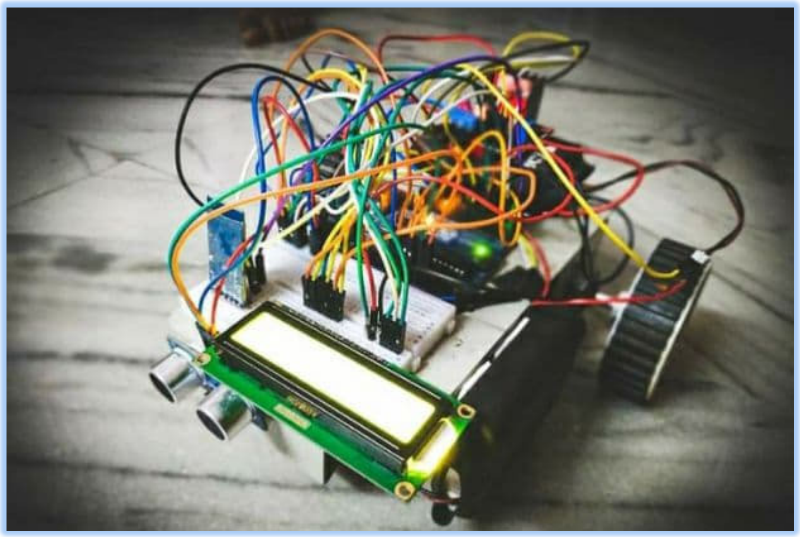
OBSTACLE AVOIDER BOT USING ARUINO UNO

Kalyani Government Engineering College

Kalyani, Nadia, West Bengal

July 2021



Submitted By –

Rudraprasad Debnath

Electronics and communication Engineering 3rd Year

Roll -- 10200318038

CERTIFICATE

This is to certify that the project report entitled **Obstacle Avoider Bot Using Arduino UNO** submitted by **RUDRAPRASAD DEBNATH** to the **Kalyani Government Engineering College** in partial fulfillment for the award of the degree of **B. Tech** in **Electronics and Communication Engineering** is a bona fide record of project work carried out by him under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

Rudraprasad Debnath | [illegible]

Student's Signature | Signature of Supervisor

INTRODUCTION

Bluetooth is a short-range wireless technology standard that is used for exchanging data between fixed and mobile devices over short distances. We can even control devices by giving commands through Bluetooth communication.

The aim of this project is to build a Bot using **Arduino UNO** and **HC-05 Bluetooth Module** that can be controlled through an android application. Now what if our bot is stuck due to any obstacle? Oh yes, it must learn to avoid that by giving the controller a warning about the obstacle. Here we have designed a system using ultrasonic sensor to measure the distance of the obstacle from the bot. If the distance is quite small (< 40 cm), then our bot displays a warning on a 16x2 LCD display screen and also displays the distance.

Aim: To build a manually controlled obstacle avoider bot that displays the distance of an obstacle.

PROJECT PLANNING

Hardware Used:

1) PC – Windows 10 Intel ® Pentium 4

2) RAM – 4 GB 256 GB HDD

Software Used:

1) Arduino IDE

Components Used:

1) Arduino UNO

2) Ultrasonic Sensor

3) 16x2 LCD Display

4) L298N Motor Driver

5) HC-05 Bluetooth Module

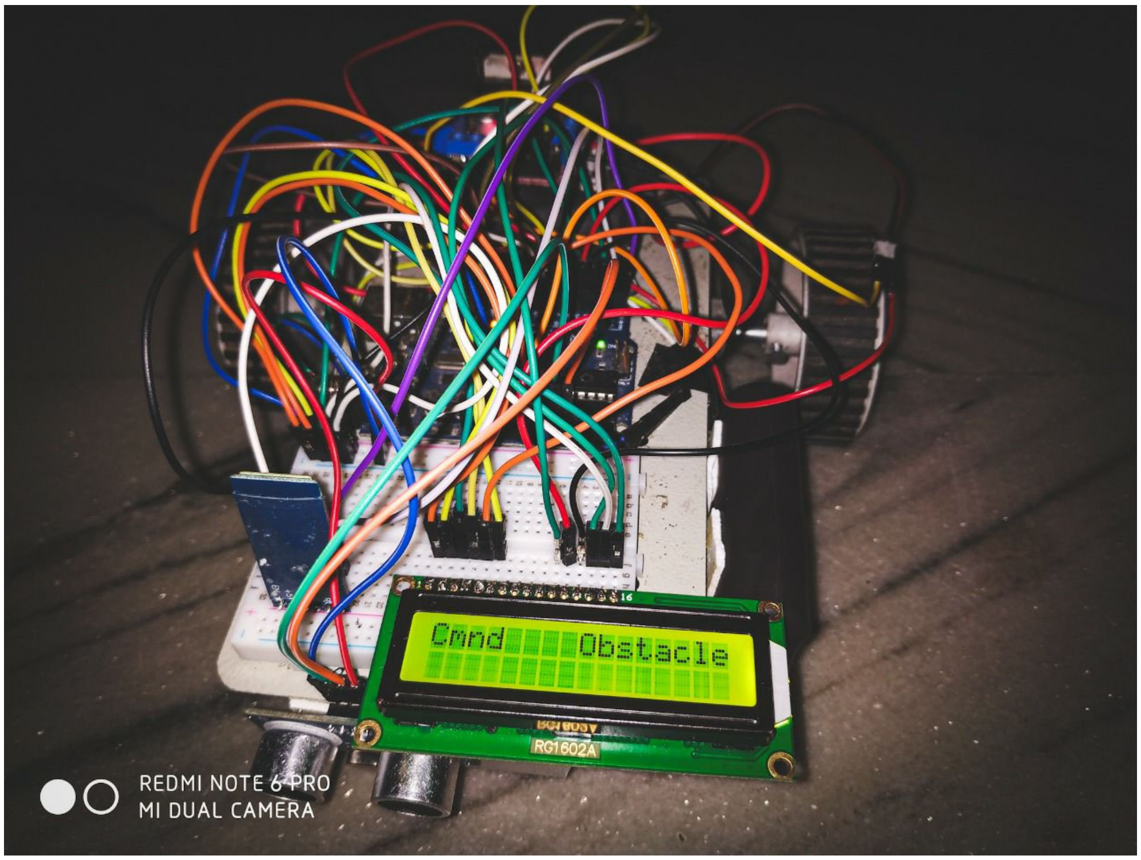
6) Jumper Wires

7) Metal Chassis

8) Breadboard (Small)

IMPLEMENTATION OF THE PROJECT

i) The bot:



ii) Description of Components used in the project:

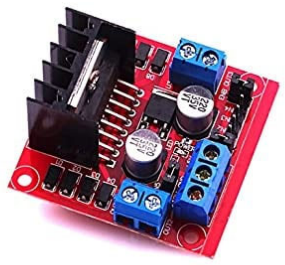
a) Arduino UNO: This is the latest revision of the basic Arduino USB board. It connects to the computer with a standard USB cable and contains everything else you need to program and use the board. It can be extended with a variety of shields: custom daughter-boards with specific features.

It is similar to the Duemilanove, but has a different
USB-to-serial chip the ATMega8U2, and newly
designed labeling to make inputs and outputs easier to identify.

b) Ultrasonic Sensor: An **ultrasonic sensor** is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity. High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

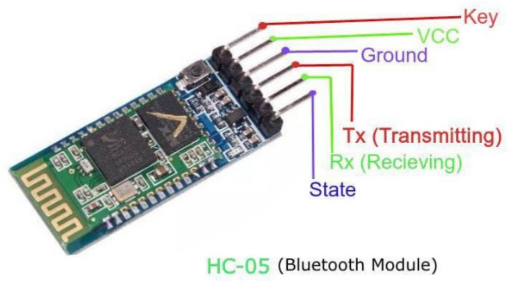
c) 16x2 LCD Display: LCD is defined as the **diode** that uses **small cells** and the **ionized gases** for the **production of images**. The LCD works on the **modulating property of light.** The light modulation is t**he technique of sending and receiving the signal** through the **light**. The **liquid crystal consumes** a small amount of energy because they are the **reflector and the transmitter of light.** It is normally used for **seven segmental display.**

d) L298N Motor Driver: This **L298N Motor Driver Module** is a high-power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. **L298N Module** can control up to 4 DC motors, or 2 DC motors with directional and speed control.



e) HC-05 Bluetooth Module: HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.



It has 6 pins,

1. **Key/EN:** It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.

HC-05 module has two modes,

1. **Data mode:** Exchange of data between devices.
2. **Command mode:** It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.

2. **VCC:** Connect 5 V or 3.3 V to this Pin.
3. **GND:** Ground Pin of module.
4. **TXD:** Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)
5. **RXD:** Receive data serially (received data will be transmitted wirelessly by Bluetooth module).
6. **State:** It tells whether module is connected or not.

HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner.

ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

PROCEDURE

Step 1: At first we have to interface the L298N motor driver with Arduino UNO. We use the PWM pins of the Arduino to control the speed of the motor according to the input.

Step 1: Then the 16x2 LCD display is interfaced to the Arduino using appropriate pin connections.

Step 3: Next, the BT module is connected to the Arduino through the Rx-Tx pins.

Step 4: The ultrasonic sensor is also connected to the Arduino to measure the distance of the obstacle.

Step 5: It should be made sure that the Rx-Tx pins of the BT module is kept open while uploading the code to the Arduino.

Step 6: The **Arduino IDE** is opened in PC and program for the operation is written in embedded C language.

Step 7: Now the bot is ready to run and to be controlled through an android app from any smartphone.

Code used for the operation:

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2);


#define trig A1
#define echo A0
#define in1 10
#define in2 9
#define in3 8
```

```
#define in4 7

int distance=0;
int duration=0;
// SoftwareSerial  BT(10,11);
 //TX,  RX respeticvely
 String readvoice;

void printFwd(){
  //lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Frwd");
  if(distance<40)
      {
        lcd.setCursor(8,0);
        lcd.print("Obstacle");
        lcd.setCursor(8,1);
        lcd.print(distance);
        lcd.setCursor(14,1);
        lcd.print("cm");
        lcd.setCursor(0,1);
        lcd.print("Warning");
      }
}


void Initialize(){
  //lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Connected");
  delay(1000);
  lcd.clear();
  if(distance<40)
      {
        lcd.setCursor(8,0);
        lcd.print("Obstacle");
        lcd.setCursor(8,1);
        lcd.print(distance);
        lcd.setCursor(14,1);
        lcd.print("cm");
          lcd.setCursor(0,1);
        lcd.print("Warning");
      }
}

/*void dis(){
  //lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Disconnected");
```

```
}*/

void printBwd(){
  //lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Down");
  if(distance<40)
      {
        lcd.setCursor(8,0);
        lcd.print("Obstacle");
        lcd.setCursor(8,1);
        lcd.print(distance);
        lcd.setCursor(14,1);
        lcd.print("cm");
          lcd.setCursor(0,1);
        lcd.print("Warning");
      }
}

void printRight(){
  //lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Rght");
  if(distance<40)
      {
        lcd.setCursor(8,0);
        lcd.print("Obstacle");
        lcd.setCursor(8,1);
        lcd.print(distance);
        lcd.setCursor(14,1);
        lcd.print("cm");
          lcd.setCursor(0,1);
        lcd.print("Warning");
      }
}

void printLeft(){
  //lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Left");
  if(distance<40)
      {
        lcd.setCursor(8,0);
        lcd.print("Obstacle");
        lcd.setCursor(8,1);
        lcd.print(distance);
        lcd.setCursor(14,1);
        lcd.print("cm");
```

```
          lcd.setCursor(0,1);
        lcd.print("Warning");
      }
}

void printStop(){
  //lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Stop");
  if(distance<40)
      {
        lcd.setCursor(8,0);
        lcd.print("Obstacle");
        lcd.setCursor(8,1);
        lcd.print(distance);
        lcd.setCursor(14,1);
        lcd.print("cm");
          lcd.setCursor(0,1);
        lcd.print("Warning");
      }
}

 void setup() {
//  BT.begin(9600);
  Serial.begin(9600);
     pinMode(in1, OUTPUT);
     pinMode(in2, OUTPUT);
     pinMode(in3, OUTPUT);
     pinMode(in4, OUTPUT);
     pinMode(trig, OUTPUT);
     pinMode(echo, INPUT);
     lcd.begin(16,2);
     lcd.print("Hello");
     analogWrite(6,125);
      lcd.clear();
      lcd.setCursor(2,0);
      //lcd.print("Connected!!!");
      delay(800);
      lcd.setCursor(1,1);
      lcd.print("Ready To Go!!!");
      delay(1000);
      lcd.clear();
      lcd.setCursor(8,0);
      lcd.print("Obstacle");
      lcd.setCursor(0,0);
      lcd.print("Cmnd");
 }
  //----------------------------------------------//
  void loop() {
//    while  (Serial.available()){
//  //check if there is an available byte to read
//    delay(10); //Delay added to make thing stable
//    char c = Serial.read();
//  //Conduct a serial read
//    readvoice += c; //build the string- "forward", "reverse","left" and "right"
//    }
    analogWrite(trig, 0);
    delayMicroseconds(2);
    analogWrite(trig, 200);
    delayMicroseconds(10);
    analogWrite(trig, 0);

    duration=pulseIn(echo,255);
    distance=duration*0.017;



    if (Serial.available()) {
      analogWrite(trig, 0);
    delayMicroseconds(2);
    analogWrite(trig, 200);
    delayMicroseconds(10);
    analogWrite(trig, 0);

    duration=pulseIn(echo,255);
    distance=duration*0.017;
      //Initialize();
      //delay(1000);
     // lcd.setCursor(8,0);
     // lcd.print("Obstacle");
      lcd.setCursor(8,1);
      lcd.print(distance);
      lcd.setCursor(14,1);
      lcd.print("cm");*/

     if(distance<40)
      {
        lcd.setCursor(8,1);
        lcd.print(distance);
        lcd.setCursor(14,1);
        lcd.print("cm");
        lcd.setCursor(0,1);
        lcd.print("Warning");
        delay(150);
        lcd.print("");
```

```
       delay(100);
       lcd.print("Warning");
     }

     lcd.clear();
       char c = Serial.read();
       Serial.println(c);

    if(c == 'f')
    {
     Serial.println("UP");
     //left
      digitalWrite(in1, LOW); //bck
      digitalWrite(in2, HIGH); //fwd
     //right
      digitalWrite(in3, HIGH); //fwd
      digitalWrite(in4, LOW); //bck
      printFwd();
      delay(100);

    }

    else if(c == 'b'){
       digitalWrite(in1, HIGH);
       digitalWrite(in2, LOW);
       digitalWrite(in3, LOW);
       digitalWrite(in4,HIGH);
       printBwd();
       delay(100);
    }

 else if (c == 'r')
 {
   digitalWrite (in1,LOW);
   digitalWrite (in2,HIGH);
   digitalWrite (in3,LOW);
   digitalWrite (in4,LOW);
   printRight();
  delay (100);


 }

else if ( c == 'l')
{
  digitalWrite (in1, LOW);
  digitalWrite (in2, LOW);
  digitalWrite (in3, HIGH);
  digitalWrite (in4, LOW);
  printLeft();
  delay (100);

}

else if (c == 's')
{
  digitalWrite (in1, LOW);
  digitalWrite (in2, LOW);
  digitalWrite (in3, LOW);
  digitalWrite (in4, LOW);
  printStop();
  delay (100);
}
}
/*else{
 dis();
}*/ //Reset the variable
 }
```

Here is a link for the bot to visualize how it works:

https://www.youtube.com/watch?v=2WsRTiPn-G4

References

1) www.google.com

2) www.wikipedia.com

3) www.electronicshub.com

4) www.geeksforgeeks.org