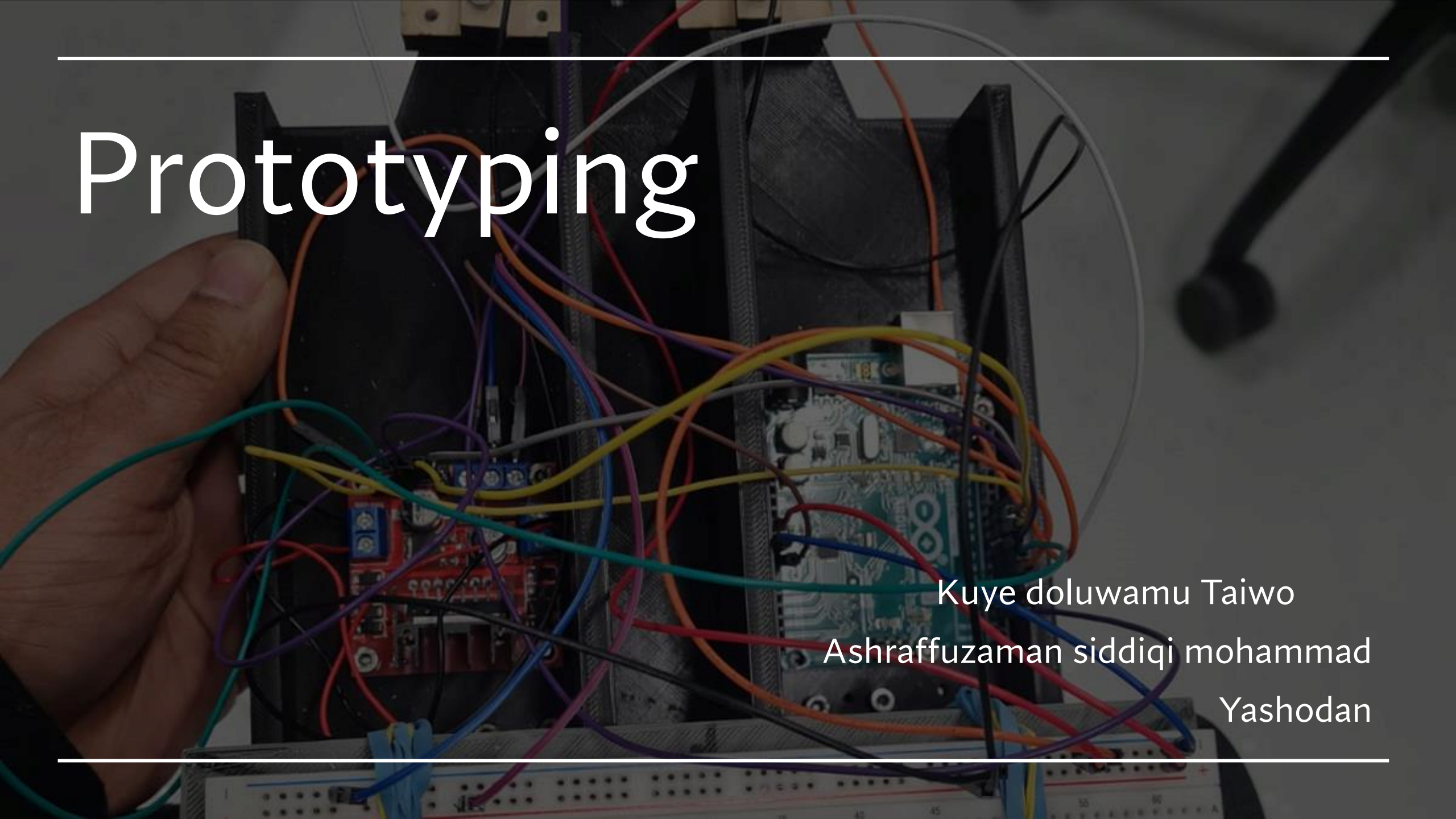
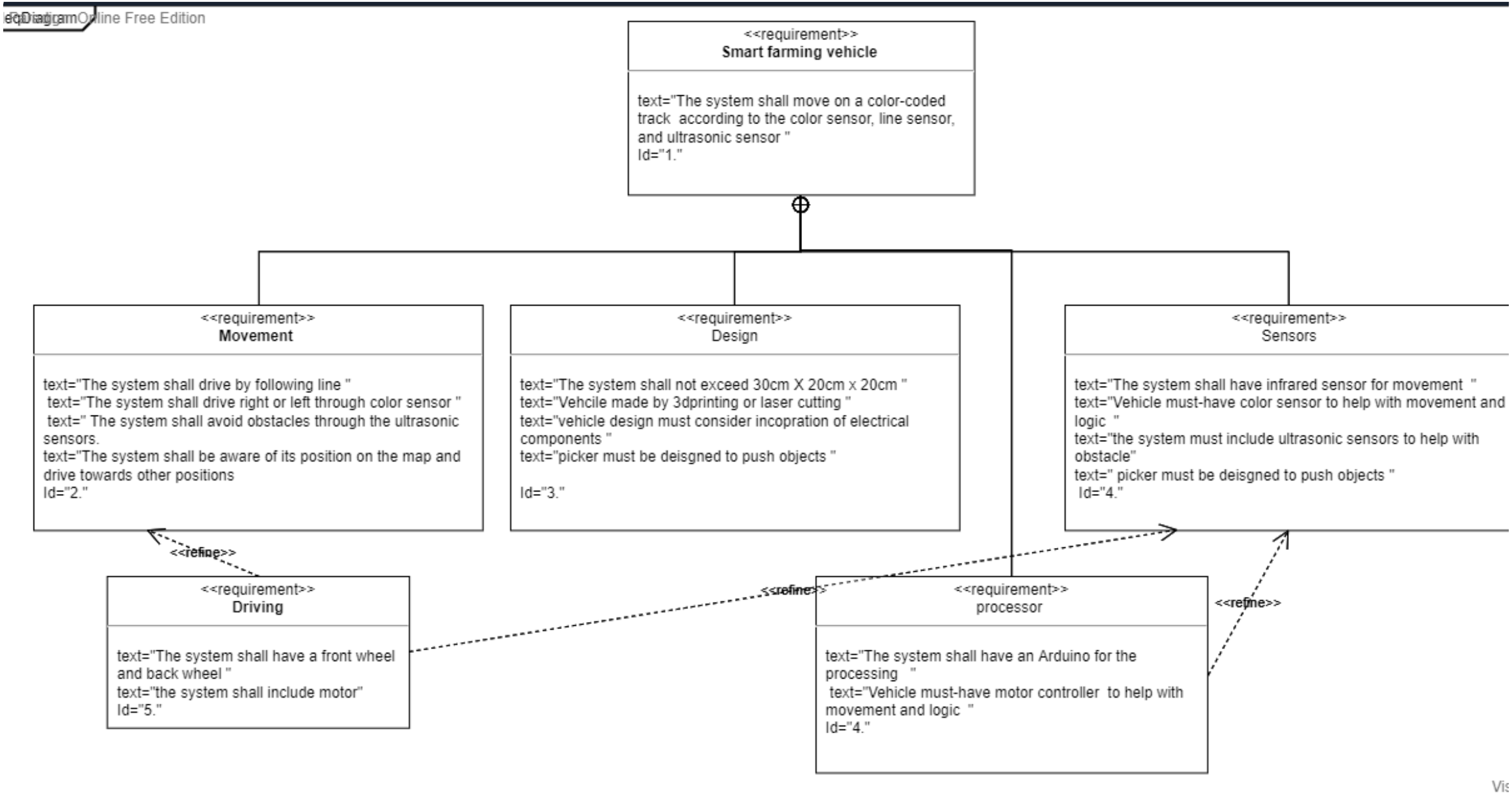
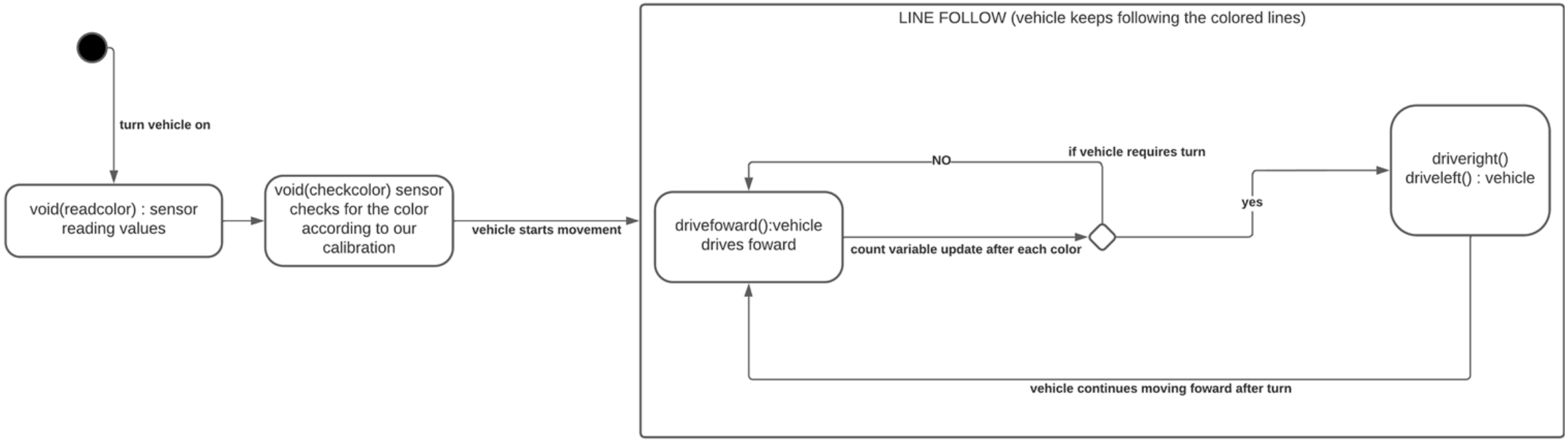


Prototyping

Kuye doluwamu Taiwo
Ashraffuzaman siddiqi mohammad
Yashodan

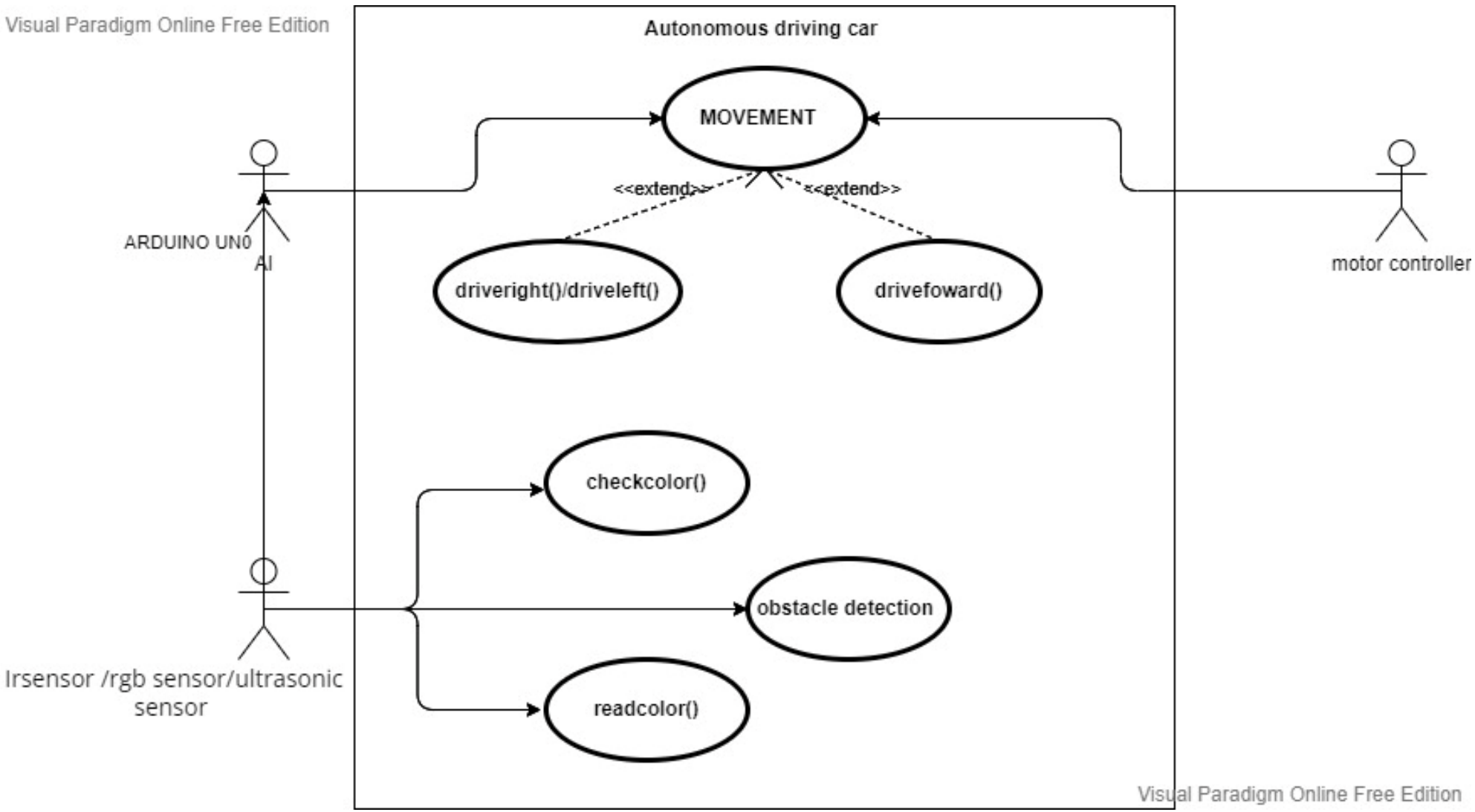
Systems Requirement



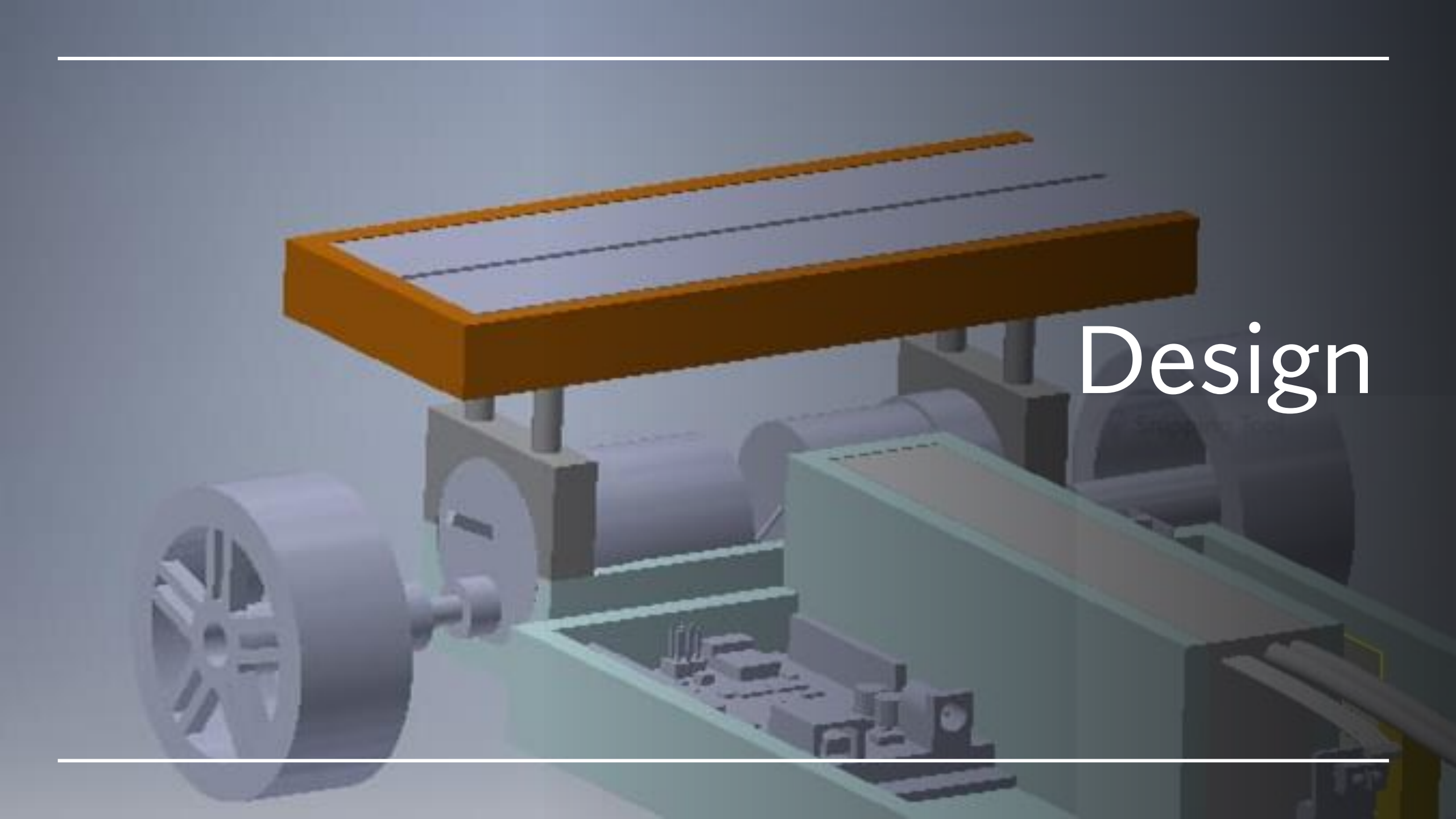


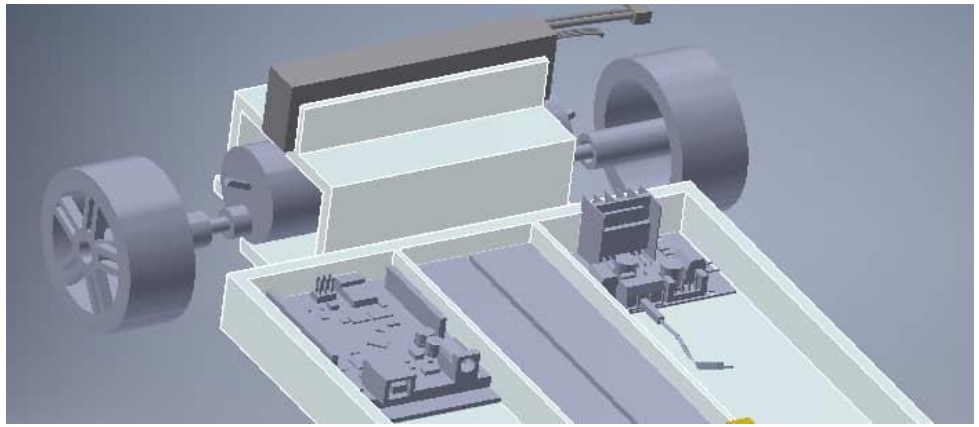
State machine

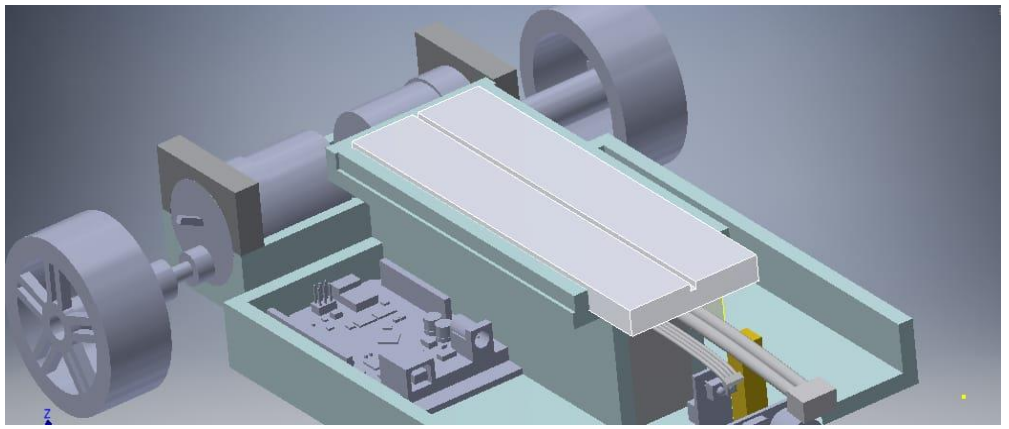
Use case


Visual Paradigm Online Free Edition
Autonomous driving car
MOVEMENT
<<extend>>
<<extend>>
ARDUINO UN0
AI
driveright()/driveleft()
drivefoward()
motor controller
checkcolor()
obstacle detection
readcolor()
Irsensor /rgb sensor/ultrasonic sensor
Visual Paradigm Online Free Edition

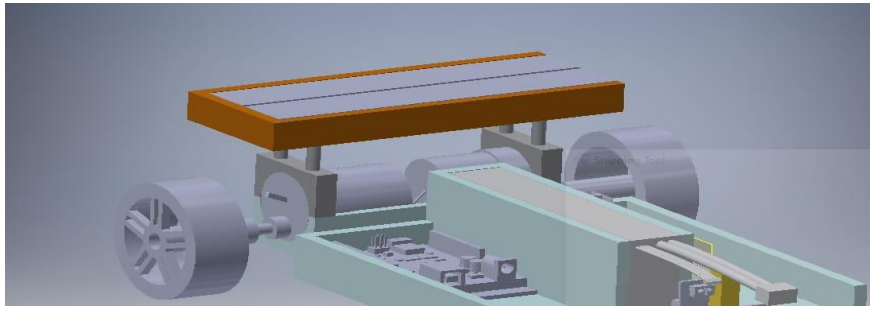
Design

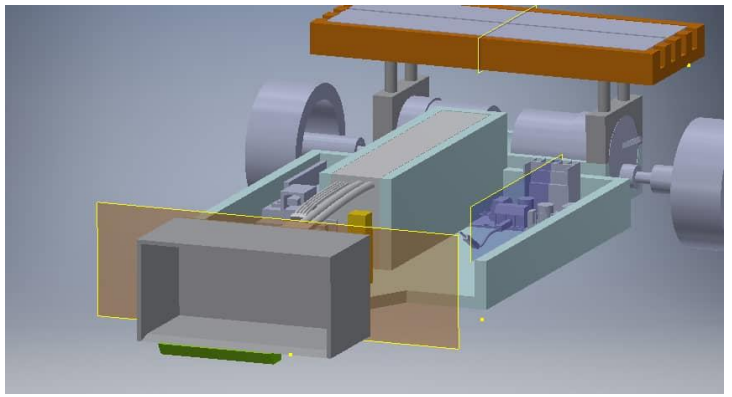




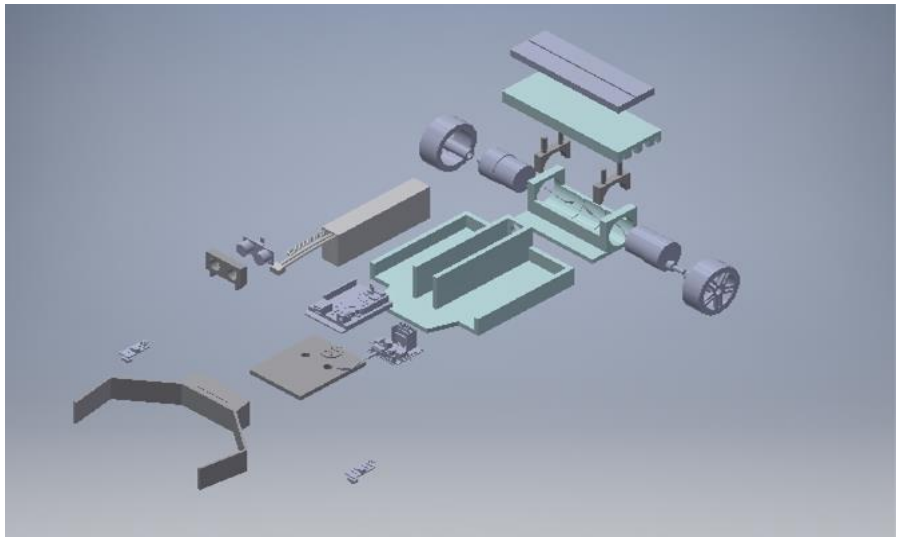


Initial design

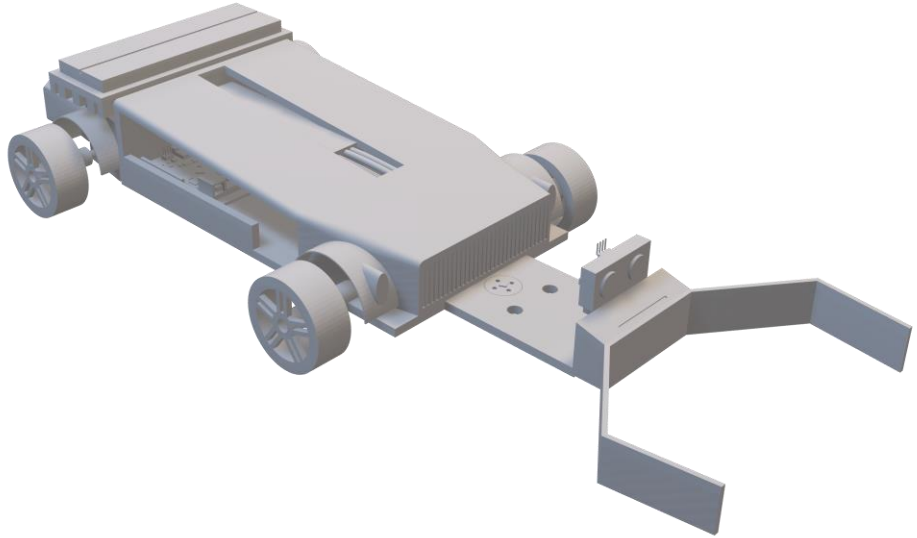


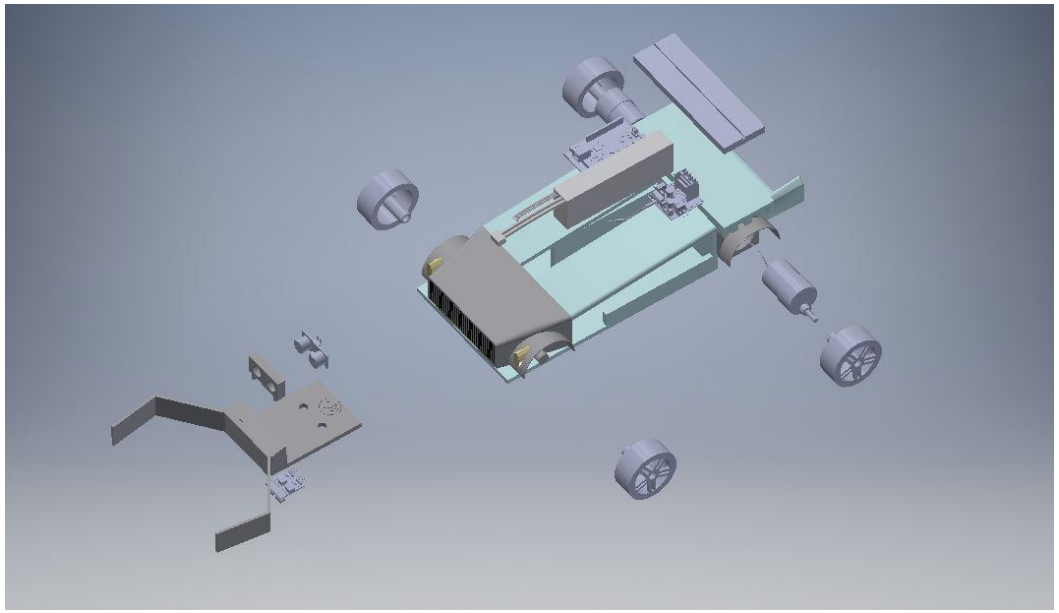


FINAL DESIGN(FUNCTIONAL)



FINAL DESIGN (RETRO)





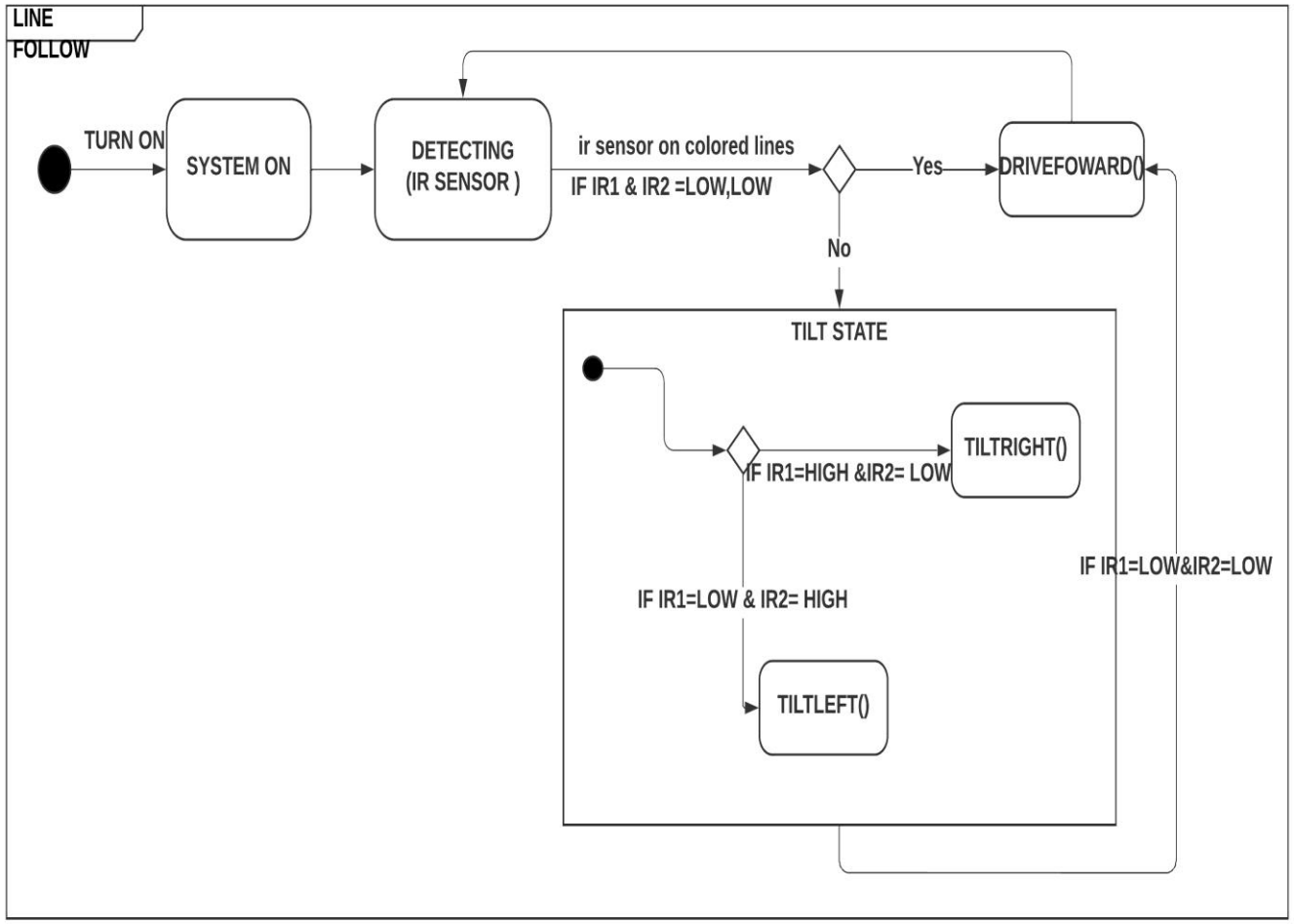
PROGRAMMING

CALIBRATION

```
int S0 = 6;
int S1 = 7;
int S2 = 8;
int S3 = 9;
int outPin = 11;

int rColourStrength;
int gColourStrength;
int bColourStrength;

unsigned int pulseWidth;

void setup() {
  // put your setup code here, to run once:

Serial.begin(9600);
pinMode(S0,OUTPUT);
pinMode(S1,OUTPUT);
pinMode(S2,OUTPUT);
pinMode(S3,OUTPUT);
pinMode(outPin,INPUT);
digitalWrite(S0,HIGH);
digitalWrite(S1,LOW);
}
```

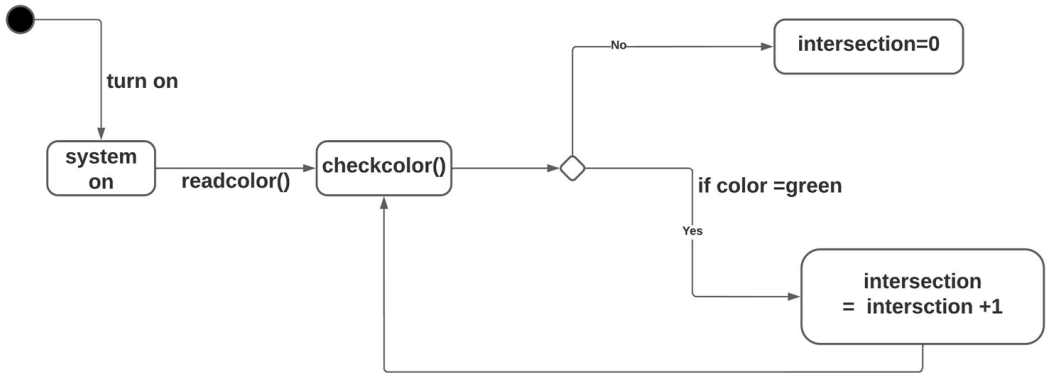
```
void loop() {
  // put your main code here, to run repeatedly:

//Reading RED components,s2 and s3 are LOW
digitalWrite(S2,LOW);
digitalWrite(S3,LOW);

pulseWidth = pulseIn(outPin,LOW);
rColourStrength = pulseWidth;

digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
pulseWidth = pulseIn(outPin,LOW);
gColourStrength = pulseWidth;

//////////////////////////////////////////////////////////

//Reading BLUE components,s2 is LOW and S3 is HIGH
//////////////////////////////////////////////////////////
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
pulseWidth = pulseIn(outPin,LOW);
bColourStrength = pulseWidth;


//////////////////////////////////////////////////////////
```

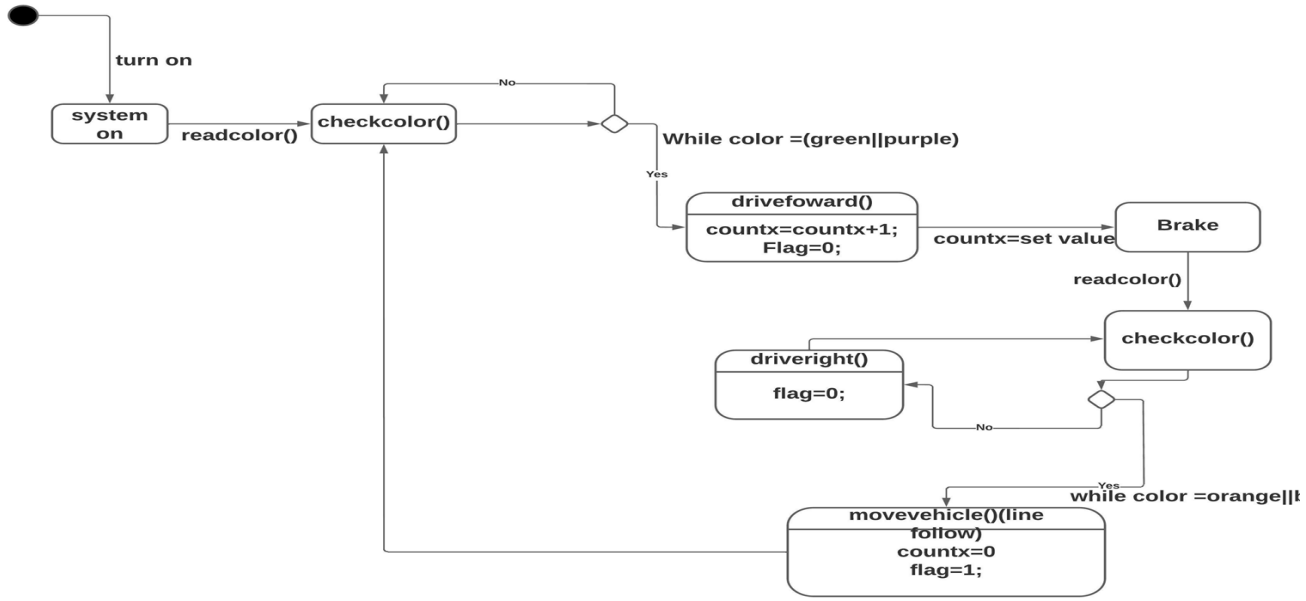
Tasks 1



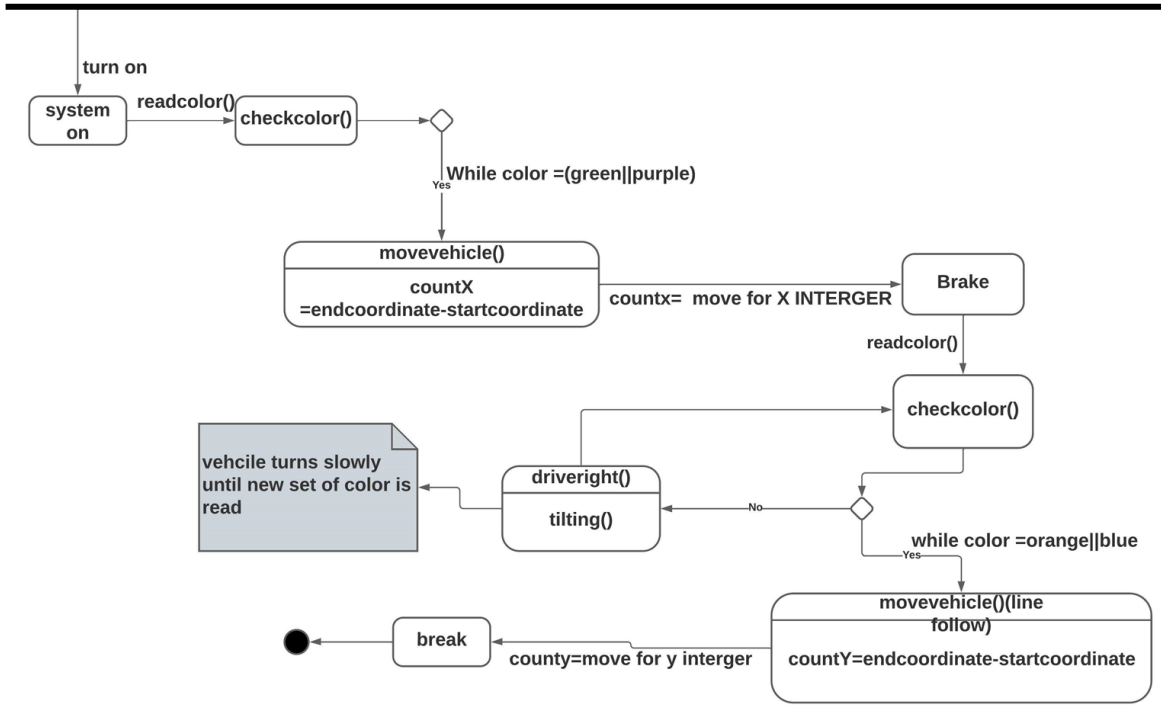

```
void line sensor()
{
 if(digitalRead(IR1)==HIGH && digitalRead(IR2)==HIGH) //IR will not glow on black lin
 {
  //Stop both Motors
  analogWrite(In1,0);
 analogWrite(In2,0);
 analogWrite(In3,0);
 analogWrite(In4,0);
 }

 else if(digitalRead(IR1)==LOW && digitalRead(IR2)==LOW)  //IR not on black line
 {
  //Move both the Motors
  analogWrite(In1,255);
 analogWrite(In2,0);
 analogWrite(In3,255);
 analogWrite(In4,0);
 }

 else if(digitalRead(IR1)==LOW && digitalRead(IR2)==HIGH)
 {
  //Tilt robot towards left by stopping the left wheel and moving the right one
  analogWrite(In1,255);
 analogWrite(In2,0);
 analogWrite(In3,0);
 analogWrite(In4,255);
  delay(100);
 }

 else if(digitalRead(IR1)==HIGH && digitalRead(IR2)==LOW)
 {
  //Tilt robot towards right by stopping the right wheel and moving the left one
```

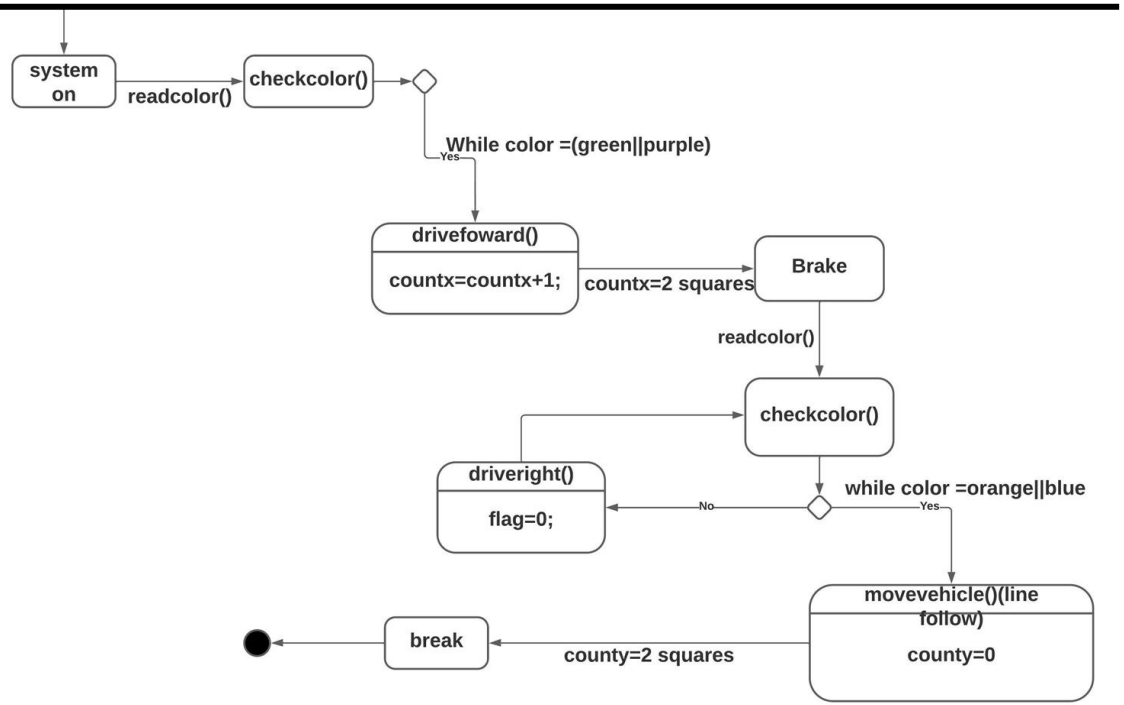
Task 3 and 4


turn on
system on
readcolor()
checkcolor()
No
intersection=0
if color =green
Yes
intersection = intersction +1


turn on
system on
readcolor()
checkcolor()
No
While color =(green||purple)
Yes
drivefoward()
countx=countx+1;
Flag=0;
countx=set value
Brake
readcolor()
checkcolor()
driveright()
flag=0;
No
Yes
while color =orange||
movevehicle()(line follow)
countx=0
flag=1;

Task 5 and 6


turn on
system on
readcolor()
checkcolor()
While color =(green||purple)
Yes
movevehicle()
countX =endcoordinate-startcoordinate
countx= move for X INTERGER
Brake
readcolor()
checkcolor()
vehcile turns slowly until new set of color is read
driveright()
tilting()
No
while color =orange||blue
Yes
movevehicle()(line follow)
countY=endcoordinate-startcoordinate
county=move for y interger
break


system on
readcolor()
checkcolor()
While color =(green||purple)
Yes
drivefoward()
countx=countx+1;
countx=2 squares
Brake
readcolor()
checkcolor()
driveright()
flag=0;
No
while color =orange||blue
Yes
movevehicle()(line follow)
county=0
county=2 squares
break

Thanks