Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektonsko poslovanje

TEMA PROJEKTNOG RADA

LED Strip Traka

Nastavnik: dr Aleksandra Labus Studenti:

Saradnik: Tamara Naumović Milan Gligorijević 297/16

Ajša Baković 195/16

Janko Gašić 134/16

Aleksandra Vučičević 126/16

Beograd, 2020.

LED Strip traka

# 1. Projektni zadatak

LED Strip Traka :

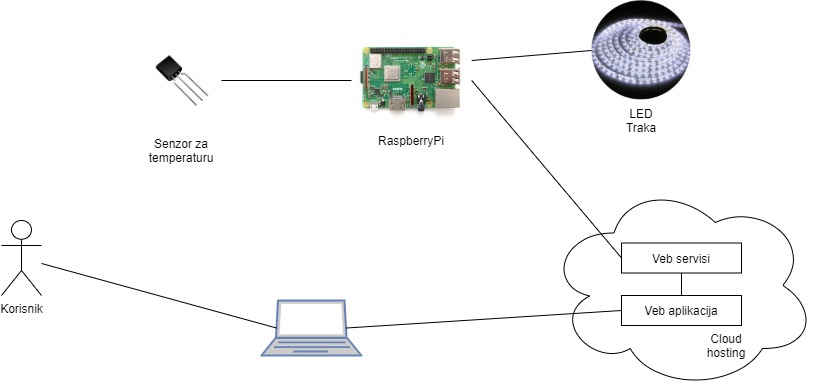
Napraviti aplikaciju za upravljanje LED strip trakom, po ugledu na:  <https://pimylifeup.com/raspberry-pi-led-strip-apa102/>

-  projektovati scenarije rada trake i kroz aplikaciju omogućiti izbor jednog od predefinisanih scenarija

-  omogućiti automatsko prilagođavanje boje LED trake, na osnovu temperature okruženja, izmerene pomoću temperaturnog senzora.

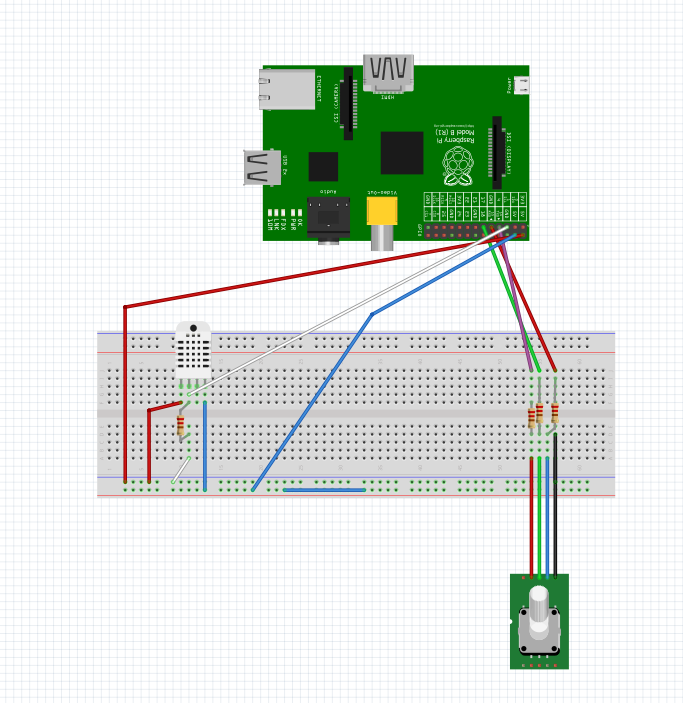
# 2.Rešenje zadatka

## 2.1 Projektovanje i arhitektura sistema



Slika 1: Arhitektura sistema 1

Na slici je dat grubi prikaz projektovanog sistema, uz ključne hardverske komponente korišćene na samom projektu. Pored prikaza arhitekture, na sledećoj slici dat je i nacrt povezivanja uredjaja i senzora potrebnih za realizaciju zahteva. Prvobitni fokus biće na samom hardverskom sklopu, a zatim na realizaciji potrebnih funkcija putem python-a i adekvatnih web alata.

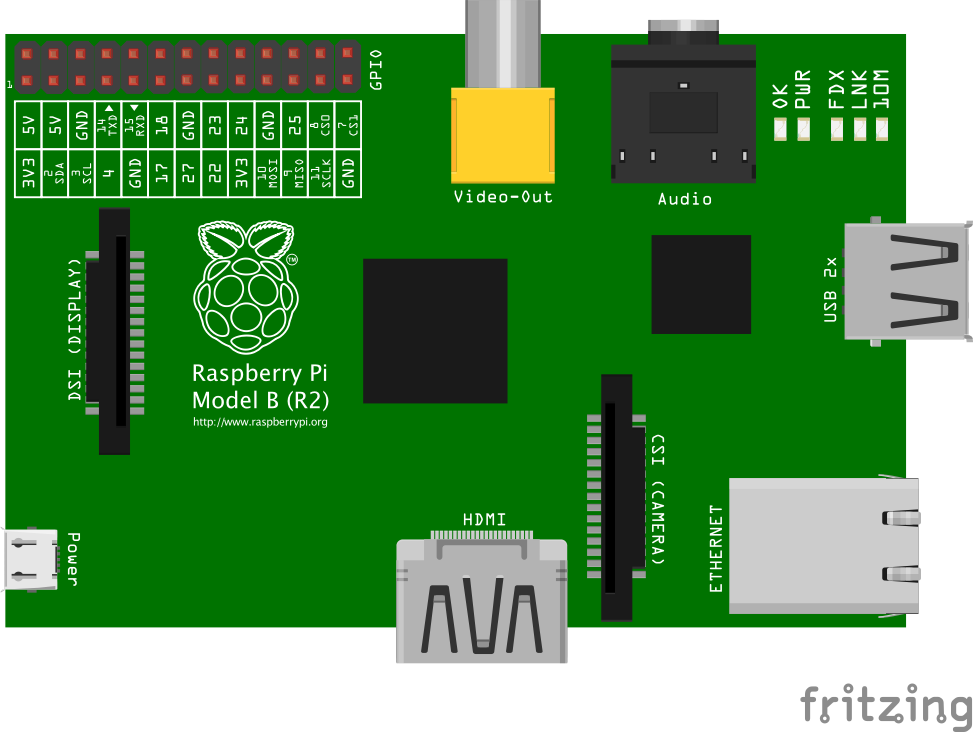


Slika 2: Prikaz povezanih hardverskih komponenti

Prethodna slika nacrtana je uz pomoć Fritzing alata. Ključne komponente koje se mogu odmah uočiti jesu RaspberryPi, senzor za merenje temperature i vlažnosti vazduha, led dioda (zamena usled nedostatka LED trake), otpornici, kao i brojni kablovi potrebni za direktno povezivanje. Data led lampica ima 4 pina pri čemu su tri rezervisana za RGB boje, a četvrti pin za ground. Senzor korišćen pri radu jeste DHT11 senzor sa 4 pina (konkretan prikaz dat je na slici 5). Kako prilikom povezivanja ne bi došlo do pregorevanja nekih od elemenata, korišćeni su potrebni otpornici.

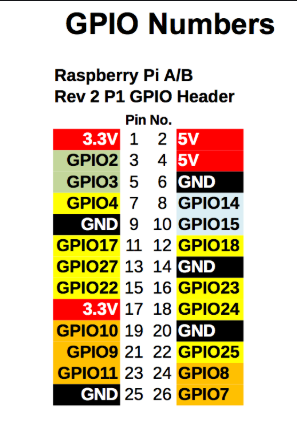
Različite boje kablova korišćene su radi lakšeg i preglednijeg snalaženja.

Raspored pinova na Raspberry Pi modelu B bliže je prikazan na slici 4.

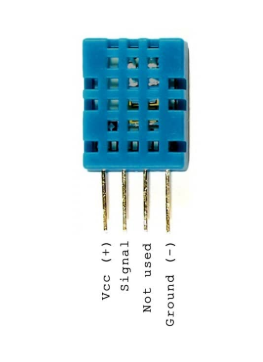


Slika 3: Prikaz RPi model B

Pinovi idu redom od spolja ka unutrašnjosti RaspberryPi-a, gledano vodoravno kao na prethodnoj slici.



Slika 4: Raspored pinova na RPi modelu B



Prilikom izrade ovog projekta korišćen je DHT11 senzor za merenje temperature i vlažnosti vazduha. U nastavku fokus je na temperaturi očitanoj pomoću njega, međutim radi lakšeg rada i manipulisanja vrednostima, upotrebljivani su i podaci o vlažnosti vazduha.

Slika 5: Prikaz DHT11 senzora

## 2.2 Pametni uređaji

Sledi detaljan spisak korišćenih uređaja za razvoj IoT sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Naziv komponente** | **Izgled komponente** | **Opis** | **Količina** |
| Raspberry Pi mikroračunar | rpi.png | Model B ili B+ | 1 |
| Proto ploča | 12002-01-500x500 | Za povezivanje elektronskih komponenti | 1 |
| Kablovi | https://content.solarbotics.com/products/photos/03e0f1ccebb02b4dc5cc17e395d3049b/lrg/45040-dscn0624.jpg | Muško-Muški | 5 |
| Kablovi | Untitled-2.png | Muško-Ženski | 10 |
| MicroUSB punjač | http://www.world-of-accessories.co.uk/images/user/11-120313233630.jpg | 2A | 1 |
| LED traka/LED lampica | Thomsen STRIP-1M-144-RGB-IP68 LED strip EEC: A++ (A++ - E) + plug ...led lampica.PNG |  | 1 |
| Napajanje za LED traku | led-napajanje-12v-3a-36w | 12v | 1 |
| Senzor temperature i vlažnosti | dht11.jpg | DHT11 | 1 |
| Otpornik |  | 4.7kΩ | 4 |

Tabela 1: Prikaz komponenti sistema

## 2.3 Scenario

Za potrebe ovog projekta potrebno je projektovati IoT sistem koji omogućava merenje temperature vazduha u pametnom okruženju. Temperatura vazduha će služiti kao indikator odnosno određivaće u kojoj boji će svetleti LED strip traka/LED lampica. Podršku IoT sistemu predstavlja Veb aplikacija koja omogućava praćenje detektovanih parametara izmerenih senzorom i potom aktiviranje određenog scenarija LED strip trake/LED lampice. Ideja i vizija primene jednog ovakvog projekta jeste u terarijumima, kako bi se kontrolisala i upravljala temperatura potrebna reptilima za život. Pomoću pomenutog DHT11 senzora na određeni vremenski period mere se temperatura i vlažnost vazduha i dobijeni podaci se šalju, obrađuju i koriste na RPi. Shodno potrebnoj temperaturi u terarijumu određene su vrednosne granice. Ukoliko senzor detektuje manje od 21ºC, LED traka/lampica svetleće u plavoj boji, međutim ukoliko je temperatura između 21ºC i 30ºC prikazuje se bela boja. Svaka pokupljena vrednost preko 30ºC biće iskazana crvenom bojom.

Uz pomoć implementirane web aplikacije može se upravljati radom LED trake/lampice. Rad svetala može se uključiti/isključiti, a takođe se može i upaliti/ugasiti režim treperenja. Korisnik u toku korišćenja aplikacije sve vreme ima uvid u trenutnu temperaturu, prikazanu u jednom od polja interfejsa.

Detaljan prikaz implementacije pomenutog biće dat u nastavku rada kroz python, html, css i javascript kod, uključujući i neophodne komentare za lakše snalaženje i razumevanje.

## 2.4 Prikaz rešenja

Za potrebe ovog projekta prilikom kodiranja korišćeni su Visual Studio Code, PuTTy i WinSCP. Celokupni funkcionalni kod otkucan je u VSC, dok je namena PuTTy-ja bila za lakše povezivanje sa RaspberryPi-jem i otvaranje potrebne konzole kako bi se pokrenule python skripte. WinSCP nam je omogućavao jednostavan i siguran prenos datoteka sa računara na RaspberryPi.

Funkcije za paljenje i gašenje LED trake/LED dioda, kao i posebni režim treperenja u potpunosti su implementirani pomoću python-a. Što se tiče web stranice, front-end je odrađen putem HTML-a i CSS-a, dok je funkcionalnost obezveđena JavaScript-om. Za back-end je korišćen paython-ov framework Flask. Nastavak rada obuhvata pomenuti kod (uz potrebne komentare) i implementaciju.

Slika 6: PuTTy Slika 7: WinSCP

Prilog 1: Prikaz koda server.py

import time, json

# Biblioteka za senzor

import Adafruit\_DHT

import RPi.GPIO as GPIO

from flask\_cors import CORS

# Flask web framework

from flask import Flask

app = Flask(\_\_name\_\_)

CORS(app)

# Funkcija koja manipulise LED lampicom u zavisnoti od temperature

# Prima tip senzora, pin koji salje vrednosti, kao i 3 pina za 3 boje

def funkcija(sensor, DHT\_PIN, red, green, blue):

    humidity, temperature = Adafruit\_DHT.read\_retry(sensor, DHT\_PIN)

    print("Vlaznost "+ str(humidity)+" Temperatura "+ str(temperature))

    if temperature==None:

        print("ne radi")

        GPIO.output(green, 1)

        GPIO.output(blue, 1)

        GPIO.output(red, 1)

    elif temperature <= 21.0 :

        print("blue")

        GPIO.output(blue, 1)

        GPIO.output(green,0)

        GPIO.output(red, 0)

    elif temperature >= 30.0:

        print("red")

        GPIO.output(red, 1)

        GPIO.output(green,0)

        GPIO.output(blue, 0)

    else:

        print("white")

        GPIO.output(green, 1)

        GPIO.output(blue, 1)

        GPIO.output(red, 1)

# Home ruta gde je lampica ugasena

@app.route('/')

def home():

    red = 27

    green = 22

    blue = 17

    GPIO.output(green,0)

    GPIO.output(blue, 0)

    GPIO.output(red, 0)

    GPIO.cleanup()

# Ruta koja setuje nacin na koji citamo pinove sa RPi-a i koja pokrece funkciju paljenja

@app.route('/uredjajON')

def upali():

    sensor = Adafruit\_DHT.DHT11

    DHT\_PIN = 4

    GPIO.setmode(GPIO.BCM)

    red = 27

    green = 22

    blue = 17

    GPIO.setup(red, GPIO.OUT)

    GPIO.setup(blue, GPIO.OUT)

    GPIO.setup(green, GPIO.OUT)

    while True:

        funkcija(sensor, DHT\_PIN, red, green, blue)

#Ruta koja gasi uredjaj

@app.route('/uredjajOFF')

def hello():

    red = 27

    green = 22

    blue = 17

    GPIO.output(green,0)

    GPIO.output(blue, 0)

    GPIO.output(red, 0)

    GPIO.cleanup()

    return "Uredjaj ugasen"

#Ruta koja pali treperenje

@app.route('/treperenjeON')

def upaliTreperenje():

    sensor = Adafruit\_DHT.DHT11

    DHT\_PIN = 4

    GPIO.setmode(GPIO.BCM)

    red = 27

    green = 22

    blue = 17

    GPIO.setup(red, GPIO.OUT)

    GPIO.setup(blue, GPIO.OUT)

    GPIO.setup(green, GPIO.OUT)

# Petlja koja omogucava treperenje u intervalu od 1 sekunde

    while True:

        funkcija(sensor, DHT\_PIN, red, green, blue)

        time.sleep(1)

        GPIO.output(green,0)

        GPIO.output(blue, 0)

        GPIO.output(red, 0)

        time.sleep(1)

# Ruta koja uzima vrednosti i koja se koristi kako bi menjali boje okvira polja preko ajaxa

@app.route('/getVrednosti')

def getVrednosti():

    humidity, temperature = Adafruit\_DHT.read\_retry(Adafruit\_DHT.DHT11, 4)

    return json.dumps({"vlaznost":humidity,"temperatura":temperature})

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True, host='0.0.0.0')

Po završetku izrade koda treba ga i pokrenuti. Ovom prilikom korišćena je konzola PuTTy-ja. Prvo se treba pristupiti na RaspberryPi unošenjem defaultnih vrednosti. Korisnicko ime je ***pi***, a šifra ***raspberry***.

Nakon uspešnog povezivanja i prijavljivanja potrebno je pokrenuti skriptu: ***python server.py***

Ukoliko je sve u redu RaspberryPi će krenuti sa radom i obradom podataka primljenih sa temperaturnog senzora. Prekid programa može se izvršiti lokalno pritiskom **Ctrl + C**.

U nastavku dat je deo koda vezan za Veb aplikaciju i korisnički interfejs za rad sa RPi.

Prilog 2: Prikaz koda sajt.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Temperatura terarijuma</title>

    <link rel="stylesheet" href="css.css">

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.4.1.js" integrity="sha256-WpOohJOqMqqyKL9FccASB9O0KwACQJpFTUBLTYOVvVU=" crossorigin="anonymous"></script>

    <link href="https://gitcdn.github.io/bootstrap-toggle/2.2.2/css/bootstrap-toggle.min.css" rel="stylesheet">

    <script src="https://gitcdn.github.io/bootstrap-toggle/2.2.2/js/bootstrap-toggle.min.js"></script>

</head>

<body>

      <div class="card" style="width: 21.5rem; border-radius: 20px">

        <ul class="list-group list-group-flush">

                <div class="card-body">

                    <h5 class="card-title">Temperatura terarijuma</h5>

                </div>

                <li class="list-group-item">Temperatura

                    <div class="switch" style="position: absolute; right: 2%;">  <!-- css-->

                        <span  id="polje">

                            <div id="poljetext"></div>

                        </span>   <!-- ovde dolazi temperatura koju povlacimo sa raspberija-->

                    </div>

                </li>

                <li class="list-group-item" >Treperenje

                    <label class="switch" style="position: absolute; right: 2%;">  <!-- css-->

                        <input type="checkbox" id='toggle-treperenje'>

                        <span class="slider round" id="dugme1" onclick="upaliTreperenje()"></span> <!--na klik se pali treperenje-->

                    </label>

                </li>

                <li class="list-group-item" style=" border-radius: 20px" >Aktiviraj

                    <label class="switch" style="position: absolute; right: 2%;"> <!-- css-->

                        <input type="checkbox" id="toggle-upaljeno">

                        <span class="slider round" id="dugme2" onclick="upali()"></span> <!--na klik se pali sistem-->

                    </label>

                </li>

        </ul>

      </div>

    <script src="script.js"></script>

</body>

</html>

Prilog 3: Prikaz koda css.css

body{

    background-image: url("kameleon.jpg");

    background-size: cover;

    margin: 0;

    background-position: center top;

    object-fit: cover;

  }

  .list-group-item{

      font-size: 20px;

      color:#8f4326;

  }

  .card{

    margin-left: 2%;

    margin-top: 200px;

    background-color: #d7c79e;

  }

  .card-title{

    font-size: 27px;

    color: #8f4326;

    font-weight: 700;

  }

  /\* #dugme1{

      position: fixed;

  } \*/

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

.switch {

    position: relative;

    display: inline-block;

    width: 60px;

    height: 34px;

  }

  /\* Hide default HTML checkbox \*/

  .switch input {

    opacity: 0;

    width: 0;

    height: 0;}

 /\* The slider \*/

  .slider {

    position: absolute;

    cursor: pointer;

    top: 0;

    left: 0;

    right: 0;

    bottom: 0;

    background-color: #d7c79e;

    -webkit-transition: .4s;

    transition: .4s; }

  .slider:before {

    position: absolute;

    content: "";

    height: 26px;

    width: 26px;

    left: 4px;

    bottom: 4px;

    background-color: white;

    -webkit-transition: .4s;

    transition: .4s;

  }

  input:checked + .slider {

    background-color: #af7c5b;

  }

  input:focus + .slider {

    box-shadow: 0 0 1px #af7c5b;

  }

  input:checked + .slider:before {

    -webkit-transform: translateX(26px);

    -ms-transform: translateX(26px);

    transform: translateX(26px);

  }

  /\* Rounded sliders \*/

  .slider.round {

    border-radius: 34px;

  }

  .slider.round:before {

    border-radius: 50%;}

#polje{

    position: absolute;

    cursor: pointer;

    top: 0;

    left: 0;

    right: 0;

    bottom: 0;

    background-color: #d7c79e;

    -webkit-transition: .4s;

    transition: .4s;

    border-radius: 34px;

    border-style: solid;

  }

  #poljetext{

    margin-top: -1px;

    text-align: center;

  }

Prilog 4: Prikaz koda script.js

otvoreno = false

// funkcija koja pali sistem

function upali() {

  // varijabli dodeljujemo checkbox

  var checkBox = document.getElementById("toggle-upaljeno");

  // if uslov koji nas baca na rutu koja pokrece odnosno gasi uredjaj

  if (checkBox.checked == false){

    $.getJSON('http://172.20.222.231:5000/uredjajON', function(data) {

      console.log("upali");

    });

  }

  else {

    $.getJSON('http://172.20.222.231:5000/uredjajOFF', function(data) {

    console.log("ugasi");

  });

}

}

function upaliTreperenje() {

  var checkBox = document.getElementById("toggle-treperenje");

  if (checkBox.checked == false){

    $.getJSON('http://172.20.222.231:5000/treperenjeON', function(data) {

    });

  }

  else {

    $.getJSON('http://172.20.222.231:5000/uredjajOFF', function(data) {

  });

}

}

// Funkcija koja cita temperaturu i na osnovu nje menja boju okvira polja u kome se ona ispisuje

$(function citaj(){

  $.ajaxSetup ({

    cache: false,

    complete: function() {

    // Ucitavanje temperature na svaku sekundu

      setTimeout(citaj, 1000);

    }

  });

  if(!otvoreno){

    $.getJSON('http://172.20.222.231:5000/getVrednosti', function(data) {

        $('#poljetext').html(data.temperatura);

        if (data.temperatura>30) {

            $("#polje").css("border-color", "#DC3232");

        }else if (data.temperatura<21) {

            $("#polje").css("border-color", "#3271DC");

        } else {

            $("#polje").css("border-color", "#d7c79e");

        }

        console.log(data.temperatura);

    });

  }

});

Veb aplikacija ovog projekta sastoji iz polja za prikaz trenutne izmerene temperature u terarijumu, dugmeta za paljenje/gašenje režima treperenja i dugmeta za uključivanje/isključivanje rada LED trake/dioda. Takođe okvir polja za temperaturu svetli u različitim adekvatnim bojama u zavisnosti od temperature. Prilično jednostavan interfejs u potpunosti olakšava korisniku rad sa povezanim uređajima.

Prikaz interfejsa veb aplikacije nalazi se na sledećoj slici.



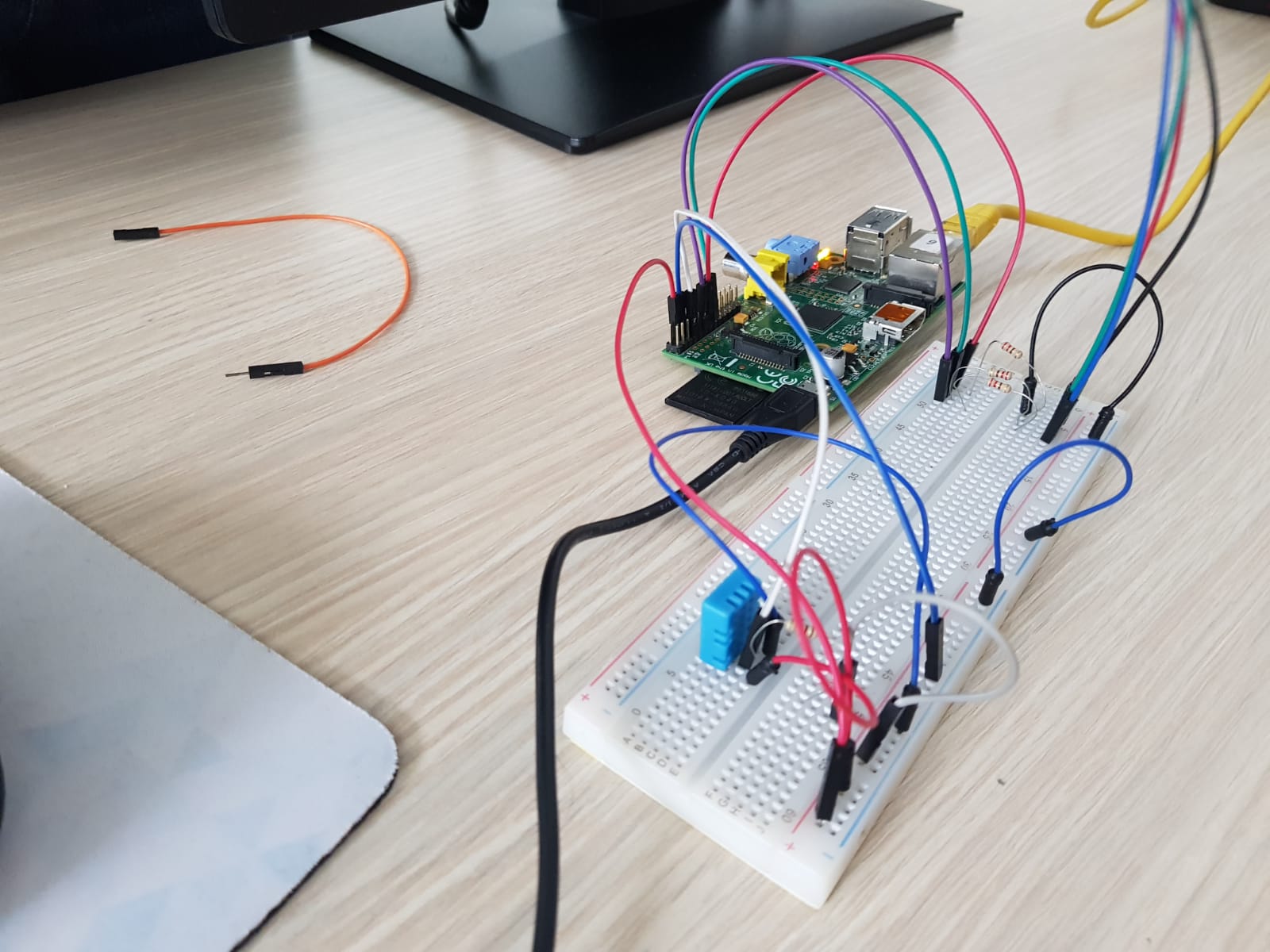
Slika 8: Prikaz interfejsa veb aplikacije

Linkovi ka video prikazu projektnog rešenja:

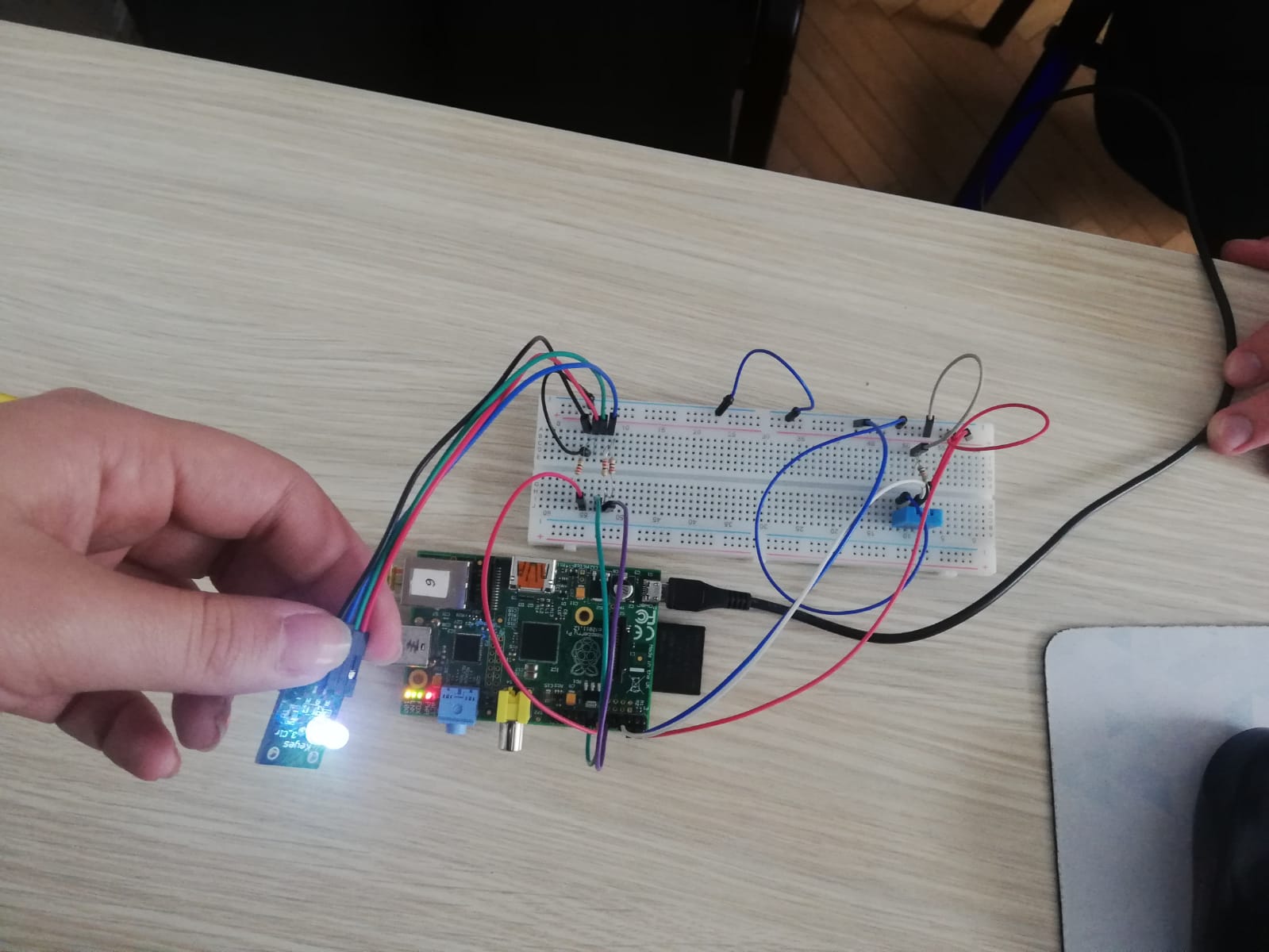
<https://youtu.be/GBNVoZ9VDiA>

<https://youtu.be/2e8VMFWWFyc>

Radi jasnoće i boljeg uvida u izgled celokupnog hardverskog sistema, a zatim i integrisanog sistema u celosti, u nastavku su priložene slike i video rada na licu mesta.



Slika 9: Prikaz hardverskog sklopa



Slika 10: Prikaz hardverskog sklopa