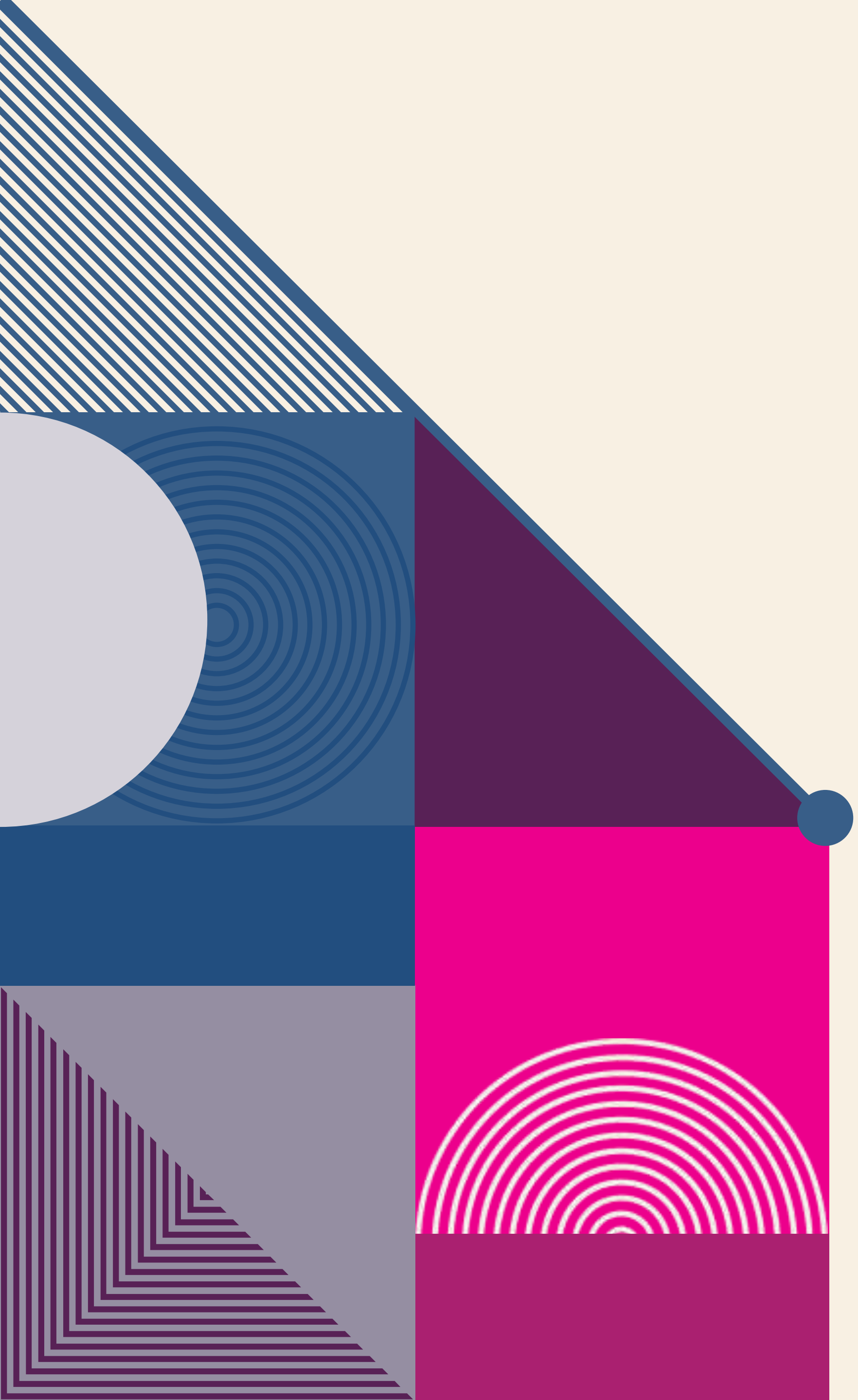
VEHICLE AUTOMATION

DIY Team Project Group – **M**

Hricheek Mondal	**21MA10023**
Dipankar Yadav	**21AG30014**
Suvendu Dalbehera	**21ME10087**
Sabariswaran	**21EC30030**
Sukanth	**21CS10066**



OBJECTIVE AND SCOPE

This project is an attempt to build an autonomous vehicle that can detect obstacles in its path and safely navigate a way out to reach its final pre-programmed destination. The work can further be enhanced and upgraded by introducing a camera sensor and an object recognition software through computer vision to make it more realistic such that it can obey traffic lights and even stop in front of 'STOP' signs.

This unmanned vehicle can be used as a mail/food delivery system, and even for advertising purposes. The same logic can be extended to self-driving vehicles by training a model accordingly.

PRESENT PROBLEMS

LAGGING DELIVERY SYSTEM

The present delivery systems are not up to the pace of modernization. It is a somewhat neglected with very few upgrades.

WORKER DEPENDENCY

The current delivery systems is completely dependent upon human resources and thus failure is bound to happen from time to time.

COSTS

The few companies and start-ups providing autonomous solutions are way too expensive and complex.

SOLUTIONS

CAPABILITY

This product makes human lives easier, and the delivery systems will be revamped. For a self driving car the code can be modified and the model can be trained accordingly.

TARGET SECTORS

Delivery systems, self-driving vehicles and off-roading.

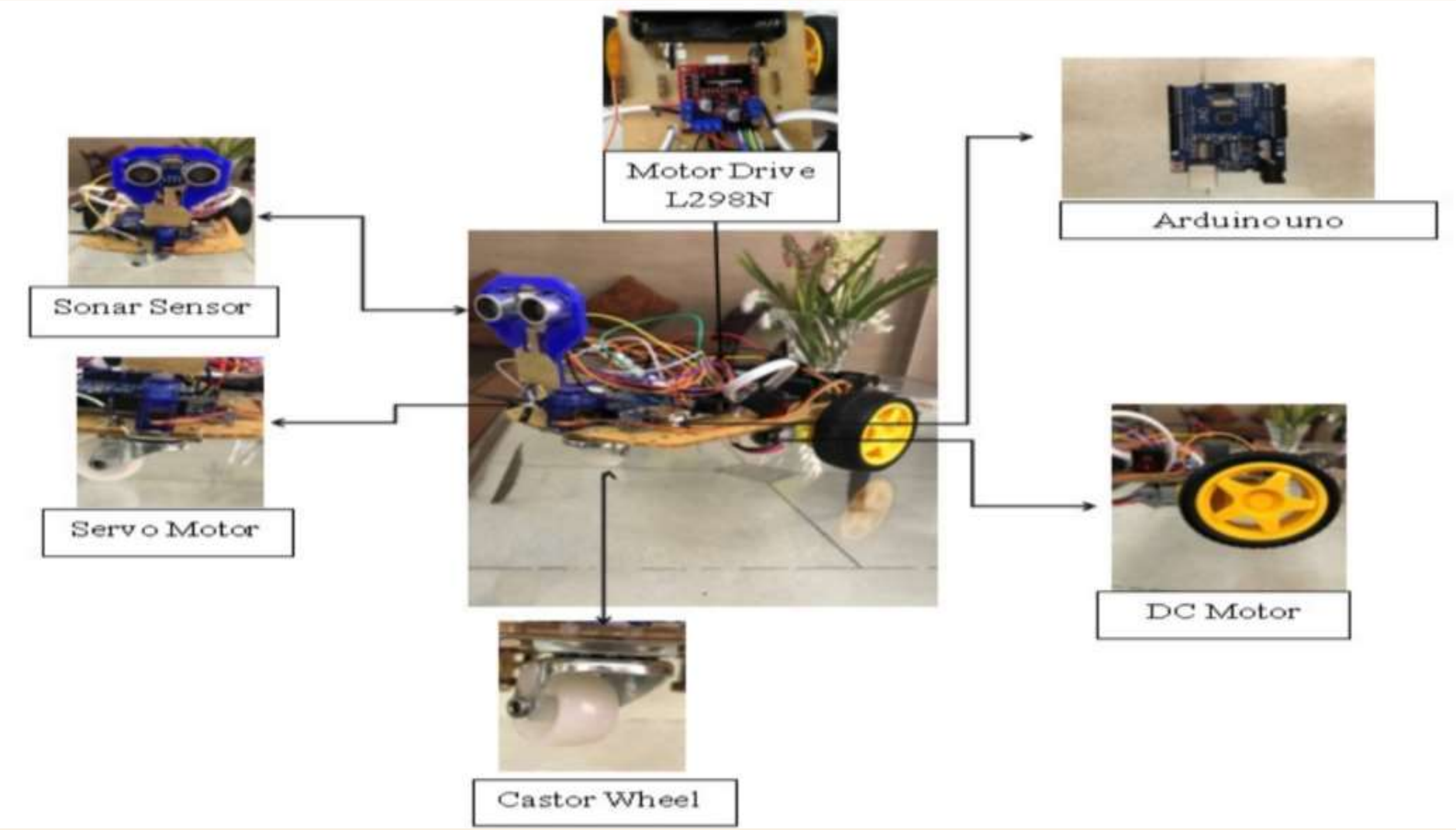
COST SAVINGS

The simple and cost-efficient obstacle detecting mobile robot might just be the solution!

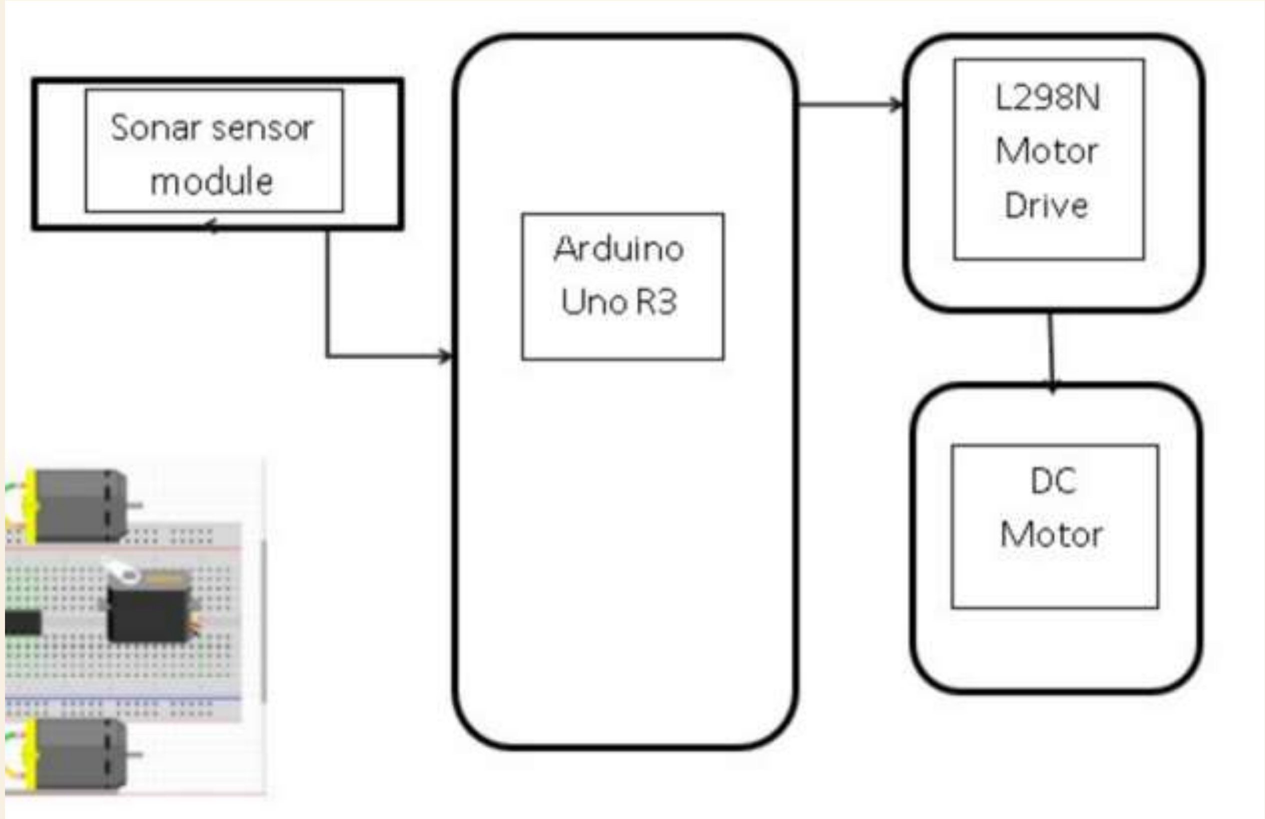
EASY TO USE

The simple design and ready-to-use objective makes it easier to control and supervise. No worker dependency as a result, no delays.

SYSTEM ARCHITECTURE

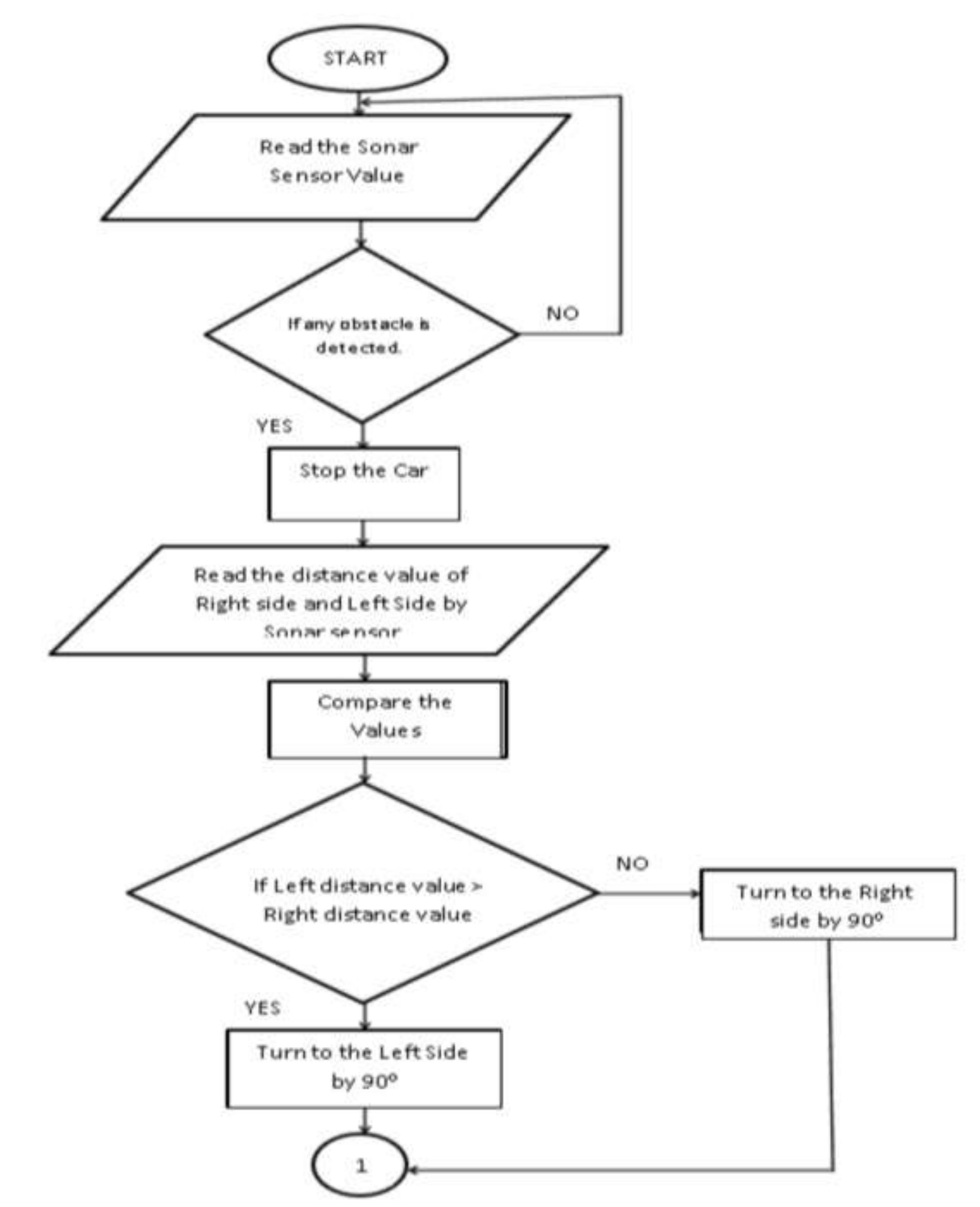


SYSTEM DESIGN

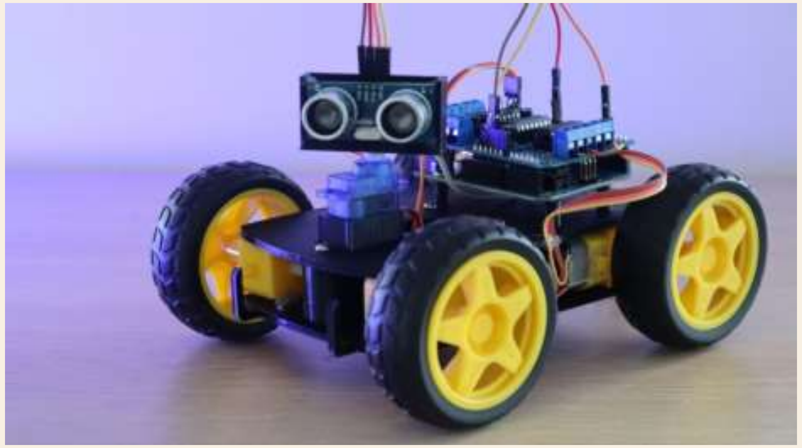


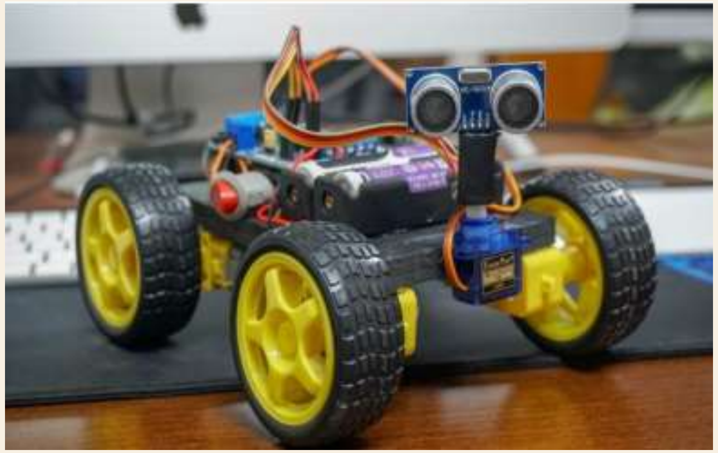
CIRCUIT FOR OBSTACLE SENSING ROBOT PROCESS

OBSTACLE AVOIDANCE OPERATION


START
Read the Sonar Sensor Value
If any obstacle is detected.
NO
YES
Stop the Car
Read the distance value of Right side and Left Side by Sonar sensor
Compare the Values
If Left distance value > Right distance value
NO
Turn to the Right side by 90°
YES
Turn to the Left Side by 90°
1

PROPOSED MODEL





DESIGN

In this project the ultrasonic sensor sends the received signal to the Arduino to process the information according to the codes. The final output sends the vital information to the dc motors to rotate accordingly and thus safely navigate.

MATERIALS REQUIRED

Here's the components that will be used for the model.

Sl. no.	Part Name	Quantity	Description	Specifications	Cost (Rs.)
1	Arduino Uno R3	1	Microcontroller board - Plastic	Op. Voltage-5V Digital pins-14	500
2	Ultrasonic Sensor	1	Measures distance to obstacle	HC-SR04 Ultrasonic Freq:40kHz, Range:2-400cm	100
3	Servo Motor	1	Rotary actuator for precise control of angular position	SG90 Servo, Torque: 2.0–2.2Kg/cm, Speed: 0.09s/cm	500
4	Wheels	4	For movement of the vehicle	Project specific, Hard plastic	100
5	DC Motors	4	DC BO Gear Motor	Speed: 100 RPM Voltage: 9V-3V	200
6	Motor Driver Chip	1	Controls speed of 4 wheels acc. to code.	L293D Pulsed current 1.2 A/driver	250
7	Chassis	1	Light weight but rigid base	Plastic/ Metal	-
8	Jumping cables	TBC	For multiple connections	M-M, M-F, F-F	50

SOFTWARES

Tinkercad

The Tinkercad software will be used for planning and making a circuit layout that we would be going to implement upon the working model.

Arduino IDE

The basis behind proper functioning of the vehicle lies in the code. It is indeed the backbone of any autonomous robot. For this purpose we are going to use C/C++ coding in Arduino IDE.

WORK STRATEGY

Here’s how the work has been divided.

PHASE 1: PREPARATION OF WORKING MODEL

SUVENDU DALBEHERA	21ME10087

PHASE 2: TINKERCAD AND CODE GENERATION

DIPANKAR YADAV	21AG30014
SUKANTH	21CS10066

PHASE 3: PRESENTATIONS AND REPORT

HRICHEEK MONDAL	21MA10023
SABARISWARAN	21EC30030

TIMELINE

17th Jan – 23rd Jan 2022:

Rigorous planning of the entire work and sharing work strategy.

24th Jan – 30st Jan 2022:

Purchasing all the required components and making a circuit layout in Tinkercad.

31st Jan – 6th Feb 2022:

Assembling the components and finishing the working model. Beginning work on the codes.

14th Feb – 21st Feb 2022:

Completing the code and debugging it.

Implementing it in our working model and further fine tunings.

THANK YOU

GROUP - **M**

Hricheek Mondal	**21MA10023**
Dipankar Yadav	**21AG30014**
Suvendu Dalbehera	**21ME10087**
Sabariswaran	**21EC30030**
Sukanth	**21CS10066**

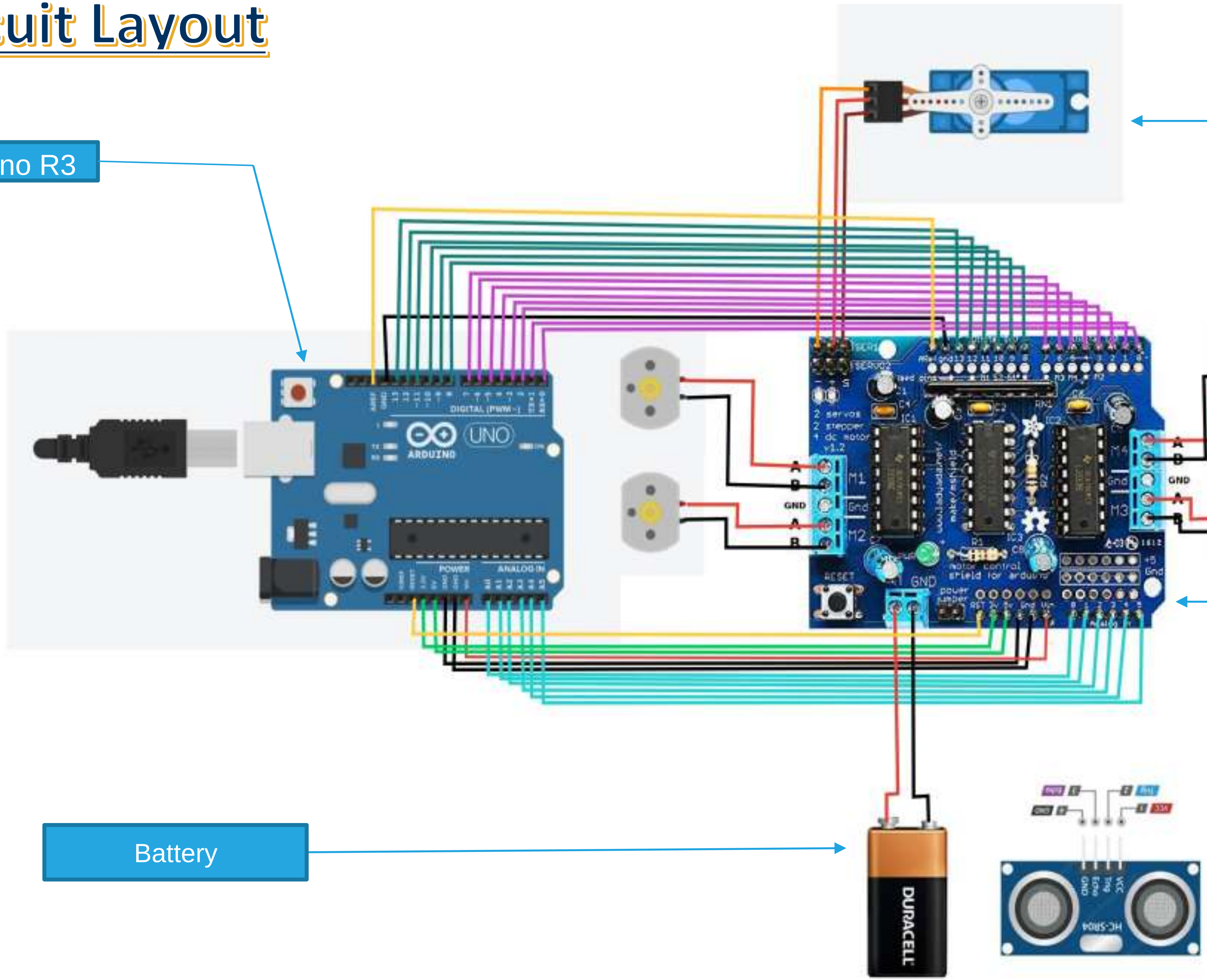
Vehicle automation

Week -2

DIY Team Project Group – **M**

Hricheek Mondal	**21MA10023**
Dipankar Kumar	**21AG30014**
Suvendu Dalbehera	**21ME10087**
Sabariswaran	**21EC30030**
Sukanth	**21CS10066**

Circuit Layout



Arduino Uno R3

Servo motor

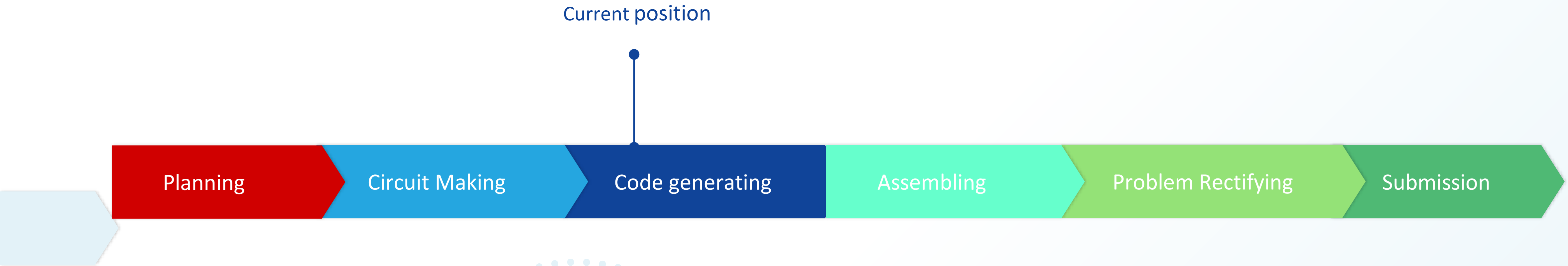
DC motor

Motor Driver chip

Battery

Ultrasonic sensor

Timeline



Current position

Planning

Circuit Making

Code generating

Assembling

Problem Rectifying

Submission

Work strategy

Here’s how the work has been divided.

PHASE 1: PREPARATION OF WORKING MODEL

SUVENDU DALBEHERA	21ME10087

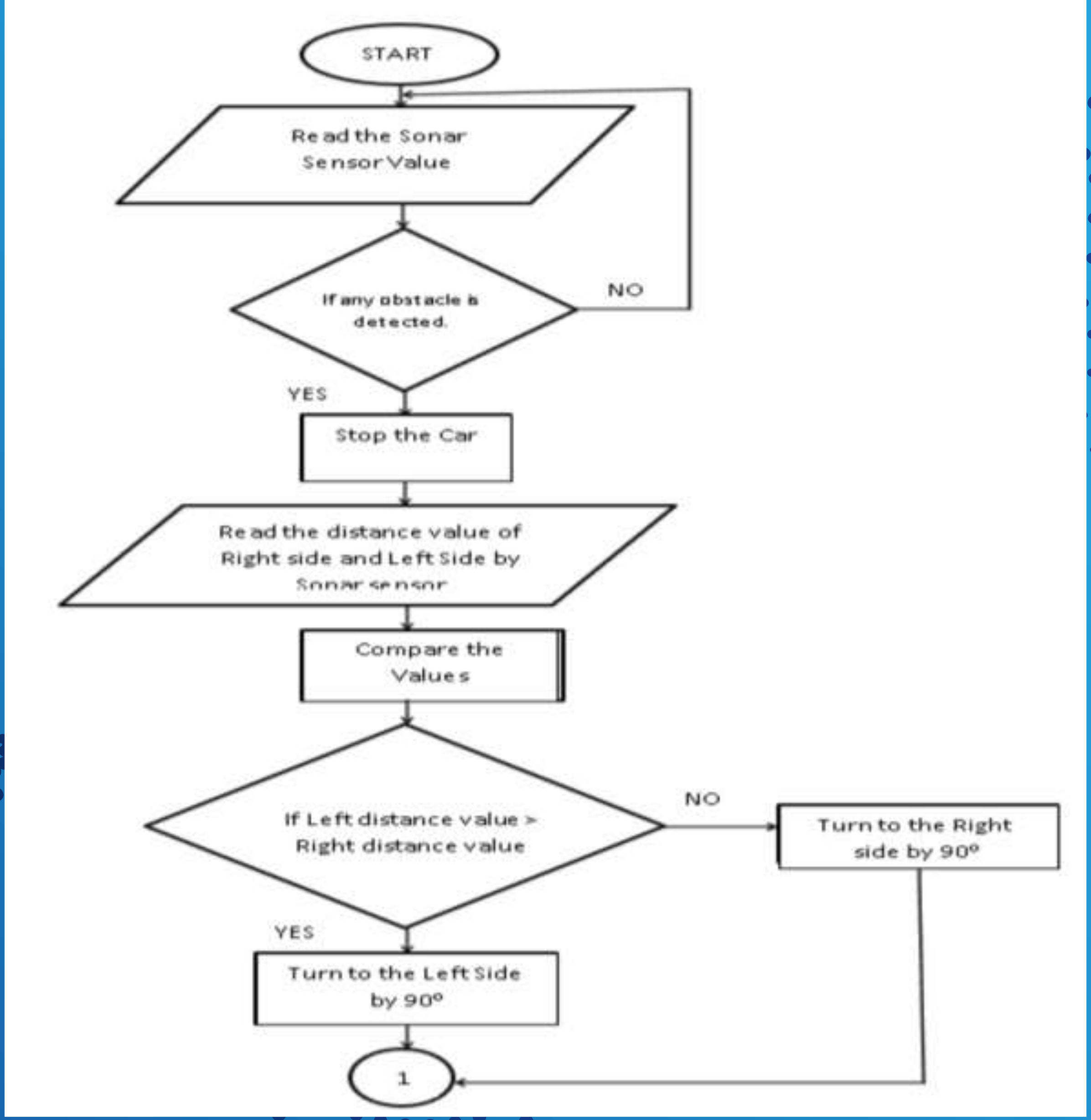
PHASE 2: TINKERCAD AND CODE GENERATION

DIPANKAR YADAV	21AG30014
SUKANTH	21CS10066

PHASE 3: PRESENTATIONS AND REPORT

HRICHEEK MONDAL	21MA10023
SABARISWARAN	21EC30030

Obstacle avoidance operation

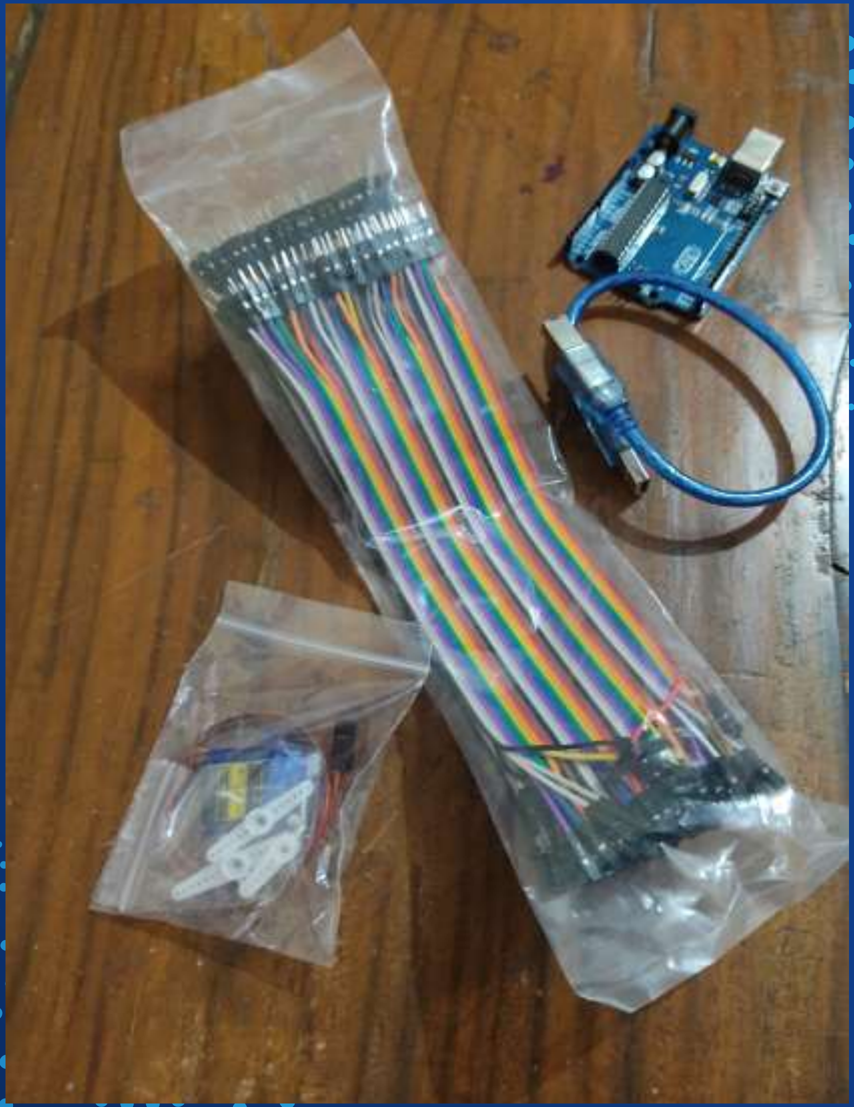

START
Read the Sonar Sensor Value
If any obstacle is detected.
NO
YES
Stop the Car
Read the distance value of Right side and Left Side by Sonar sensor
Compare the Values
If Left distance value > Right distance value
NO
Turn to the Right side by 90º
YES
Turn to the Left Side by 90º
1

Materials required

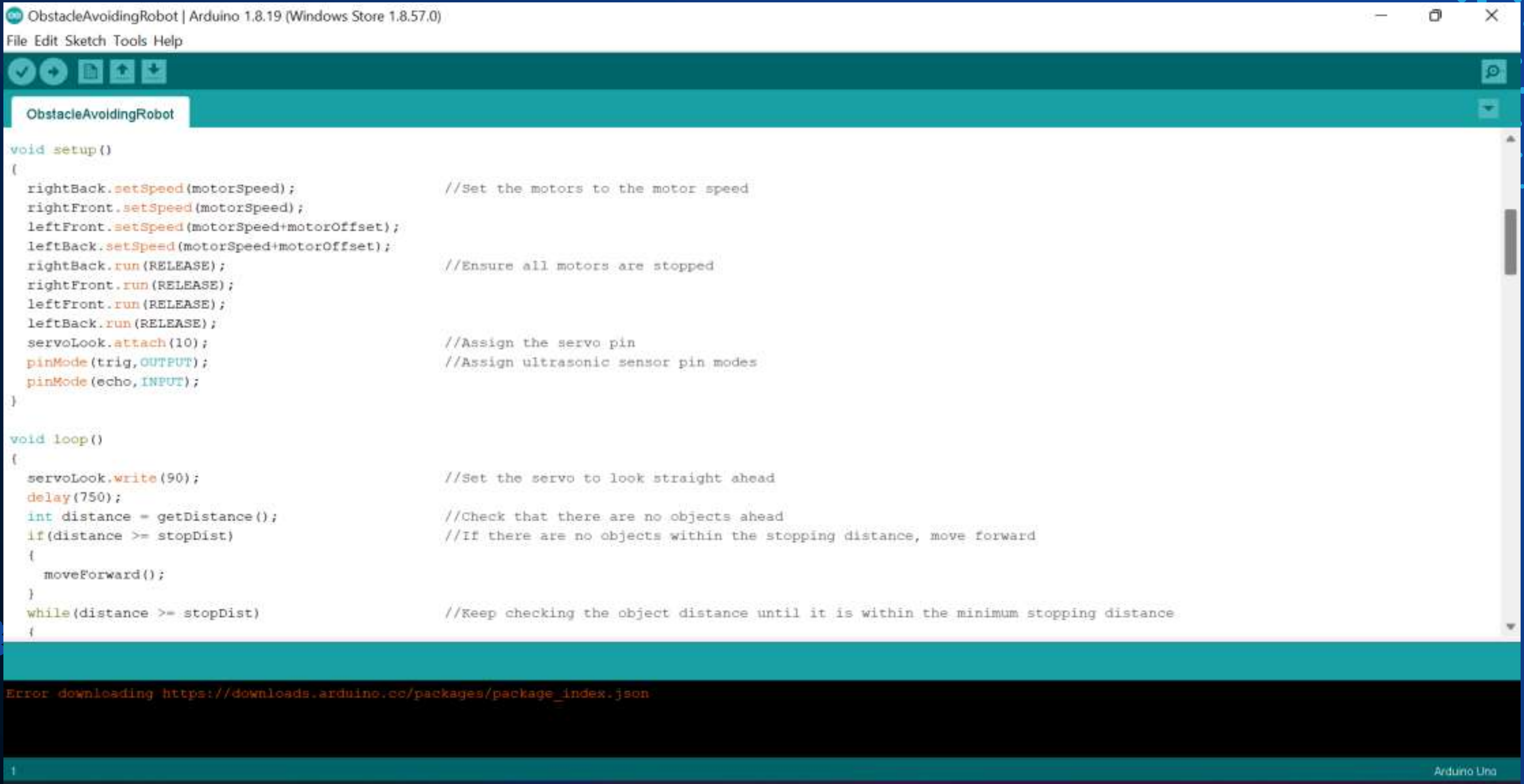
Here’s the components that will be used for the model.

Sl. no.	Part Name	Quantity	Description	Specifications	Cost (Rs.)
1	Arduino Uno R3	1	Microcontroller board - Plastic	Op. Voltage-5V Digital pins-14	500
2	Ultrasonic Sensor	1	Measures distance to obstacle	HC-SR04 Ultrasonic Freq:40kHz, Range:2-400cm	100
3	Servo Motor	1	Rotary actuator for precise control of angular position	SG90 Servo, Torque: 2.0–2.2Kg/cm, Speed: 0.09s/cm	500
4	Wheels	4	For movement of the vehicle	Project specific, Hard plastic	100
5	DC Motors	4	DC BO Gear Motor	Speed: 100 RPM Voltage: 9V-3V	200
6	Motor Driver Chip	1	Controls speed of 4 wheels acc. to code.	L293D Pulsed current 1.2 A/driver	250
7	Chassis	1	Hard plastic	Cardboard	-
8	Jumping cables	TBC	For multiple connections	M-M, M-F, F-F	50
9	Battery	1	To power		

Products received till now



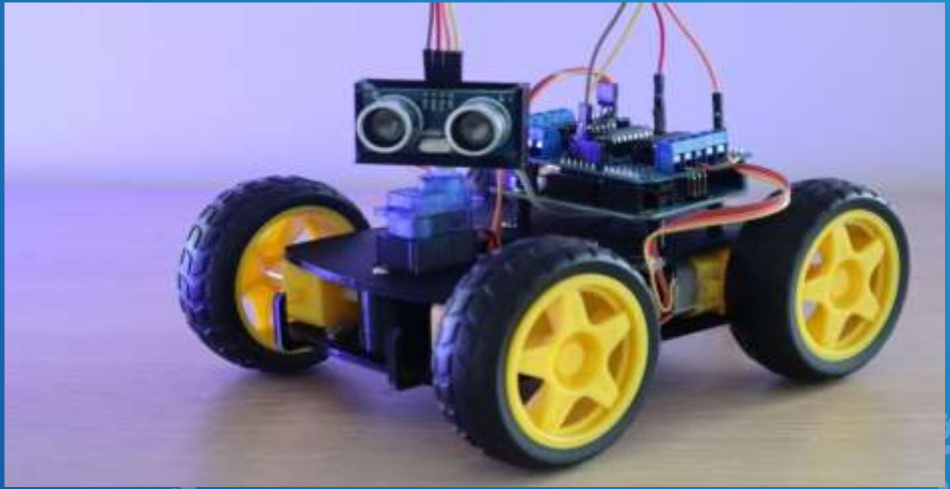
Glimpse of coding that has been done till now


ObstacleAvoidingRobot | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
ObstacleAvoidingRobot
1
Arduino Uno

```
void setup()
{
  rightBack.setSpeed(motorSpeed);                 //Set the motors to the motor speed
  rightFront.setSpeed(motorSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);                         //Ensure all motors are stopped
  rightFront.run(RELEASE);
  leftFront.run(RELEASE);
  leftBack.run(RELEASE);
  servoLook.attach(10);                           //Assign the servo pin
  pinMode(trig,OUTPUT);                           //Assign ultrasonic sensor pin modes
  pinMode(echo,INPUT);
}

void loop()
{
  servoLook.write(90);                            //Set the servo to look straight ahead
  delay(750);
  int distance = getDistance();                   //Check that there are no objects ahead
  if(distance >= stopDist)                        //If there are no objects within the stopping distance, move forward
  {
    moveForward();
  }
  while(distance >= stopDist)                     //Keep checking the object distance until it is within the minimum stopping distance
  {
```

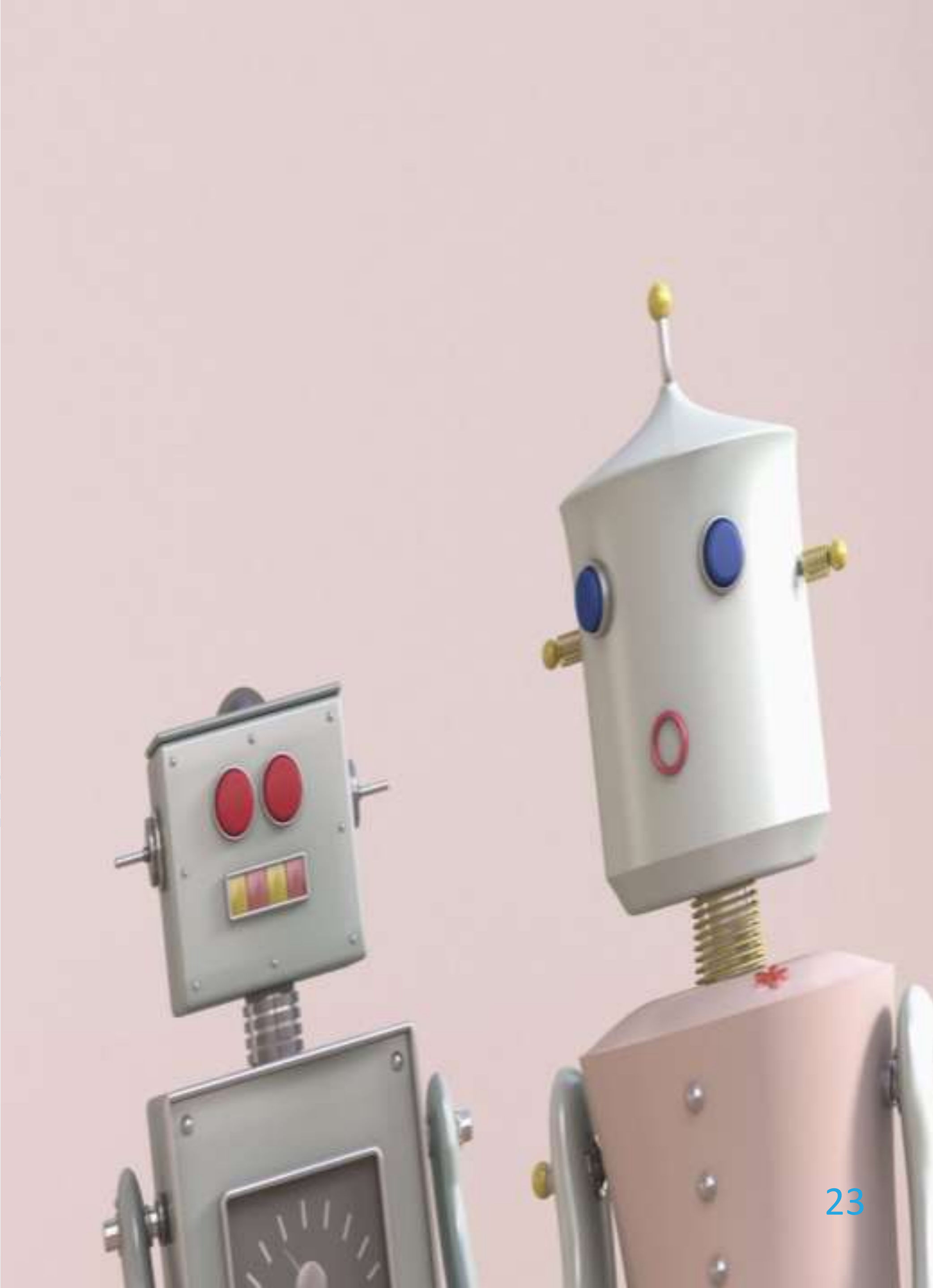
```
Error downloading https://downloads.arduino.cc/packages/package_index.json
```

Proposed Model



ThankYou

Any questions?

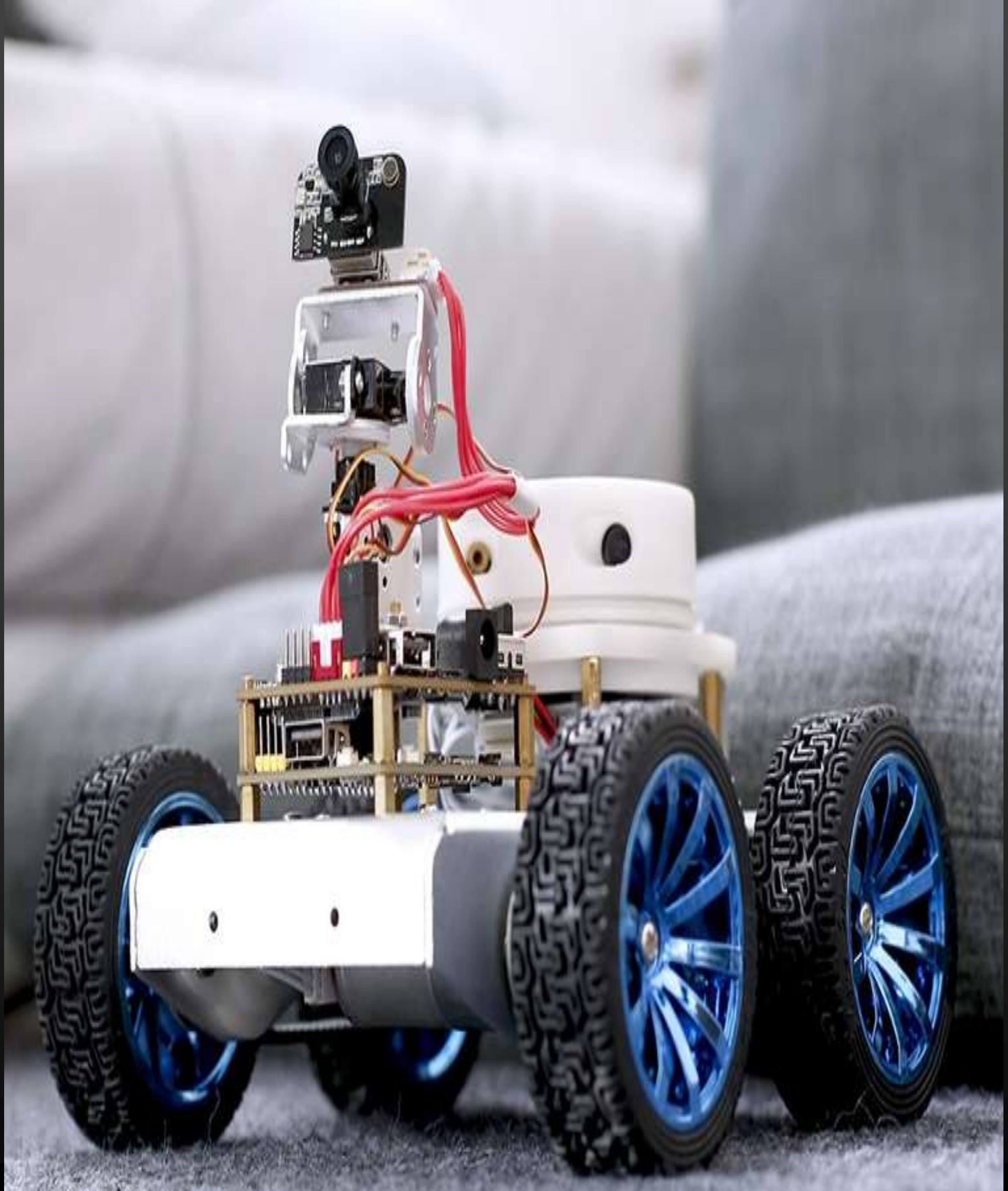


VEHICLE AUTOMATION

WEEK – 3

DIY Team Project Group – **M**

Hricheek Mondal	**21MA10023**
Dipankar Yadav	**21AG30014**
Suvendu Dalbehera	**21ME10087**
Sabariswaran	**21EC30030**
Sukanth	**21CS10066**

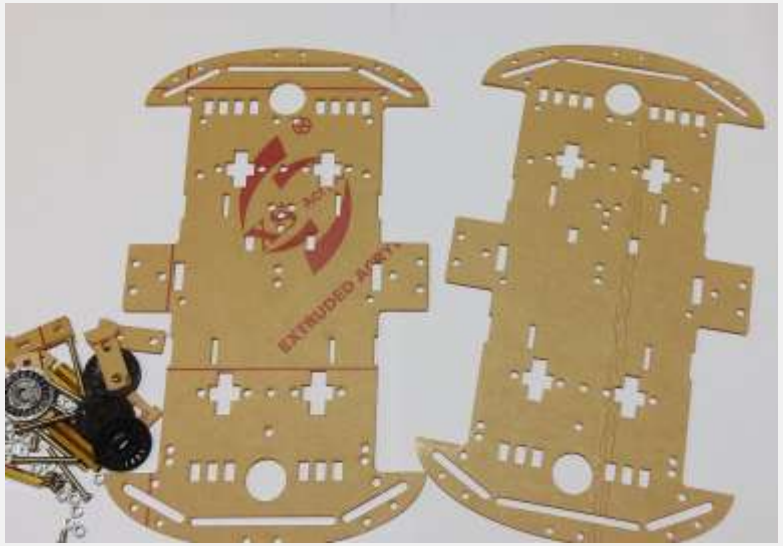


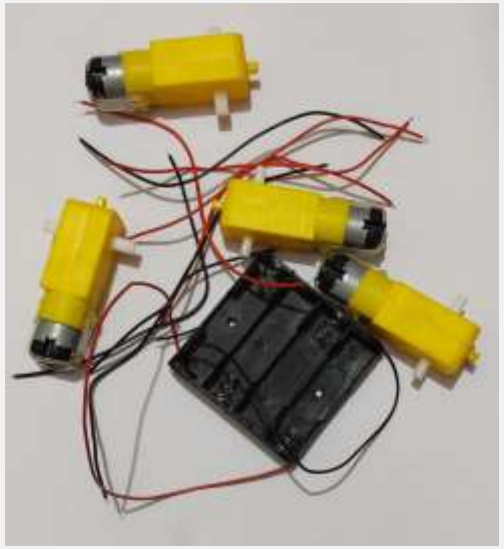
CHALLENGES FACED

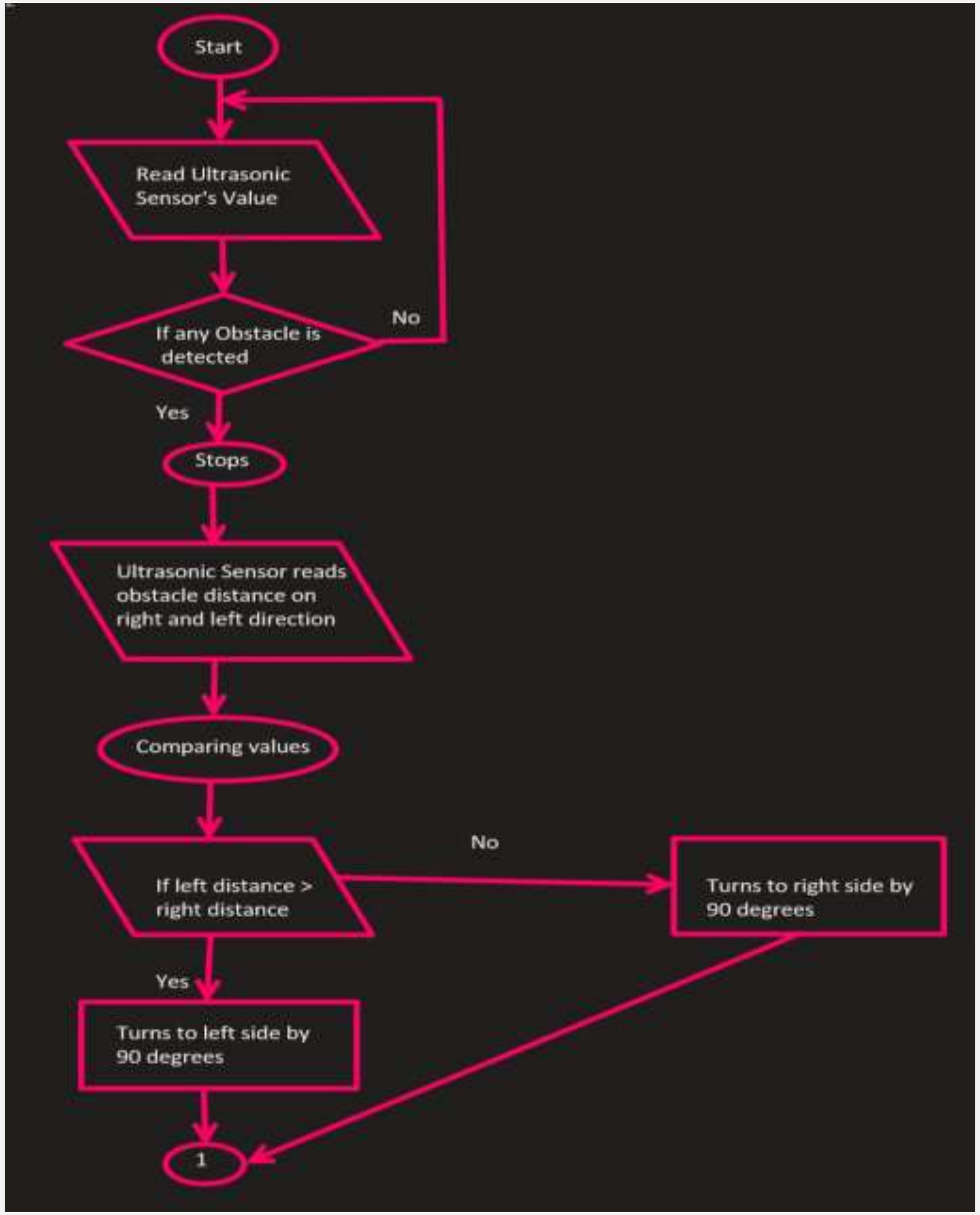
❑ The delayed arrival of the motor driver hampered our progress with the working model.

❑ The motor driver shield lacked some connections for jumper wires as a result they had to be connected to the pin holes with the help of soldering which delayed our work by a couple of days.

PRODUCTS USED





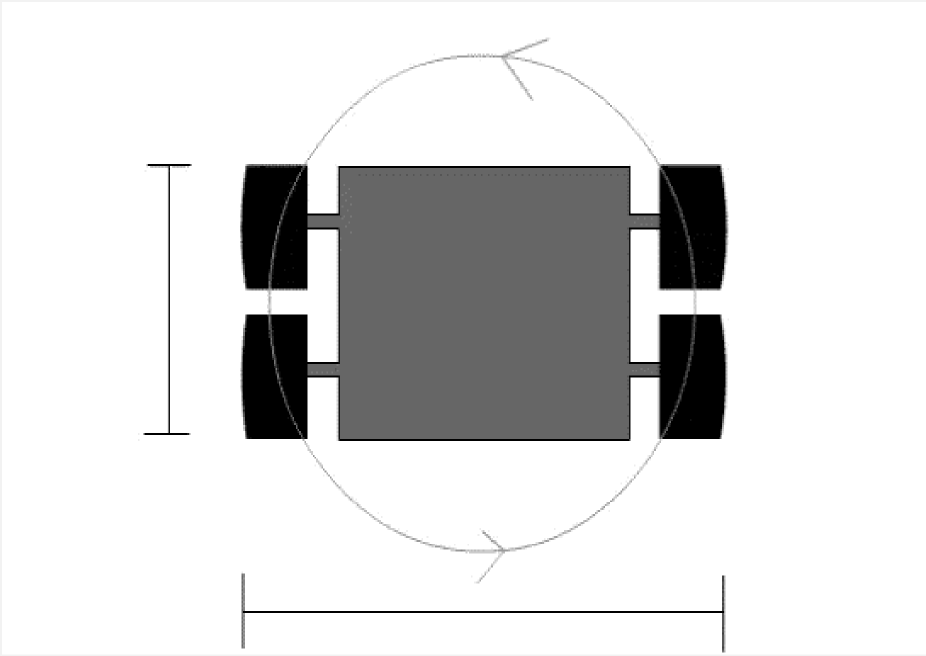

Start
Read Ultrasonic Sensor's Value
If any Obstacle is detected
No
Yes
Stops
Ultrasonic Sensor reads obstacle distance on right and left direction
Comparing values
If left distance > right distance
No
Turns to right side by 90 degrees
Yes
Turns to left side by 90 degrees
1

SCHEMATIC

A brief logic diagram behind the working of the physical model.

ROTATION EXPLAINED

We are concentrating on pivot turning for directional change of the vehicle to make the motion more efficient as it consumes lesser space. The wheels of the car needs to rotate in opposite directions. The vehicle takes a zero radius 360 degree turn, the front wheels give almost negligible resistance to the overall circular motion generated by the rear wheels. Hence the torque applied by motors on each wheels gets properly harnessed, resulting in better performance without any drag or wheel slip while taking turn. Motors has been attached to each wheels to operate each wheels independently and control the robocar motion in an even better way. Thus we don't require omnidirectional wheels, and a set of ordinary rubber wheels would do just fine (cost saving).

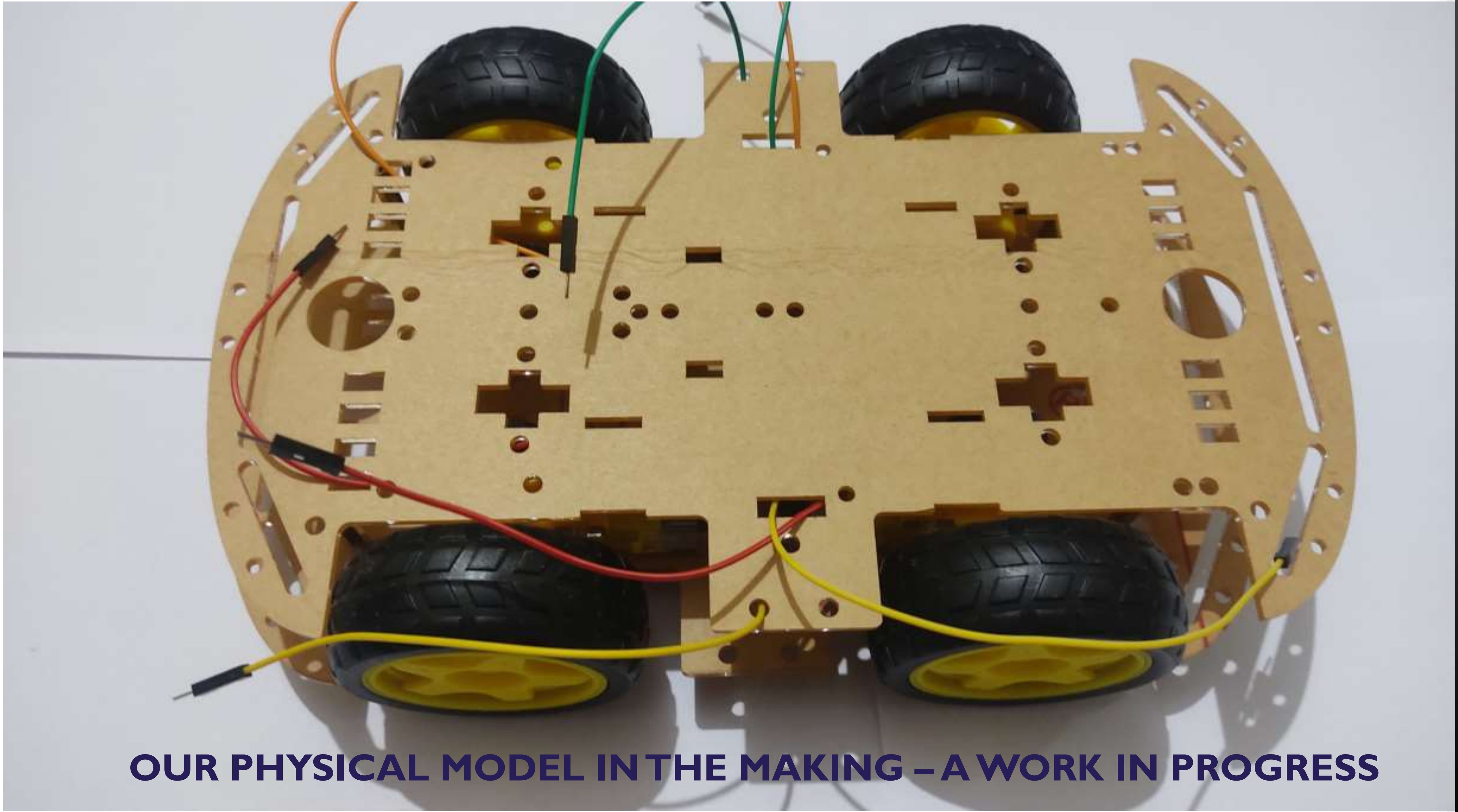


Jens Martensson

CURRENT WORK STATUS

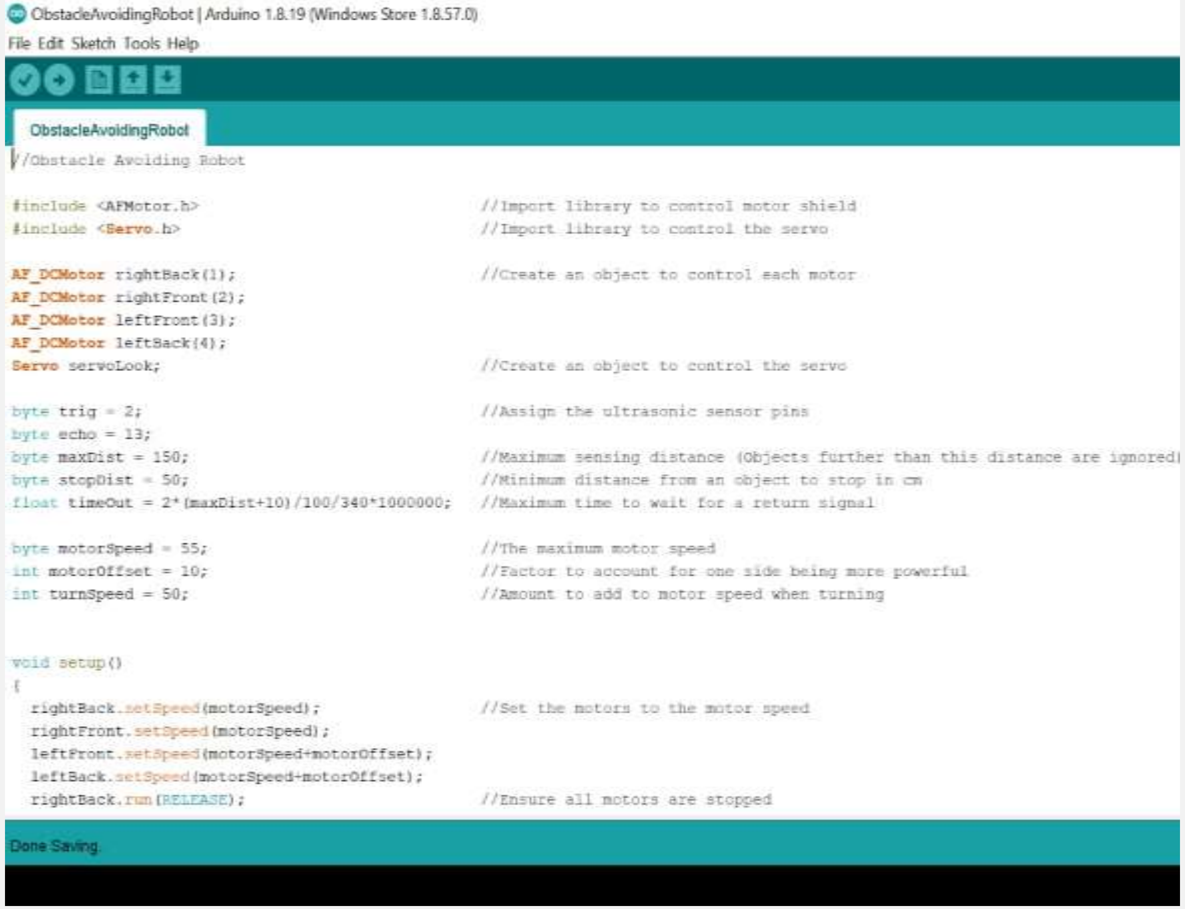
The code has been completed and suitably debugged.

The physical model almost done barring some connections pending due to some soldering works.



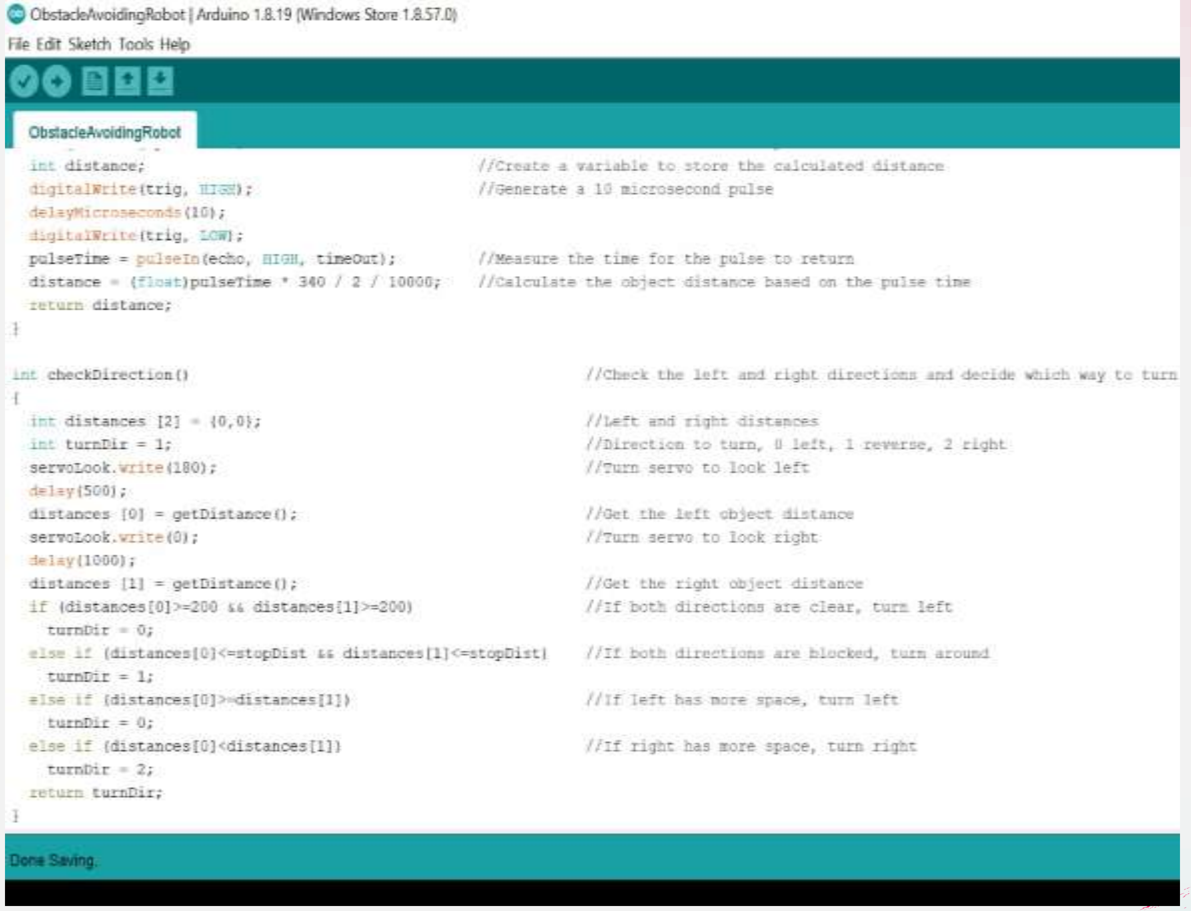
OUR PHYSICAL MODEL IN THE MAKING – A WORK IN PROGRESS

CODE SNIPPETS


ObstacleAvoidingRobot | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
ObstacleAvoidingRobot

```
//Obstacle Avoiding Robot

#include <AFMotor.h>                            //Import library to control motor shield
#include <Servo.h>                              //Import library to control the servo

AF_DCMotor rightBack(1);                        //Create an object to control each motor
AF_DCMotor rightFront(2);
AF_DCMotor leftFront(3);
AF_DCMotor leftBack(4);
Servo servoLook;                                //Create an object to control the servo

byte trig = 2;                                  //Assign the ultrasonic sensor pins
byte echo = 13;
byte maxDist = 150;                             //Maximum sensing distance (Objects further than this distance are ignored)
byte stopDist = 50;                             //Minimum distance from an object to stop in cm
float timeOut = 2*(maxDist+10)/100/340*1000000; //Maximum time to wait for a return signal

byte motorSpeed = 55;                           //The maximum motor speed
int motorOffset = 10;                           //Factor to account for one side being more powerful
int turnSpeed = 50;                             //Amount to add to motor speed when turning


void setup()
{
  rightBack.setSpeed(motorSpeed);               //Set the motors to the motor speed
  rightFront.setSpeed(motorSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);                       //Ensure all motors are stopped
```

Done Saving.


ObstacleAvoidingRobot | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
ObstacleAvoidingRobot

```
  int distance;                                   //Create a variable to store the calculated distance
  digitalWrite(trig, HIGH);                       //Generate a 10 microsecond pulse
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  pulseTime = pulseIn(echo, HIGH, timeOut);       //Measure the time for the pulse to return
  distance = (float)pulseTime * 340 / 2 / 10000;  //Calculate the object distance based on the pulse time
  return distance;
}

int checkDirection()                                              //Check the left and right directions and decide which way to turn
{
  int distances [2] = {0,0};                                      //Left and right distances
  int turnDir = 1;                                                //Direction to turn, 0 left, 1 reverse, 2 right
  servoLook.write(180);                                           //Turn servo to look left
  delay(500);
  distances [0] = getDistance();                                  //Get the left object distance
  servoLook.write(0);                                             //Turn servo to look right
  delay(1000);
  distances [1] = getDistance();                                  //Get the right object distance
  if (distances[0]>=200 && distances[1]>=200)                     //If both directions are clear, turn left
    turnDir = 0;
  else if (distances[0]<=stopDist && distances[1]<=stopDist)      //If both directions are blocked, turn around
    turnDir = 1;
  else if (distances[0]>=distances[1])                            //If left has more space, turn left
    turnDir = 0;
  else if (distances[0]<distances[1])                             //If right has more space, turn right
    turnDir = 2;
  return turnDir;
}
```

Done Saving.

TASKS COMPLETED

MODEL CONSTRUCTION AND ASSEMBLY – Suvendu Dalbehera

CODE CREATION – Sukanth, Dipankar Yadav
SCHEMATIC AND FLOWCHARTS – Dipankar Yadav
DEBUGGING – Dipankar Yadav, Sukanth

WEEK 1 PRESENTATION – Hricheek Mondal
WEEK 2 PRESENTATION – Sabariswaran
WEEK 3 PRESENTATION – Hricheek Mondal

REGULAR MEETS and GROUP DISCUSSION SESSIONS were arranged every week.

FINANCE – Suvendu Dalbehera
Sabariswaran
Hricheek Mondal
Sukanth
Dipankar Yadav
***equally divided among all team members**

TASKS LEFT

Some connections of the model is left. Otherwise everything is going as per plan.

The final implementation of the code and uploading it on the Arduino UNO board is left.
Based on what we perceive in the working of our physical model we would make necessary changes in the code and modify it accordingly.

Week 4 presentation will be prepared once the week progresses and alongside we would be preparing the final report of our entire project.

WORK PLAN – LAST WEEK

FINAL CONNECTIONS IN THE PHYSICAL MODEL AND TESTING – Suvendu Dalbehera

CODE UPLOADING AND FINAL MODIFICATIONS/TUNINGS – Dipankar Yadav, Sukanth

WEEK 4 PRESENTATION – Sabariswaran

FINAL REPORT PREPARATION – Hricheek Mondal, Sabariswaran

Thank You !

DIY Team Project Group – **M**

Hricheek Mondal	**21MA10023**
Dipankar Yadav	**21AG30014**
Suvendu Dalbehera	**21ME10087**
Sabariswaran	**21EC30030**
Sukanth	**21CS10066**

Vehicle automation

DIY Team Project Group - M

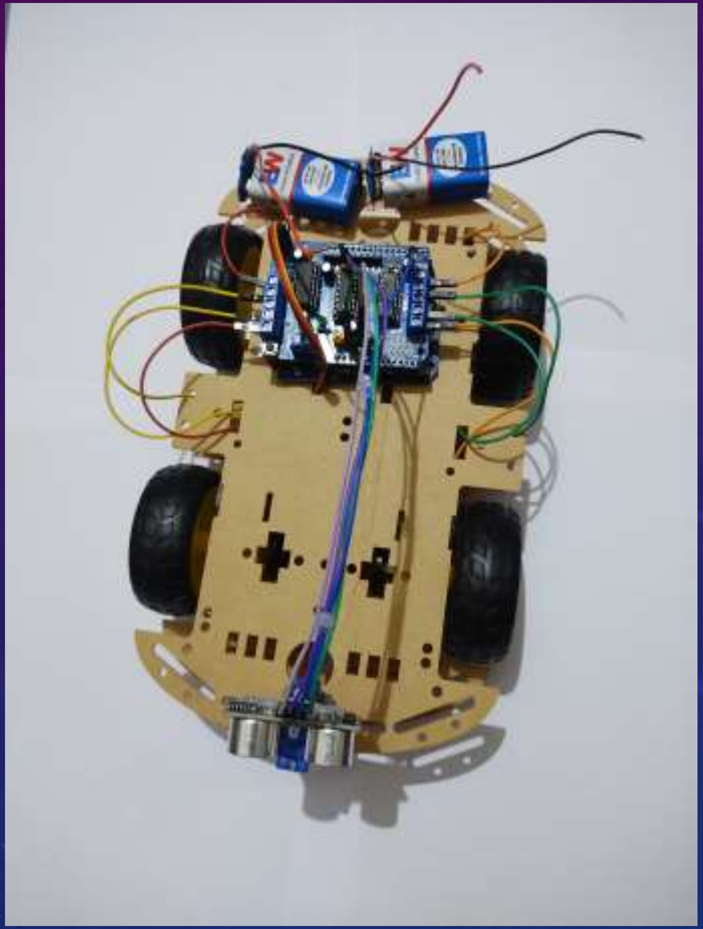
Hricheek Mondal	21MA10023
Dipankar Yadav	21AG30014
Suvendu Dalbehera	21ME10087
Sabariswaran	21EC30030
Sukanth	21CS10066

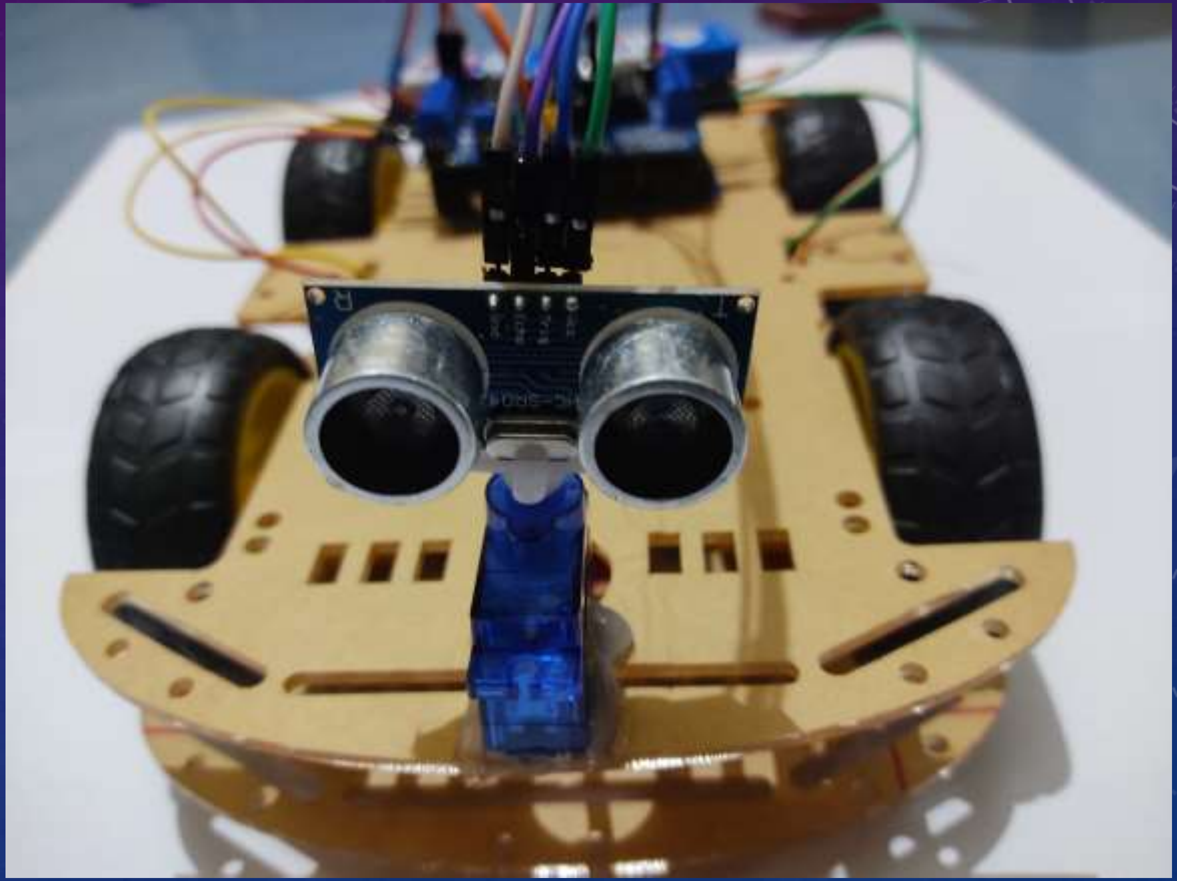
Objective and Scope

This project is an attempt to build an autonomous vehicle that can detect obstacles in its path and safely navigate a way out to reach its final pre-programmed destination. The work can further be enhanced and upgraded by introducing an object recognition software through computer vision to make it more realistic such that it can obey traffic lights and even stop in front of ‘STOP’ signs.

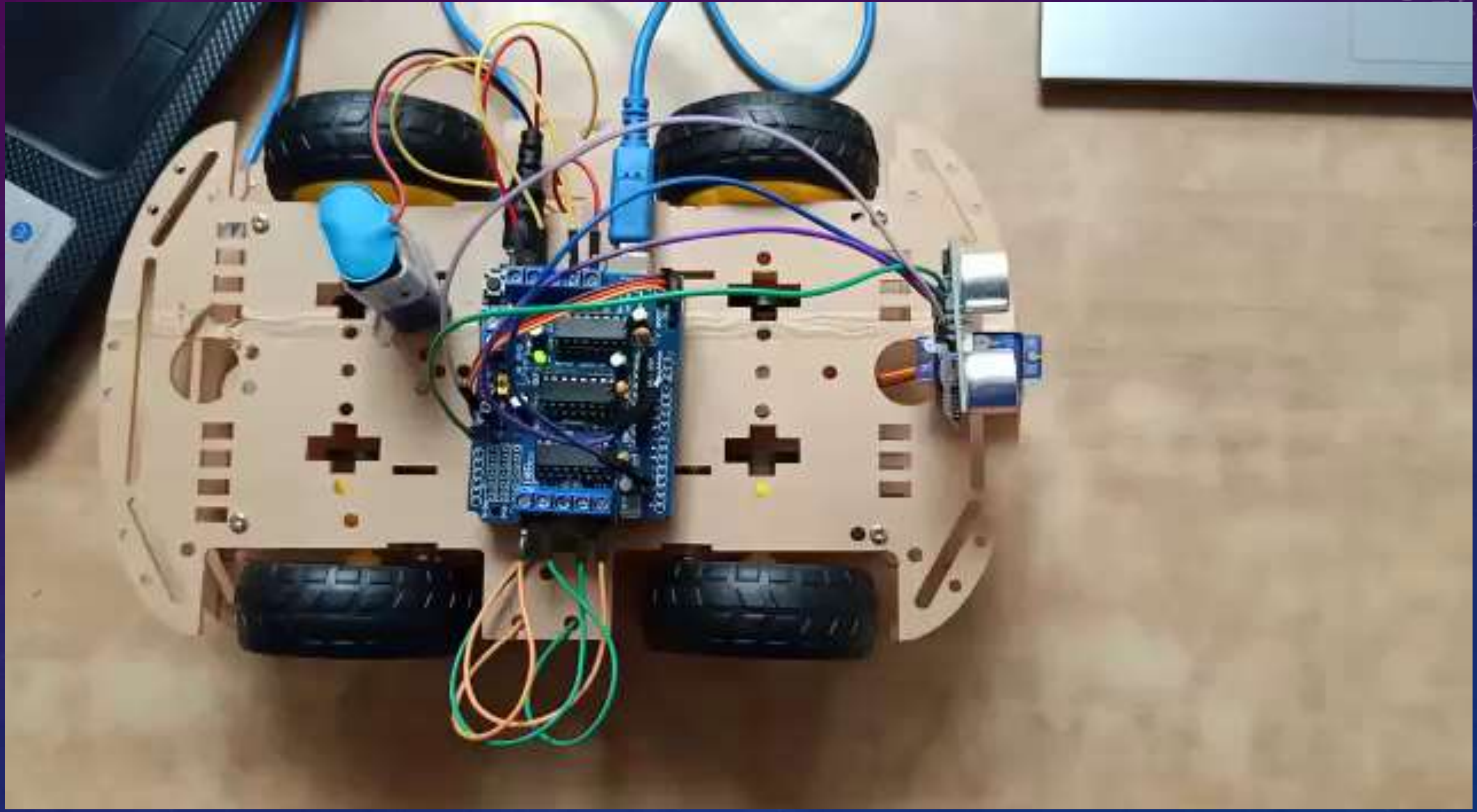
This unmanned vehicle can be used as a mail/food delivery system, and even for advertising purposes. The same logic can be extended to self-driving vehicles by training a model accordingly.

Current Photos



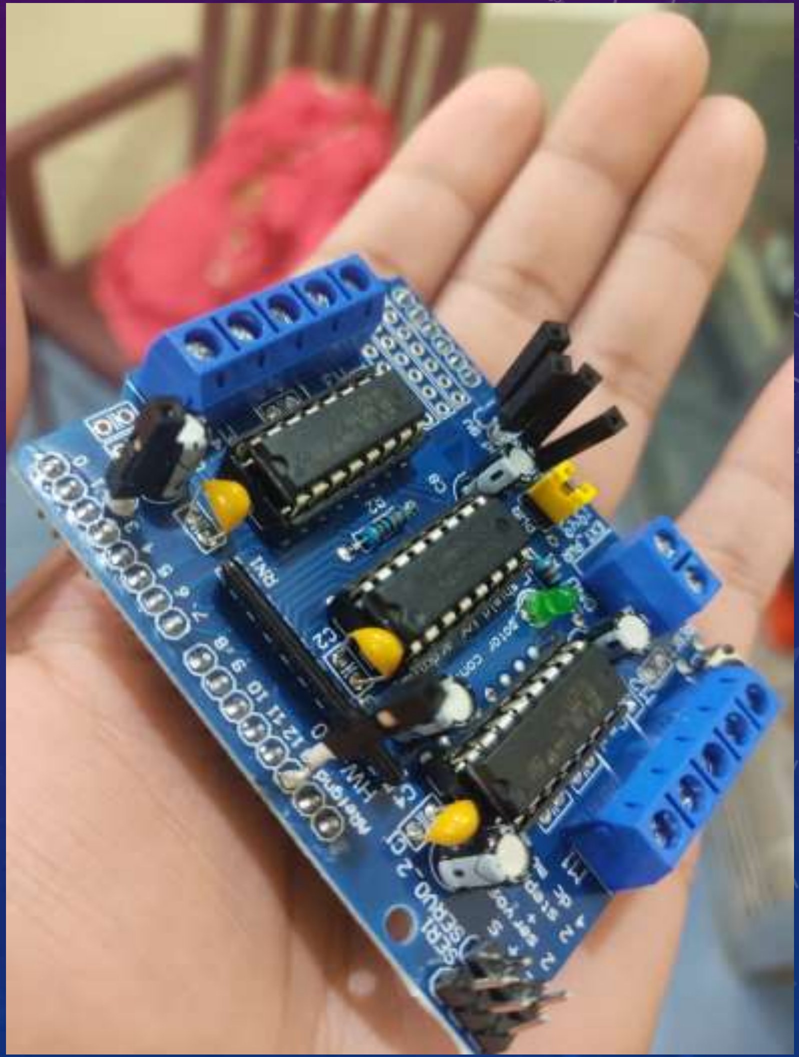


Video Presentation



Problems Faced with model:

1. Late delivery
2. A female pin had to be soldered on motor driver for connection
3. Faulty Arduino received
4. Took 10 days to ship the model to another teammate.
5. Low rotation speed of DC motor (which is insufficient to bring turning motion).

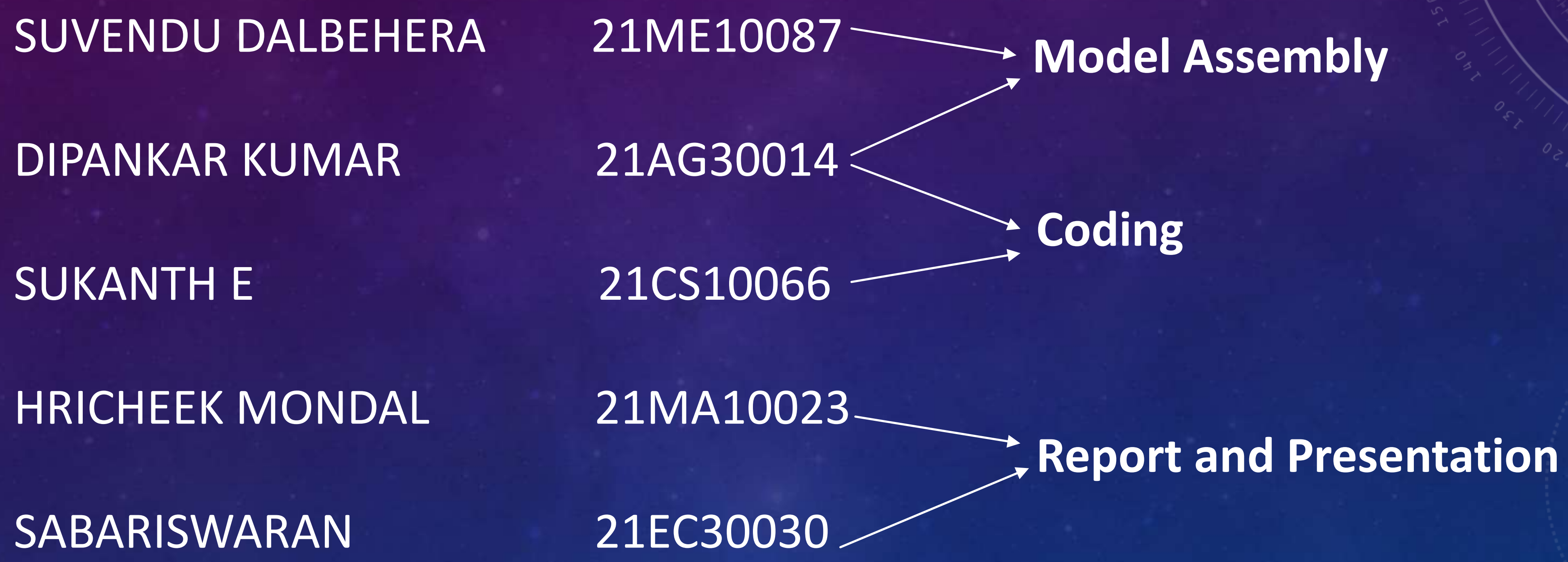


Final Appeal

1. it's been really difficult working on this hardware project solely from home
2. We tried our best. Even shipped the model from Suvendu’s house (Bargarh, Odisha) to Dipankar’s (Kanpur, Uttar Pradesh) in a hope to complete the project since he was facing difficulty in resolving the issues and uploading code to Arduino.

We will be completing our project and also adding more advancements to our project, when we’re on campus!

Work distribution:



SUVENDU DALBEHERA	21ME10087
DIPANKAR KUMAR	21AG30014
SUKANTH E	21CS10066
HRICHEEK MONDAL	21MA10023
SABARISWARAN	21EC30030

Model Assembly

Coding

Report and Presentation

Thank You!

The DIY Projects have given us great knowledge of the things required in our real life to implement our ideas.