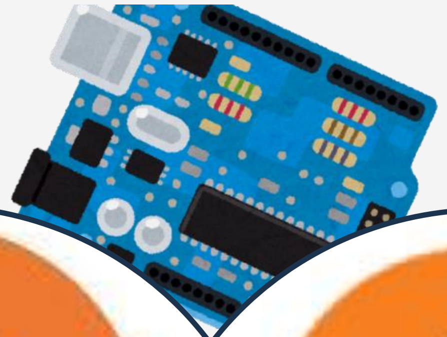
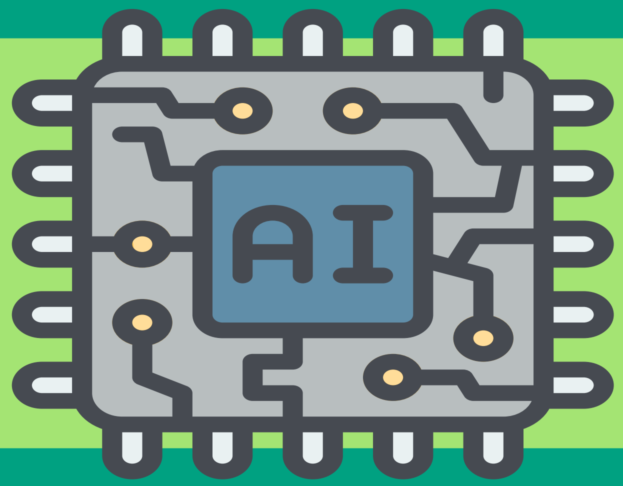
Graduation Project



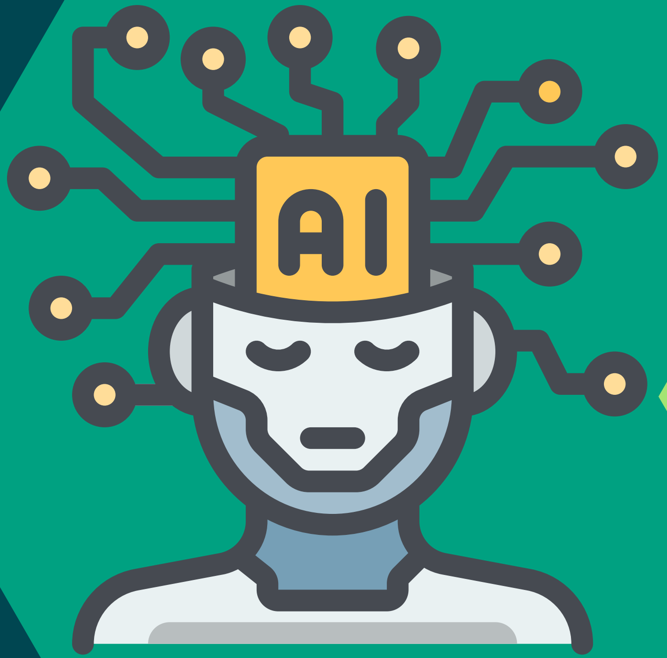


Graduation Project

جامعة ٦ أكتوبر
October 6 University

علوم الحاسب ونظم المعلومات
Info Sys & Comp Science

Object Detection System On Controlled Vehicle



Supervised by

Prof. Dr . Mohamed Fouad

Co-Supervision

Eng . Shady Bedier

Presented By

Mario Emad Edward	CS
Omar Mohamed Ahmed	CS
Moataz Ibrahim Gaber	CS
Ali Farouk Ali	CS
Nancy Khaled Sayed	CS
Mohamed Eid Mahmoud	IS
Yasmeen Mahdy Faheem	IS

AGENDA

- 1. Introduction
- 3. Components Required
- 5. Circuit Diagram
- 6. Software App
- 7. Arduino Coding Platform
- 8. Object Detection System
- 10. Code

Design and Methodology:

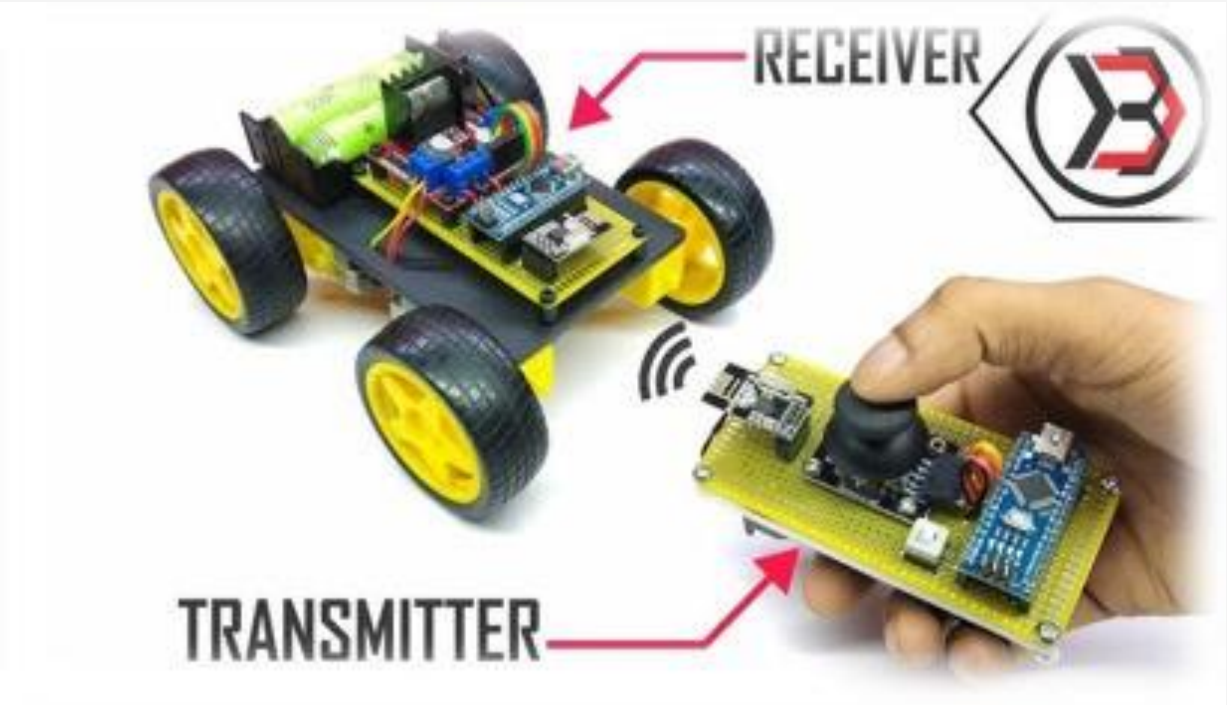
- 18. Activity Diagram
- 19. Use Case Diagram
- 20. Sequence Diagram
- 21. Applications
- 24. Future Work

INTRODUCTION

introduction

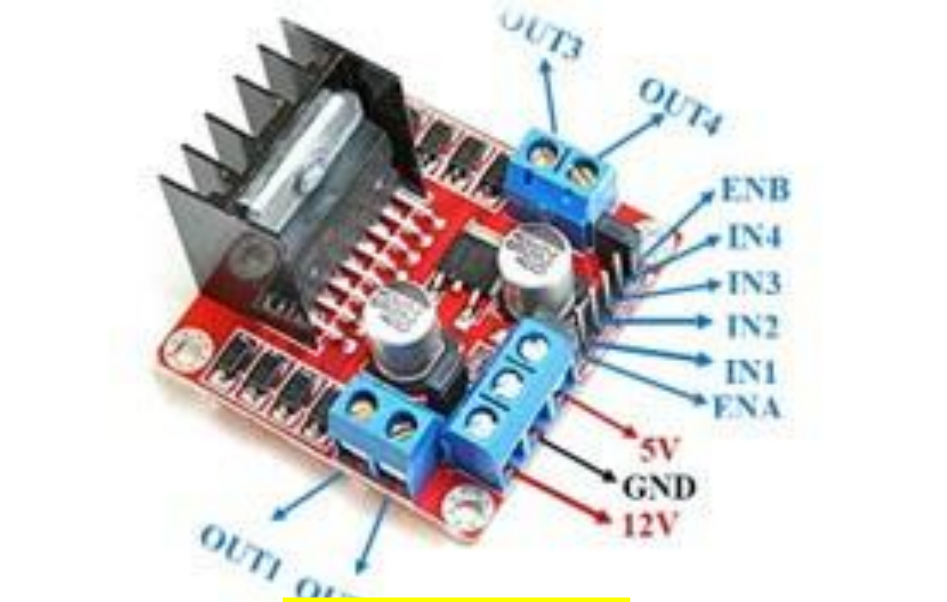
An Arduino car is a small robotic vehicle powered by an Arduino microcontroller. It's a popular project that combines electronics, programming, and mechanics to create a programmable and customizable vehicle, These projects are a great way to learn about robotics and hone programming skills in a hands-on manner

In this project, an Arduino-based car was built and equipped with an object detection camera. employs a camera for real-time object detection and live streaming. Controlled via a dedicated application, the car uses computer vision algorithms to identify many objects like Human, cell phone, chair and desk. This setup provides an excellent hands-on learning experience in integrating mobile app control with image processing and machine learning for robotics.

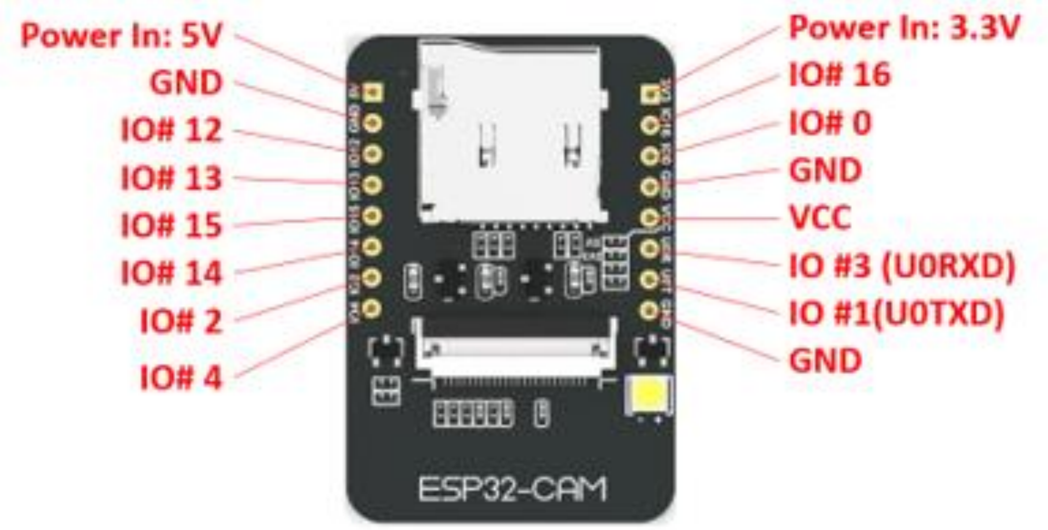


Components Required:

-

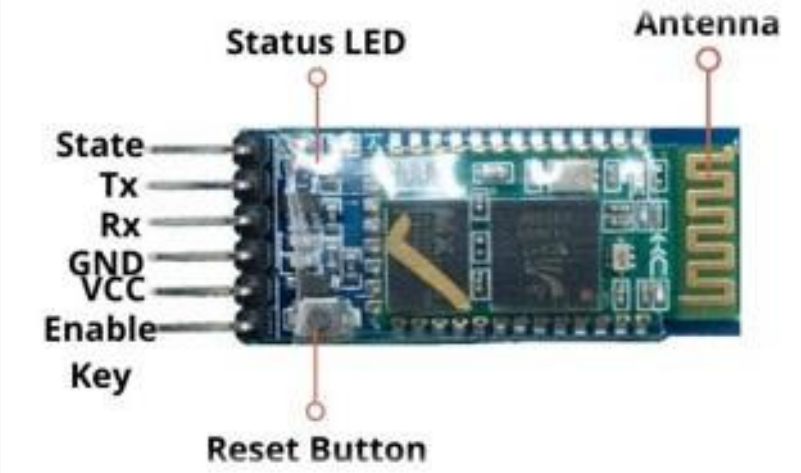


L298 Driver



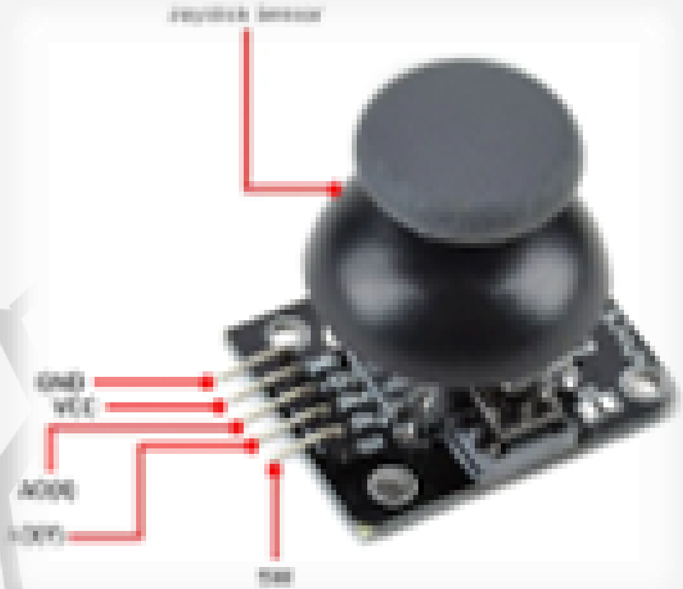
ESP – 32 Cam Microcontroller

RF Sender & Reciver



Bluetooth Module HC-05

Arduino Uno



Joystick

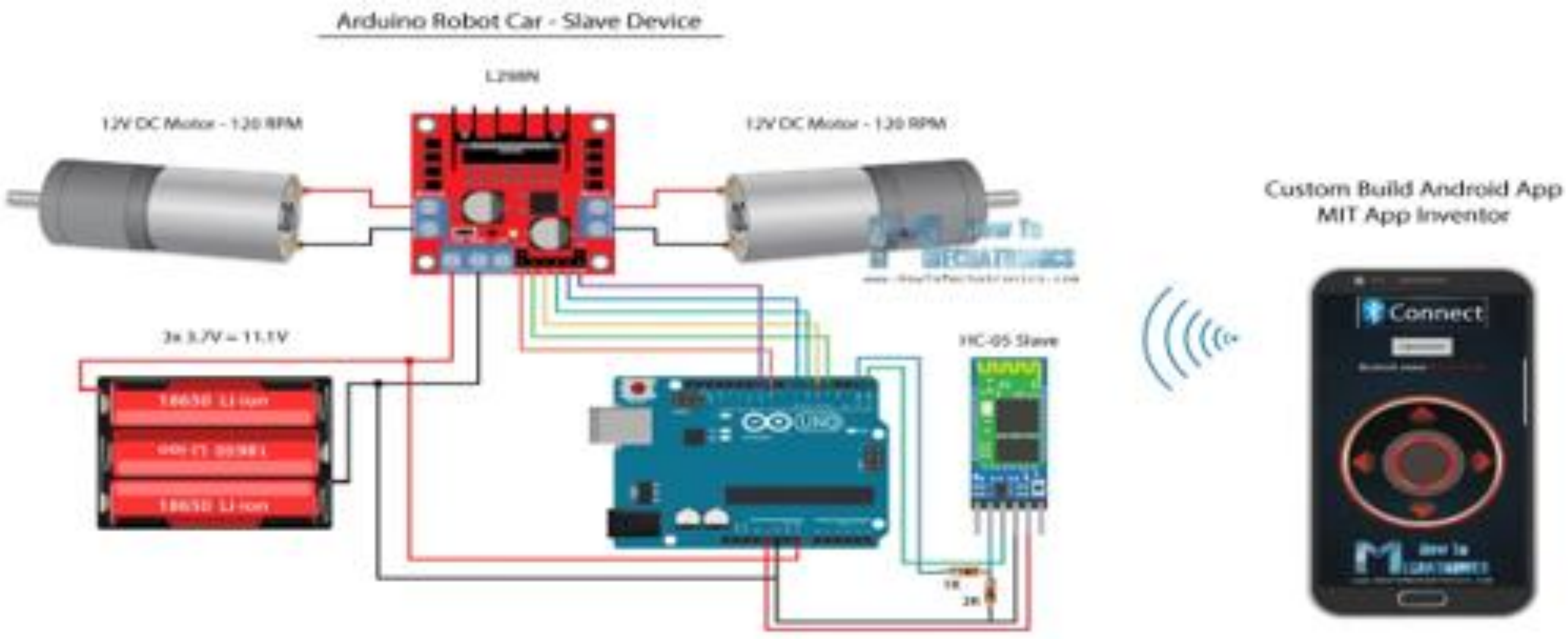
12V Battery

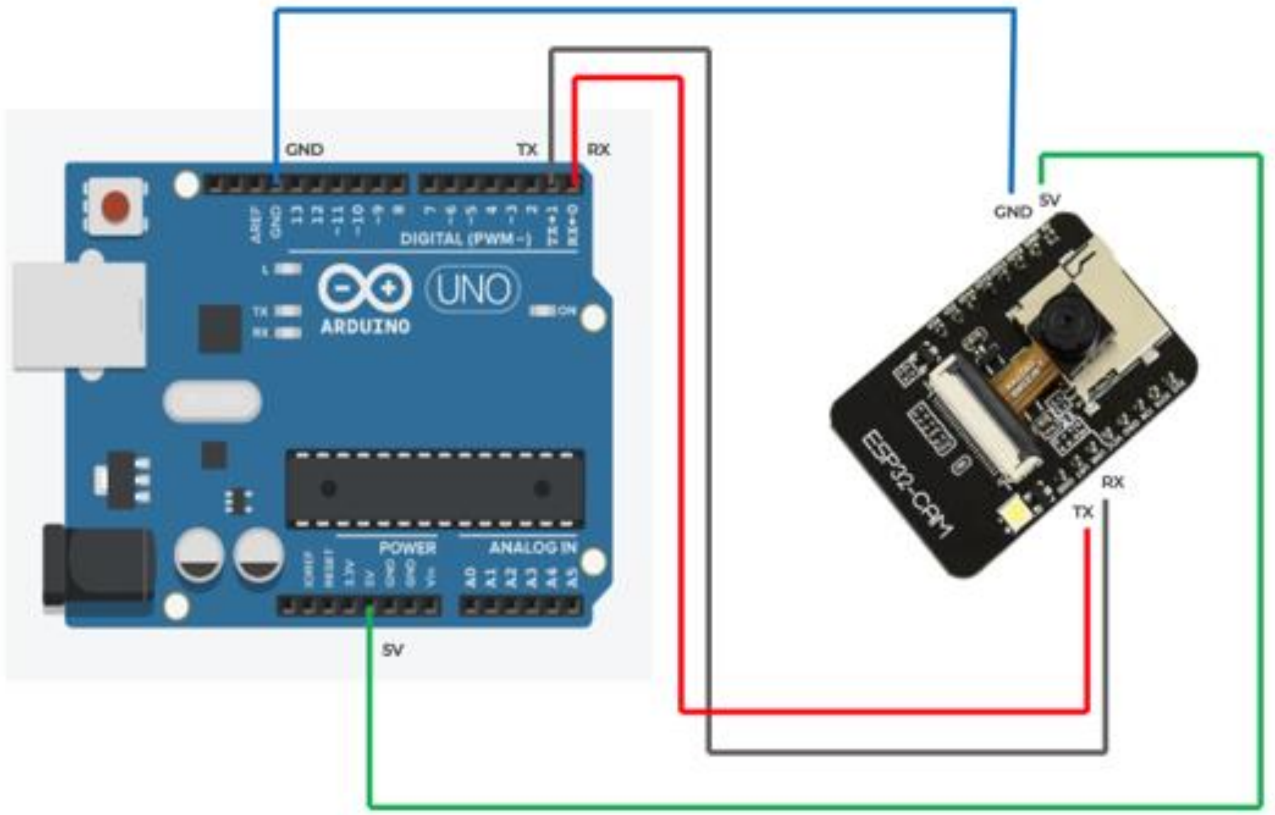
Car Kit

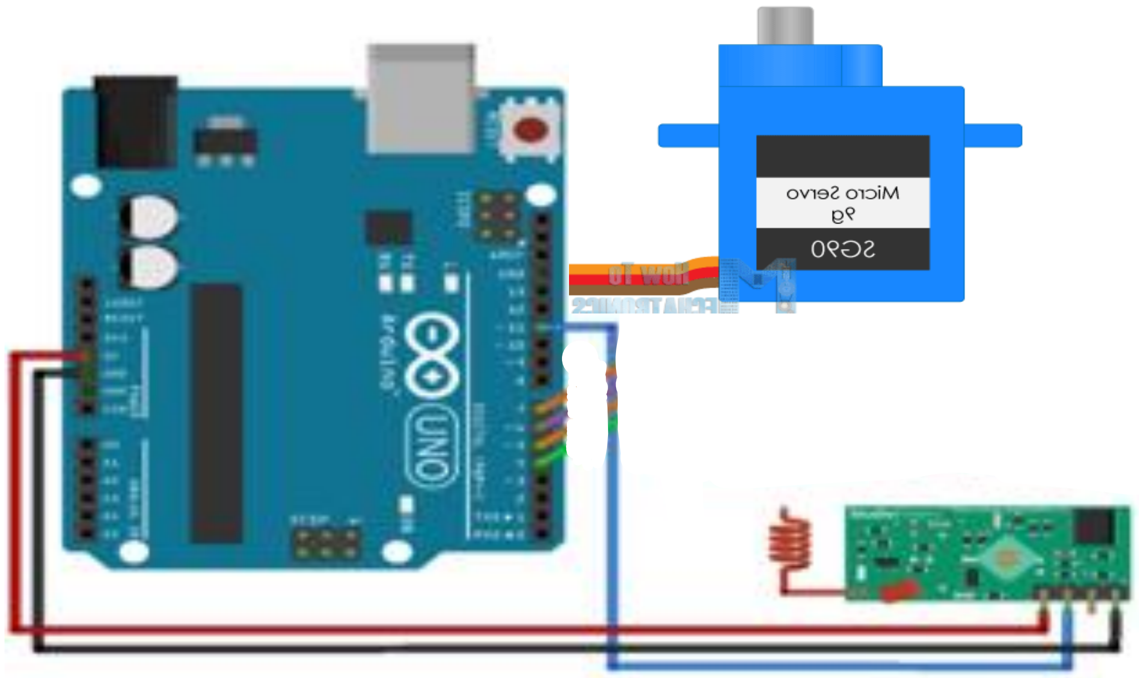
DC Motors

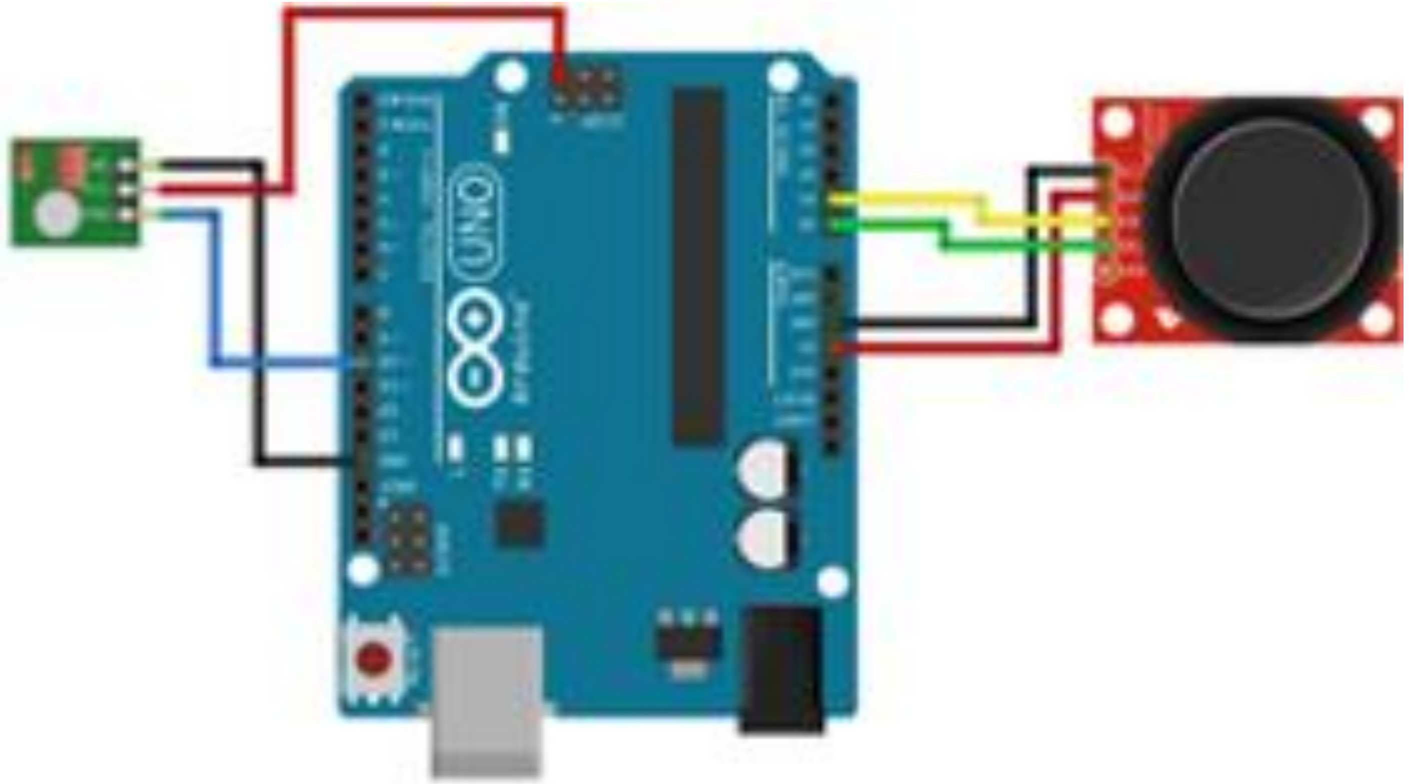
Servo Motor

Circuit Diagram:

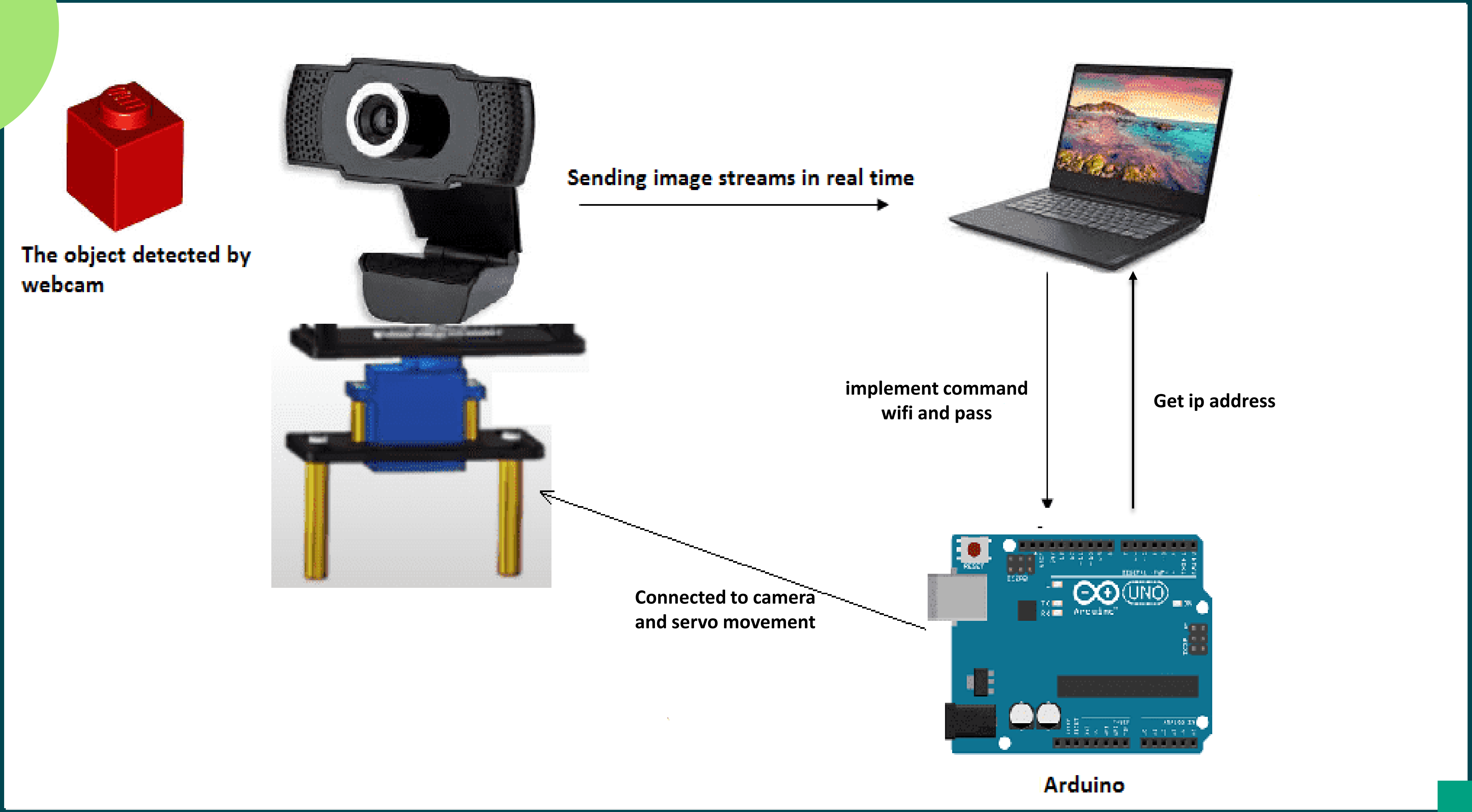




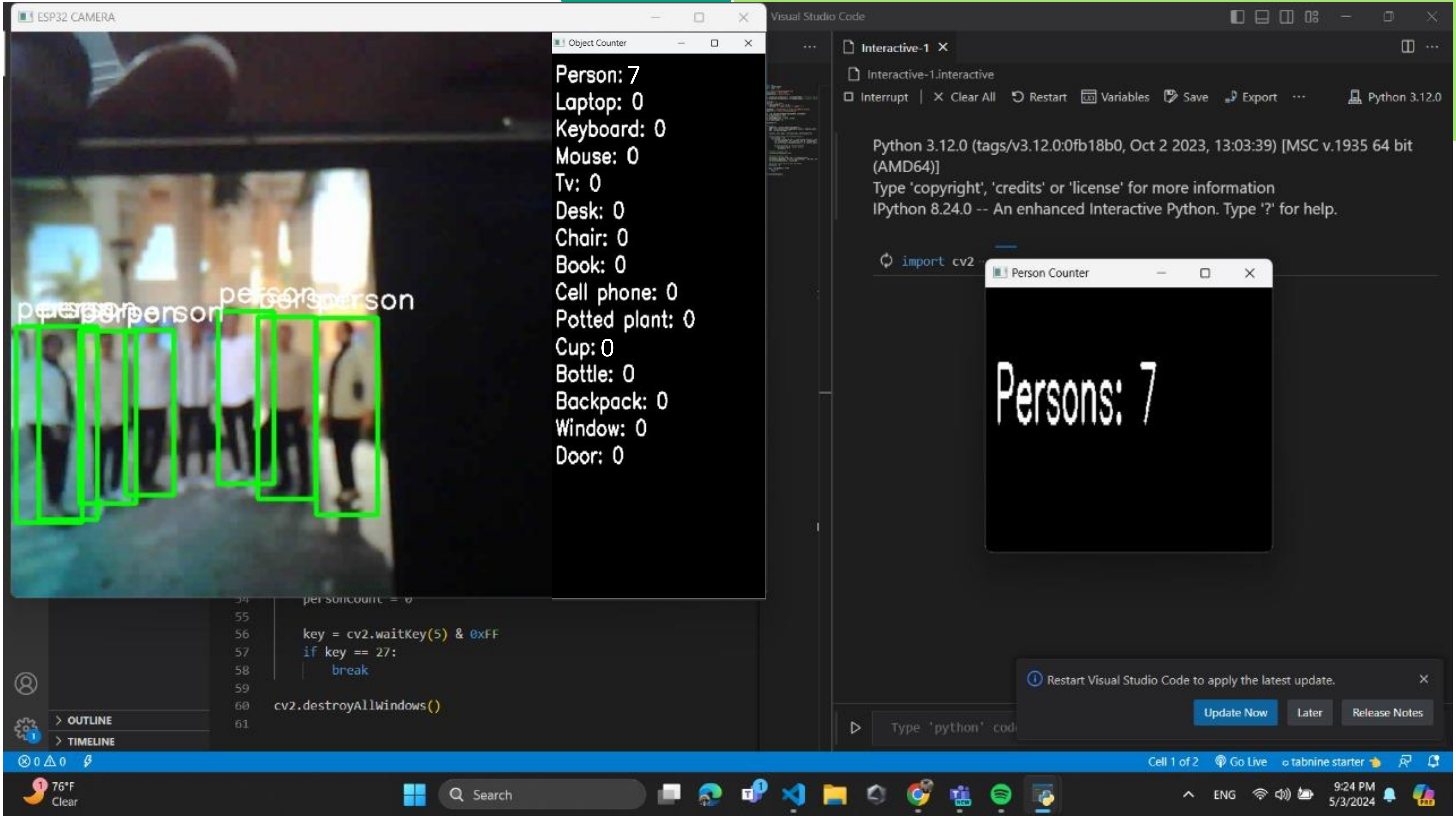




Detection Circuit:


The object detected by webcam
Sending image streams in real time
implement command wifi and pass
Get ip address
Connected to camera and servo movement
Arduino

Live Stream Detection:



```
Person: 7
Laptop: 0
Keyboard: 0
Mouse: 0
Tv: 0
Desk: 0
Chair: 0
Book: 0
Cell phone: 0
Potted plant: 0
Cup: 0
Bottle: 0
Backpack: 0
Window: 0
Door: 0
```

```
Persons: 7
```

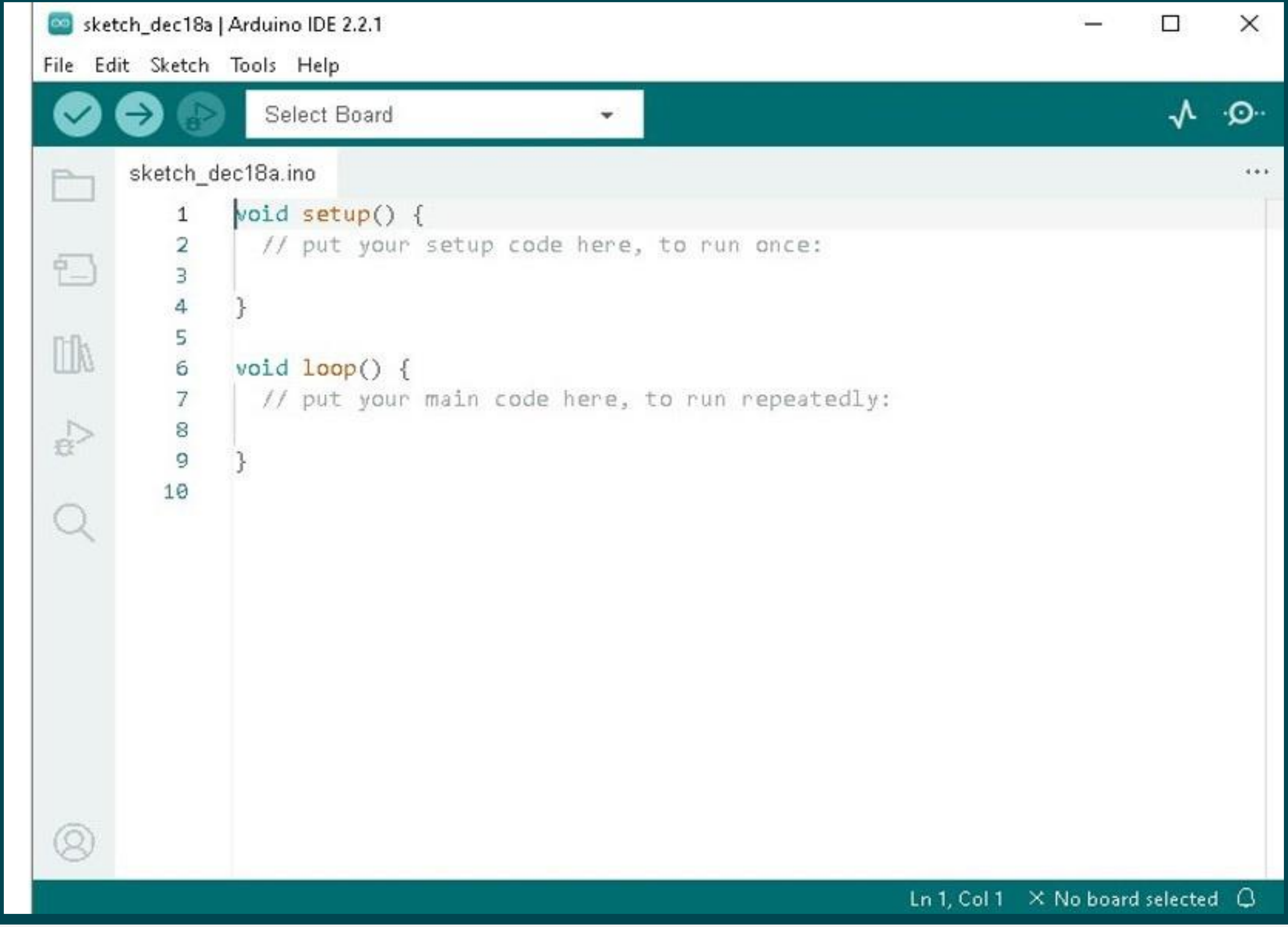
```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.24.0 -- An enhanced Interactive Python. Type '?' for help.
```

```
import cv2
```

```
    personcount = 0

    key = cv2.waitKey(5) & 0xFF
    if key == 27:
        break

cv2.destroyAllWindows()
```

Restart Visual Studio Code to apply the latest update.

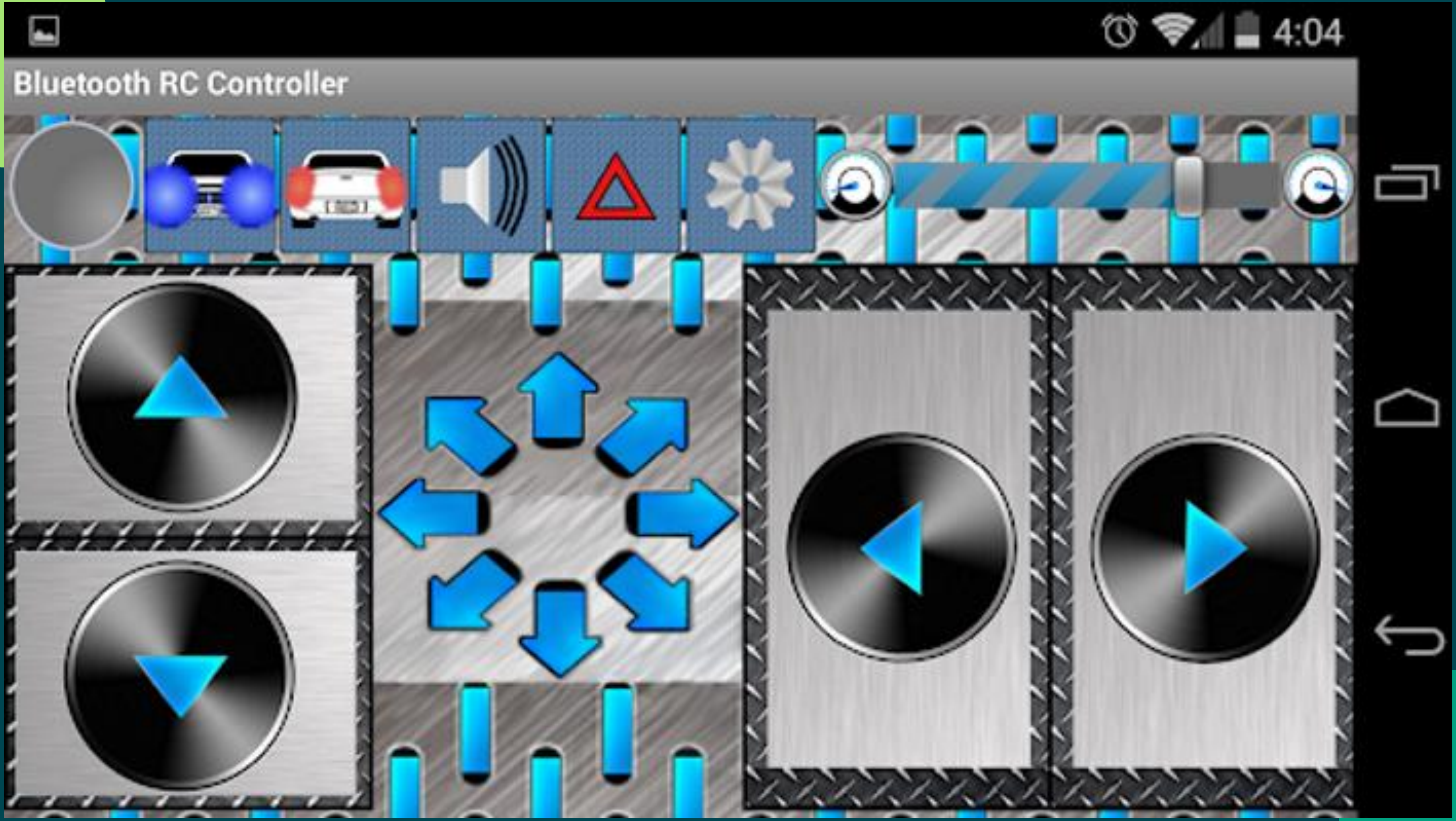
Arduino Coding Platform

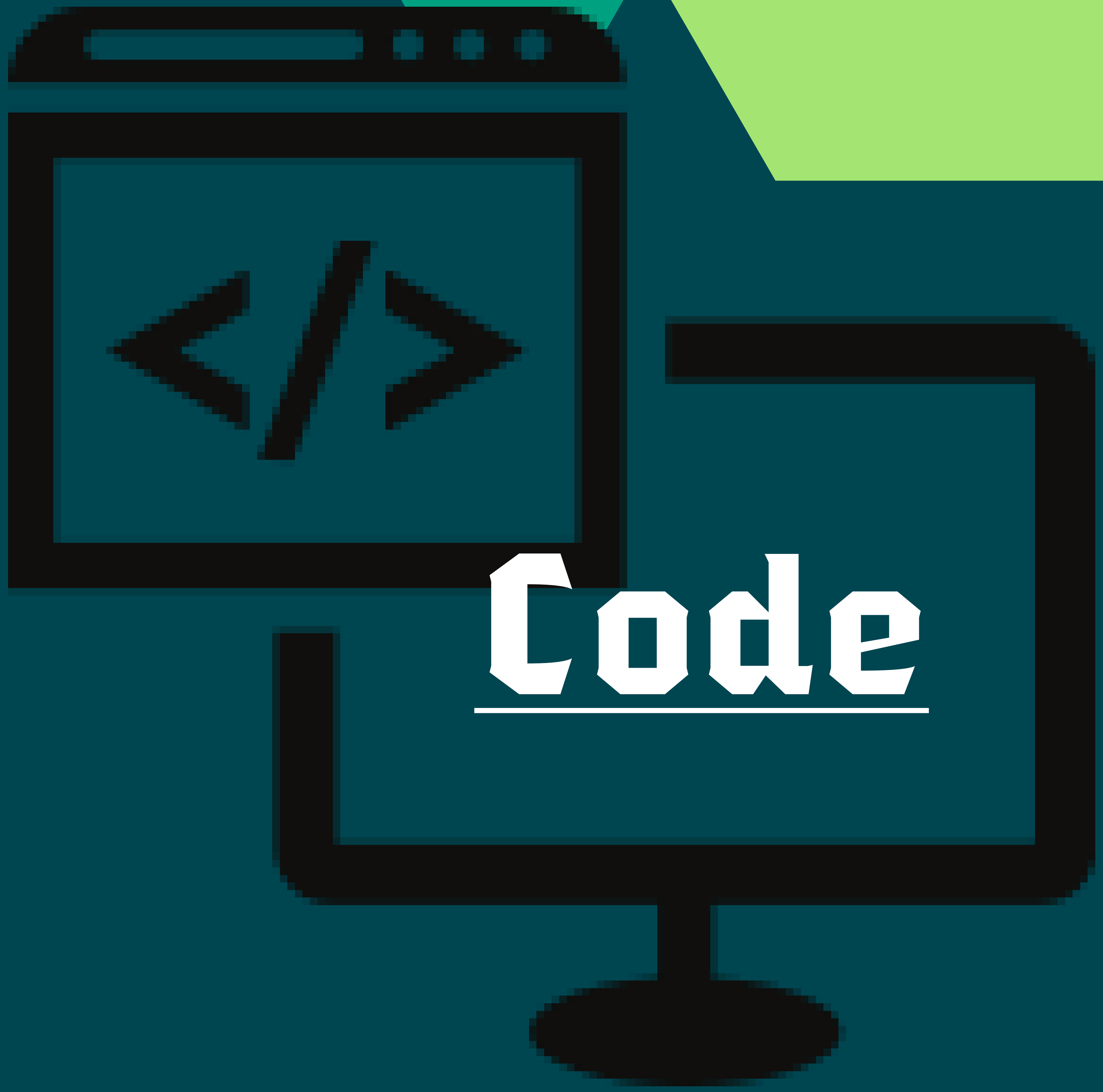


sketch_dec18a | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Select Board
sketch_dec18a.ino
Ln 1, Col 1
No board selected

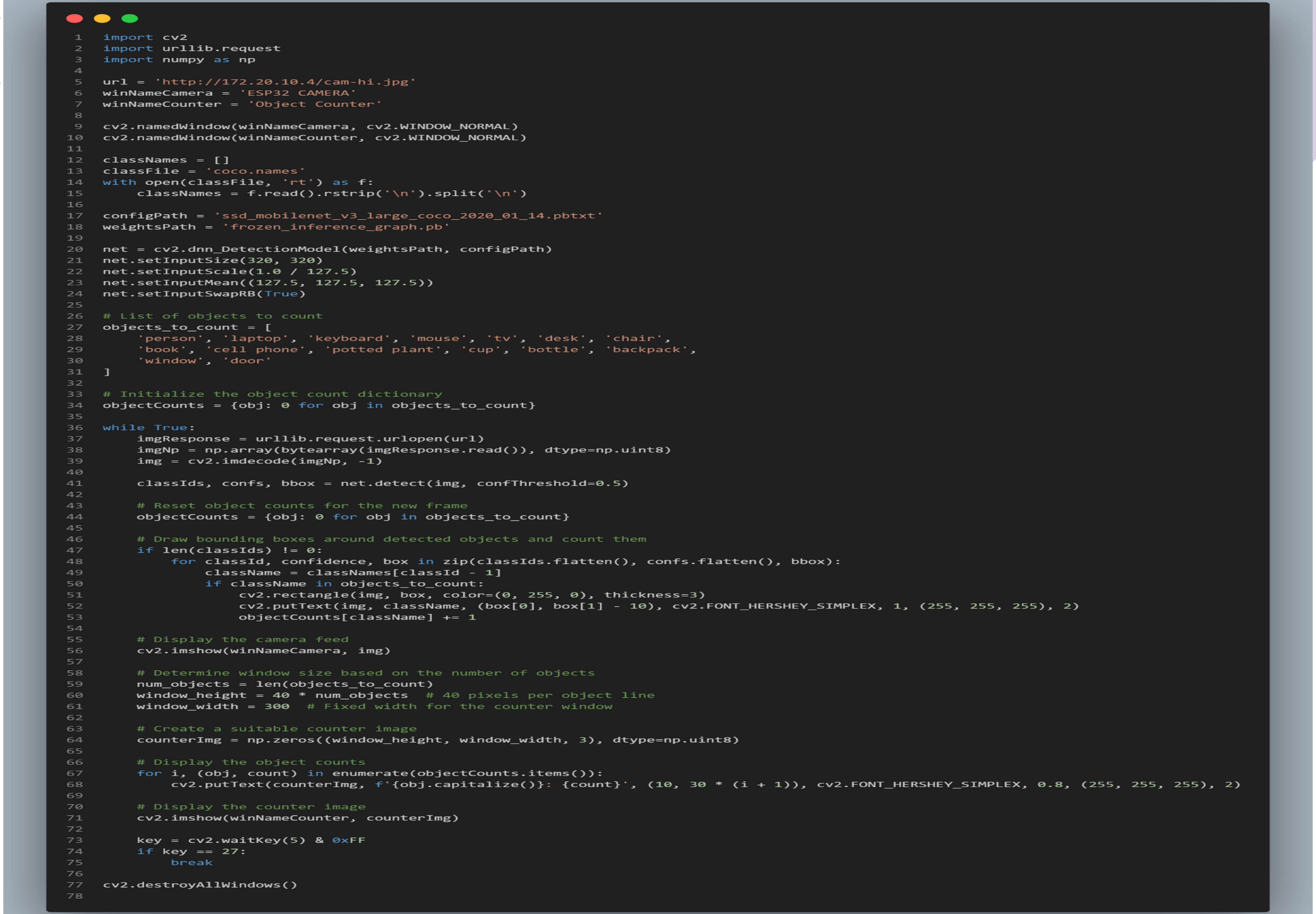
```
void setup() {
  // put your setup code here, to run once:

}

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

}
```

Bluetooth RC Controller Application:

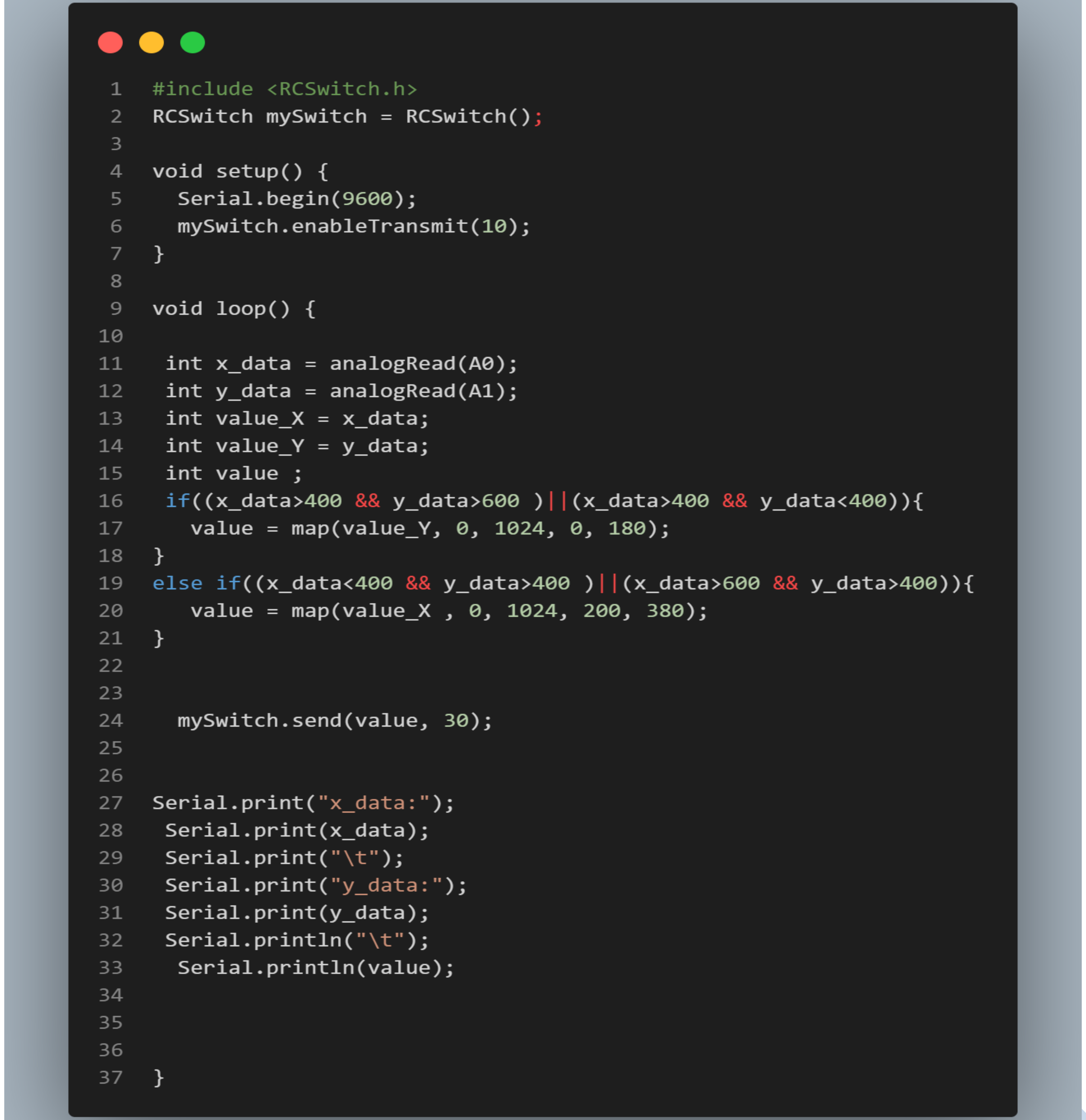




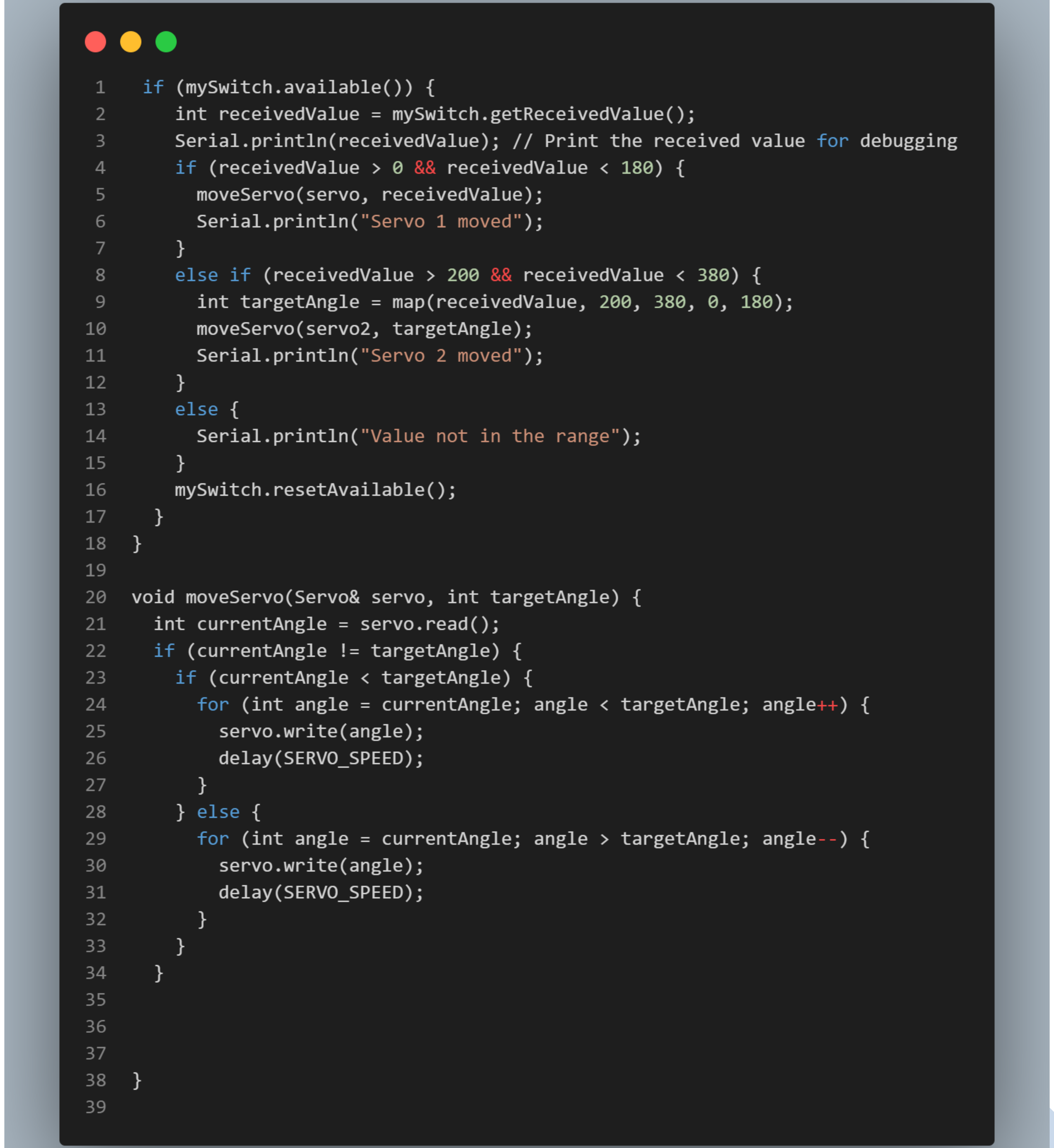
Code



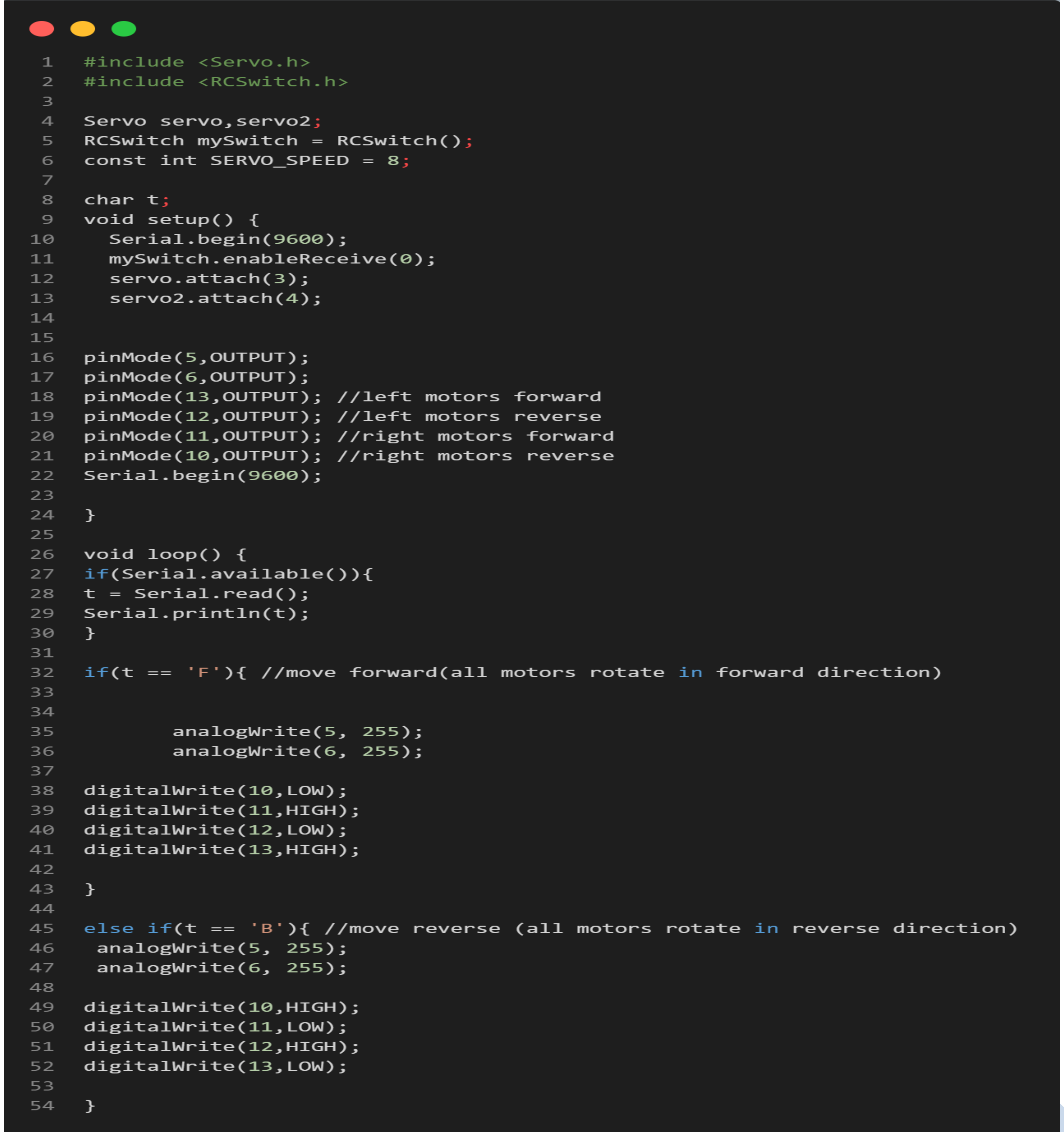
```python
import cv2
import urllib.request
import numpy as np

url = 'http://172.20.10.4/cam-hi.jpg'
winNameCamera = 'ESP32 CAMERA'
winNameCounter = 'Object Counter'

cv2.namedWindow(winNameCamera, cv2.WINDOW_NORMAL)
cv2.namedWindow(winNameCounter, cv2.WINDOW_NORMAL)

classNames = []
classFile = 'coco.names'
with open(classFile, 'rt') as f:
    classNames = f.read().rstrip('\n').split('\n')

configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
weightsPath = 'frozen_inference_graph.pb'

net = cv2.dnn_DetectionModel(weightsPath, configPath)
net.setInputSize(320, 320)
net.setInputScale(1.0 / 127.5)
net.setInputMean((127.5, 127.5, 127.5))
net.setInputSwapRB(True)

# List of objects to count
objects_to_count = [
    'person', 'laptop', 'keyboard', 'mouse', 'tv', 'desk', 'chair',
    'book', 'cell phone', 'potted plant', 'cup', 'bottle', 'backpack',
    'window', 'door'
]

# Initialize the object count dictionary
objectCounts = {obj: 0 for obj in objects_to_count}

while True:
    imgResponse = urllib.request.urlopen(url)
    imgNp = np.array(bytearray(imgResponse.read()), dtype=np.uint8)
    img = cv2.imdecode(imgNp, -1)

    classIds, confs, bbox = net.detect(img, confThreshold=0.5)

    # Reset object counts for the new frame
    objectCounts = {obj: 0 for obj in objects_to_count}

    # Draw bounding boxes around detected objects and count them
    if len(classIds) != 0:
        for classId, confidence, box in zip(classIds.flatten(), confs.flatten(), bbox):
            className = classNames[classId - 1]
            if className in objects_to_count:
                cv2.rectangle(img, box, color=(0, 255, 0), thickness=3)
                cv2.putText(img, className, (box[0], box[1] - 10), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
                objectCounts[className] += 1

    # Display the camera feed
    cv2.imshow(winNameCamera, img)

    # Determine window size based on the number of objects
    num_objects = len(objects_to_count)
    window_height = 40 * num_objects  # 40 pixels per object line
    window_width = 300  # Fixed width for the counter window

    # Create a suitable counter image
    counterImg = np.zeros((window_height, window_width, 3), dtype=np.uint8)

    # Display the object counts
    for i, (obj, count) in enumerate(objectCounts.items()):
        cv2.putText(counterImg, f'{obj.capitalize()}: {count}', (10, 30 * (i + 1)), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 255, 255), 2)

    # Display the counter image
    cv2.imshow(winNameCounter, counterImg)

    key = cv2.waitKey(5) & 0xFF
    if key == 27:
        break

cv2.destroyAllWindows()
```



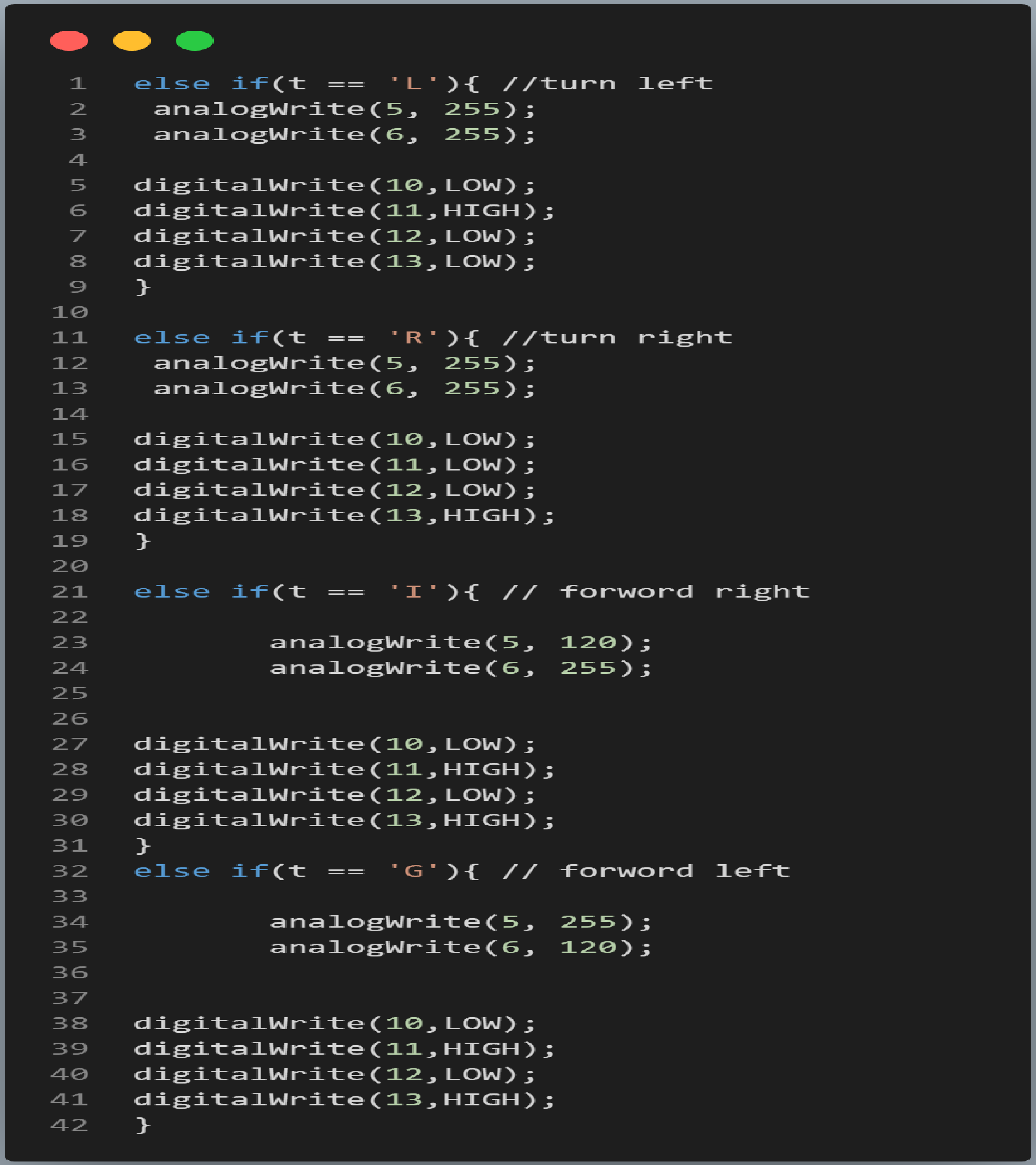
```
#include <RCSwitch.h>
RCSwitch mySwitch = RCSwitch();

void setup() {
  Serial.begin(9600);
  mySwitch.enableTransmit(10);
}

void loop() {

 int x_data = analogRead(A0);
 int y_data = analogRead(A1);
 int value_X = x_data;
 int value_Y = y_data;
 int value ;
 if((x_data>400 && y_data>600 )||(x_data>400 && y_data<400)){
   value = map(value_Y, 0, 1024, 0, 180);
}
else if((x_data<400 && y_data>400 )||(x_data>600 && y_data>400)){
   value = map(value_X , 0, 1024, 200, 380);
}


  mySwitch.send(value, 30);


Serial.print("x_data:");
 Serial.print(x_data);
 Serial.print("\t");
 Serial.print("y_data:");
 Serial.print(y_data);
 Serial.println("\t");
  Serial.println(value);



}
```



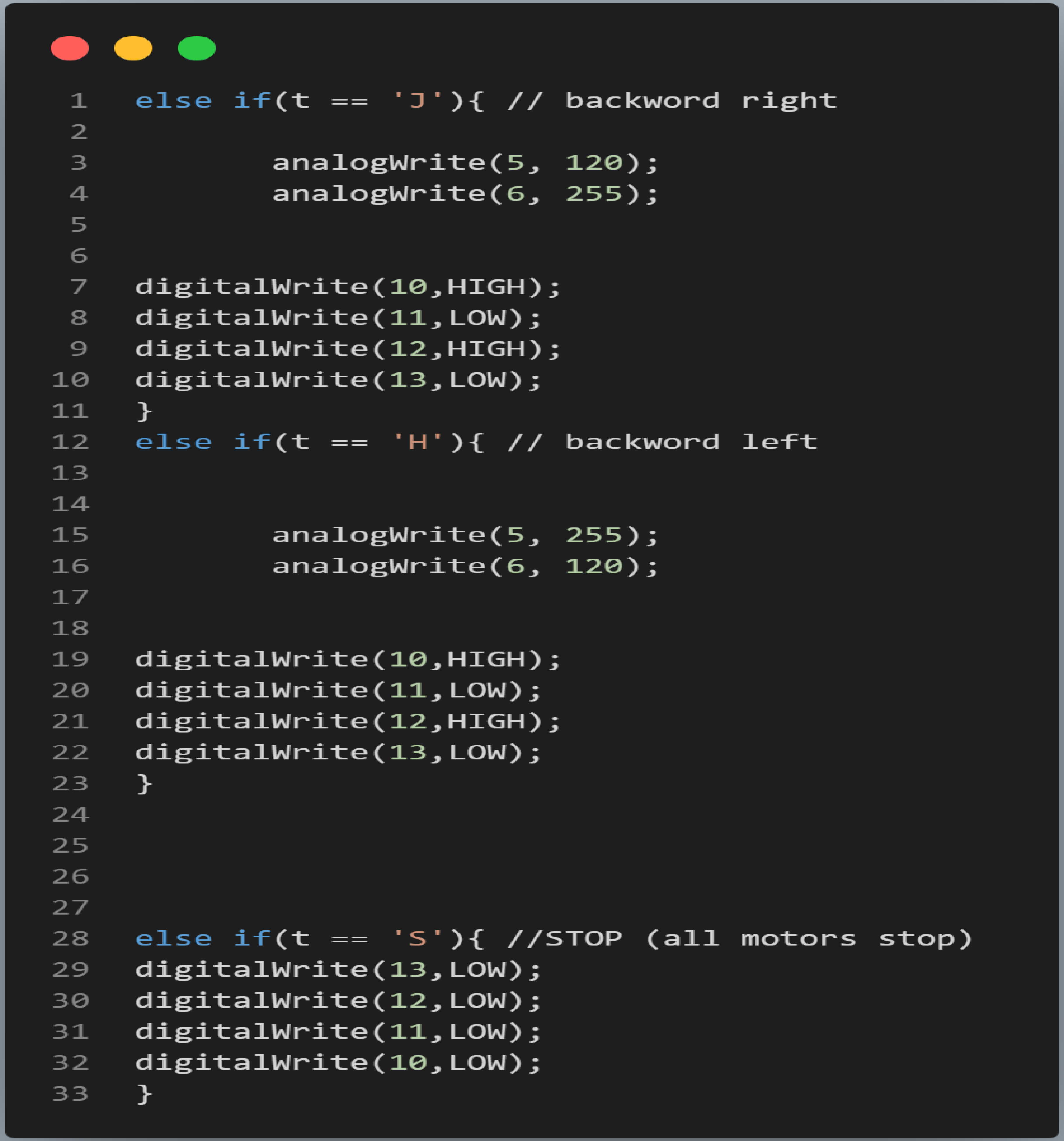
```
 if (mySwitch.available()) {
    int receivedValue = mySwitch.getReceivedValue();
    Serial.println(receivedValue); // Print the received value for debugging
    if (receivedValue > 0 && receivedValue < 180) {
      moveServo(servo, receivedValue);
      Serial.println("Servo 1 moved");
    }
    else if (receivedValue > 200 && receivedValue < 380) {
      int targetAngle = map(receivedValue, 200, 380, 0, 180);
      moveServo(servo2, targetAngle);
      Serial.println("Servo 2 moved");
    }
    else {
      Serial.println("Value not in the range");
    }
    mySwitch.resetAvailable();
  }
}

void moveServo(Servo& servo, int targetAngle) {
  int currentAngle = servo.read();
  if (currentAngle != targetAngle) {
    if (currentAngle < targetAngle) {
      for (int angle = currentAngle; angle < targetAngle; angle++) {
        servo.write(angle);
        delay(SERVO_SPEED);
      }
    } else {
      for (int angle = currentAngle; angle > targetAngle; angle--) {
        servo.write(angle);
        delay(SERVO_SPEED);
      }
    }
  }



}

```



```
#include <Servo.h>
#include <RCSwitch.h>

Servo servo,servo2;
RCSwitch mySwitch = RCSwitch();
const int SERVO_SPEED = 8;

char t;
void setup() {
  Serial.begin(9600);
  mySwitch.enableReceive(0);
  servo.attach(3);
  servo2.attach(4);


pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(13,OUTPUT); //left motors forward
pinMode(12,OUTPUT); //left motors reverse
pinMode(11,OUTPUT); //right motors forward
pinMode(10,OUTPUT); //right motors reverse
Serial.begin(9600);

}

void loop() {
if(Serial.available()){
t = Serial.read();
Serial.println(t);
}

if(t == 'F'){ //move forward(all motors rotate in forward direction)


       analogWrite(5, 255);
       analogWrite(6, 255);

digitalWrite(10,LOW);
digitalWrite(11,HIGH);
digitalWrite(12,LOW);
digitalWrite(13,HIGH);

}

else if(t == 'B'){ //move reverse (all motors rotate in reverse direction)
 analogWrite(5, 255);
 analogWrite(6, 255);

digitalWrite(10,HIGH);
digitalWrite(11,LOW);
digitalWrite(12,HIGH);
digitalWrite(13,LOW);

}
```



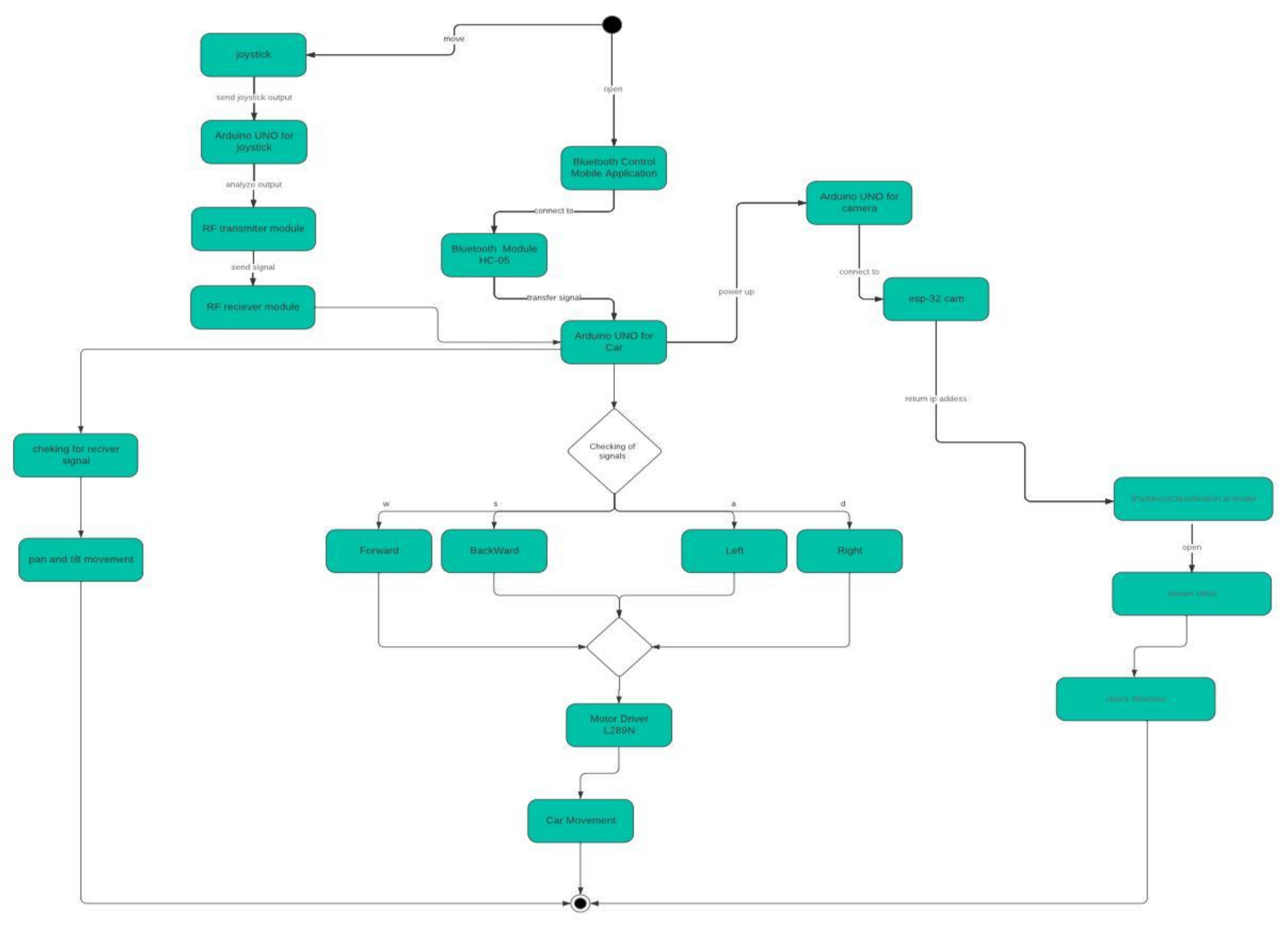
```
else if(t == 'L'){ //turn left
 analogWrite(5, 255);
 analogWrite(6, 255);

digitalWrite(10,LOW);
digitalWrite(11,HIGH);
digitalWrite(12,LOW);
digitalWrite(13,LOW);
}

else if(t == 'R'){ //turn right
 analogWrite(5, 255);
 analogWrite(6, 255);

digitalWrite(10,LOW);
digitalWrite(11,LOW);
digitalWrite(12,LOW);
digitalWrite(13,HIGH);
}

else if(t == 'I'){ // forword right

       analogWrite(5, 120);
       analogWrite(6, 255);


digitalWrite(10,LOW);
digitalWrite(11,HIGH);
digitalWrite(12,LOW);
digitalWrite(13,HIGH);
}
else if(t == 'G'){ // forword left

       analogWrite(5, 255);
       analogWrite(6, 120);


digitalWrite(10,LOW);
digitalWrite(11,HIGH);
digitalWrite(12,LOW);
digitalWrite(13,HIGH);
}
```



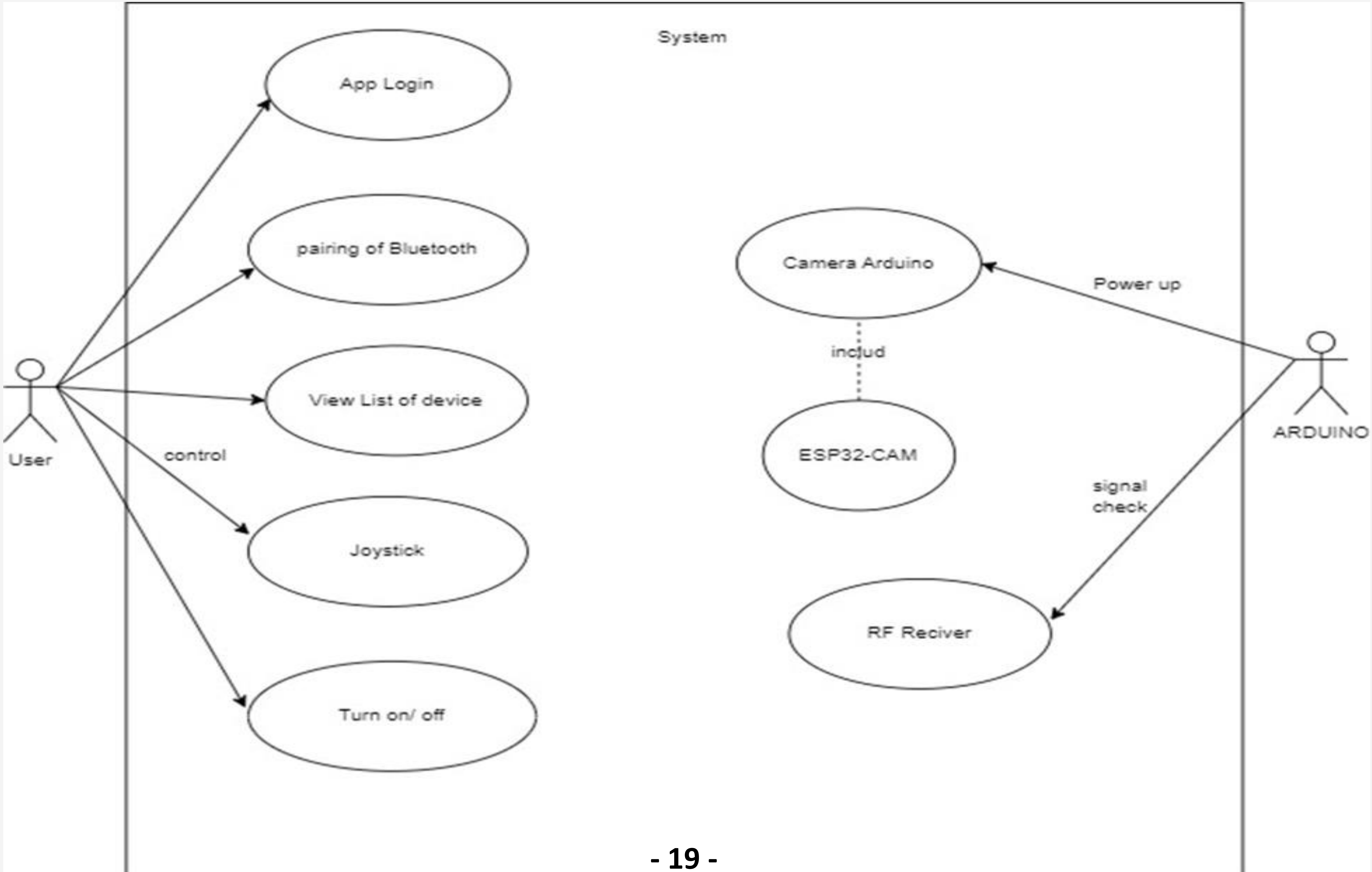
```
else if(t == 'J'){ // backword right

       analogWrite(5, 120);
       analogWrite(6, 255);


digitalWrite(10,HIGH);
digitalWrite(11,LOW);
digitalWrite(12,HIGH);
digitalWrite(13,LOW);
}
else if(t == 'H'){ // backword left


       analogWrite(5, 255);
       analogWrite(6, 120);


digitalWrite(10,HIGH);
digitalWrite(11,LOW);
digitalWrite(12,HIGH);
digitalWrite(13,LOW);
}




else if(t == 'S'){ //STOP (all motors stop)
digitalWrite(13,LOW);
digitalWrite(12,LOW);
digitalWrite(11,LOW);
digitalWrite(10,LOW);
}
```

Design and Methodology

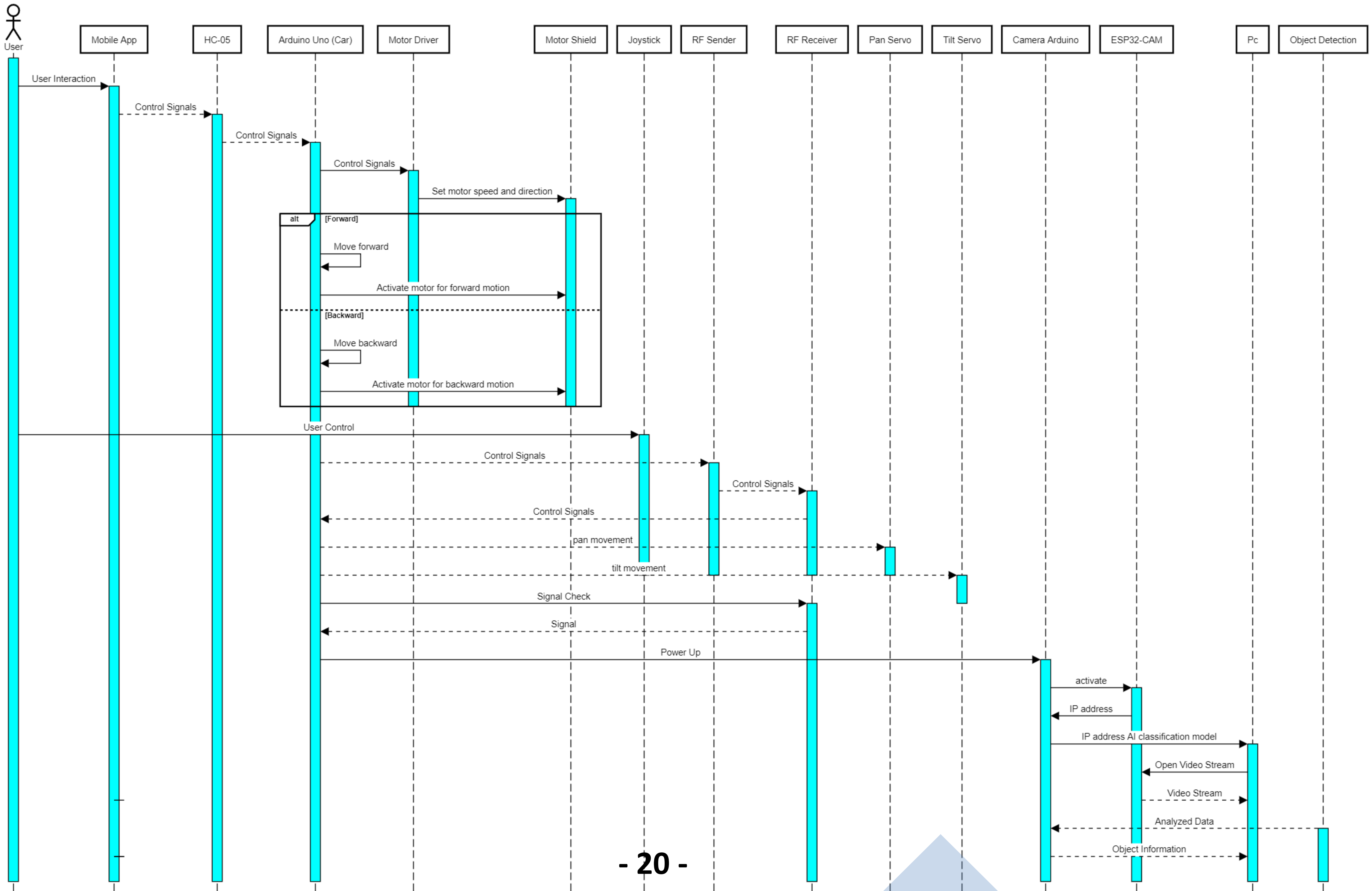
Activity Diagram

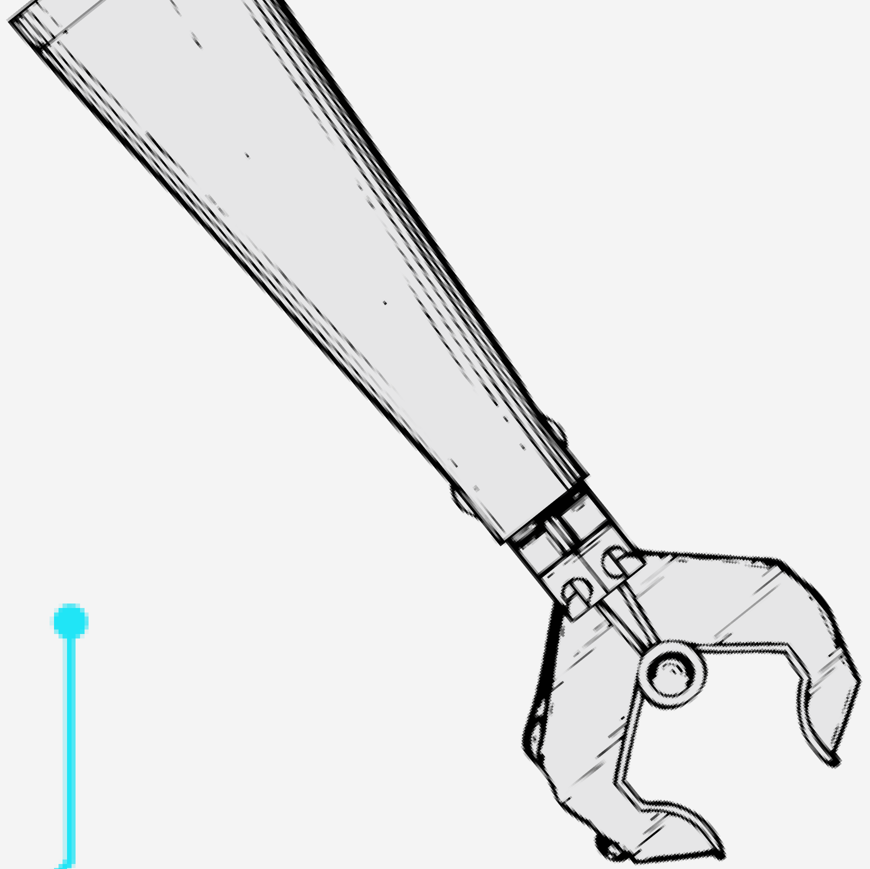


UseCase Diagram

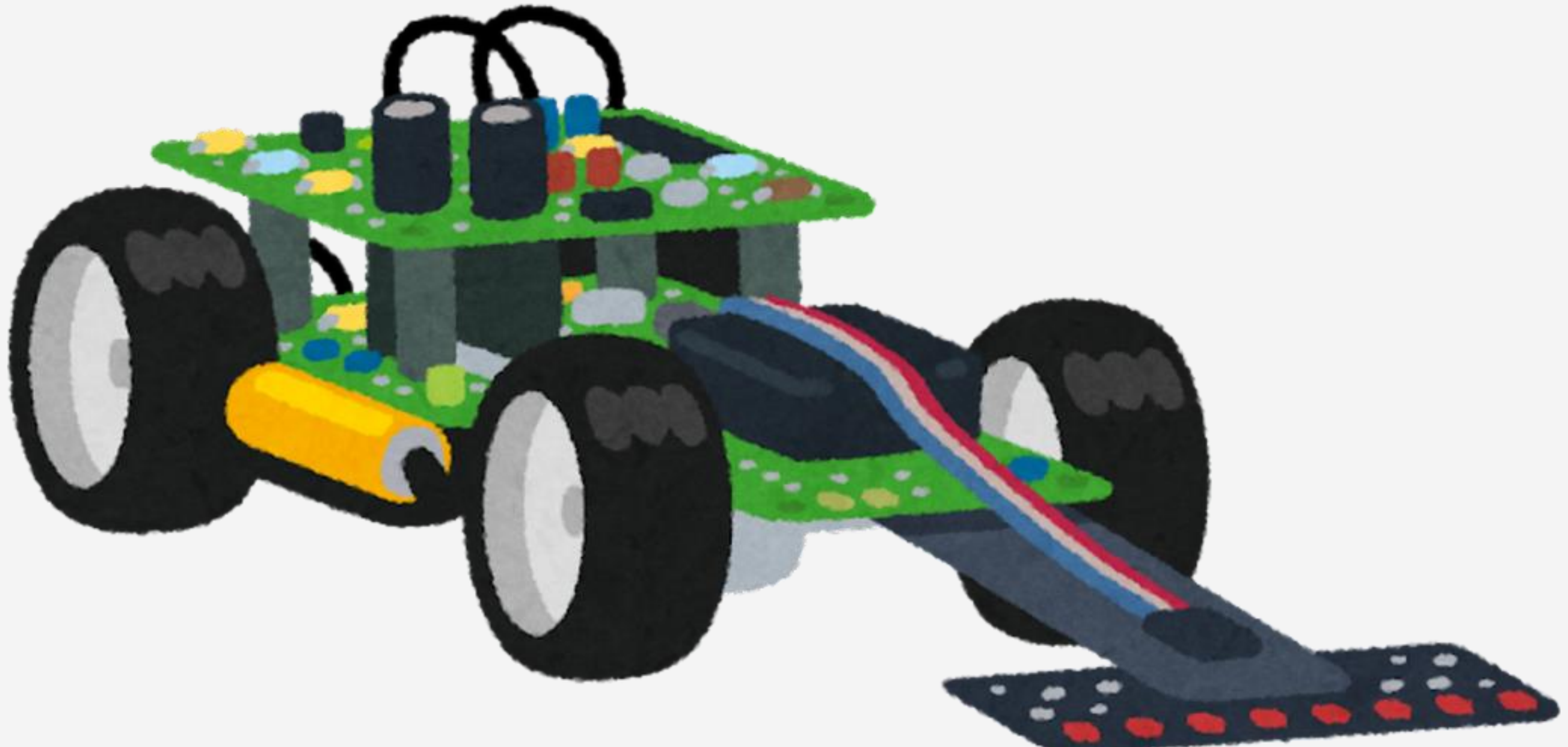


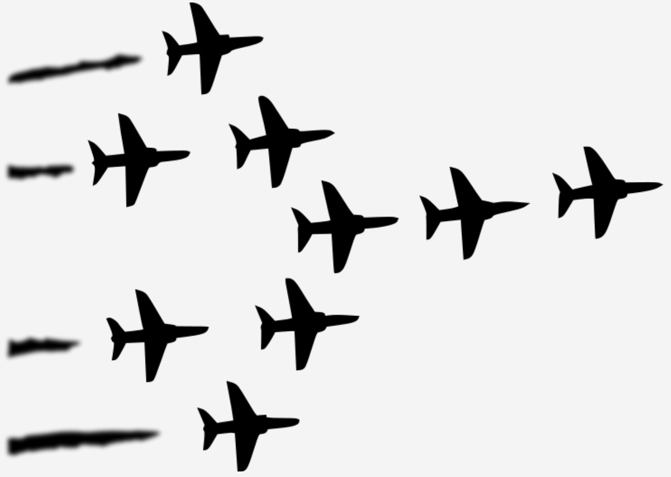
Sequence Diagram



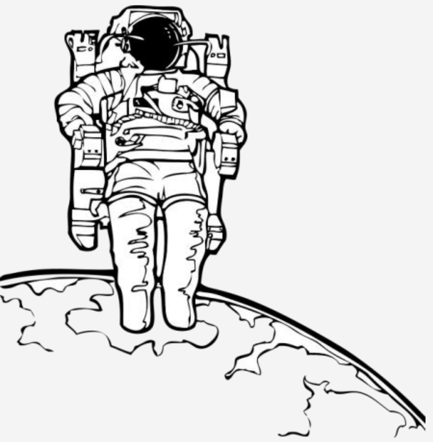


Applications





Military and Law Enforcement:



Space Exploration:

Future Work

- making application that offer live streaming ,car movement and servo movement at the same time

- Incorporating additional Item “Raspberry Pi4” for improving object detection system.

- Improved wireless communication protocols such as Wi-Fi 6 or 5G for more reliable and faster data transmission between the vehicle and the control system

- Incorporating additional sensors like ultrasonic sensors, LiDAR, or Infrared sensors can provide complementary data to enhance object detection and navigation.

Omar Mohamed

ANY QUESTIONS?

Mario Emad

Moataz Ibrahim

Thank You

Ali Farouk

Mohamed Eid

Nancy Khaled

Yasmeen Mahdy