



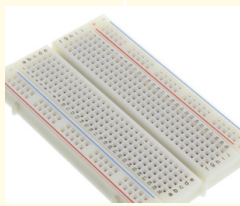
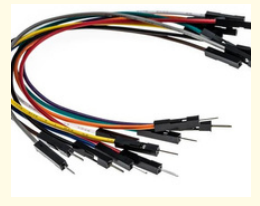
LED CONTROL USING FINGERS

Project Overview

The "LED Control Using Fingers" project aims to create an innovative and interactive system that allows users to control LED lights through hand gestures. This project leverages gesture recognition technology to provide a hands-free and user-friendly interface for controlling lighting systems.

In a world driven by technological advancements, the integration of interactive and intuitive interfaces has become increasingly essential. The "LED Control Using Fingers" project represents a pioneering venture into the realm of gesture-based control systems, offering a novel and engaging method for managing LED lighting. By harnessing the power of gesture recognition technology, this project seeks to redefine the way individuals interact with their surroundings, providing a seamless and hands-free experience in controlling LED lights.

Hardware Components

1**2****3****4**

- 1. Arduino UNO**
- 2. LED Light (5-pcs)**
- 3. Breadboard**
- 4. Jumper Wire**

Software Components

1. Visual Studio

- mediapipe
- cvzone
- pyfirmata

2. Arduino IDE

- firmata library

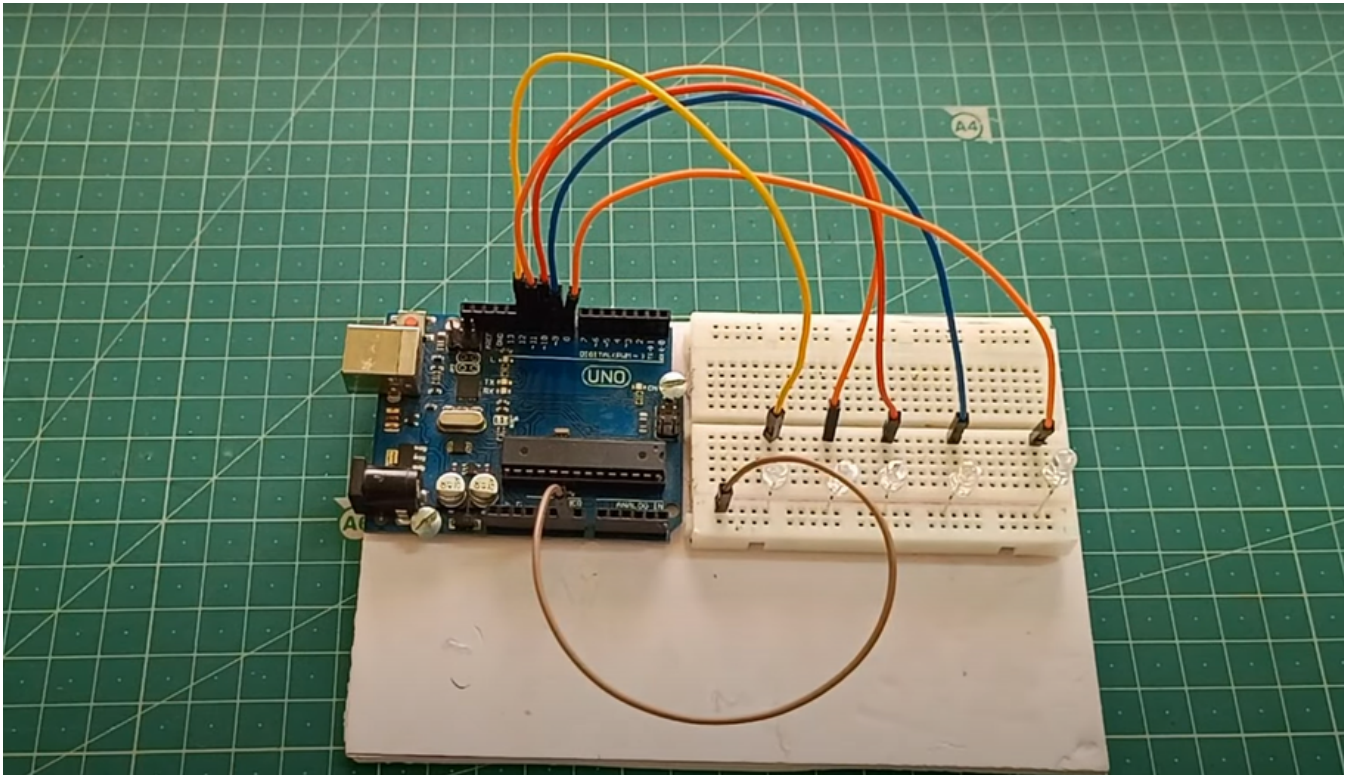
Connections

Arduino UNO - BreedBoard

- -ve To GND
-

5- LED Light - BreedBoard

- Long Leg (ANODE) +ve TO Centre
- Short Leg (CATHODE) +ve TO -ve side

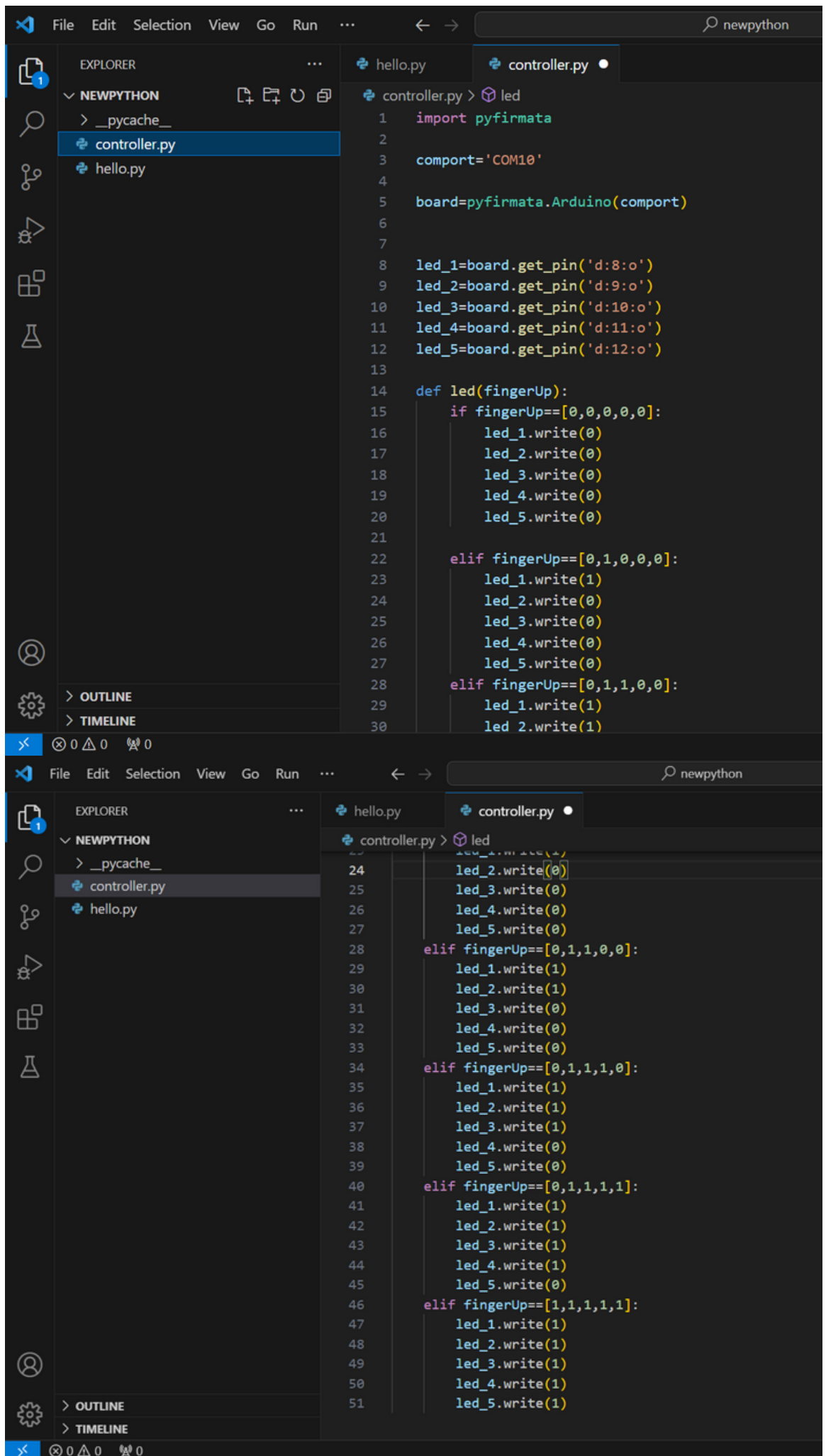


: Circuit Diagram

Usage

- **Home Automation:** Control lighting in different rooms with a wave of the hand.
- **Art Installations:** Create interactive and visually appealing light displays in public spaces or galleries.
- **Entertainment:** Enhance gaming experiences by incorporating gesture-based controls for ambient lighting.

Source Code :



The image displays two screenshots of a Visual Studio Code editor interface, showing the source code for a Python program that controls LEDs using a finger sensor. The top screenshot shows the initial setup, including importing the PyFirmata library, setting the COM port to 'COM10', and initializing the Arduino board. It also shows the definition of the 'led' function, which takes a 'fingerUp' list as input and writes values to five LEDs (led_1 to led_5) based on the sensor's state. The bottom screenshot shows a zoomed-in view of the 'led' function, highlighting the logic for different finger positions (e.g., [0,0,0,0,0], [0,1,0,0,0], [0,1,1,0,0], [0,1,1,1,0], [0,1,1,1,1], [1,1,1,1,1]) and the corresponding LED states (0 or 1).

```
File Edit Selection View Go Run ... newpython

EXPLORER
NEWPYTHON
  > __pycache__
  controller.py
  hello.py

controller.py > led
1 import pyfirmata
2
3 comport='COM10'
4
5 board=pyfirmata.Arduino(comport)
6
7
8 led_1=board.get_pin('d:8:o')
9 led_2=board.get_pin('d:9:o')
10 led_3=board.get_pin('d:10:o')
11 led_4=board.get_pin('d:11:o')
12 led_5=board.get_pin('d:12:o')
13
14 def led(fingerUp):
15     if fingerUp==[0,0,0,0,0]:
16         led_1.write(0)
17         led_2.write(0)
18         led_3.write(0)
19         led_4.write(0)
20         led_5.write(0)
21
22     elif fingerUp==[0,1,0,0,0]:
23         led_1.write(1)
24         led_2.write(0)
25         led_3.write(0)
26         led_4.write(0)
27         led_5.write(0)
28
29     elif fingerUp==[0,1,1,0,0]:
30         led_1.write(1)
31         led_2.write(1)
32         led_3.write(0)
33         led_4.write(0)
34         led_5.write(0)
35
36     elif fingerUp==[0,1,1,1,0]:
37         led_1.write(1)
38         led_2.write(1)
39         led_3.write(1)
40         led_4.write(0)
41         led_5.write(0)
42
43     elif fingerUp==[0,1,1,1,1]:
44         led_1.write(1)
45         led_2.write(1)
46         led_3.write(1)
47         led_4.write(1)
48         led_5.write(0)
49
50     elif fingerUp==[1,1,1,1,1]:
51         led_1.write(1)
52         led_2.write(1)
53         led_3.write(1)
54         led_4.write(1)
55         led_5.write(1)
```

File Edit Selection View Go Run ... newpython

EXPLORER

NEWPYTHON

> __pycache__

controller.py

hello.py

controller.py > led

```
24 led_2.write(0)
25 led_3.write(0)
26 led_4.write(0)
27 led_5.write(0)
28 elif fingerUp==[0,1,1,0,0]:
29     led_1.write(1)
30     led_2.write(1)
31     led_3.write(0)
32     led_4.write(0)
33     led_5.write(0)
34 elif fingerUp==[0,1,1,1,0]:
35     led_1.write(1)
36     led_2.write(1)
37     led_3.write(1)
38     led_4.write(0)
39     led_5.write(0)
40 elif fingerUp==[0,1,1,1,1]:
41     led_1.write(1)
42     led_2.write(1)
43     led_3.write(1)
44     led_4.write(1)
45     led_5.write(0)
46 elif fingerUp==[1,1,1,1,1]:
47     led_1.write(1)
48     led_2.write(1)
49     led_3.write(1)
50     led_4.write(1)
51     led_5.write(1)
```

> OUTLINE

> TIMELINE

hello.py

```
hello.py x controller.py
hello.py > ...
1 import cv2
2 import controller as cnt
3 from cvzone.HandTrackingModule import HandDetector
4
5 detector=HandDetector(detectionCon=0.8,maxHands=1)
6
7 video=cv2.VideoCapture(0)
8
9 while True:
10     ret,frame=video.read()
11     frame=cv2.flip(frame,1)
12     hands,img=detector.findHands(frame)
13     if hands:
14         lmList=hands[0]
15         fingerUp=detector.fingersUp(lmList)
16
17         print(fingerUp)
18         cnt.led(fingerUp)
19         if fingerUp==[0,0,0,0,0]:
20             cv2.putText(frame,'Finger count:0',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
21         elif fingerUp==[0,1,0,0,0]:
22             cv2.putText(frame,'Finger count:1',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
23         elif fingerUp==[0,1,1,0,0]:
24             cv2.putText(frame,'Finger count:2',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
25         elif fingerUp==[0,1,1,1,0]:
26             cv2.putText(frame,'Finger count:3',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
27         elif fingerUp==[0,1,1,1,1]:
28             cv2.putText(frame,'Finger count:4',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
29         elif fingerUp==[1,1,1,1,1]:
30             cv2.putText(frame,'Finger count:5',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
31
32     print(fingerUp)
33     cnt.led(fingerUp)
34     if fingerUp==[0,0,0,0,0]:
35         cv2.putText(frame,'Finger count:0',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
36     elif fingerUp==[0,1,0,0,0]:
37         cv2.putText(frame,'Finger count:1',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
38     elif fingerUp==[0,1,1,0,0]:
39         cv2.putText(frame,'Finger count:2',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
40     elif fingerUp==[0,1,1,1,0]:
41         cv2.putText(frame,'Finger count:3',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
42     elif fingerUp==[0,1,1,1,1]:
43         cv2.putText(frame,'Finger count:4',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
44     elif fingerUp==[1,1,1,1,1]:
45         cv2.putText(frame,'Finger count:5',(20,460),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),1,cv2.LINE_AA)
46
47     cv2.imshow("frame",frame)
48     k=cv2.waitKey(1)
49     if k==ord("q"):
50         break
51
52 video.release()
53 cv2.destroyAllWindows()
54
```

Arduino Code :

Step 1 :

Install firmata Library from,
INCLUDE LIBRARY ---> MANAGE LIBRARY

Step 2 :

File --> Examples --> Firmata --> Standard Firmata

Step 3 :


Connect the Port and Board.


Step 4 :


Upload the Code.


References :


tannukumari742/
Led_control_using_fingers




 1
Contributor

 0
Issues


 0
Stars

 0
Forks



tannukumari742/Led_control_using_fingers

Contribute to tannukumari742/Led_control_using_fingers development by creating an account on GitHub.

 GitHub