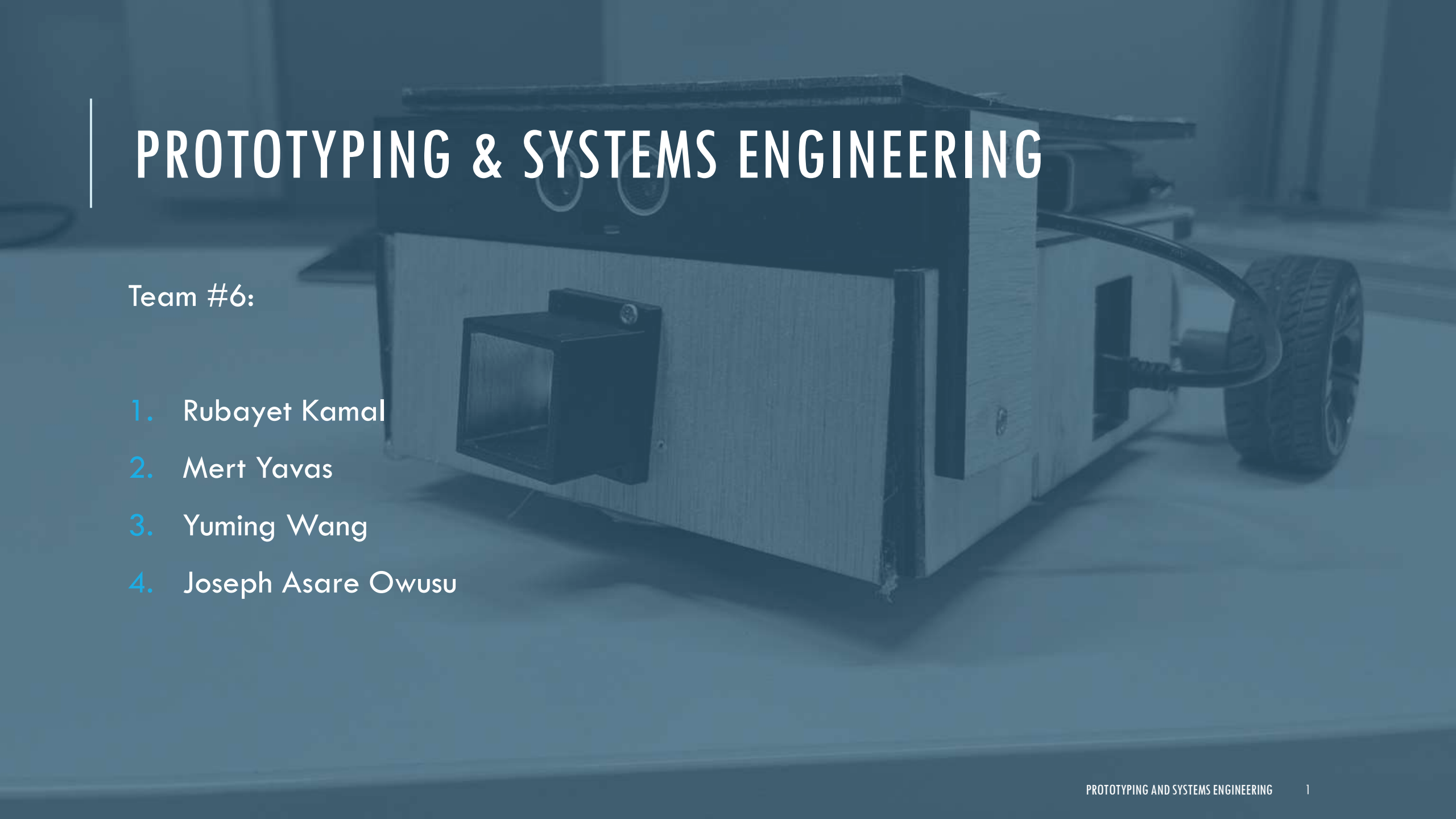
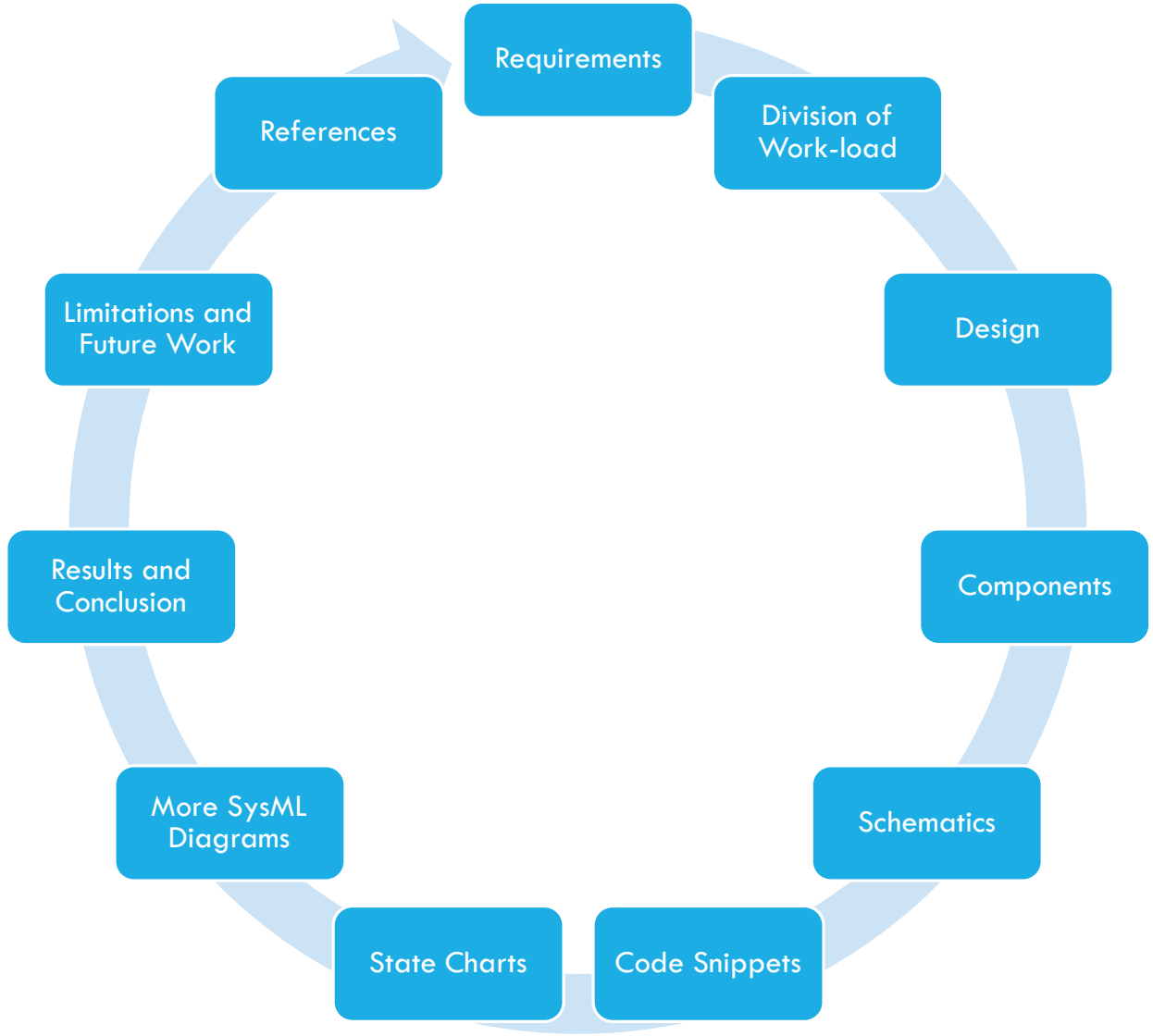
PROTOTYPING & SYSTEMS ENGINEERING

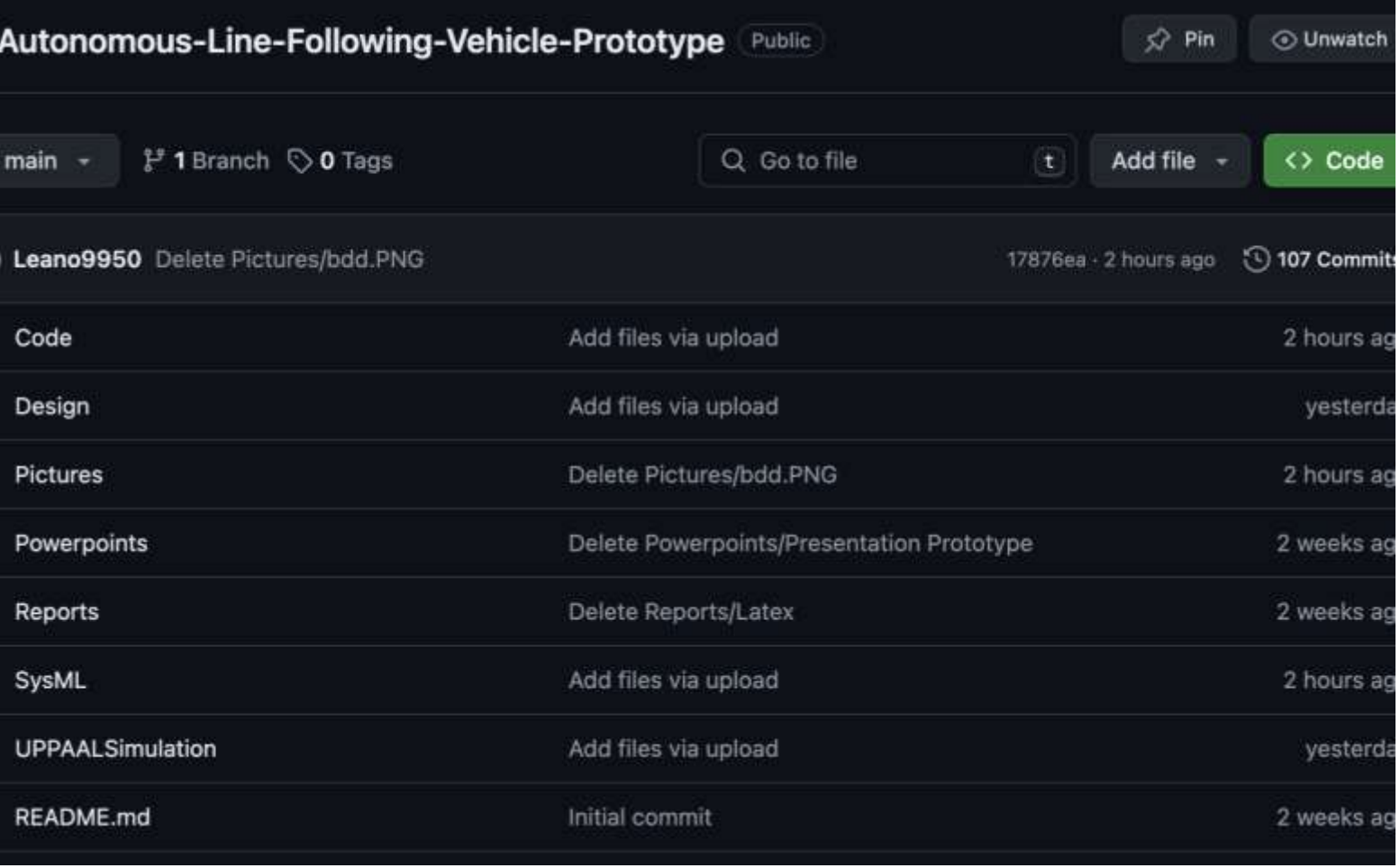


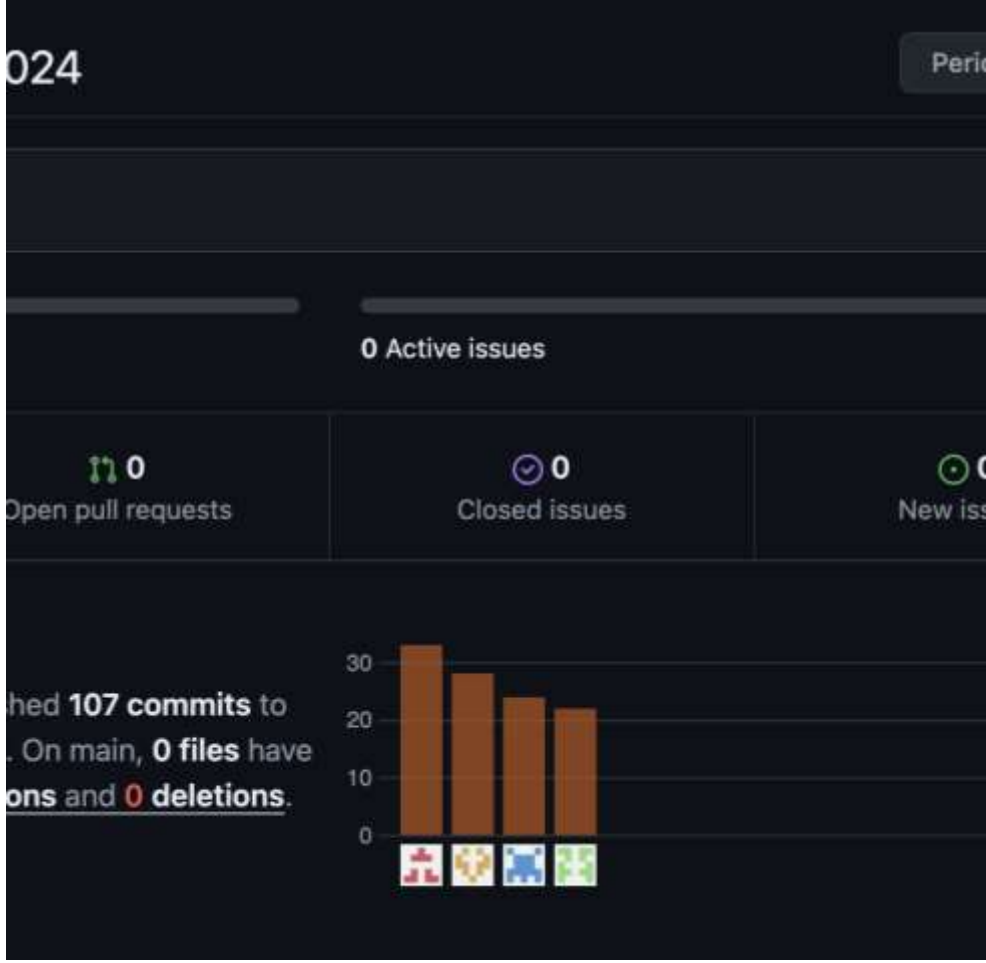
Team #6:

1. Rubayet Kamal
2. Mert Yavas
3. Yuming Wang
4. Joseph Asare Owusu

AGENDA


Requirements
Division of Work-load
Design
Components
Schematics
Code Snippets
State Charts
More SysML Diagrams
Results and Conclusion
Limitations and Future Work
References



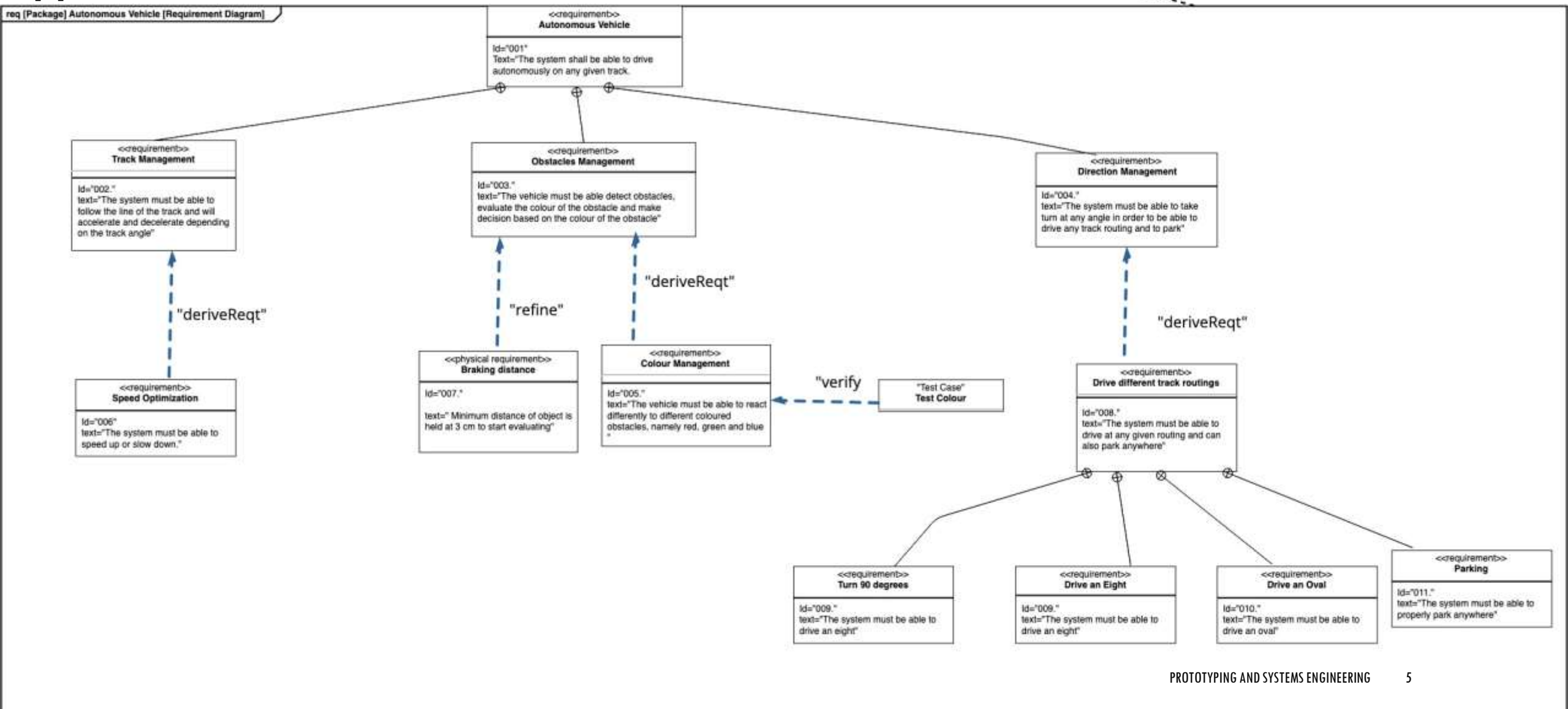


WORK-LOAD DIVISION

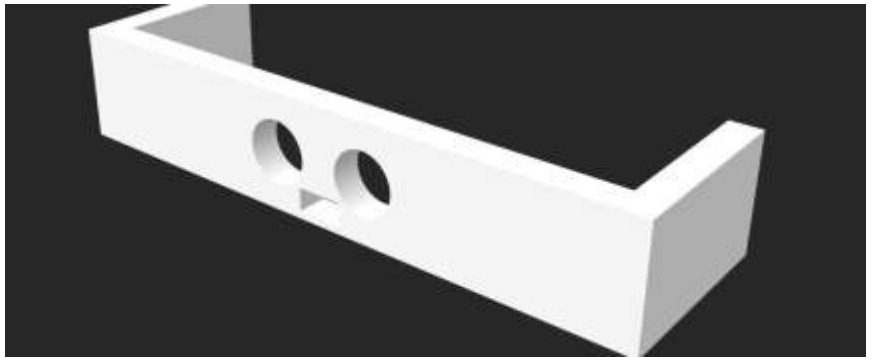
REQUIREMENTS:

Requirements
Can Follow a Line?
Can Take Turns at any angle (i.e 90 degree turn)?
Can drive different routings (i.e Oval)?
Can optimize speed ?
Can detect obstacles?
Can evaluate colour of Object?
Can take 180 degree turn?
Can overtake an obstacle?
Can park?

[1]

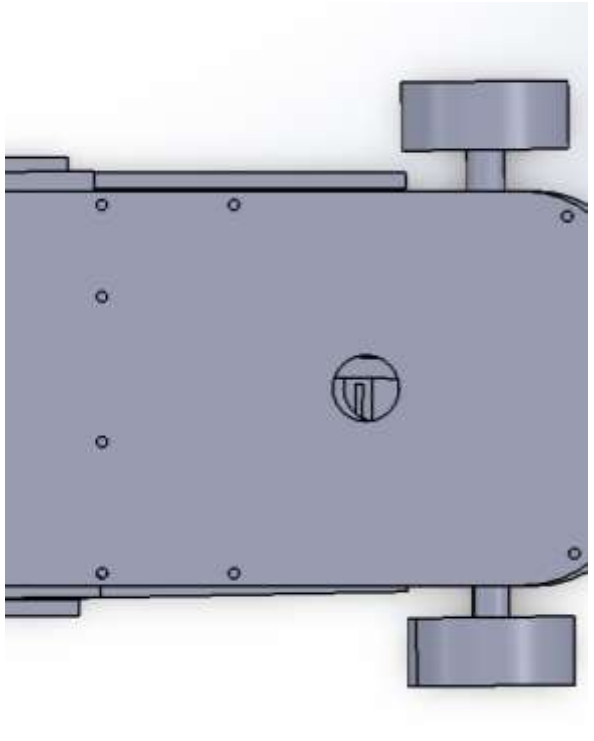


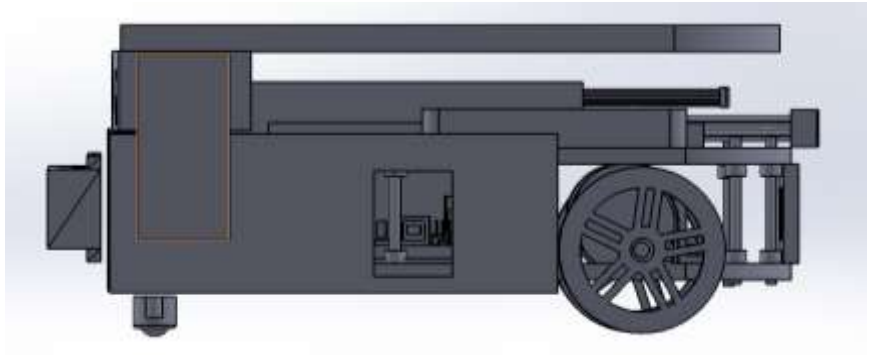
- Designed in Solidworks and Fusion 360.
- .dxf files to laser cut with wood.
- .stl to 3D print via PLA.

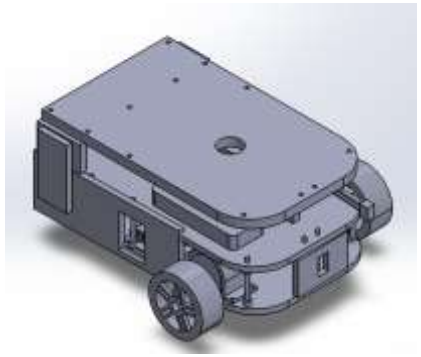


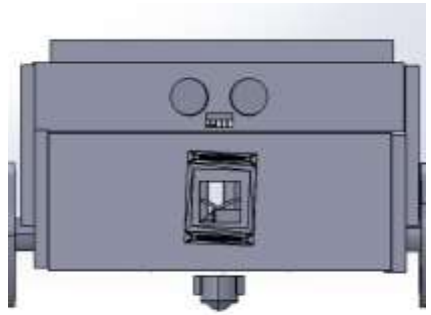
PARTS DESIGNED VIA SOLIDWORKS AND FUSION 360

- No components are glued.
- All components are screwed in.
- 100 percent modifiable.



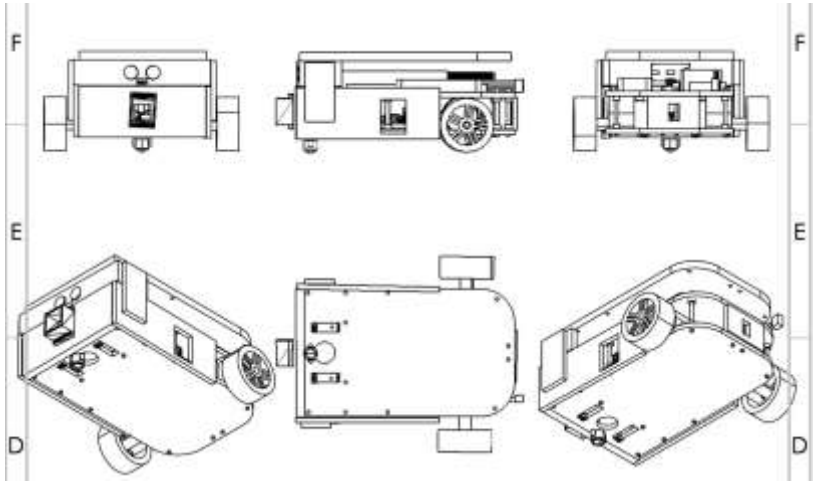




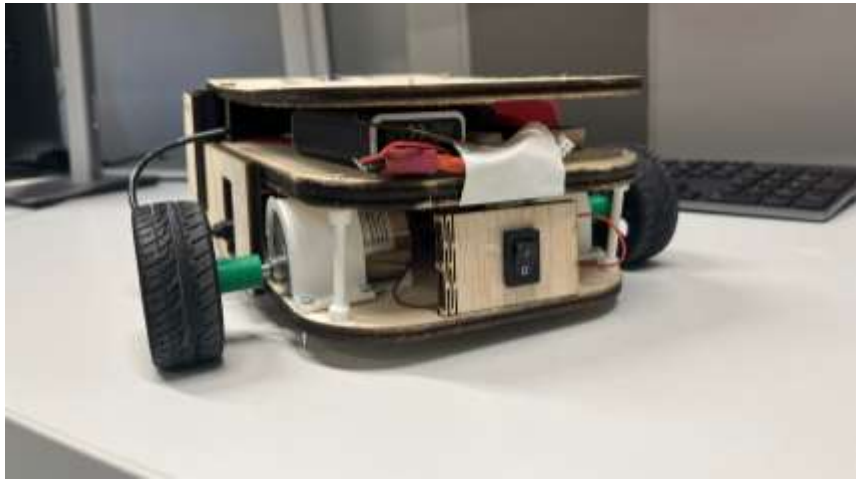


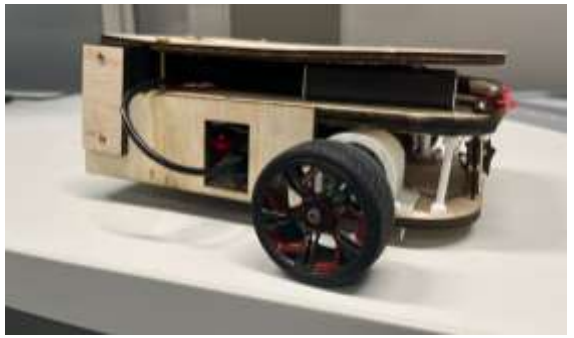
[2]

FINAL DESIGN



[2]





SENSORS:

To Detect Obstacles

To Detect Change in Line

To Detect Obstacle Color

[3]

HC SR-04 Ultrasonic Sensor
(Maximum range: 14 cm)

ST 1140 Line Sensor [5]

[4]

TCS3200 Color Sensor

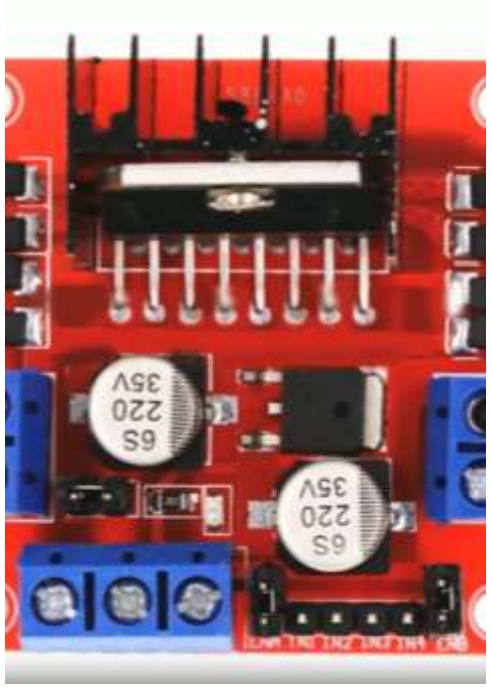
ACTUATORS:

To move the car in any direction.

[7]

- DC Motors (2x)

- L298N Motor Driver
- Input of 12V from Battery.
- Output of 5V to Arduino
- Used to control speed and direction of DC Motors.



[6]

MICROCONTROLLER AND POWER SOURCE:

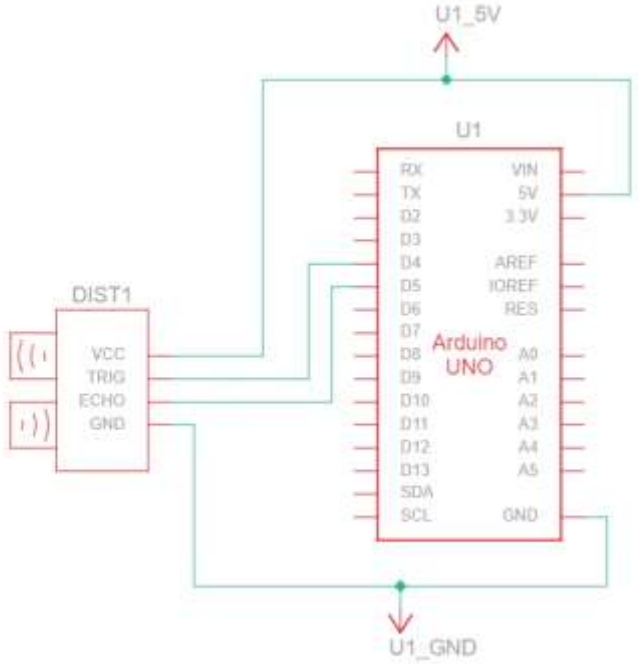
- Brain and Energy source of the car.

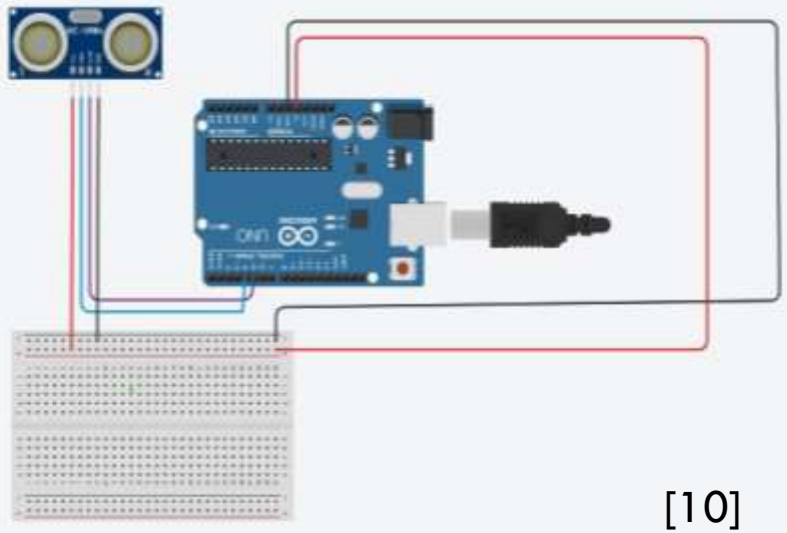
- 5V Input
- 14 Digital Pins
- 6 Analog Pins
- Used with Arduino IDE

[8]

- Technology LiPo
- Cell number 2
- Tension 7.4V
- Capacity 3000mAh
- Resilience 20C
- Weight 210g

[9]

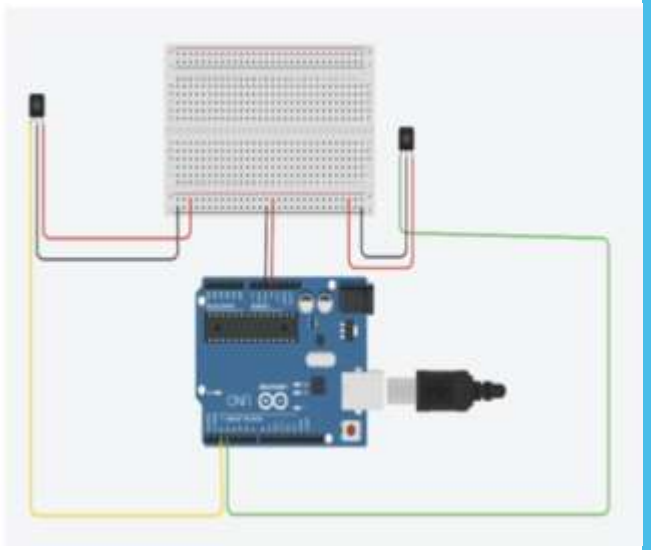


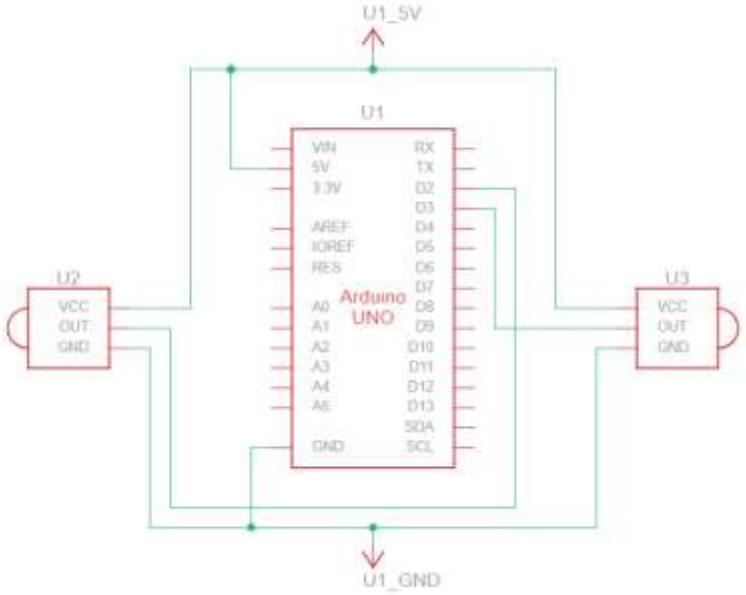


[10]

SCHEMATICAL VIEW FOR HCSR04

Pin Configuration on Ultrasonic Sensor	Pin Configuration on Arduino
Vcc	5V
TRIG	4
ECHO	5
GND	GND



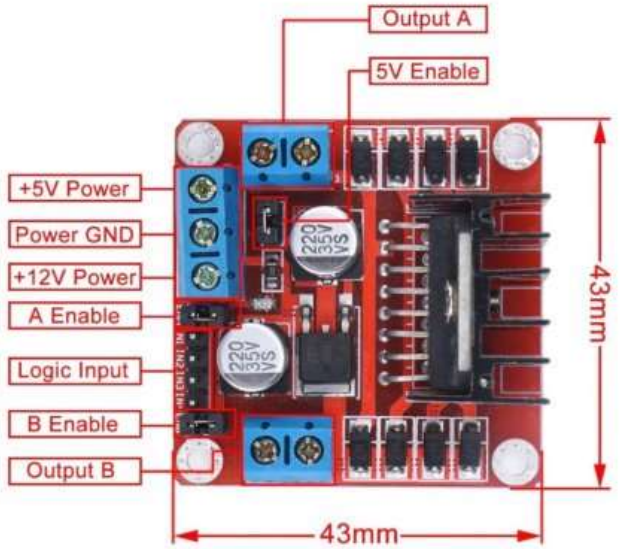


[10]

SCHEMATICAL VIEW FOR LINE DETECTOR:

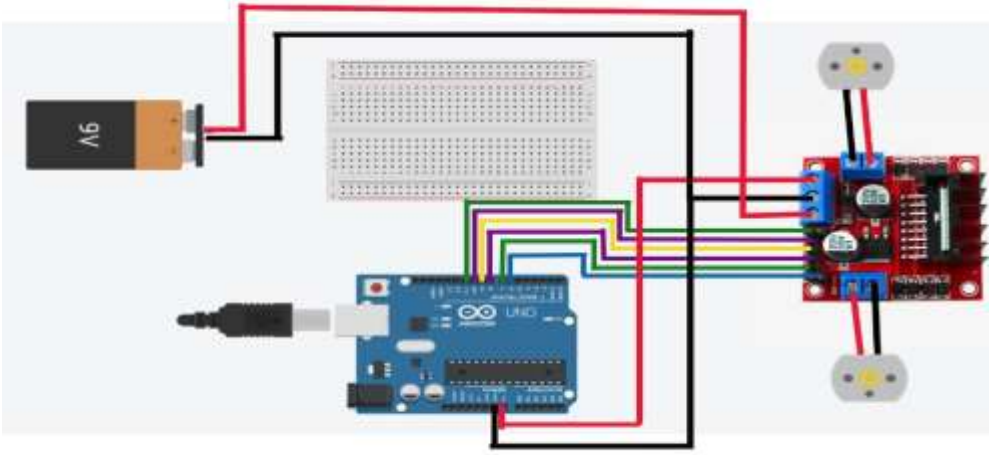
Pin on Sensor	Pin Configuration on Arduino
Vcc	5V
GND	GND
OUTPUT 1	2
OUTPUT 2	3

SCHEMATICAL VIEW OF L298N



[6]

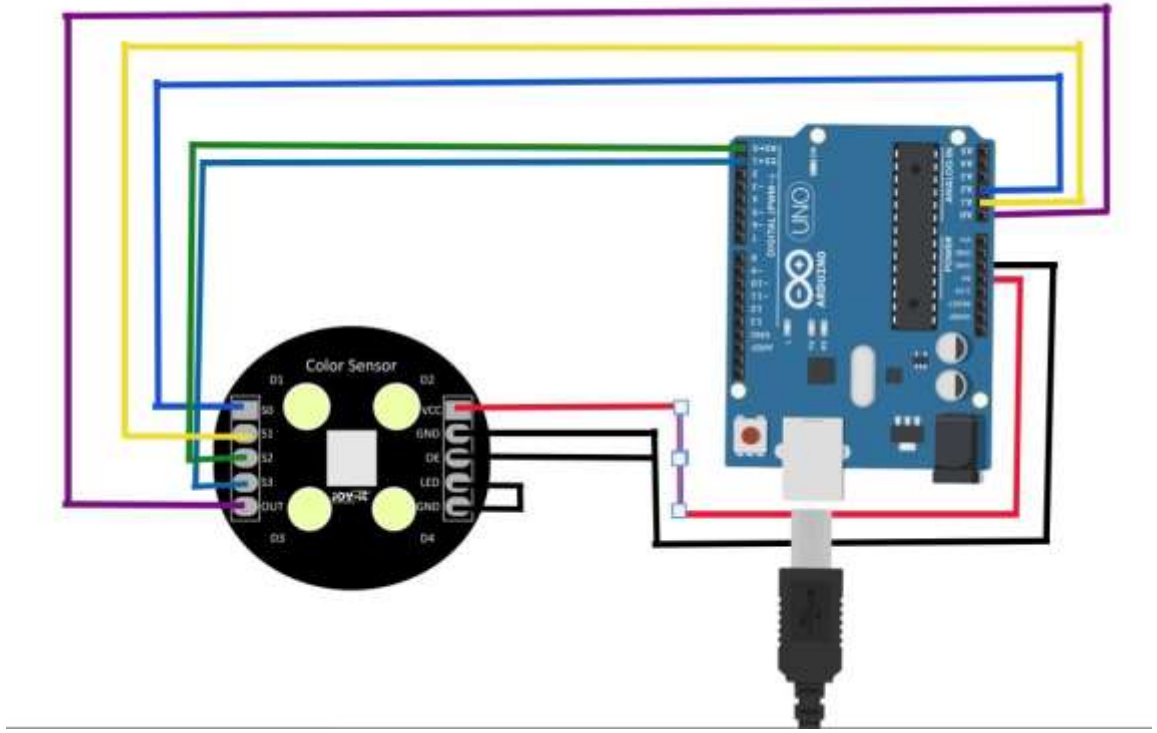
Pin on Sensor	Pin on Configuration on Arduino and components
IN1	10
IN2	9
IN3	8
IN4	7
ENABLE A	11
ENABLE B	6
OUTPUT A	MOTOR
OUTPUT B	MOTOR
5V POWER	Vcc
POWER GND	GND
12V POWER	BATTERY



[10]

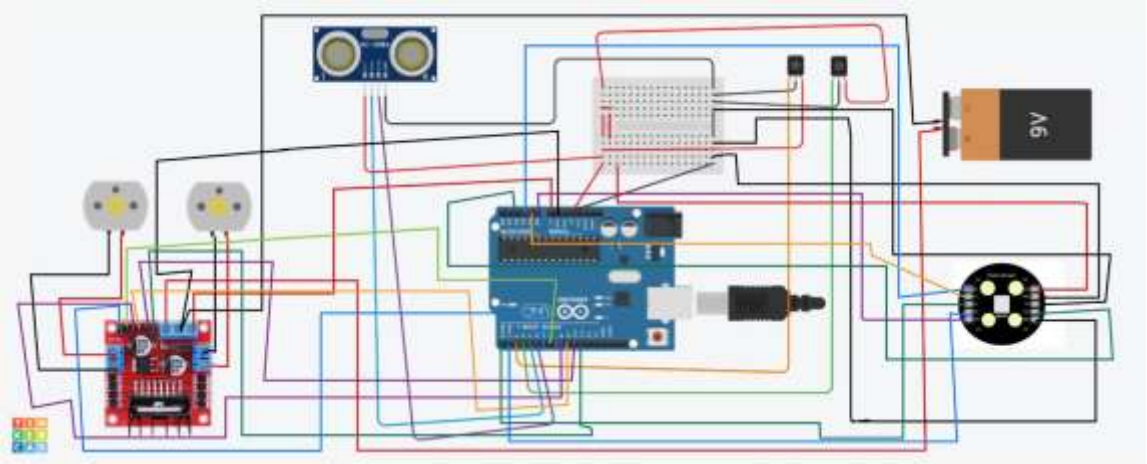
SCHEMATICAL VIEW OF TCS3200

Pin on Sensor	Pin Configuration on Arduino
S0	A2
S1	A1
S2	0
S3	1
OUT	A0
V_{cc}	5V
GND	GND
OE	GND
LED	GND
GND	LED



[10]

COMPLETE SCHEMATIC



[10]

```
267  void InfraRedManager({
268    RightInfraRedValue = digitalRead(RightInfraRed);
269    LeftInfraRedValue = digitalRead(LeftInfraRed);
270  }
271
```

INFRARED MANAGEMENT:

```
void activateTriggerPin() {
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
}
float measureDistance() {
  activateTriggerPin();
  long travelTime = pulseIn(echoPin, HIGH, 30000);
  return travelTime * 0.34 / 2;  // Distance in mm
}
```

HCSR04 MANAGEMENT

```
void motorAForward(int speedA) {
  analogWrite(motorAenable, speedA);
  digitalWrite(motorAcontrolA, LOW);
  digitalWrite(motorAcontrolB, HIGH);
}

void motorBForward(int speedB) {
  analogWrite(motorBenable, speedB);
  digitalWrite(motorBcontrolA, HIGH);
  digitalWrite(motorBcontrolB, LOW);
}

void forward(int carSpeed) {
  motorAForward(carSpeed);
  motorBForward(carSpeed);
}
```

```
153  void motorABackward(int speedA) {
154    analogWrite(motorAenable, speedA);
155    digitalWrite(motorAcontrolA, HIGH);
156    digitalWrite(motorAcontrolB, LOW);
157  }
158
159  void motorBBackward(int speedB) {
160    analogWrite(motorBenable, speedB);
161    digitalWrite(motorBcontrolA, LOW);
162    digitalWrite(motorBcontrolB, HIGH);
163  }
164  void backward(int speed) {
165    motorABackward(speed);
166    motorBBackward(speed);
167  }
```

L298N MANAGEMENT

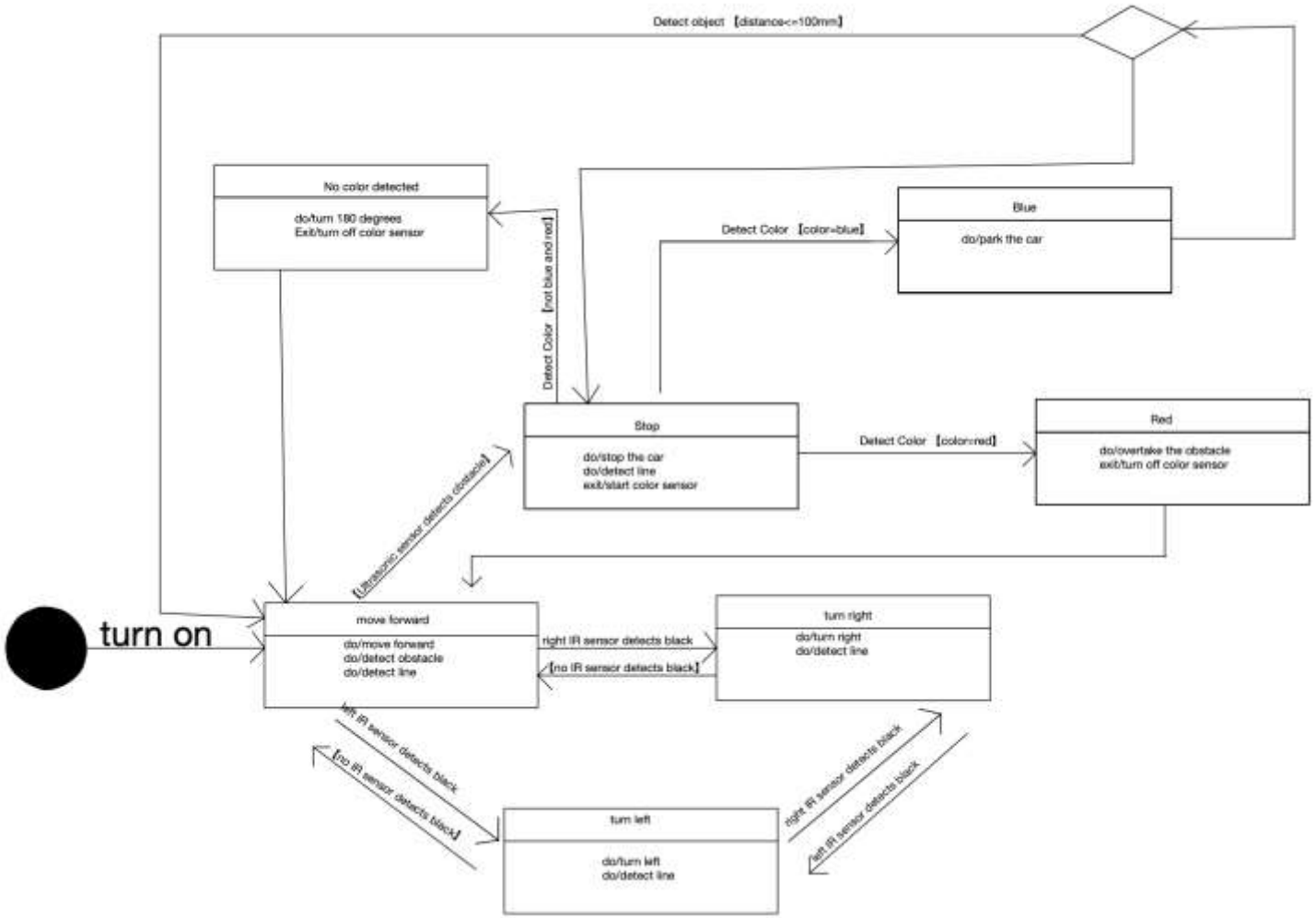
```
if (redFlag) {

  backward(baseSpeed);
  delay(500);

  stop();
  delay(10);

  overtake();
  delay(10);

}
```

```
else if (blueFlag) {
  backward(baseSpeed
  delay(500);

  stop();
  delay(10);

  parkCar();
  delay(10);
}
```

```
else {
  backward(baseSpeed
  delay(500);

  stop();
  delay(10);

  turn180();
  delay(10);
}
```

COLOUR EVALUATION MANAGEMENT

State Machine Diagram



```
enum Direction { Straight,
                 Left,
                 Right };
Direction currentDirection = Straight;

enum Colour { Red,
              Blue,
              Unknown,
              NoObject };
Colour colourIdentified = NoObject;

enum State { Moving,
             Stopped,
             Idle };
State currentState = Moving;

enum LineType {
  WhiteLine,
  BlackLine
};
LineType lineType = BlackLine;
```

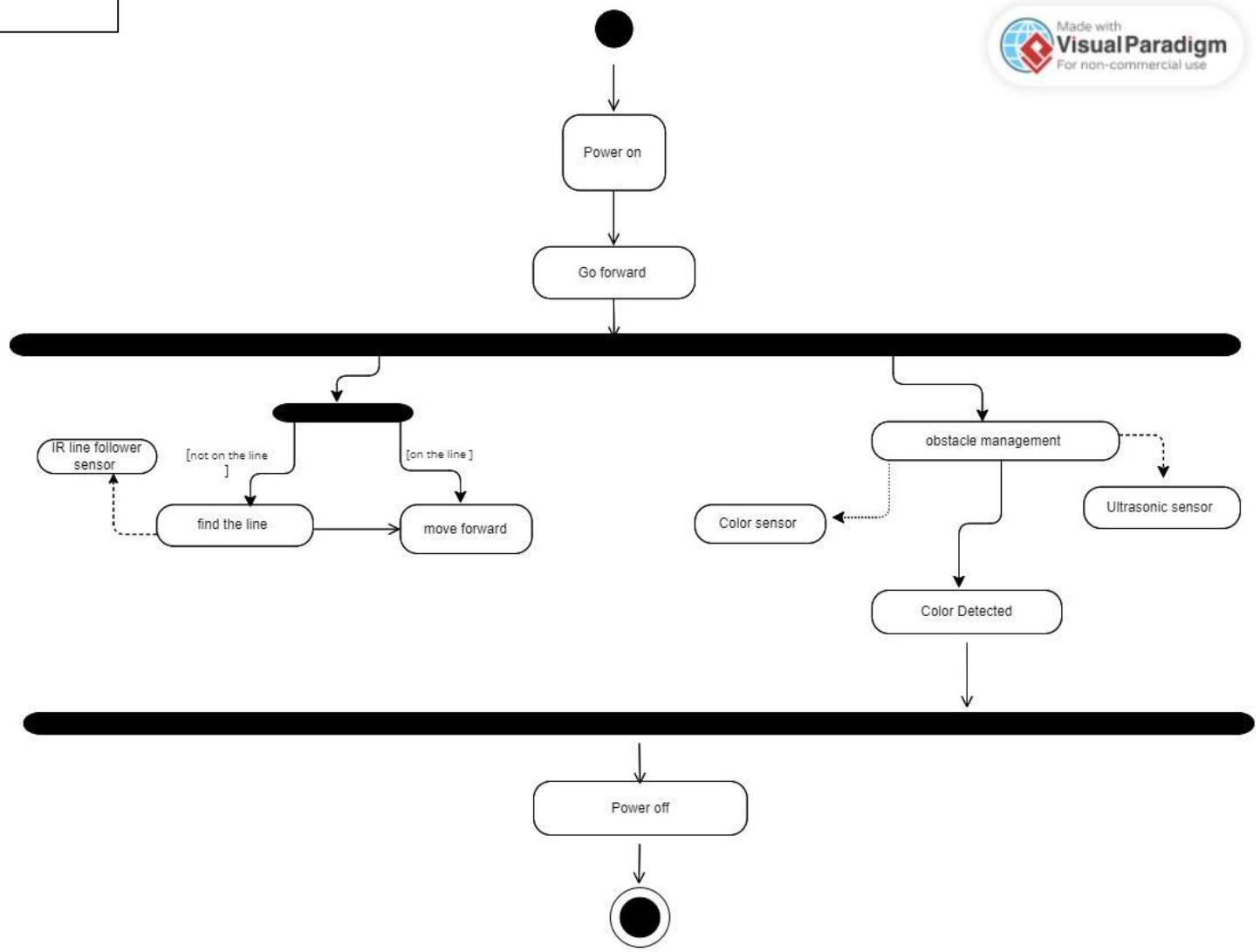
```
case Idle:
  {

    do {
      distanceFromObject = measureDistance();
      Serial.println(distanceFromObject);
    } while (distanceFromObject >= 100);

    turnRight45();

    currentState = Moving;
    break;
  }
```

```
case Stopped:
  {
    stop();
    delay(2000);

    evaluatingColour();
    evaluatingColour();
    evaluatingColour();

    if (redFlag) {
      colourIdentified = Red;
    } else if (blueFlag) {
      colourIdentified = Blue;
    } else {
      colourIdentified = Unknown;
    }
```

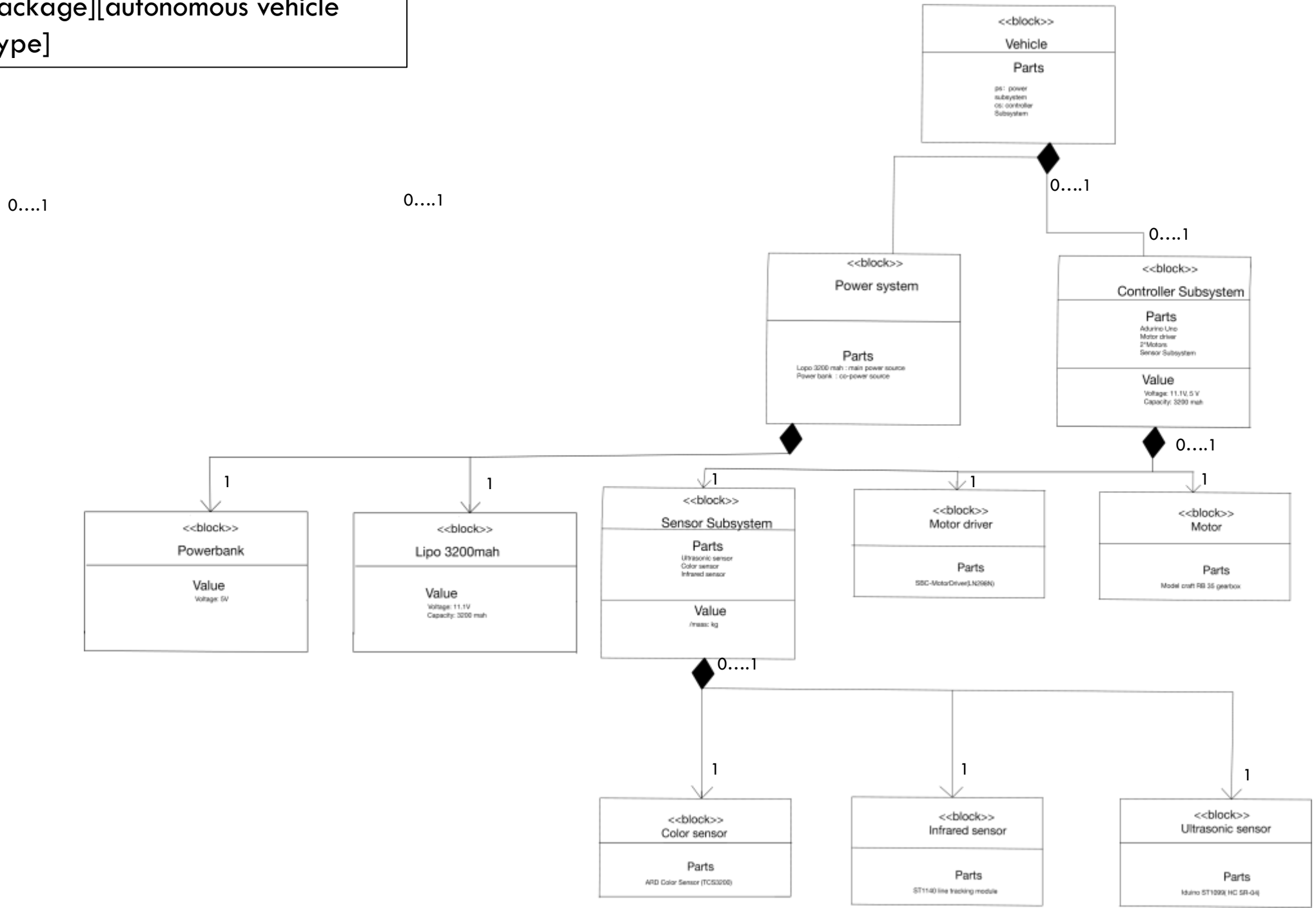
STATE MACHINE CODE USING SWITCH CASE

Activity Diagram

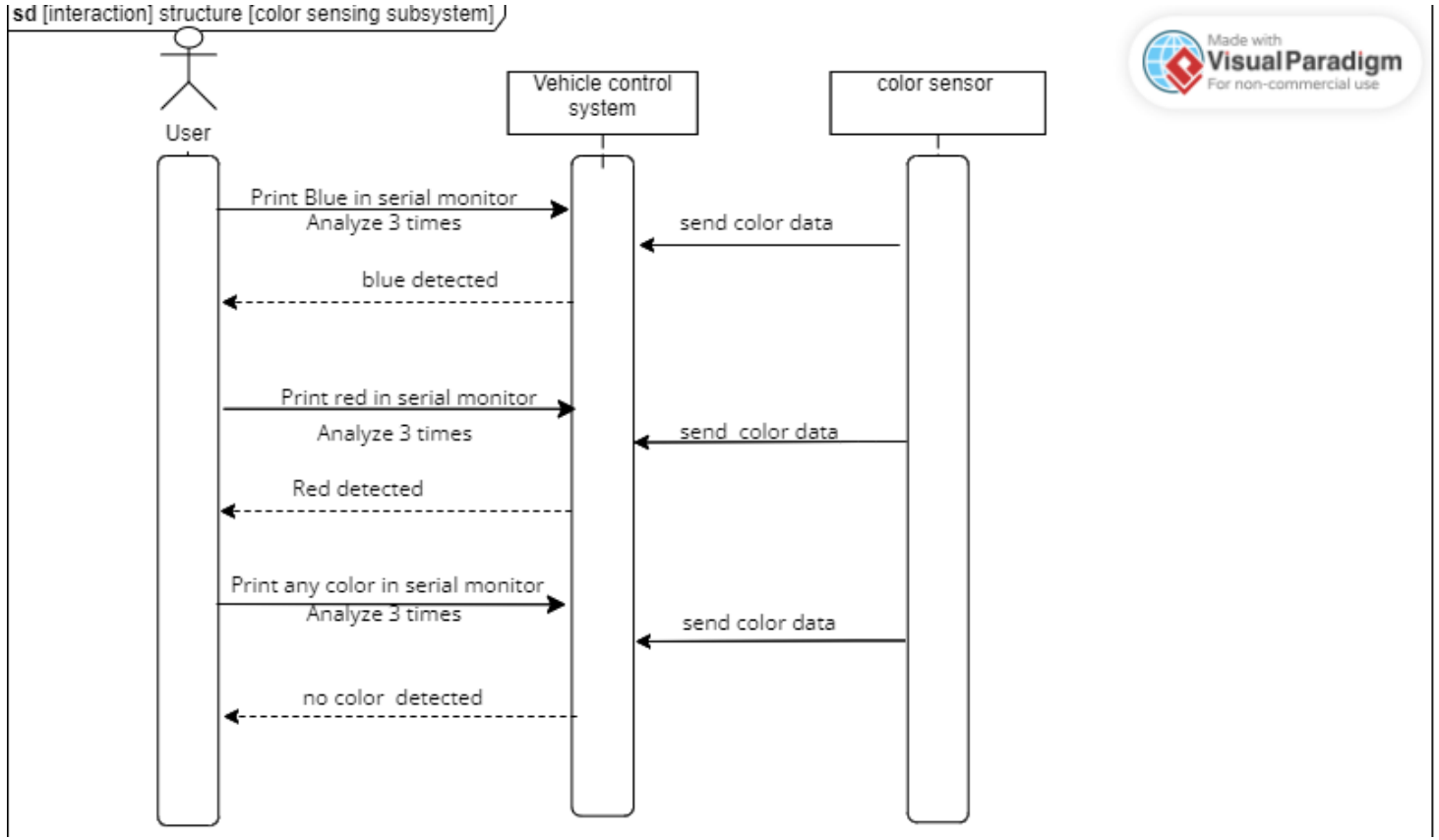
Operate Vehicle



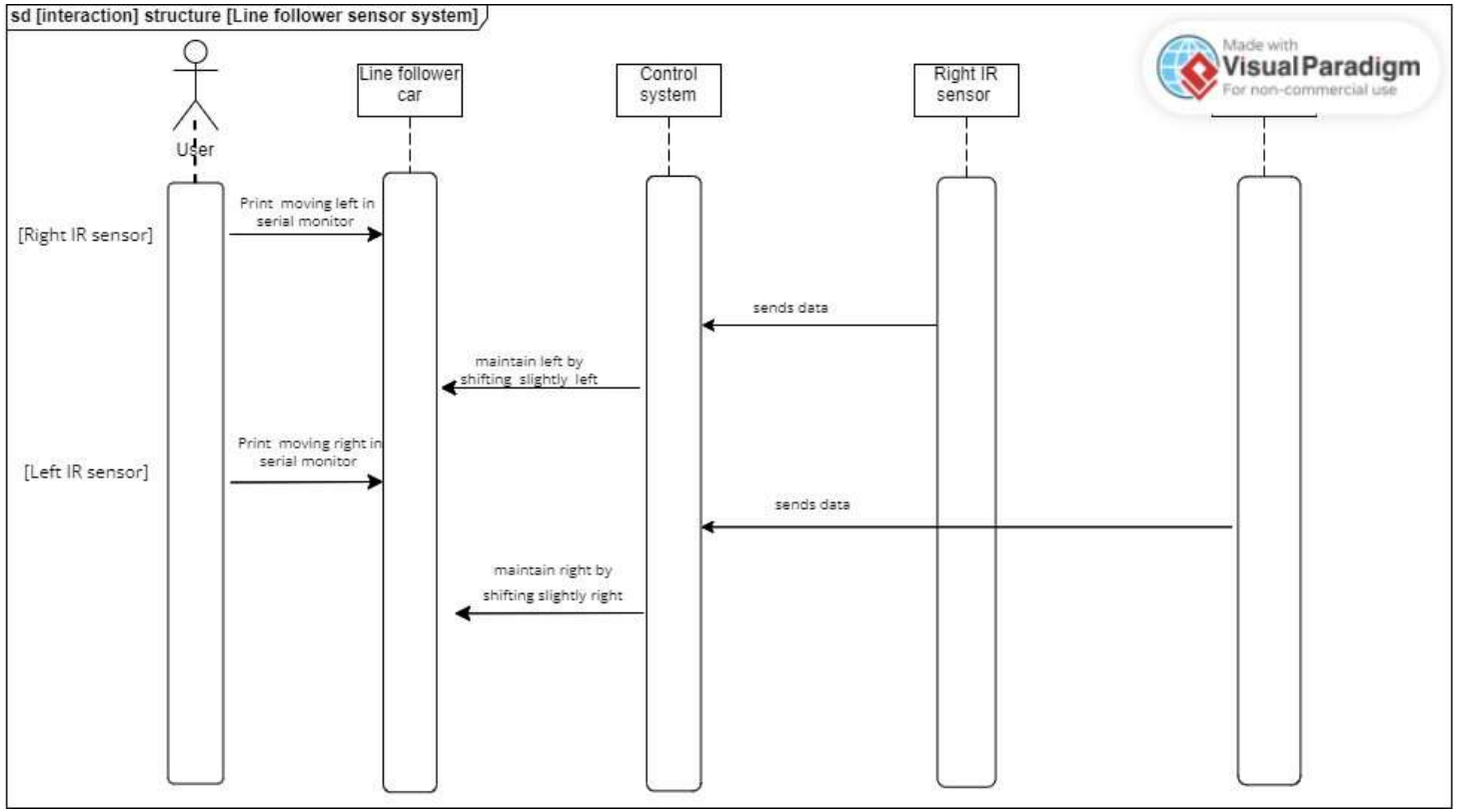
bdd[package][autonomous vehicle prototype]



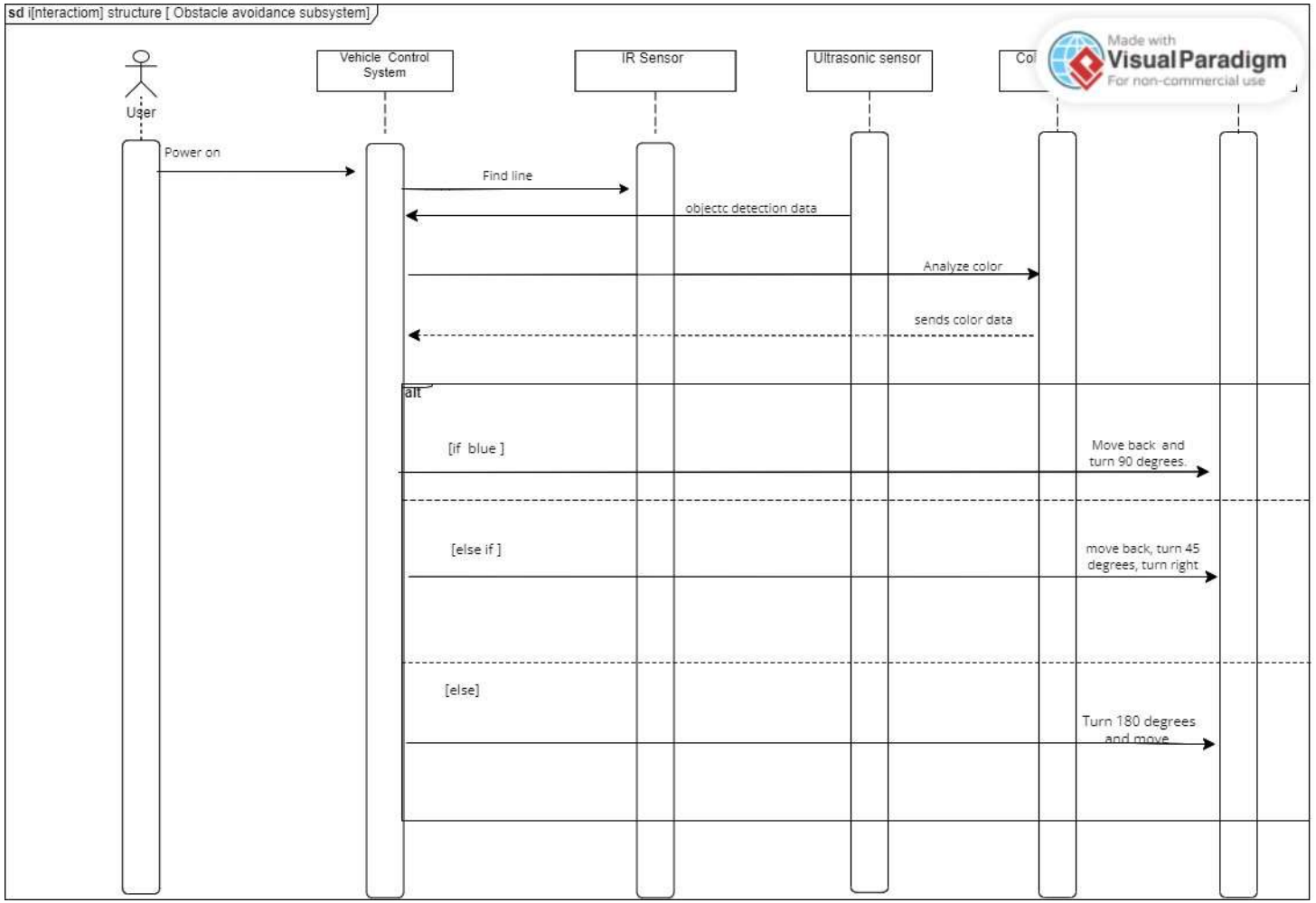
INTERACTION STRUCTURE FOR COLOUR EVALUATION



INTERACTION STRUCTURE FOR FOLLOWING LINE

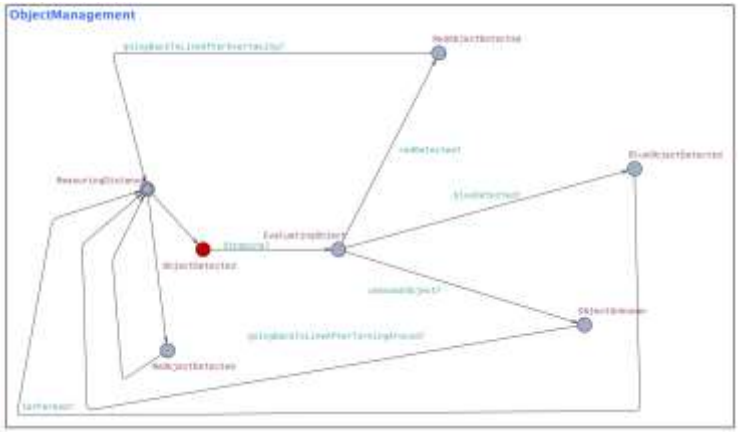


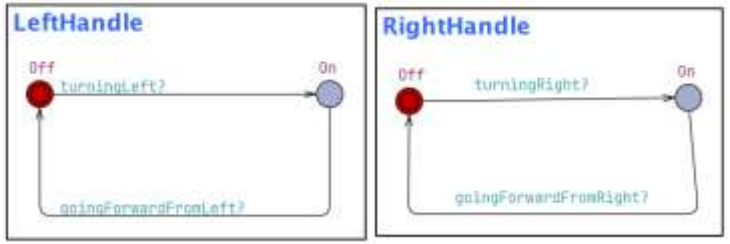
INTERACTION STRUCTURE FOR OBSTACLE MANAGEMENT

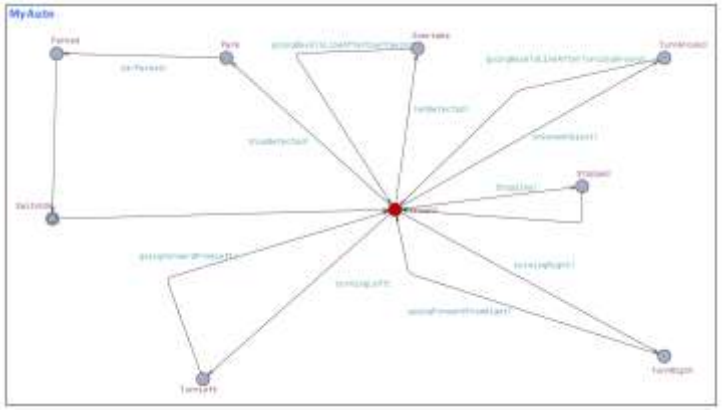


RESULTS AND ACCOMPLISHMENT

Requirements	Result
Can Follow a Line?	✓
Can Take Turns at any angle (i.e 90 degree turn)?	✓
Can drive different routings (i.e Oval)?	✓
Can optimize speed ?	✓
Can detect obstacles?	✓
Can evaluate colour of Object?	✓
Can take 180 degree turn?	✓
Can overtake an obstacle?	✓
Can park?	✓

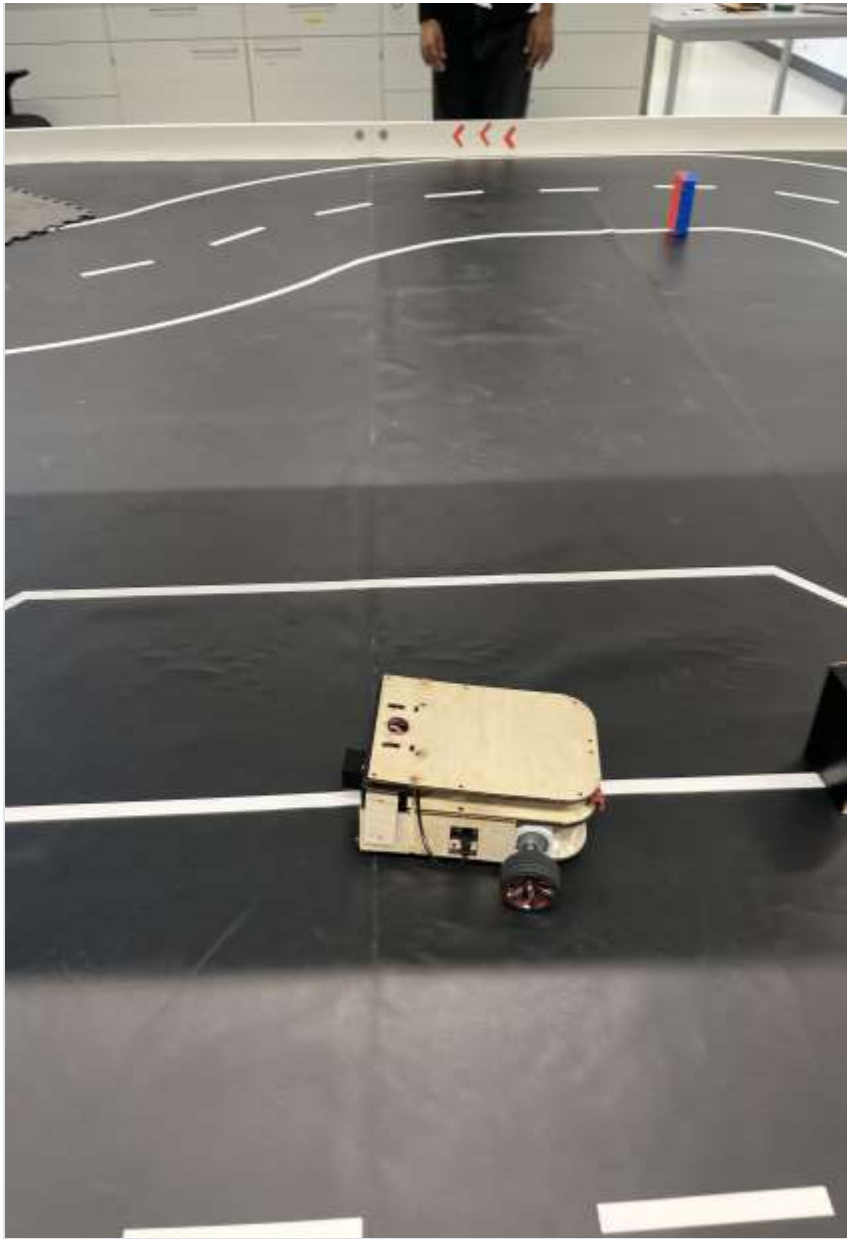




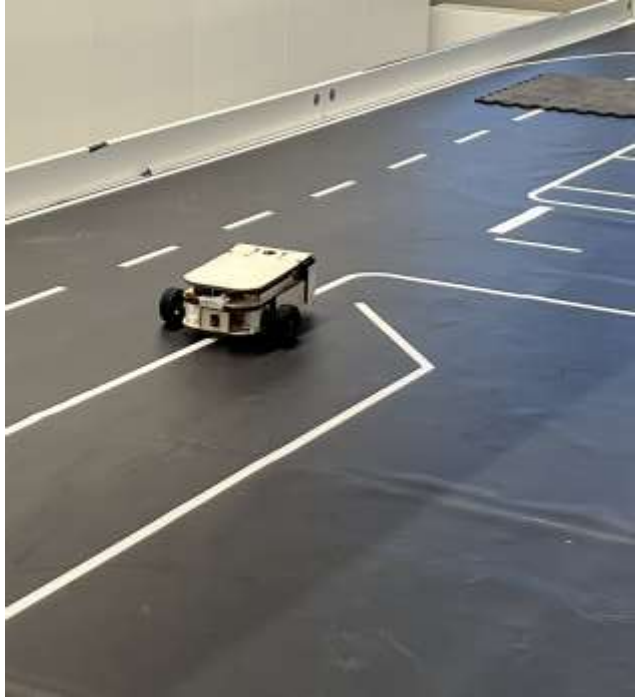


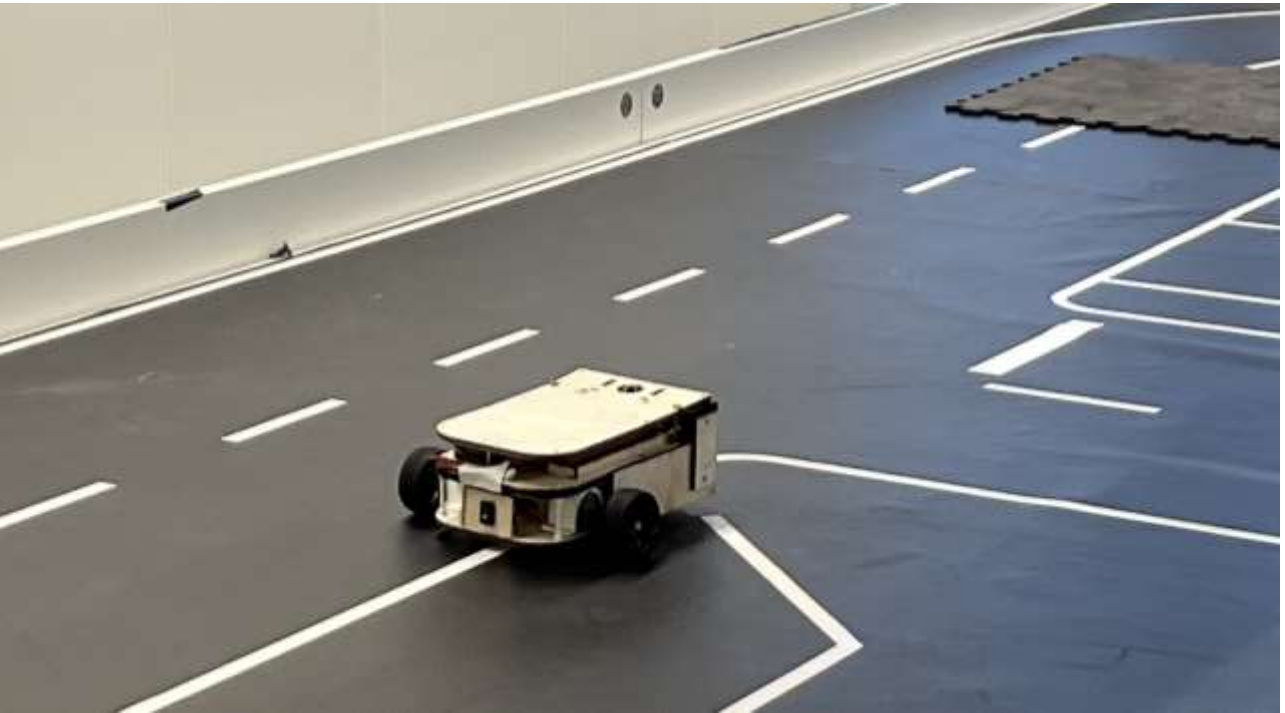
SIMULATION:

[1]



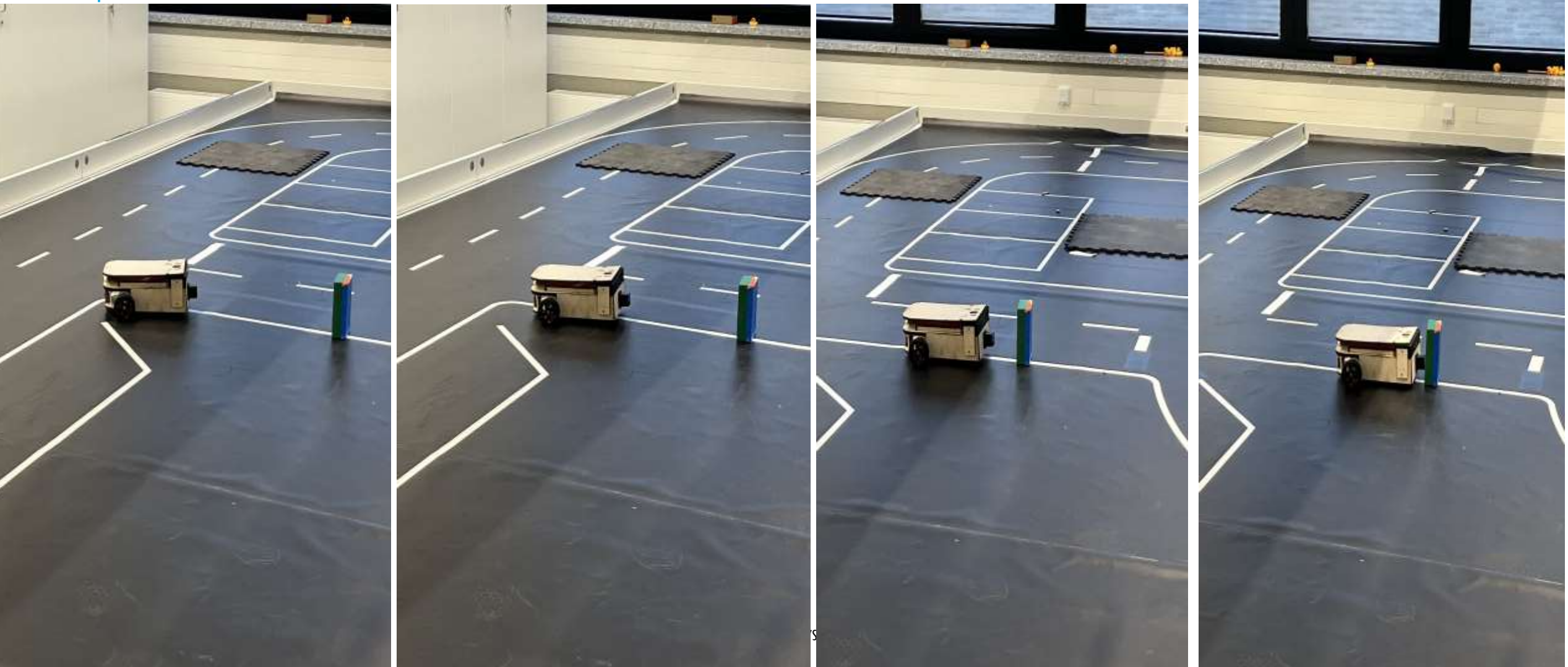
CAR FOLLOWING LINE:



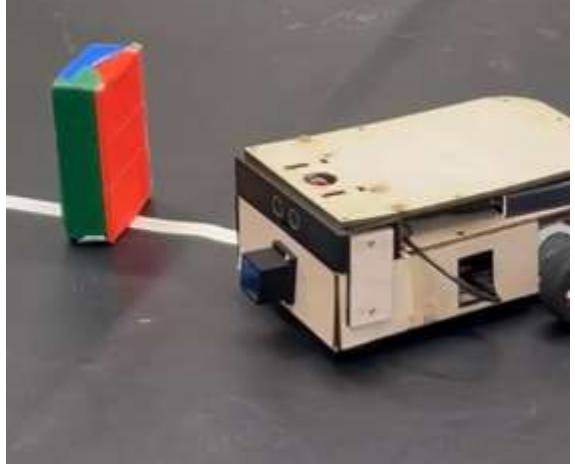
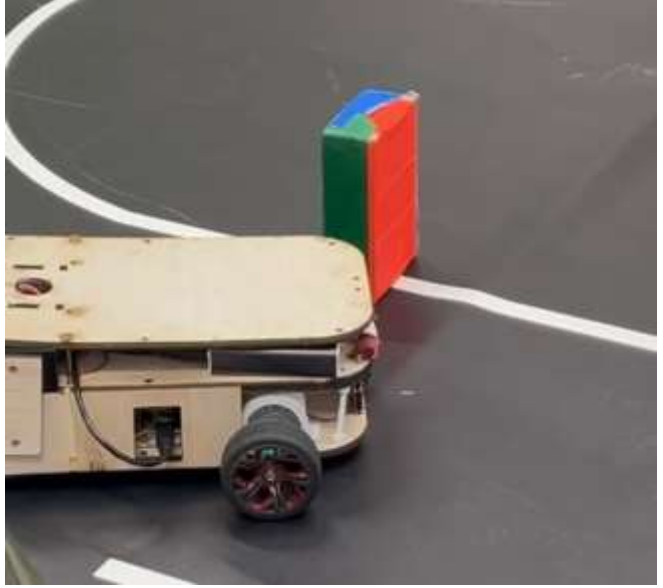


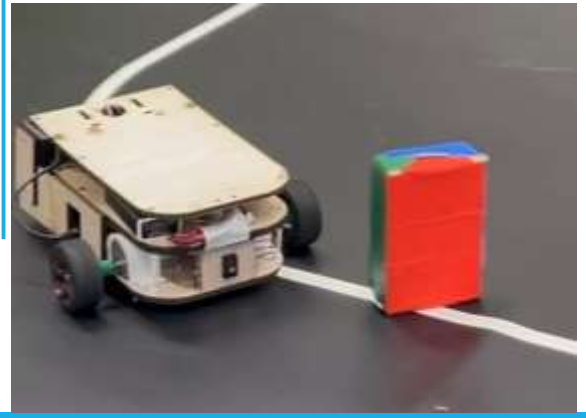
CAR TAKING TURN AT 90 DEGREES:

CAR DETECTING OBSTACLE:



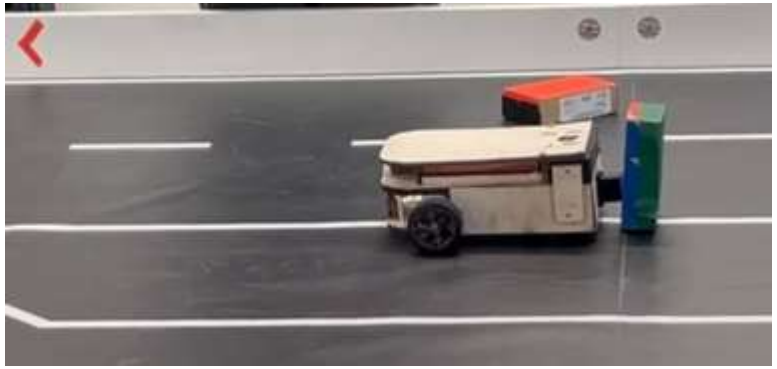


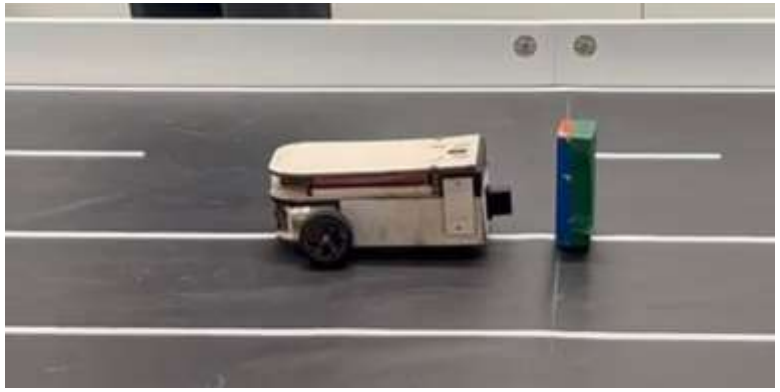



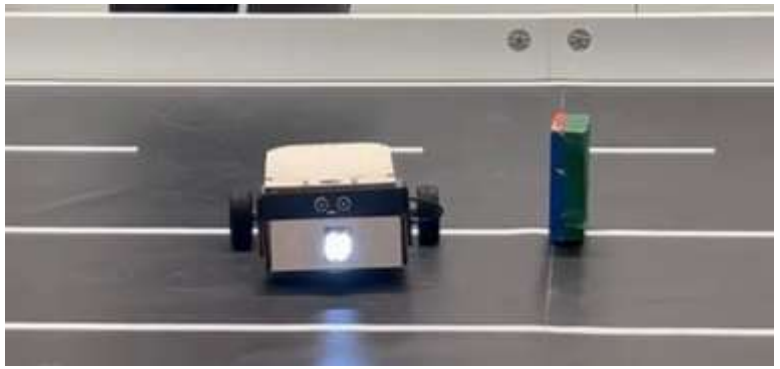


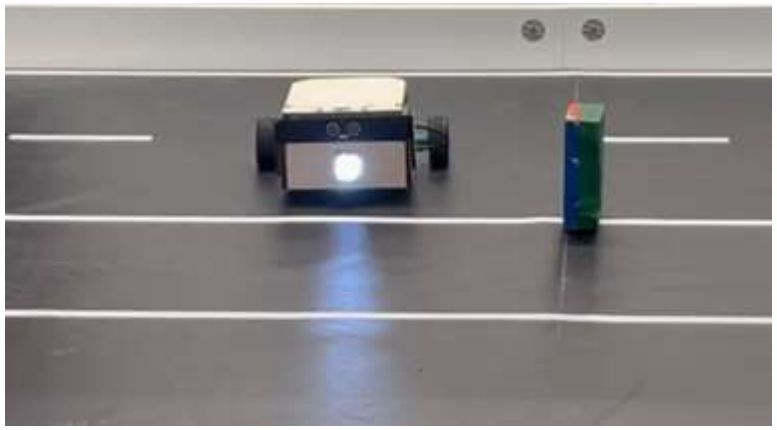
OVERTAKING OBJECT

PARKING THE CAR

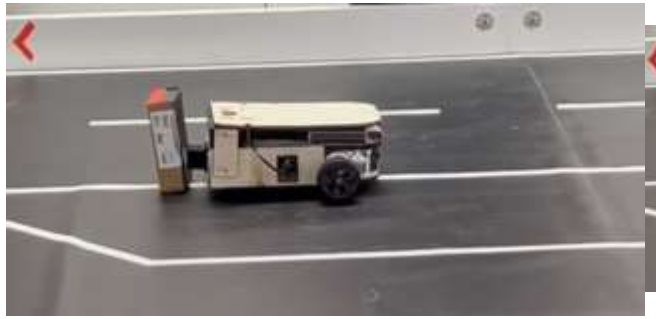
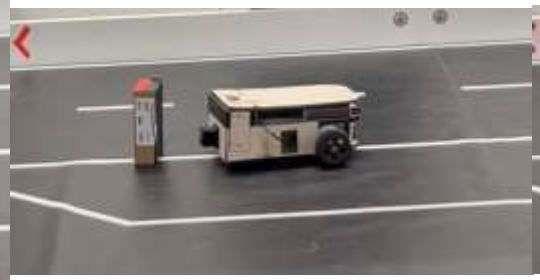
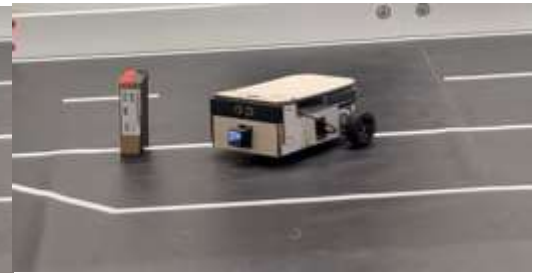
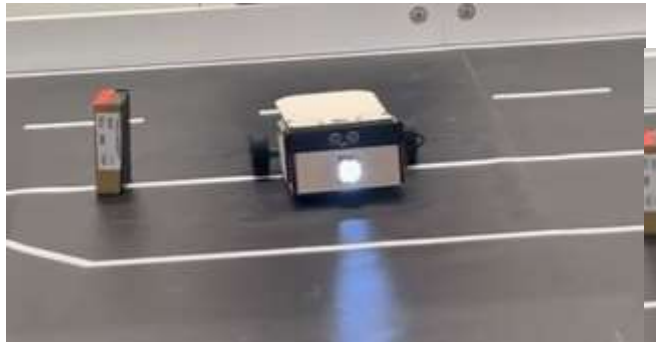








TURNING 180

FUTURE WORK:

4 wheeled autonomous vehicle prototype.

Using Servo front and back to move the ultrasonic so that it can scan surroundings.

Will probably need to switch to a different microcontroller as digital pins in Arduino are limited.

Maybe use with a Real-Time Operating System for better response.

[12]

REFERENCES:

[1] “Uppaal,” UPPAAL, https://uppaal.org/ (accessed Jun. 20, 2024).

[2] “3D CAD Design Software,” SOLIDWORKS, https://www.solidworks.com/ (accessed Jun. 20, 2024).

[3] “Ultrasonic sensor,” Joy, https://joy-it.net/files/files/Produkte/SEN-US01/SEN-US01_Manual_2024-04-15.pdf (accessed Jun. 19, 2024).

[4] “Colorimeter,” https://joy-it.net, https://www.joy-it.net/files/files/Produkte/SEN-Color/SEN-Color_Manual_2021-08-18.pdf (accessed Jun. 19, 2024).

[5] “KY-033 linetracking sensor,” SensorKit, https://sensorkit.joy-it.net/en/sensors/ky-033 (accessed Jun. 20, 2024).

[6] “L298N motor driver module,” Components101, https://components101.com/modules/l293n-motor-driver-module (accessed Jun. 20, 2024).

[7] DC Motor, https://www.handsontec.com/dataspecs/775-Motor.pdf (accessed Jun. 21, 2024).

[8] T. A. Team, “Arduino hardware,” Arduino, https://www.arduino.cc/en/hardware (accessed Jun. 20, 2024).

[9] “Buy Conrad Energy Scale Model Battery Pack (LIPO) 14.8 V 5500 mah no. of cells: 4 20 C Softcase XT90,” Conrad Electronic, https://www.conrad.com/en/p/conrad-energy-scale-model-battery-pack-lipo-14-8-v-5500-mah-no-of-cells-4-20-c-softcase-xt90-1344151.html (accessed Jun. 20, 2024).

[10] Tinkercad, https://www.tinkercad.com/dashboard (accessed Jun. 20, 2024).

[11] “Online productivity suite,” Visual Paradigm - Online Productivity Suite, https://online.visual-paradigm.com/ (accessed Jun. 20, 2024).

[12] Explore 685+ free racing cars illustrations: Download now - pixabay, https://pixabay.com/illustrations/ai-generated-mouse-racing-car-car-8674235/ (accessed Jun. 20, 2024).