Seven Segment Controller

Abstract

In this project , we control seven segment display counter through 3 ways : hardware (push buttons and IR sensors) ,mobile application and web application . The project based on ESP32 with micro python firmware and code is written with python . Web application is connected with program using socket module . Push buttons and IR sensors are connected to pins with interrupt listener so we can control with buttons although socket blocking . We use IR sensors to control through hardware without touching . Mobile application is made using MIT inventor and control counter through accessing the web application in a hidden web browser included . Web application and mobile application can synchronize with each other and hardware by clicking refresh button on web and mobile app .

Hardware used

1. Breadboard 840pins * 2
2. ESP32 espressif 30pins * 1
3. Push Buttons * 3
4. IR Sensor Modules * 3
5. 7 Segment comn anode * 1
6. 1/8 watt 220 resistors * 7
7. 1/8 watt 10K resistors * 3
8. 18650 Lithium ion batteries * 2
9. 18650 2cell Battery Holder * 1
10. Jumpers

11. Micro USB cable * 1

Software used

1. Thonny IDE
2. Micro python firmware
3. MIT inventor

Get ready to start coding steps

1. Download Thonny IDE through **official website** and install it
2. Download USB to UART bridge though **official website** and install it
3. Download Micro python firmware v1.13 through **official website**
4. Burn Micro python firmware to ESP flash though these **steps**
5. Create account on MIT inventor through **This Link**

Project Code

```
 #_____________________The Project Team_____________________#

#_____________________________________________________________#
draw_digit=[[0,0,0,0,0,0,1]#0
    ,[1,0,0,1,1,1,1]#1
    ,[0,0,1,0,0,1,0]#2
    ,[0,0,0,0,1,1,0]#3
    ,[1,0,0,1,1,0,0]#4
    ,[0,1,0,0,1,0,0]#5
    ,[0,1,0,0,0,0,0]#6
    ,[0,0,0,1,1,1,1]#7
    ,[0,0,0,0,0,0,0]#8
    ,[0,0,0,0,1,0,0]]#9
zero_slow=[[0,1,1,1,1,1,1]
```

```
    ,[0,0,1,1,1,1,1]
    ,[0,0,0,1,1,1,1]
    ,[0,0,0,0,1,1,1]
    ,[0,0,0,0,0,1,1]
    ,[0,0,0,0,0,0,1]
    ,[0,0,0,0,0,0,1]]
nine_slow=[[1,1,1,1,1,1,0]
    ,[1,1,1,1,1,0,0]
    ,[0,1,1,1,1,0,0]
    ,[0,0,1,1,1,0,0]
    ,[0,0,0,1,1,0,0]
    ,[0,0,0,0,1,0,0]
    ,[0,0,0,0,1,0,0]]
data_pins=[17,5,18,19,21,22,23]#A__B__C__D__E__F__G#
input_pins=[33,32,35]#increase__decrease__reset#

#SET INPUT & OUTPUT PINS
from machine import Pin

for k in data_pins:
    Pin(k,Pin.OUT)
for k in input_pins:
    Pin(k,Pin.IN)
#_____________________________________________________________#
Counter=0#Initialize Counter
from time import sleep_ms

def draw(Counter):#Draw Counter on 7 Segment
    if Counter>=0 and Counter<=9:
        for k in range(7):#Draw Counter Normally
            Pin(data_pins[k],value=draw_digit[Counter][k])
        sleep_ms(500)
    elif Counter<0:
        for k in range(7):#Draw Nine Slow Motion
            for l in range(7):
                Pin(data_pins[l],value=nine_slow[k][l])
            sleep_ms(500)
    else :
        for k in range(7):#Draw Zero Slow Motion
            for l in range(7):
                Pin(data_pins[l],value=zero_slow[k][l])
            sleep_ms(500)
draw(0)#Initialize 7 Segment Dispaly to ZERO

def increase():#Increase Counter Function__is CALLED Any where we need
increase counter
    global Counter
```

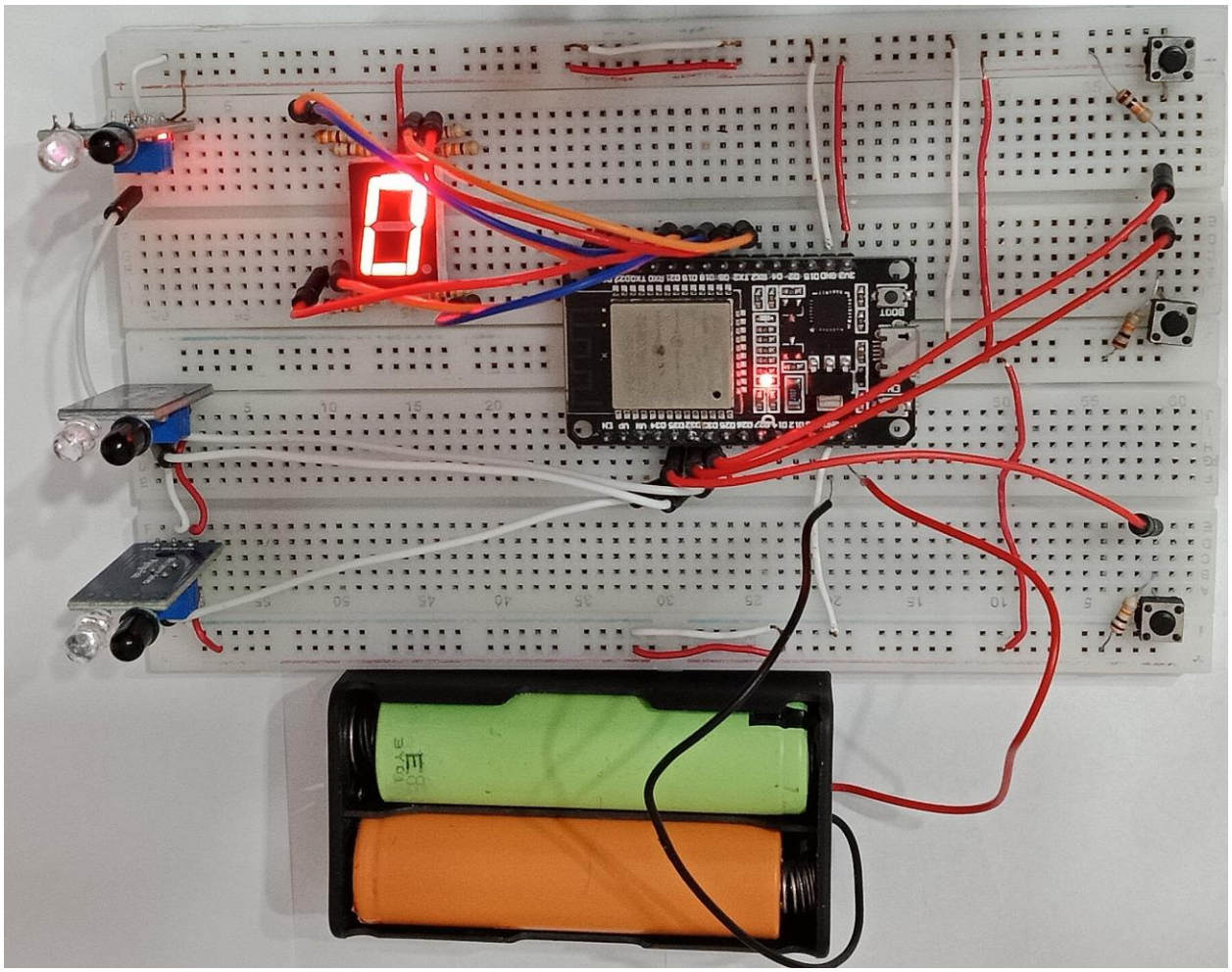
```
    Counter+=1
    draw(Counter)#Send Counter to show on 7 Segment
    if(Counter>9):
        Counter=0

def decrease():#Decrease Counter Function__is CALLED Any where we need
decrease counter
    global Counter
    Counter-=1
    draw(Counter)#Send Counter to show on 7 Segment
    if(Counter<0):
        Counter=9

def reset():#Reset Counter Function__is CALLED Any where we need reset
counter
    global Counter
    Counter=0
    draw(Counter)#Send Counter to show on 7 Segment

def debounce(pin):#Debouncing Function To Avoid Non Perfect Contact
    previous_value = None#Initial A Temp Variable
    for k in range(10):#Take 10 Samples of Signal
        current_value = pin.value()#Take Sample
        if previous_value != None and previous_value != current_value:
            return None #If Rippled Return None
        previous_value = current_value
    return previous_value#AFTER 10 Samples return New State

def increase_interrupt(pin):#Interrupt Routine for Increment
    d = debounce(pin)#Check Bouncing
    if not d:
        increase()#After 10 Samples Execute Increment

def decrease_interrupt(pin):#Interrupt Routine for Decrement
    d = debounce(pin)#Check Bouncing
    if not d:
        decrease()#After 10 Samples Execute Decrement

def reset_interrupt(pin):#Interrupt Routine for Reset
    d = debounce(pin)#Check Bouncing
    if not d:
        reset()#After 10 Samples Execute Reset
# SET Interrupts Pins
handlers=[increase_interrupt,decrease_interrupt,reset_interrupt]
for k in range(3):
    Pin(input_pins[k]).irq(trigger=Pin.IRQ_FALLING,
handler=handlers[k])
```

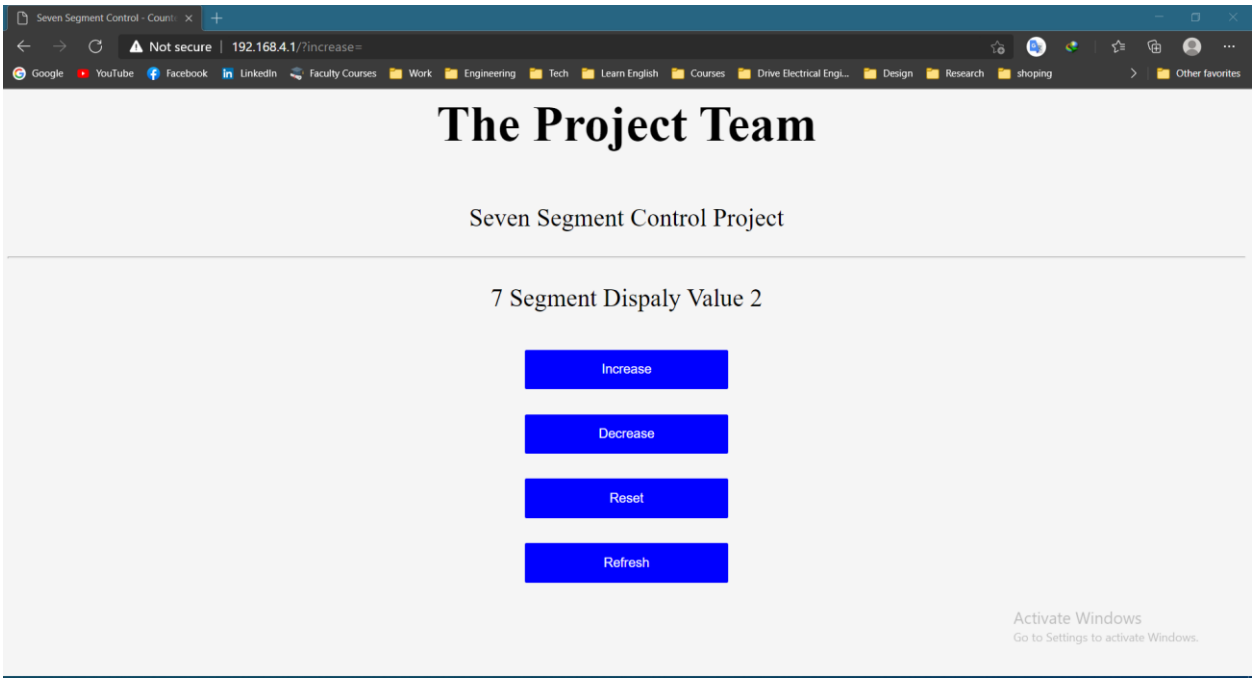
```
#_____________________________________________________________________#

def web_page(Counter):
    title="Seven Segment Control - Counter "+str(Counter)#Page Title
    html_page = """<html>
    <head>
        <title>""" + title + """</title>
    <meta content="width=device-width, initial-scale=1"
name="viewport"></meta>
    <style>
        .button {
            background-color:blue;
            width:250px;
            border:none;
            color:white;
            padding:15px 32px;
            margin:1vw;
            font-size:16px;
            }

    </style>
    </head>
       <center><body style="background-color:f5f5f5">
        <div>
            <p style="font-size:4vw;font-weight:bold;">The Project
Team</p>
            <p style="font-size:2vw">Seven Segment Control Project</p>
        </div>
        <hr/>
        <p style="font-size:2vw">7 Segment Dispaly Value """ +
str(Counter) + """</p>
        <div><form>
            <button class="button" name="increase" type="submit"
value="">Increase</button>
            <br>
            <button class="button" name="decrease" type="submit"
value="">Decrease</button>
            <br>
            <button class="button" name="reset" type="submit"
value="">Reset</button>
            <br>
            <button class="button" name="refresh" type="submit"
value="">Refresh</button>
        </form></div>
    </body></center>
</html>"""
```

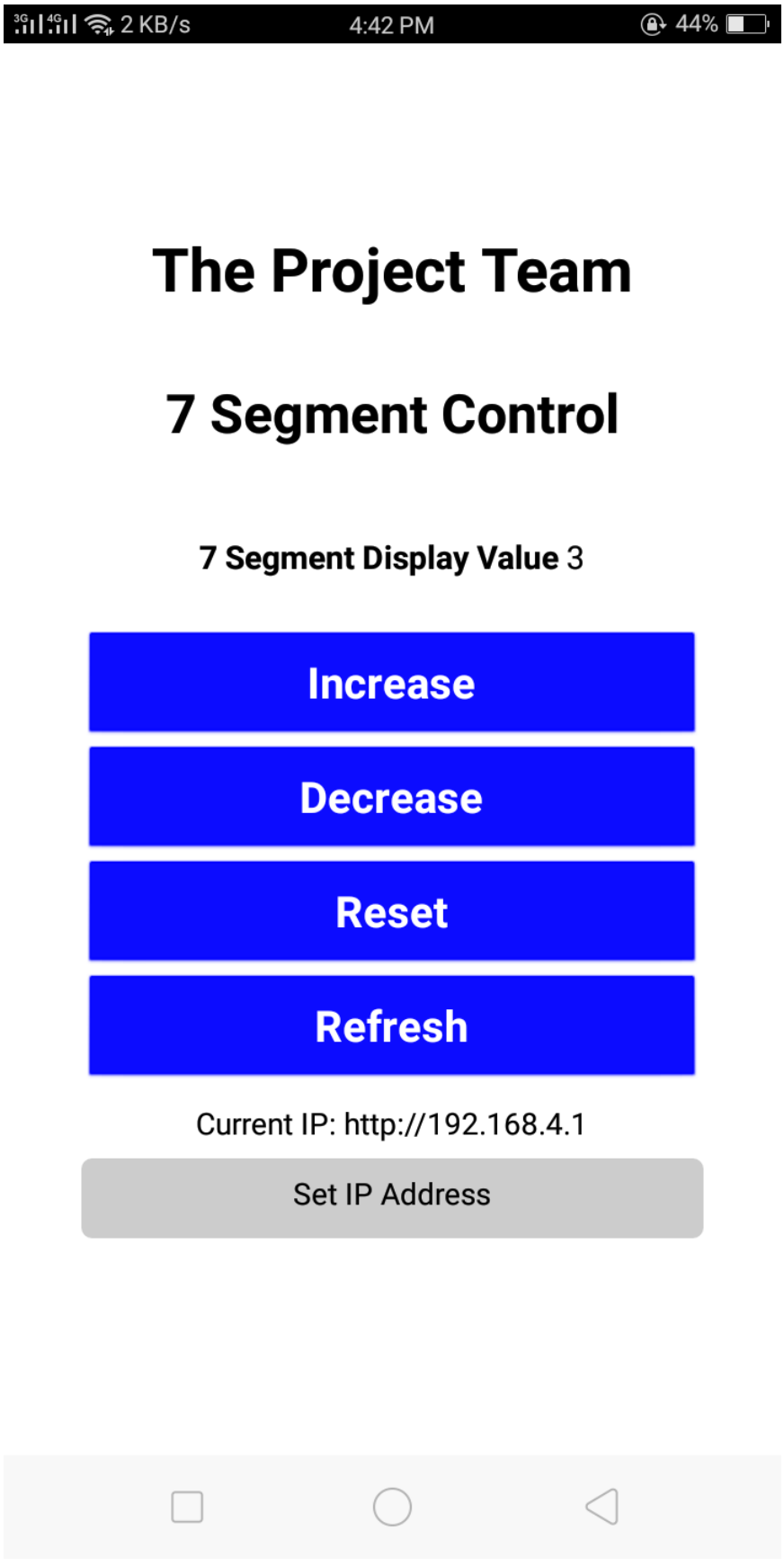
```
    return html_page

from network import WLAN , AP_IF#Import WirelessLAN and AccessPoint
Internet Family
WIFI = WLAN(AP_IF)#Create Object of WLAN Class as AccessPoint
#Configure Access Point Name , Encryption and Password
WIFI.config(essid='Seven Segment
Controller',password='7777*7777',authmode=4)
WIFI.active(True)#Turn Access Point on
while not WIFI.isconnected():
        pass#Don't Skip untill Connection Success
#Note That ESP IP is 192.168.4.1 in Default

from socket import socket,AF_INET,SOCK_STREAM
#Create Object of scoket Class
s = socket(AF_INET,SOCK_STREAM)#AddressFamily:IP v4 | TCP Protocol
s.bind(('',80))#Assign socket to ESP Address on Port 80 (HTTP PORT)
s.listen(10)#Start accepting TCP connections with maximum 10
connections

while(1):
    try:
        #Start Accepting New connection and make new accept object to
use and take client address and port
        connection,sender_address=s.accept()
        connection.settimeout(3)#Set Connection timeout to 3 Seconds
        request=connection.recv(1024)#recieve data with maximum 1024
Bytes
        connection.settimeout(None)#Unlimit Timeout
        request = str(request)#Cast Byte Object to String
        increase_request =request.find('GET /?increase')#Search for
increase parameter
        decrease_request = request.find('GET /?decrease')#Search for
decrease parameter
        reset_request = request.find('GET /?reset')#Search for reset
parameter

        #Excute operation according to the parameter found
        if(increase_request != -1):
            increase()

        elif(decrease_request != -1):
            decrease()

        elif(reset_request != -1):
            reset()
```

```
        #send web page after updating counter
        connection.sendall(web_page(Counter))
        connection.close()#close connection

    except :
        connection.close()#In case error close connection
```

Project Photos




Seven Segment Control - Counte
Not secure | 192.168.4.1/?increase=
Google
YouTube
Facebook
LinkedIn
Faculty Courses
Work
Engineering
Tech
Learn English
Courses
Drive Electrical Engi...
Design
Research
shoping
Other favorites
The Project Team
Seven Segment Control Project
7 Segment Dispaly Value 2
Increase
Decrease
Reset
Refresh
Activate Windows
Go to Settings to activate Windows.


2 KB/s
4:42 PM
44%
The Project Team
7 Segment Control
7 Segment Display Value 3
Increase
Decrease
Reset
Refresh
Current IP: http://192.168.4.1
Set IP Address

Contact

GitHub

LinkedIn

Engineeroo FB page

Engineeroo YouTube channel