## horizontal line



Obstacle Avoiding Bluetooth Controlled Car

05.04.2024

**─**

Group - 22 / Section - 17

CHARNIYA KRISH

KAVYAN SHAH

KEVIN JAIS

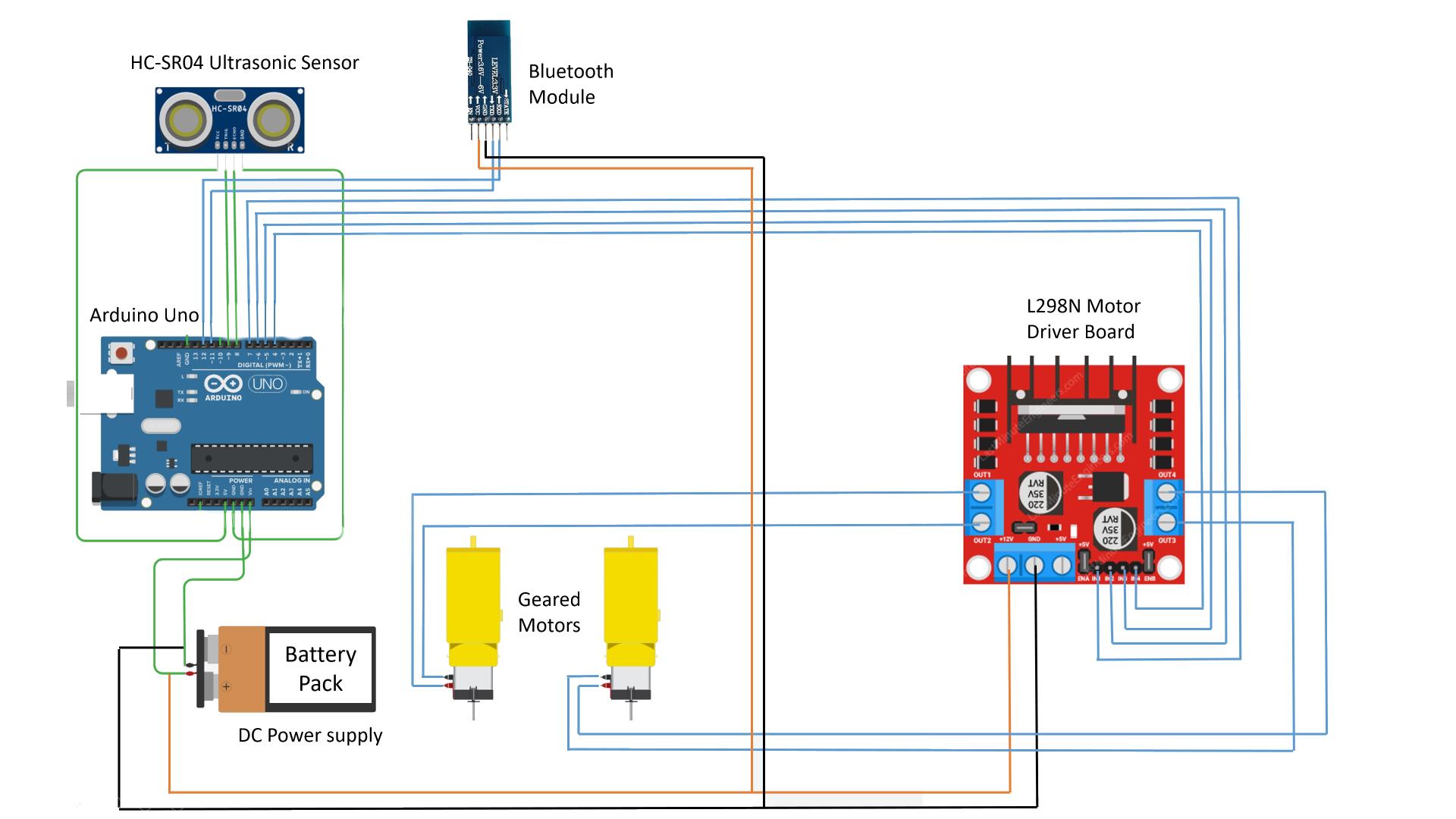
# Motivation

* The need of an automated power driven electric mode of transport for traveling small distances like inside IIT Kharagpur campus. This would reduce human efforts and hence, the driver traveling through it would only need to control the directions and our device would itself control the braking.
* This system could be also utilized as an additional safety feature of automatic braking in all other modes of road transport to reduce accidents due to human error.
* All the existing automatic electric vehicles out there in the market are very expensive so this prototype of ours provides affordability by only providing automatic braking and giving manual handling.

# Work Methodology

The obstacle-avoiding car with Bluetooth control operates by integrating various hardware components and software algorithms. The Arduino UNO R3 microcontroller serves as the brain of the system, receiving inputs from the HC-SR04 Ultrasonic Sensor to detect obstacles in the car's path by Measuring distance to avoid collisions by controlling the TT Gear Motors via the L298N Motor Driver. Additionally, the HC-05 Bluetooth module enables remote control of the car from a paired device such as a smartphone, allowing the user to override autonomous navigation and steer the car manually. This fusion of obstacle avoidance and Bluetooth control functionalities offers versatility and adaptability to different scenarios, making the car suitable for both autonomous exploration and user-directed tasks.

# Schematic of Setup



# 

# List of Items

* Arduino UNOR3
* L298N Motor Driver
* HC-SR04 Ultrasonic Sensor
* HC-05 Bluetooth
* TT Gear Motor
* 18650 Battery
* 18650 Battery Holder
* 65MM Wheels for TT Motors
* Jumper Wires

# Item Specifications

## Arduino UNO R3:

* + Microcontroller: ATmega328P
  + Operating Voltage: 5V
  + Digital I/O Pins: 14 (including 6 PWM outputs)
  + Analog Input Pins: 6
  + Flash Memory: 32 KB (0.5 KB used by bootloader)
  + SRAM: 2 KB
  + EEPROM: 1 KB
  + Clock Speed: 16 MHz

## L298N Motor Driver:

* + Input Voltage: 5V - 35V
  + Output Current: 2A (continuous) / 3A (peak)
  + Logic Voltage: 5V
  + Logic Current: 0 - 36mA

## HC-SR04 Ultrasonic Sensor:

* + Operating Voltage: 5V
  + Operating Current: < 15mA
  + Operating Frequency: 40kHz
  + Detection Range: 2cm - 400cm
  + Resolution: 0.3cm

## HC-05 Bluetooth Module:

* + Bluetooth Version: Bluetooth 2.0+EDR
  + Operating Voltage: 3.6V - 6V
  + Default Baud Rate: 9600 bps
  + Operating Current: < 40mA
  + Transmission Range: Up to 10 meters

# Code

#include <NewPing.h>

//sensor pins

#define trig\_pin A1 //analog input 1

#define echo\_pin A2 //analog input 2

//in1 in2 in3 in4 4 5 6 7

#define maximum\_distance 200

boolean goesForward = false;

int distance = 100;

NewPing sonar(trig\_pin, echo\_pin, maximum\_distance); //sensor function

char getstr;

const int LeftMotorForward = 7;

const int LeftMotorBackward = 6;

const int RightMotorForward = 5;

const int RightMotorBackward = 4;

void \_mForward()

{

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

void \_mBack()

{

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorForward, LOW);

}

void \_mleft()

{

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

void \_mright()

{

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorBackward, LOW);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, LOW);

}

void \_mStop()

{

digitalWrite(RightMotorForward, LOW);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

digitalWrite(LeftMotorBackward, LOW);

}

void setup()

{

Serial.begin(9600);

pinMode(RightMotorForward, OUTPUT);

pinMode(LeftMotorForward, OUTPUT);

pinMode(LeftMotorBackward, OUTPUT);

pinMode(RightMotorBackward, OUTPUT);

distance = readPing();

delay(100);

}

void loop()

{

int distanceRight = 0;

int distanceLeft = 0;

delay(50);

if (distance <= 20){

\_mStop();

getstr=Serial.read();

if(getstr=='B'){

\_mBack();

}

delay(300);

}

else{

getstr=Serial.read();

if(getstr=='F')

{

\_mForward();

}

else if(getstr=='B')

{

\_mBack();

}

else if(getstr=='R')

{

\_mleft();

}

else if(getstr=='L')

{

\_mright();

}

else if(getstr=='X')

{

\_mStop();

}

}

distance = readPing();

}

int readPing(){

delay(70);

int cm = sonar.ping\_cm();

if (cm==0){

cm=250;

}

return cm;

}

# Challenges

* In the very Beginning we faced a problem where we weren’t able to connect the Bluetooth Module to the Smartphone.
* Initially our project was to make a model which could be used by a real human but due to its high cost we had to switch to a small scale model of the same.
* The Ultrasonic Sensor was not measuring distance and printing in the Serial Monitor. Later we found out we were using the libraries in the code wrong.
* We made a huge mistake during the preparation of the project. When we were connecting the battery, accidentally the positive terminal and negative terminal of the battery were connected and a short circuit happened, we quickly removed the batteries from it.
* We Learn a lot of things through searching and digging Google every time we faced a Problem.
* Sometimes there was a Fault in a component as these components are made in China, Sometimes we were lacking the important logic in the code.
* We changed our Code multiple times to try multiple Solutions to the Problem Statement and We Finally arrived at our Final Conclusion!
* We watched Multiple Arduino Tutorials and read a dozen of articles to get working with Arduino.

# Learnings

## Teamwork

This project allowed us to appreciate the essence of teamwork as while working on the project, we worked together while distributing the tasks amongst ourselves and depending on others' work as well as ours.

## Arduino

Arduino is a very essential software which none of us was familiar with at the start of the semester. Working on this project allowed us to delve deeper into its nuances and learn how to use it most efficiently.

## Electronic Components

During this project, we worked with various electronic components. This allowed us to get used to the practicalities of theirs and we learnt how to best utilize them in our project.

# References

GitHub Repo containing all the Details:

https://github.com/chrisrex007/arduino\_diy/