**MINOR PROJECT-I REPORT**

**On**

**Title of Project**

**Automated Braking System**

**Submitted by:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Roll No** | **Branch** | **SAP ID** |
| Mayank Rajput | R200219026 | BCA- IOT | 500076142 |

**Under the guidance of**

Dr.Inder Singh

School Of Computer Science



**SCHOOL OF COMPUTER SCIENCE**

UNIVERSITY OF PETROLEUM & ENERGY STUDIES

Bidholi Campus, Energy Acres, Dehradun – 248007.

**September - 2021**

 **School of Computer Science**

**University of Petroleum & Energy Studies, Dehradun**

**Project Proposal Approval Form (2019-20)**

I

**Minor**

**PROJECT TITLE:**AUTOMATED BREAKING SYSTEM

**ABSTRACT**

Automated Breaking System was designed, constructed and programmed which may be potentially used for educational and research purposes. The developed robot will move in a particular direction once the infrared (IR) and the PIR passive infrared (PIR) sensors sense a signal while avoiding the obstacles in its path. The robot can also perform desired tasks in unstructured environments without continuous human guidance. The hardware was integrated in one application board as embedded system design. The software was developed using C++ and compiled by Arduino IDE 1.6.5. The main objective of this project is to provide simple guidelines to the polytechnic students and beginners who are interested in this type of research. It is hoped that this robot could benefit students who wish to carry out research on IR and PIR sensors..

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Title** | **Page Number** |
| 1. | Abstract | (ii) |
| 2. | Introduction | 1 |
| 3. | Problem Statement | 2 |
| 4. | Literature Review | 3 |
| 5. | Objectives | 4 |
| 6. | Design and Methodology | 5-9 |
| 7. | Implementation  -Results/ Output Screen | 9-10  18-20 |
| 8. | Important Theory | 10-11 |
| 9.  10. | Conclusion and Future Scope  Refrences | 21  22 |

**INTRODUCTION**

Many robots for automation and navigation have been developed in recent years like wall-following, edge-following, human following and obstacle avoiding robots. The Automated Breaking robot will evade obstacles it encounters in its path towards its operational goal. Due to the reliability, accessibility and cost effectiveness of using mobile robot in industry and technical applications, the Automated Breaking robots are very important in factory floor. On the other hand, Unmanned Aerial Vehicles (UAVs) are playing a vital role in defence as well as civilian applications [14]. The military applications include reconnaissance, surveillance, battle damage assessment and communications. Meanwhile, civilian applications include disaster management, remote sensing, traffic monitoring, etc. Many of the UAVs applications need the capability to navigate in urban environment or unknown terrains that have many obstacles of different types and sizes. Basic requirement of autonomous UAVs is to detect obstacles in its path and avoid them. This proposed an example of the Automated Breaking robot algorithm and design of the robot base using IR and PIR sensors. The developed robot can be used as a platform for several applications in educational, research or industrial.

**PROBLEM STATEMENT**

Now a days sensors are used in some of the cars (e.g) Tesla . But not in every car some of the car face problems like :-

* Some times breaks do not work .
* When some one suddenly appears in front of the car it is unable to stop the car and accident may happen .
* Breaks can not be applied immediately .
* The major Drawback of previous breaking system was that you have to apply them immediately as you look on road and move your toes to apply the break it can take time.
* And the advantage of using this system is breaks can be applied with the help of signals sent by sensors before the obstacle by some distance and accidents can reduce.

**LITERATURE REVIEW**

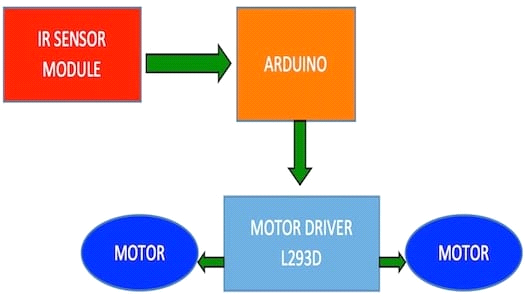
Robots need miscellaneous of sensors to obtain information about the world around them. Sensors will help detect position, velocity, acceleration and range for the object in the robot's workspace. There is a variety of sensors used to detect the range of an object. One of the most common range finders is the ultrasonic transducer. Vision systems are also used to greatly improve the robot’s versatility, speed and accuracy for its complex and difficult task. Varun et al. developed obstacle avoidance equipped with pan-tilt mounted vision system. In this case, the robot uses histograms of images obtained from a monocular and monochrome camera to detect and avoid obstacles in a dynamic environment. Kim et al. presented the detection of moving obstacles (particularly walking humans) using single camera attached to a mobile robot. Detection of object that moves near the robot is searched by block-based motion estimation. Shoval et al. described the use of mobile robot obstacle avoidance system as a guidance device for blind and visually impaired people. Electronic signals are sent to a mobile robot’s motor controllers and auditory signals can guide the blind traveller around the obstacles. Kumar proposed an alternative design for cost effective and simplified version of the obstacle avoidance robot using three ultrasonic sensors. Borenstein et al. built a mobile robot system, capable of performing various tasks for the physically disabled people. The developed robot uses ultrasonic range finder for detection and mapping to avoid collision with the unexpected obstacles. Sathiyanaranayan et al. developed self-controlled robots for military purposes. It uses GPS and magnetic compass, and adjusts strategies based on the surroundings using path planning and obstacle detection algorithms. Bhanu et al. built a system for obstacle detection during rotorcraft low altitude flight. The requirements of an obstacle detection system for rotorcraft in low altitude flight based on various rotorcraft motion constraints is analysed in details. Huballi et al. discussed on Smart Distance Measurement using IR sensor. From the research, they found that a major drawback of IR based sensors is their capability of detecting only in short distance.

**OBJECTIVES**

* The  main objective of this Project is to prevent accidents and better control to the user .
* The project is built at administrator end and thus only the administrator is guaranteed the access.
* The purpose of the project is to provide a safe driving to the user and better control over vehicle so that they can avoid accidents .

**DESIGN AND METHODOLOGY**

* REQUIRMENTS
* ANALYSIS
* DESIGN
* CONSTRUCTION
* TESTING
* DEPLOYMENT & MAINTENANCE



**Analysis:-**

For this project we have searched some research papers to know that if there is some drawbacks are there in this project or not. We have used this research to modify this project and make it better for users .

**System Requirements:-**

**Hardware Requirments:-**

* Arduino Uno Board
* L239D Motor Driver
* Jumper Wires
* IR Sensors
* HDMI cable
* 9v battery

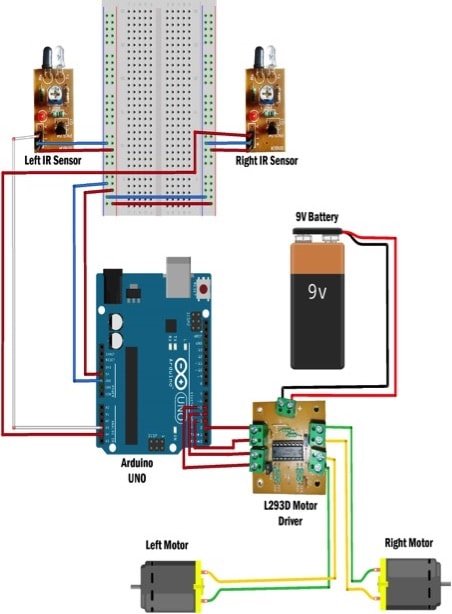
**Software Requirements:-**

* Arduino Platform 1.8.6
* Operating System : Windows 10

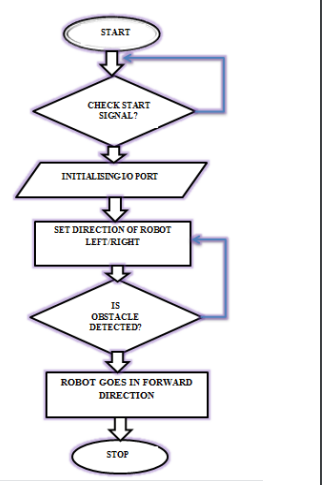
**Design:-**

In this phase we have designed a circuit diagram, use-case diagram, and a pert chart .

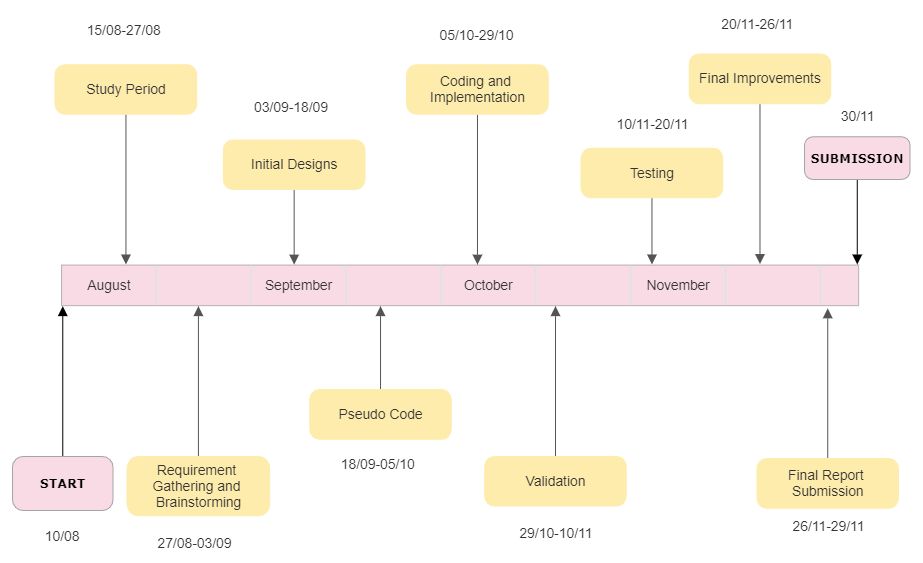
**Circuit diagram:-**



**Use case Diagram:-**

****

**Pert Chart:-**

****

**IMPLEMENTATION:-**

After this step we implemented the circuit diagram and connected the Arduino Board to the Laptop through HDMI Cable and then connected the sensors with the board and then connected the board to the l239d driver and connect the driver with a 9v battery to supply the power. We have to connect the motors to the l239d driver. We have to attach the wheels with the motors.

**Testing:-**

After the implementation we have to test this system by putting obstacle in its way and chek whether the sensors are working properly or not.

**IMPLEMENTATION**

* When the system is well connected and set up you need to supply the power to the system.
* Once the power is given the light on the motor driver will glow .
* If senors detect object in front of it the light will glow .
* Once they catch the signal then they will send the signals to the arduino board .
* Once the signal reaches the arduino board it will send the command according to the code .
* The l239d driver will accept the command and give command to the motors to work accordingly.

**Important Theory**

Check whether the connected IR sensors are working with the Arduino. To do this, connect the IR module to the Arduino's analog pins and check the values received in the serial monitor. Then find the corresponding values when an object is in front of the IR sensor.The two terminals on the motors are connected to the four output terminals on the board. The motors then, based on the command from the Arduino are powered by the 9V battery. The logic for controlling the motors from the Arduino is as given below:

Here, HIGH means 5V signal or digital 1 and LOW is 0V signal or digital 0. Eg: digitalWrite(5, HIGH), this command sends a HIGH signal (digital 1) to pin 5 on the Arduino. Thus, each motor’s direction can be controlled by writing HIGH/LOW signals through two digital pins on the Arduino.

**DEVICE DETAILS**

**L293 MOTOR DRIVER:** The driver has 2 inputs for power, 4 points for motor control inputs, and 4 points for motor control outputs. That is a set of 2 inputs and 2 outputs for each motor. To control the motors, connect the four control points to the Arduino and the 4 output points to the motors. The 9V battery can be connected to the driver using the power input pins. The input and output power points are indicated on the board.  
  
**IR module**: IR stands for Infrared, which is a wavelength of light not visible to the human eye (but can be seen with our smartphone cameras!). These modules consist of a pair of receiver and transmitter IR LEDs. When an object gets in front of the IR sensor, the surface of the object reflects a part of the IR light back to the receiver, the receiver then outputs a LOW signal notifying that an object is in front of the sensor.This Arduino robot uses two IR sensor modules which can detect objects within a range of 5-6cm. This sensor outputs a digital LOW (0V) signal when there is an object within its range and outputs a digital HIGH (5V) signal otherwise.

**ARDUINO BOARD:-**

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

Technical specifications

* Operating Voltage: 5 Volts
* Input Voltage: 7 to 20 Volts
* Digital I/O Pins: 14 (of which 6 can provide PWM output)
* PWM Pins: 6 (Pin # 3, 5, 6, 9, 10 and 11)[[9]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-9)
* UART: 1
* I2C: 1
* SPI: 1
* Analog Input Pins: 6
* DC Current per I/O Pin: 20 mA
* DC Current for 3.3V Pin: 50 mA
* Clock Speed: 16 MHz
* Length: 68.6 mm
* Width: 53.4 mm
* Weight: 25 g
* ICSP Header: Yes
* Power Sources: DC Power Jack & USB Port

**General pin functions**

* **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.
* **VIN**: The input voltage to the Arduino/Genuino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
* **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND**: Ground pins.
* **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.
* **Reset**: Typically used to add a reset button to shields that block the one on the board.[[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7)

**Special pin functions**

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pinMode(), digitalWrite(), and digitalRead() functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of the range using the AREF pin and the analogReference() function.[[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7)

In addition, some pins have specialized functions:

* **Serial** : pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
* **External interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* Pulse-width modulation: pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.
* Serial Peripheral Interface: pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
* **TWI** (two-wire interface) : pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
* **AREF** (analog reference): Reference voltage for the analog inputs.[[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7)

**MOTORS:-**

There are two motors used in this system. Wheels are connected to them and they are used to move the system.

**IR Sensors:-**

These sensors are IR(Infra red )sensors that are used to catch the signal and transmit them to arduino board.

**C++ Code**

#include <Arduino.h>

//Declare all the variables being used here:

int IRA;

int IRB;

int IRC;

int LFlag;

int CFlag;

int RFlag;

//Declare all the functions being used here:

void MR();

void ML();

void MF();

void RE();

void setup()

{

// put your setup code here, to run once:

pinMode(6, OUTPUT);

pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

pinMode(11, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

digitalWrite(7, HIGH);

digitalWrite(8, HIGH);

Serial.begin(115200);

}

void loop()

{

// put your main code here, to run repeatedly:

IRA = analogRead(A0);

IRB = analogRead(A1);

IRC = analogRead(A2);

if (IRB < 200)

{

LFlag = 1;

}

else

{

LFlag = 0;

}

if (IRC < 200)

{

RFlag = 1;

}

else

{

RFlag = 0;

}

if (IRA < 200)

{

CFlag = 1;

}

else

{

CFlag = 0;

}

if ((LFlag == 1) || (RFlag == 1) || (CFlag == 1))

{

if (CFlag == 1)

{

if (LFlag == 1)

{

MR();

}

else if (RFlag == 1)

ML();

else

ML();

}

else if (LFlag == 1)

{

MR();

}

else if (RFlag == 1)

{

ML();

}

else

{

ML();

}

}

else

{

MF();

}

}

//Function definitions go here:

void RE()

{

Serial.println("Backing up");

digitalWrite(9, LOW);

analogWrite(6, 75);

analogWrite(11, 75);

digitalWrite(10, LOW);

delay(1000);

}

void MR()

{

Serial.println("Right");

RE();

digitalWrite(6, LOW);

digitalWrite(9, LOW);

digitalWrite(11, LOW);

analogWrite(10, 350);

delay(300);

}

void ML()

{

Serial.println("Left");

RE();

analogWrite(9, 350);

digitalWrite(6, LOW);

digitalWrite(11, LOW);

digitalWrite(10, LOW);

delay(300);

}

void MF()

{

Serial.println("Forward");

analogWrite(9, 75);

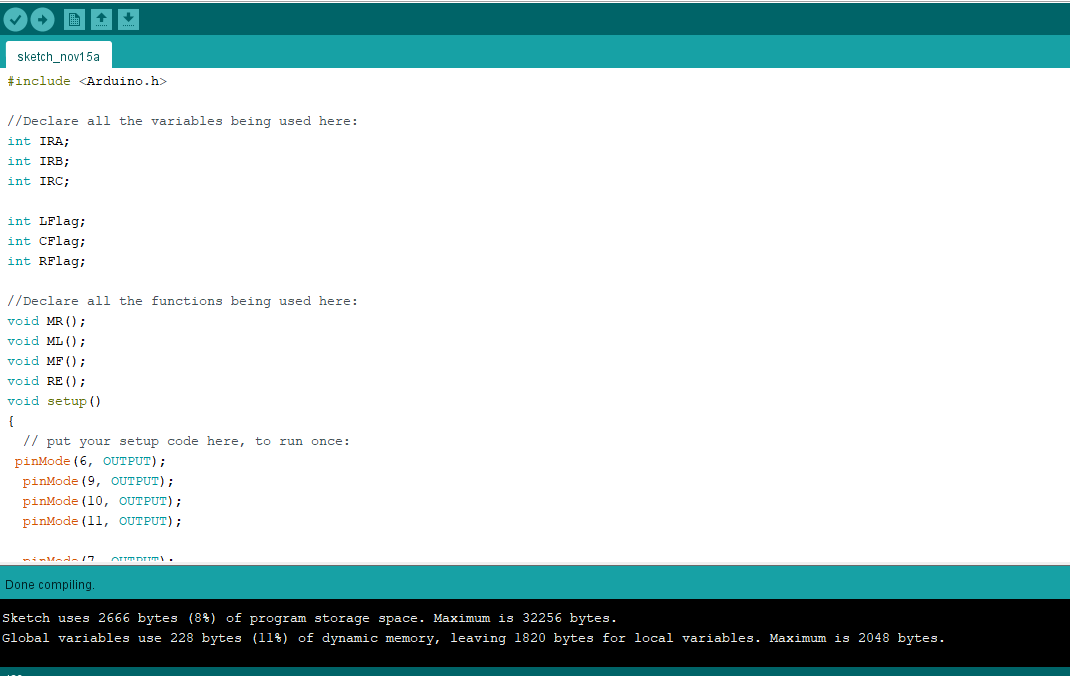
digitalWrite(6, LOW);

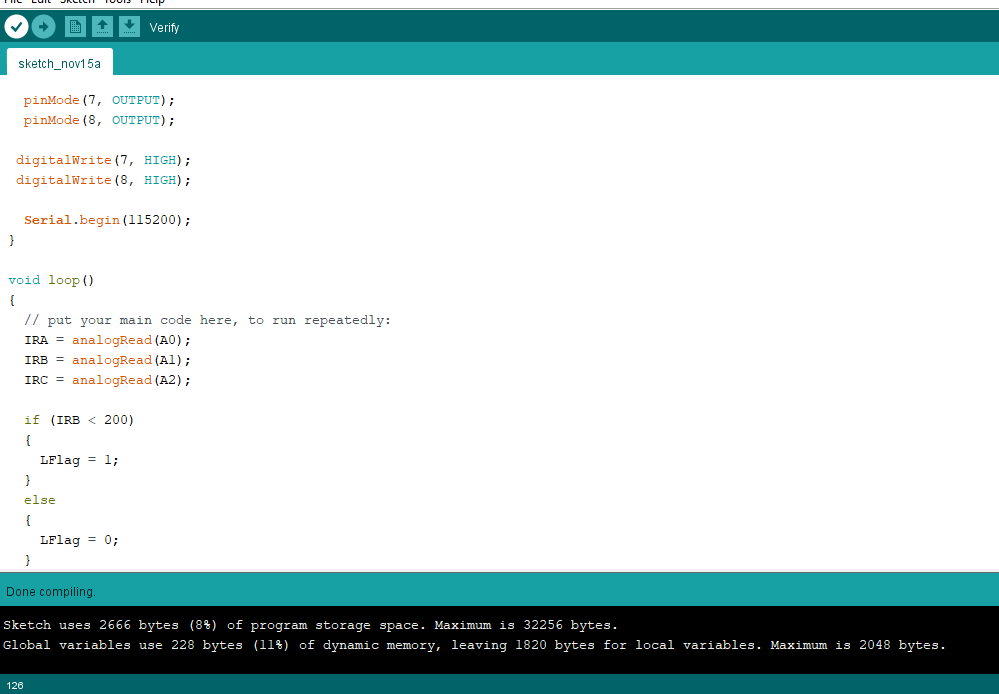
digitalWrite(11, LOW);

analogWrite(10, 75);

**}**

**RESULT/OBSERVATION/OUTPUT SCREEN**



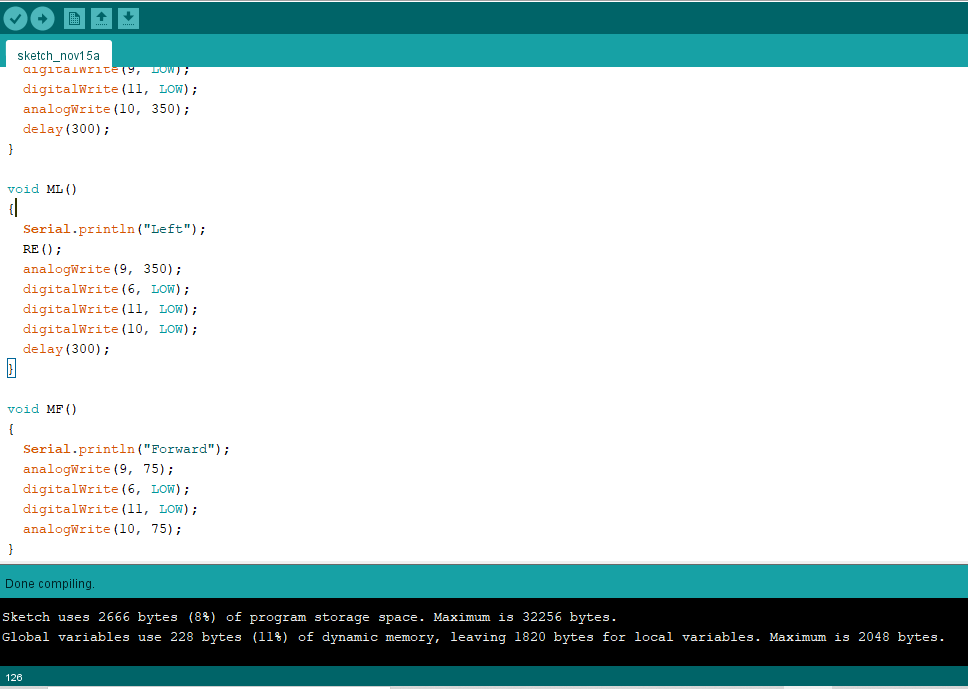


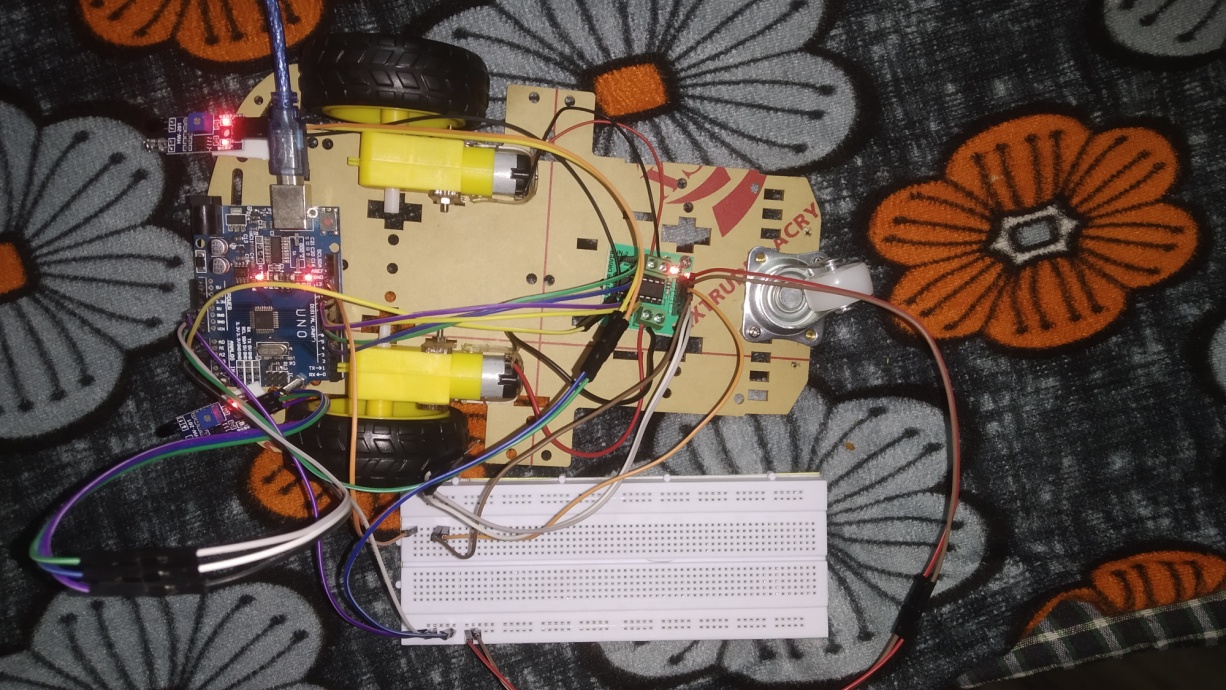


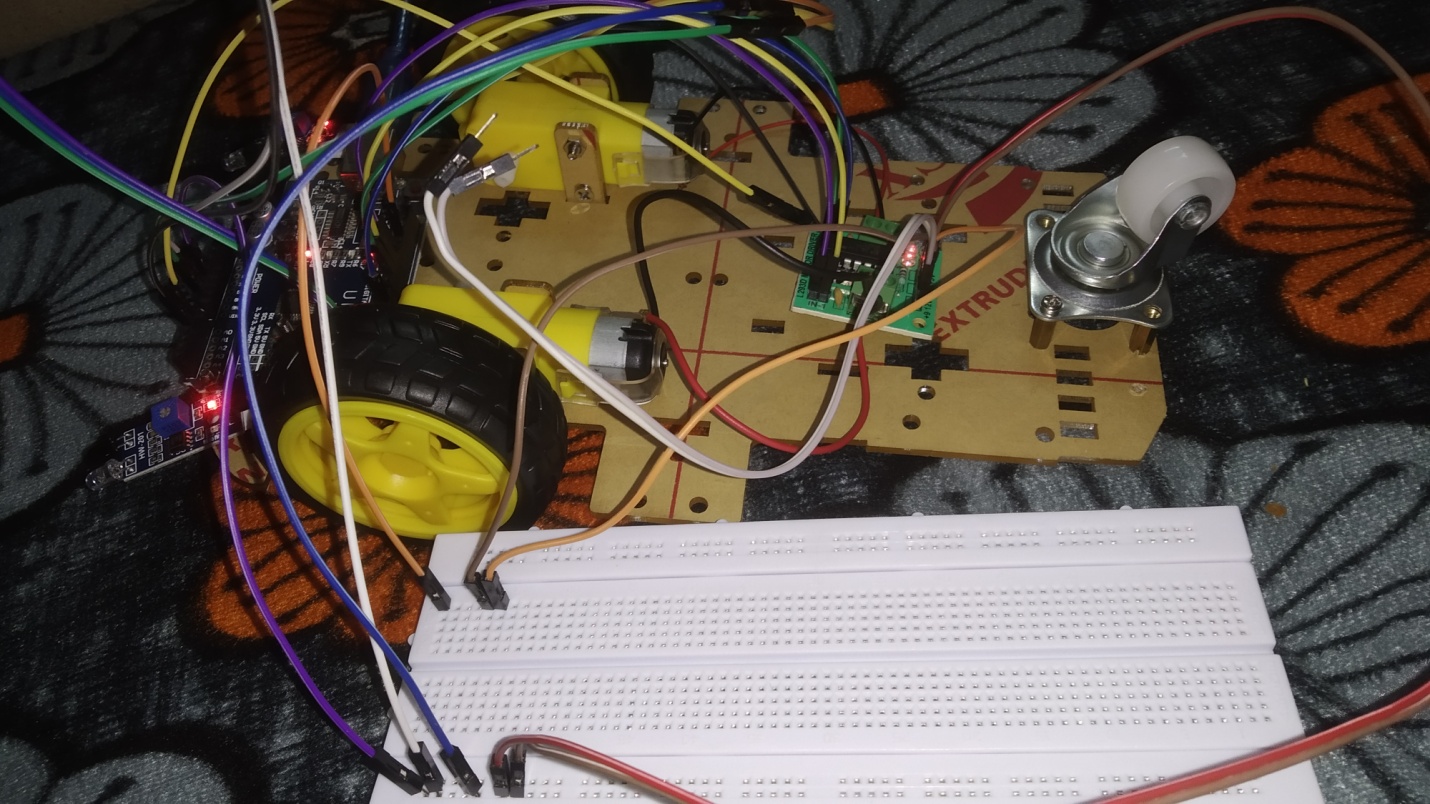












This is the how our project looks like. We can see the red lights in the devices that are indicating that it is ready to use and power is supplied. We have used breadboard to connect multiple wires to a single port avoiding the risk of short ciruit.

**CONCLUSION AND FUTURE SCOPE**

This paper is all about Obstacle Avoidance Robot using Arduino which avoids obstacles which it encounters. In future this project can be enhance by connecting Bluetooth module and a camera so that the user can see the detected obstacle on his screen by sitting at just one place.

This project is design to build an obstacle avoidance robotic vehicle using infrared sensors. A micro- controller (AT mega 8) is used to achieve the desired operation.

**REFERENCES**

Ismail, R., Z. Omar, and S. Suaibun. "Obstacle-avoiding robot with IR and PIR motion sensors." *IOP Conference Series: Materials Science and Engineering*. Vol. 152. No. 1. IOP Publishing, 2016.[1]

Tabassum, Faiza, et al. "Obstacle avoiding robot." *Global Journal of Research In Engineering* (2017).[2]

Efaz, Erteza Tawsif, et al. "Design of an indicative featured and speed controlled obstacle avoiding robot." *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*. IEEE, 2019.[3]

Molina, Miguel, et al. "Design and Construction of an Obstacle Avoiding Robot Based on Arduino Platform and Programming Too." *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2018.[4]

MISHRA, SHWETA, et al. "Real-World Simulation Through Obstacle Avoiding Robot." *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2021.

**Draft verified by**

**Project Guide: HOD**