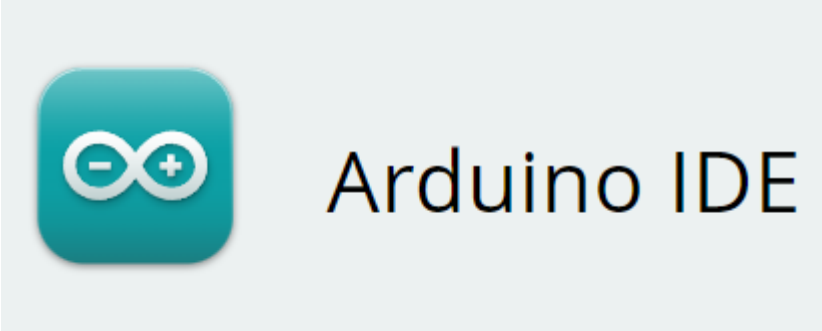
- After the detail literature survey through the books, periodical, journal, magazine, websites. The idea of the robot is well defined.
- The logic is derived for the intelligence of the robot. It is programmed and burn it to the Arduino by using the software Arduino IDE.
- The accuracy and viability of the program and electronic components is tested.
- After the successful simulation result it is implemented in the hardware.
- After the finishing the programming, electrical and electronics part, the stable, reliable and flexible mechanical design and fabrication is completed.
- Finally, system is tested and encountered error is omitted.

Software Required

For the simulation of the circuit, Arduino software is used. For coding and uploading the sketch, the Arduino is used.

Programming Approach

Coding IDE:



How its working

The concept of working of a line follower robot is based on the phenomenon of light. We know that white color reflects almost all of the light that falls on it, whereas black color absorbs most of the light. In case of a line follower robot we use IR transmitters and receivers also called photodiodes. They are used for sending and receiving light. IR transmits infrared lights. When infrared rays fall on white surface, it's reflected back and catches by photodiodes which generates some voltage changes. When IR light falls on a black surface, light is absorbing by the black surface and no rays are reflected back, thus photo diode does not receive any light or rays.

Here in this Arduino line follower robot when sensor senses white surface then Arduino gets 1, i.e., HIGH as input and when senses black line Arduino gets 0, i.e., LOW as input.

L293D Motor Driver

The H-Bridge Motor Drivers L293D is used to drive DC motors as they require much more current that the Arduino can provide.

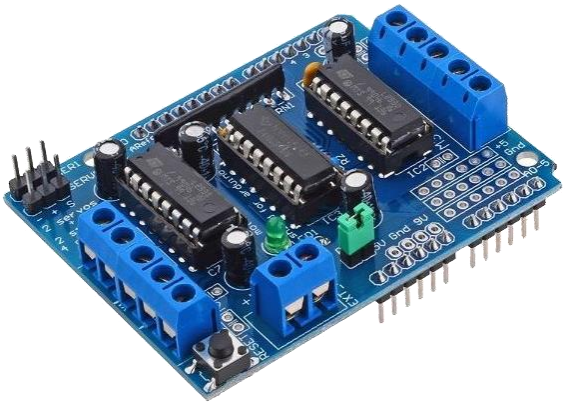
The L293D motor controller follows the H-bridge configuration, which is handy when controlling the direction of rotation of a DC motor. The other benefit of using an H-bridge is that we can provide a separate power supply to the motors. This is very significant especially when using an Arduino board where the 5V power source is simply not enough for two DC motors.

We have Motor A and Motor B terminals. These connect to the microcontroller. Motor A connects to terminals 1 and 2 while Motor B connects to terminals 3 and 4.

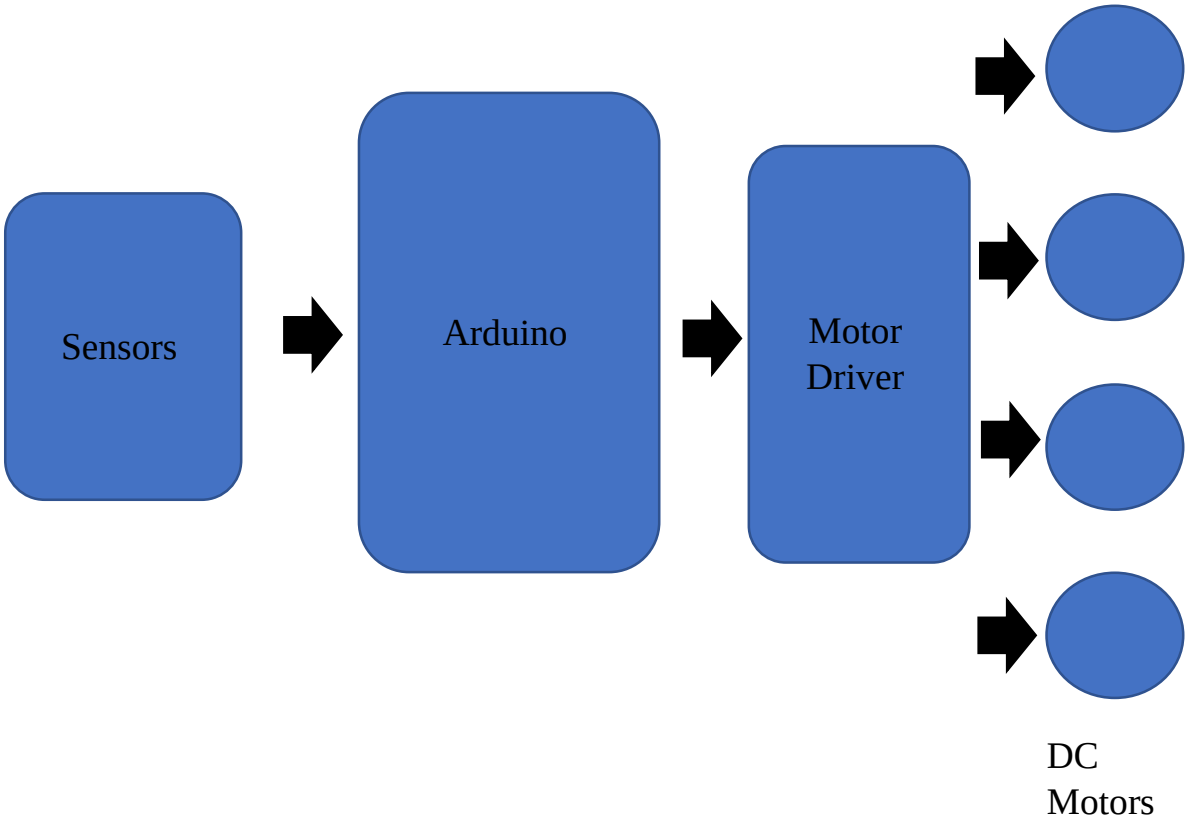
If we want the left motor to rotate in one direction, we apply a high pulse to IN1 and a low pulse to IN2. To reverse the direction, reverse the pulses to IN1 and IN2. The same applies to the right motor.

If we want the left motor to rotate in one direction, we apply a high pulse to IN1 and a low pulse to IN2. To reverse the direction, reverse the pulses to IN1 and IN2. The same applies to the right motor.

Speed control is also possible with the L293D motor driver. All we need is feed PWM (Pulse Width Modulation) signals to the motor enable pins. The speed of the motor will vary according to the width of the pulses. The wider the pulses, the faster the motor rotates.



How the line follower working



Procedure:

Step 1: Connect the circuit as following the schematic.
Step 2: Use the Arduino IDE to write own code.
Step 3: Upload the code to the Arduino and connect it to the batteries to run the Arduino.
Step 4: Finally test it on a black path.

Procedure of sensing the line:

The line follower robot is a kind of a design which is similar as a light follower robot. Here, besides sensing the light, the sensor is used for detection of a line. Therefore, by individualizing the color of line and its enclosing, any light sensitive sensor could be used for navigation of the robot to follow its designated track. The design of the robot was made like; it had one pair of Infrared ray sensor fitted underneath the robot. So Infrared ray sensor will first be sending a wavelength for detecting black line and then other infrared ray sensor will be receiving the information and take decision for following a black line on white surface. With the supply from an 9V DC power adapter the whole sensor and the motor driver IC and the motors and Arduino are powered. Making the setup less prone to power failure The outputs of the sensor circuits are connected as in the analog inputs of the Arduino board.

Arduino Mega continuously monitors the data from both the sensors and turns the robot as per the line detected by them.

Problems facing

- At the first time I cannot understand what are the pins should I work with.
- Its take time understanding the Arduino environment.
- At the beginning I have no idea to which sensors are to use.

Coding Issues

- We need to build a robot that works well. The coding part is not an easy task for us.

- We have to calibrate the code to work properly with the hardware we have built. So the first step in programming is to make a demo track. This helped us test our code then and there.

- Here is the track with (X, +, Y) that I built for our line follower. This is a challenge for us

Solutions

- ✓ We referred some of YouTube videos & tutorials to find out knowledge regarding robotics.
- ✓ Get some help from senior bachelors.
- ✓ Most of time discussing with lecturer.

Program Code

```
//ARDUINO LINE FOLLOWING CAR//
//2022-Robotics Application Development CW//


//including the libraries
#include <AFMotor.h>

//defining pins and variables

#define left A1
#define center A2
#define right A3
int HS=120;
int LS=150;
int ST=0;
//defining motors
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

void setup() {
 //declaring pin types

  pinMode(left,INPUT);
  pinMode(right,INPUT);
  pinMode(center,INPUT);
  //begin serial communication
  Serial.begin(9600);

}

void loop(){
  //printing values of the sensors to the serial monitor
  Serial.println(digitalRead(left));
  Serial.println(digitalRead(center));
  Serial.println(digitalRead(right));

  //line detected by both
  if(digitalRead(left)==0 && digitalRead(center)==1 && digitalRead(right)==0 ){
    //Forward
    motor1.run(FORWARD);
    motor1.setSpeed(HS);
```

```
  motor2.run(FORWARD);
  motor2.setSpeed(HS);
  motor3.run(FORWARD);
  motor3.setSpeed(HS);
  motor4.run(FORWARD);
  motor4.setSpeed(HS);
 }
 //line detected by left sensor
 else if(digitalRead(left)==1 && digitalRead(center)==0 && digitalRead(right)==0 ){
  //turn left
  motor1.run(BACKWARD);
  motor1.setSpeed(LS);
  motor2.run(BACKWARD);
  motor2.setSpeed(LS);
  motor3.run(FORWARD);
  motor3.setSpeed(HS);
  motor4.run(FORWARD);
  motor4.setSpeed(HS);

 }
 //line detected by right sensor
 else if(digitalRead(left)==0 && digitalRead(center)==0 && digitalRead(right)==1){
  //turn right
  motor1.run(FORWARD);
  motor1.setSpeed(HS);
  motor2.run(FORWARD);
  motor2.setSpeed(HS);
  motor3.run(BACKWARD);
  motor3.setSpeed(LS);
  motor4.run(BACKWARD);
  motor4.setSpeed(LS);

 }

 else if(digitalRead(left)==1 && digitalRead(center)==1 && digitalRead(right)==1){
  //Stop
  motor1.run(FORWARD);
  motor1.setSpeed(ST);
  motor2.run(FORWARD);
  motor2.setSpeed(ST);
  motor3.run(FORWARD);
  motor3.setSpeed(ST);
  motor4.run(FORWARD);
  motor4.setSpeed(ST);

 }
}
```