EXPERIMENT NUMBER: 3

EXPERIMENT NAME: NEB SERVER IMPLEMENTATION USING RASPBURRY PI

DATE: 03/11/2022, THURSDAY

## \* MM:

Creation of Web server and its interactions using Flack Francowork on Rapperry Pi with SenseHAT.

## \* INTEGRATED DEVELOPMENT :

Name - Thonny 40.1 Publisher - Airas Annamaa support link - htgs; Il thorny . org

- (a) Web Server Implementation using Raspberry Pi

1 From sens hat import Sense HAT python module to control the Raspberry Pi sense HAT.

- No arguments defaults to OFF and toggle the UD matrix in low light made, useful if the Sense Hot is being used in a wark
- From flask import Flask, lightweight web Server Gateway
  Interface (WSGI) web application framework.

  B Flask constructor takes the name of current module (\_name\_)

as argument.

E Initialize the colours and define the modules for "keep ant Set', 'Oh To Come In Set' and 'Off rautes.

(E) map the CRLS to a specific function that will handle the lagic

for that URL.

1 Run the app in the main function.

-> Python lade from sense hat import Sense Hat sense = SenxHar() sense clear () senx. low-light - True from flack import Flack app = Flask (\_name \_) green = (0, 255, 0) ned = (255,0,0) nothing = (0,0,0) def siene-Leep-autl): R = red scene = [ R, R) return siene des scene-come-in (): 6 = green scene = [ 6, 6, 6, 6, 6, 6, 6, 6, 9, 6, 6, 6, 6, 6, 6, 6,

6, 6, 6, 6, 6, 6, 6, 6,

9, 9, 9, 9, 9, 9, 9, 9, 9,

St all

def scene - off ():

N = nothing

scene = E

N, N, N, N, N, N, N, N, N,

N, N, N, N, N, N, N, N,

N, N, N, N, N, N, N, N,

N, N, N, N, N, N, N,

N, N, N, N, N, N, N,

N, N, N, N, N, N, N,

N, N, N, N, N, N, N,

N, N, N, N, N, N, N, N,

N, N, N, N, N, N, N, N,

retorn scene

@ app. route ('1')

def hello-world():

return 'Lighting Scene Controller!'

@ app. soute ('/keep-aut')

def keep-aut():

sense set-pixels (scene-keep-aut())

return "keep aut set'

@ app. route ('Icome-in')

def come-in():

sense. set pinels (scene-come\_in())

neturn 'or To come In Set'

-> OUTPUTS:

172.17.176.47 : 5000/

Lighting Scene Controller!

(ASI I : H2. M. D6. 47: 5000 / Reop. out } Red LED wathin

(ASI II): 170. 11. 126. 47: 5000 / come - in ) green LED waterin

Ok To come In Set

CASE IV: 172.17. 106.47; 3000/off & Default LED Wathin

I (needing)

@ app route ('Toff') def off(): sense set-pixels (scene-off ()) return off

# pi@ Raspberrypis ~ \$ &f config # ethe: flags = 4163 (UP, BROADCAST, RUNNING, MULTICAST) mtu 1500 thet [172.17.126.47] Netmask 255.255.0 broadcast \_\_ enet b fest :: 9d5a: ccdd: c209: ebab prefishen 04 \_ .

TX errors o dropped o oversuns o carrier o callisions o #

# Driver Code - Mais Function: f\_name\_ = \_ main\_ '; app. hun (hast = 172.17.126.47) # Part Number - 5000

(6) Web server based Weather Station using Raybury Pi with Sensethat

@ From sense-har impart Sensettat, python madule to control the

Raspberry Pi Sense HAT.

O No arguments defaults to OFF. Taggle the HED mathin in law light mode, useful if the Sense HAT is being used in a dark environment. environment.

O From flask import Flask, a lightneight web Server Gateway
Interface (WSGI) web application framework.

O From flask import render template, used to generate autput from a template file shat is found in she application is templates

tolder.

The flash constructor takes the name of current module [ name as argument.

Desport time, the python module providing various time-related functions. Initialize the tent and background colours for the IFD matrix display.

map the URLS to a specific function that will handle the logic

for that URL.

@ Gets the ourrent pressure in Millibars from the pressure sensor.

Return the time in seconds since the epoch as a floating point number and convert the time in seconds since the epoch to a string of a form: "Sun Tem 20 23:21:05 1993" representing local time.

Print the message in Thomas shell and show the same in the HD mathin with appropriate scrall speed and multable tent and

background colour

@ Run the app in the main praction.

Fythen Cade from sense\_hat import SenseHat
sense = SenseHat ()
sense. clear()
sense. low\_light = True

from flask import Flask from flask import nender-template app = Flask (\_name.\_)

import time

green = (0,255,0)
nothing = (0,0,0)

@app.route ('1')
def inden ():

```
pressure = sense get_pressure ()

temperature = sense get_temperature ()

seconds = time · time ()

local_time = time · clime (seconds)
```

print\_nessage = str ("Weather Manitoring System 1" + "In" +
"current fressure: " + str (round (pressure, 2)) + "millibars" + 'b" +
"Temperature: " + str (round (temperature, 2)) + "degree(s)" + "In" +
" tocal Time: " + str (local\_time))

senx shew-message ("current Pressure: {} Temperature: {} }

Letal Time: {} '. format (pressure, temperature, local-time),

stroll-speed = 0.005,

text-colour = green,

back-colour = nothing)

neatherData = ['weather': print\_message]
return render\_template ('index. html', ## weatherData)

# pi@ raspberrypi: ~ \$ if config

# etho: flags = 4163 < UP, BROADCAST, RUNNING, MULTICAST > mtu 1500

# unet \[ \frac{172.17.126.47}{196.47} \] netmark 255.255.0 ...

# ineth fe80: 9d5a: cedd: c209: elab prefixlen 64...

# ....

TX errors o dropped o everuns o carrier o ....

# Driver Cade: - Main Function:

#

app. run (hast = 172.17.126. 47') # Port Number - 5000

```
templates Folder: inder html-
 < html >
     < head >
         k set : "stylesheet" href : "/static/style . css' />
     < shead >
     < body>
         < h1> Weather: </h1>
         < hr class = "title" > { [weather]} < /hz>
    < Shody>
</html>
static Folder: - style ess -
. title [
     font family: Roboto; color: #FFFFF;
 3
bredy &
background: navy;
color: yellow;
fant-family; Arial, sans-serif;
tert-align: center;
RI E
border - bottom: I pr solid gold;
margin - battom: 40 pr
padding: 10 px;
```

index. html weather: fineacher 3) is by a somety and asserting about the contract of a new come of position is point a competition of the second of

in a proceeding of front of many in the second of the contract of the second of the se But Mark We sty wood for it was . and the second of the second of the second

- () Neb Server based Remote Device Control using Rapperry Pi with Sense HAT:
- -> Algorithm -
- 1 From sense hat import Sensetlat, pythen module to control the Raspberry Pi Sonce HAT.
- O No argument diefaults to OFF and toggle the IFD mathia in low light made, useful of the Sense HAT is being used in a dark environment.
- 3 From flask import Flash, a lightweight Web server Gateway Interface (WSGI) web application francework.
- From flark import render-template, used to generate autput from a template file that is found in the application's templates fælder.
- 1 Flack constructor takes the pame of current madule (\_rame\_) as argument.
- @ map the URLs to a specific function that will handle the lagic for that URL.
- 9 Inden () definition is Set the desice status to OFF and show the letter 'I' in the HD matrix.
  - (ii) store the dence name and status in a dictionary and neturn the same along with the html as parameter in render-tomplated,
  - action () definition (i) For the required device name and action (on/off), set the corresponding device status and show the message on the LED matrix.
  - (81) store the device name and status in a dictionary and neturn the same along with the inden. Atml as parameter in render - template ().
- @ dearl) definition (3) Set the device status as OFF.
  - (b) clear the sense HAT display mataix.
  - (iii) store the donce name and status in a dictionary and return the same along with index thoul as farameter in render-template ().

```
-> Python Codo -
   from sense-hat import Sensethat
   sense = sensetlat ()
   sense · cleans)
   sense low_ light = True
   from flask import Flash
   from feask import render - template
   ago = Flask (-rame -)
    @ app route ('/)
    def index ():
        deniests = "OFF"
        sense . show_ letter ("I")
        tenglate Data = ['device 1': devicests }
        return render_template ('inden html', ** template Data)
    of return render _ template ("index Atml", It weather Data)
    @ agg. reute (" Kderice Name > / < action > ")
    def action (deviceName, action):
        if device Name == 'dl' and action == 'on':
            dencests = "ON"
             sense. show_letter ("o") # ly Arrow
        If device Name == 'dl' and action == 'off':
             dence 8ts = "OFF"
             sense show letter ("F") # Up Arrow
        template Data = ['dence!'; dence Sts }
        neturn render template l'inder Atml, At template Data)
      @ app. route ('Soff')
      def dear():
         dericests : "OFF"
          sense. clear ()
```

template Data = {'device!': device Sts }

neturn render - template ('inden html', \*\* template Data)

# Driver Code - Main Function:
# \_ name \_ == '\_ main \_ '

app. nun (hest = \$172.17.126.471) # Nort Number - 5000

Static Falder: style.css 
body {

background: blue;

color: yellaw;

button {

font: bold 15 px Arial;

text-decoration: none;

background-color: #EFFFFF;

color: #333333

padding: 2pn 6pn 2pn 6px;

border-top: 1pn solid #CCCCCC;

border-right: 1pn solid #333333;

border-bottom: 1pn solid #333333;

border-left: 1pn solid #cccccc;

< link nel = "stylesheet" href = "/static/style.css" >> </kead>

<hr/>

OUTPUTS 3

index html soll go the service as a service as Derice Status

Derice 1 ==> {{ dence 1}}

Commands Derice CERT :-> Derice on Derice 1 OFF

The second of the second

< br>>

< h2> Commands </h1> (h3) Dence Ctrl >2> La Bref = " IdV/on " class = "button"> Device I ON x/a> (a Kref = "Salloff" class = "button"> Device 1 OFF </a> </ R3> </body>

S/Rtml>

\* RESULT:

Thus, implemented web server and its interactions using Flask Framework on Raspherry Pi with SensettAT. Thus, all the simulation results were verified successfully.