МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра медіаінженерії та інформаційних радіоелектронних систем

Дисципліна «Розробка мікропроцесорних пристроїв "розумного дому"»

Лабораторна робота №1

**«**Програмування ліній вводу-виводу Raspberry PI**»**

|  |  |  |
| --- | --- | --- |
| Виконав:  Коваленко І.Р  ст. гр. ІТШІ-19-1 |  | Прийняв:  доц. каф. МТС  О.В. Зубков |

Харків – 2020

Лабораторна робота №3.

Створення локального Web сайту

1.1Мета роботи

Оволодіння технологією створення Web сайту за допомогою серверу Flask

**Бригада №1 Мирошник, Коваленко**

Текст програми server.py

from flask import Flask

from flask import json

from flask import render\_template

from flask\_socketio import SocketIO

from heater import \*

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = 's1ckret'

socketio = SocketIO(app)

fileJSON = ''

def startDummyHeaterThread():

global heater\_map

print("Heater controller start")

threading.Timer(1.0, startDummyHeaterThread).start()

for heater in heater\_map.values():

if heater.isOn == 1:

if int(heater.curTemp) >= int(heater.maxTemp):

heater.turnOff()

else:

if int(heater.curTemp) < 100:

heater.curTemp = heater.curTemp + 1

else:

if heater.curTemp > 0:

heater.curTemp = heater.curTemp - 1

msg = {}

msg["roomName"] = heater.roomName

msg["varName"] = "temperature"

msg["varValue"] = heater.curTemp

json\_msg = json.dumps(msg)

socketio.emit('update', json\_msg)

msg["roomName"] = heater.roomName

msg["varName"] = "isHeatingOn"

msg["varValue"] = heater.isOn

json\_msg = json.dumps(msg)

print(json\_msg)

socketio.emit('update', json\_msg)

@app.route('/')

def hello\_world():

file = open("./cfg/rooms.json")

fileStr = file.read()

global fileJSON

fileJSON = json.loads(fileStr)

heaterInit(fileJSON)

startDummyHeaterThread()

return render\_template("controls.html", rooms=fileJSON)

@socketio.on('message')

def handle\_message(message):

print('received message: ' + message)

@socketio.on('sync')

def handle\_my\_custom\_event():

print('Start synchronizing...')

global fileJSON

socketio.emit('sync', fileJSON)

print('Synchronizing completed.')

@socketio.on('update')

def handle\_message(json\_msg):

print('Receive update from a client: ' + json\_msg)

fileJSON = json.loads(json\_msg)

heaterUpdate(fileJSON);

socketio.emit('update-confirm', json\_msg)

print('Emit update-confirm to a client: ' + json\_msg)

@socketio.on('update-confirm')

def handle\_message(json):

print('Receive update-confirm from a client: ' + json)

if \_\_name\_\_ == '\_\_main\_\_':

socketio.run(app, debug=True)

Текст програми heater.py

import threading

class Heater:

def \_\_init\_\_(self, roomName, curTemp, maxTemp):

self.roomName = roomName

self.curTemp = curTemp

self.maxTemp = maxTemp

self.isOn = 0

def turnOn(self):

self.isOn = 1

print("The heater in {} is On({}) to {} degrees".format(self.roomName, self.isOn, self.maxTemp))

def turnOff(self):

self.isOn = 0

print("The heater in {} is Off({})".format(self.roomName, self.isOn))

heater\_map = {}

def heaterInit(json):

global heater\_map

for room in json:

heater\_map[room["roomName"]] = Heater(room["roomName"], room["temperature"], room["maxTemperature"])

if room["isHeatingOn"] == 1:

heater\_map[room["roomName"]].turnOn()

else:

heater\_map[room["roomName"]].turnOff()

def heaterUpdate(json):

global heater\_map

print(heater\_map)

if json["varName"] == "isHeatingOn":

if json["varValue"] == 1:

print("I AM ON")

heater\_map[json["roomName"]].turnOn()

else:

print("I AM OFF")

heater\_map[json["roomName"]].turnOff()

elif json["varName"] == "maxTemperature":

heater\_map[json["roomName"]].maxTemp = json["varValue"]

Текст web сторінки

<!DOCTYPE html>

<html lang="en">

<head>

<link rel= "stylesheet" type= "text/css" href= "{{ url\_for('static',filename='styles/style.css') }}">

<link rel= "stylesheet" type= "text/css" href= "{{ url\_for('static',filename='styles/style-toggle-switch.css') }}">

<link rel= "stylesheet" type= "text/css" href= "{{ url\_for('static',filename='styles/style-slider.css') }}">

<script src="//cdnjs.cloudflare.com/ajax/libs/socket.io/2.2.0/socket.io.js" integrity="sha256-yr4fRk/GU1ehYJPAs8P4JlTgu0Hdsp4ZKrx8bDEDC3I=" crossorigin="anonymous"></script>

<script src="{{url\_for('static', filename='js/statuses.js')}}"></script>

<script src="{{url\_for('static', filename='js/connection.js')}}"></script>

<script src="{{url\_for('static', filename='js/componets.js')}}"></script>

</head>

<body>

<h1>Smart House</h1>

<div class="apartment">

{% for room in rooms %}

<div class="room">

<div class="room-title">"{{room.roomName}}"</div>

<div class="room-control-panels-collection">

<div class="room-control-panel">

<div class="room-control-panel-title">Light control panel</div>

<div class="room-control-component">

<span class="room-control-component-title">Light switch: </span>

<label class="switch">

<input type="checkbox" data-roomName="{{room.roomName}}" data-varName="isLightOn"

{% if room.isLightOn %}

checked

{% endif %}>

<span class="switch-slider round"></span>

</label>

</div>

<div class="room-status-component" data-roomName="{{room.roomName}}" data-varName="isLightOn">

The light is

<span>{{room.isLightOn}}</span>

</div>

<div class="room-control-component">

<span class="room-control-component-title">Light brightness: </span>

<div class="slidecontainer">

<input type="range" min="0" max="100" value="{{room.lightBrightness}}" class="slider" data-roomName="{{room.roomName}}" data-varName="lightBrightness">

</div>

</div>

<div class="room-status-component" data-roomName="{{room.roomName}}" data-varName="lightBrightness">

The light brightness is

<span>{{room.lightBrightness}}</span>

</div>

</div>

<div class="room-control-panel">

<div class="room-control-panel-title">Temperature control panel</div>

<div class="room-control-component">

<span class="room-control-component-title">Heating switch: </span>

<label class="switch">

<input type="checkbox" data-roomName="{{room.roomName}}" data-varName="isHeatingOn"

{% if room.isHeatingOn %}

checked

{% endif %}>

<span class="switch-slider round"></span>

</label>

</div>

<div class="room-status-component" data-roomName="{{room.roomName}}" data-varName="isHeatingOn">

The heating is

<span>{{room.isHeatingOn}}</span>

</div>

<div class="room-control-component">

<span class="room-control-component-title">Max temperature: </span>

<div class="slidecontainer">

<input type="range" min="0" max="100" value="{{room.maxTemperature}}" class="slider" data-roomName="{{room.roomName}}" data-varName="maxTemperature">

</div>

</div>

<div class="room-status-component" data-roomName="{{room.roomName}}" data-varName="maxTemperature">

The max temperature is

<span>{{room.maxTemperature}}</span>

</div>

<div class="room-status-component" data-roomName="{{room.roomName}}" data-varName="temperature">

The temperature is

<span>{{room.temperature}}</span>

</div>

</div>

</div>

</div>

{% endfor %}

</div>

</body>

</html>

Текст CSS файлу style.css

.apartment {

display: flex;

flex-wrap: wrap;

justify-content: space-around;

}

.room {

display: block;

padding: 12px 18px;

margin-bottom: 6px;

background-color: #C6C6C6;

margin: 12px;

}

.room-title {

font-size: 48px;

margin-bottom: 6px;

}

.room-control-panels-collection {

display: flex;

}

.room-control-panel {

background-color: #A2C8C8;

padding: 12px 18px;

}

.room-control-panel-title {

font-size: 32px;

white-space: nowrap;

margin-bottom: 10px;

}

.room-control-component {

background-color: #ABCDEF;

margin-bottom: 6px;

padding: 6px;

}

.room-control-component-title {

vertical-align: middle;

font-size: 24px;

}

.room-status-component {

font-size: 24px;

}

Текст CSS файлу style-slider.css

/\* Source: https://www.w3schools.com/howto/howto\_js\_rangeslider.asp \*/

/\* Example:

<div class="slidecontainer">

<input type="range" min="0" max="100" value="50" class="slider">

</div>

\*/

.slidecontainer {

width: 100%; /\* Width of the outside container \*/

}

/\* The slider itself \*/

.slider {

-webkit-appearance: none; /\* Override default CSS styles \*/

appearance: none;

width: 100%; /\* Full-width \*/

height: 25px; /\* Specified height \*/

background: #d3d3d3; /\* Grey background \*/

outline: none; /\* Remove outline \*/

opacity: 0.7; /\* Set transparency (for mouse-over effects on hover) \*/

-webkit-transition: .2s; /\* 0.2 seconds transition on hover \*/

transition: opacity .2s;

}

/\* Mouse-over effects \*/

.slider:hover {

opacity: 1; /\* Fully shown on mouse-over \*/

}

/\* The slider handle (use -webkit- (Chrome, Opera, Safari, Edge) and -moz- (Firefox) to override default look) \*/

.slider::-webkit-slider-thumb {

-webkit-appearance: none; /\* Override default look \*/

appearance: none;

width: 25px; /\* Set a specific slider handle width \*/

height: 25px; /\* Slider handle height \*/

background: #4CAF50; /\* Green background \*/

cursor: pointer; /\* Cursor on hover \*/

}

.slider::-moz-range-thumb {

width: 25px; /\* Set a specific slider handle width \*/

height: 25px; /\* Slider handle height \*/

background: #4CAF50; /\* Green background \*/

cursor: pointer; /\* Cursor on hover \*/

}

Текст CSS файлу style-toggle-switch.css

/\* Source - https://www.w3schools.com/howto/howto\_css\_switch.asp \*/

/\* Examples:

<!-- Rectangular switch -->

<label class="switch">

<input type="checkbox">

<span class="slider"></span>

</label>

<!-- Rounded switch -->

<label class="switch">

<input type="checkbox">

<span class="slider round"></span>

</label>

\*/

/\* The switch - the box around the slider \*/

.switch {

position: relative;

display: inline-block;

width: 60px;

height: 34px;

margin: 6px;

vertical-align: middle;

}

/\* Hide default HTML checkbox \*/

.switch input {

opacity: 0;

width: 0;

height: 0;

}

/\* The slider \*/

.switch-slider {

position: absolute;

cursor: pointer;

top: 0;

left: 0;

right: 0;

bottom: 0;

background-color: #ccc;

-webkit-transition: .4s;

transition: .4s;

}

.switch-slider:before {

position: absolute;

content: "";

height: 26px;

width: 26px;

left: 4px;

bottom: 4px;

background-color: white;

-webkit-transition: .4s;

transition: .4s;

}

input:checked + .switch-slider {

background-color: #2196F3;

}

input:focus + .switch-slider {

box-shadow: 0 0 1px #2196F3;

}

input:checked + .switch-slider:before {

-webkit-transform: translateX(26px);

-ms-transform: translateX(26px);

transform: translateX(26px);

}

/\* Rounded sliders \*/

.switch-slider.round {

border-radius: 34px;

}

.switch-slider.round:before {

border-radius: 50%;

}

Скріни результатів роботи програми









