Computer Vision Car

Rian Sen Majumder, Tahseen Rayhan, Luke Lam, Nehan Mohammed

Department of Technology, SATEC @ W.A Porter C.I.

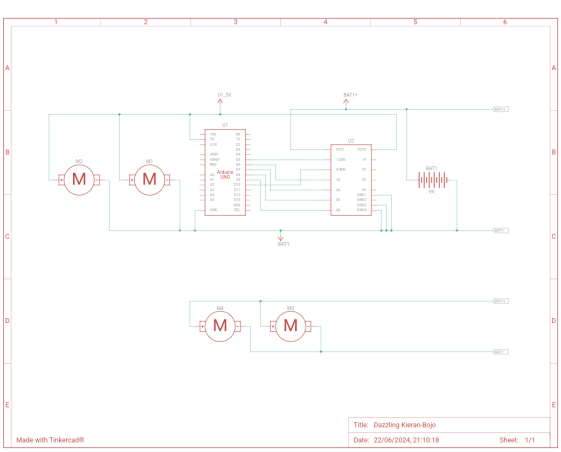
TEJ4M1: Computer Engineering

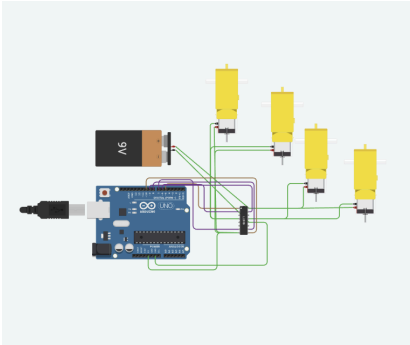
Mrs. Hana Patel

June 15, 2024

TinkerCAD Schematic:

https://www.tinkercad.com/things/k4vhEqFlVMR-dazzling-kieran-bojo/editel?sharecode=919j662hIPS5lcPoI4CWaeBI07BjqhsoedZTJSHfI1Q





Documented Arduino Code:

```
int ena = 5;              // Declaring the first/second enables, and input pins for motors
int in1 = 6;
int in2 = 7;
int in3 = 8;
int in4 = 9;
int enb = 10;

void setup() {
  Serial.begin(9600);

  pinMode(ena, OUTPUT);           // Declaring pin modes
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(enb, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  stopMotors();                   // Ensure motors are stopped at startup
}

void loop() {
  if (Serial.available() > 0) {
```

```
    String input = Serial.readStringUntil('\n');           // Receives data from the serial monitor

    // action for the car according to the data received
    if (input == "FORWARD") {
      moveForward();
    } else if (input == "STOP") {
      stopMotors();
    } else if (input == "BACKWARD") {
      moveBackward();
    } else if (input == "TURN_RIGHT") {
      turnRight();
    } else if (input == "TURN_LEFT") {
      turnLeft();
    }
  }
}

void moveForward() {

  // Start Motor A and B Clockwise and in the same direction
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  analogWrite(ena, 255);

  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  analogWrite(enb, 255);
}

void moveBackward() {

  // Start Motor A and B CCW and in the same direction
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(ena, 255);

  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(enb, 255);
}

void turnRight() {

  // Motor A CW and Motor B CCW
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
```

```
  analogWrite(ena, 255);

  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(enb, 255);
}

void turnLeft() {

  // Motor A CCW and Motor B CW
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(ena, 255);

  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  analogWrite(enb, 255);
}

void stopMotors() {

// turn off all the motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
```

Documented Python Code:

```
# importing the libraries
import cv2
import mediapipe as mp
import serial
import time
import math

# open the serial port that the Arduino is connected to
Arduino = serial.Serial('/dev/cu.usbmodem11101', 9600, timeout=0.1)

# initialize and start the camera for my laptop
wCam, hCam = 1240, 720
cam = cv2.VideoCapture(1)
```

```
cam.set(3, wCam)
cam.set(4, hCam)

class mpHands:
    def __init__(self, mode=False, modelComplexity=1, maxHands=2, TrackCon=0.5,
DetectCon=0.5):
        self.mode = mode
        self.modelComplexity = modelComplexity
        self.maxHands = maxHands
        self.TrackCon = TrackCon
        self.DetCon = DetectCon

        # using mediapipe to capture hand and map finger positions
        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands, self.modelComplexity,
                                        self.TrackCon, self.DetCon)
        self.mpDraw = mp.solutions.drawing_utils

    def Marks(self, frame):
        myHands = []
        handsType = []
        frameRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        results = self.hands.process(frameRGB)
        if results.multi_hand_landmarks:
            for hand in results.multi_handedness:
                handType = hand.classification[0].label
                handsType.append(handType)
            for handLandMarks in results.multi_hand_landmarks:
                myHand = []
                for landMark in handLandMarks.landmark:
                    h, w, c = frame.shape
                    myHand.append((int(landMark.x * w), int(landMark.y * h)))
                myHands.append(myHand)
        return myHands, handsType

    def getGesture(self, hand):       # detect which fingers are opened and closed
        fingers = []
        if hand:
            # Thumb
            if hand[4][0] > hand[3][0]:
                fingers.append(1)  # Thumb is open
            else:
                fingers.append(0)  # Thumb is closed

            # Other four fingers
            for i in range(8, 21, 4):
```

```
            if hand[i][1] < hand[i-2][1]:
                fingers.append(1)  # Finger is open
            else:
                fingers.append(0)  # Finger is closed
        return fingers

findHands = mpHands(maxHands=2)

while True:
    ret, frames = cam.read()
    if not ret:
        print("Failed to capture image")
        continue

    frame = cv2.flip(frames, 1)
    handData, handType = findHands.Marks(frame)

    command = 'NO_HAND\n'
    if handData:                            # detecting which fingers are up
        for hand in handData:
            fingers = findHands.getGesture(hand)
            if sum(fingers) == 5:  # All fingers are open
                command = 'FORWARD\n'
            elif sum(fingers) == 0:  # All fingers are closed
                command = 'STOP\n'
            elif fingers == [0, 0, 1, 0, 0]:  # Middle finger only
                command = 'BACKWARD\n'
            elif fingers == [0, 1, 0, 0, 0]:  # Pointer finger up
                command = 'TURN_RIGHT\n'
            elif fingers == [0, 1, 1, 0, 0]:  # Pointer and middle finger up
                command = 'TURN_LEFT\n'

    Arduino.write(command.encode())         # send the data to the arduino via serial

    # if 'q' pressed, then quit the program and destroy the camera instance
    cv2.imshow('my WEBcam', frame)
    if cv2.waitKey(1) & 0xff == ord('q'):
        break

cam.release()
cv2.destroyAllWindows()
```