

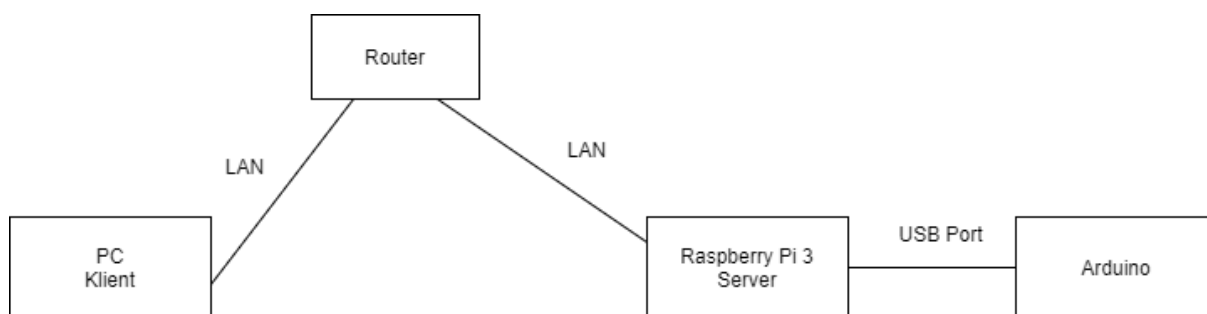
SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Záverečné zadanie

Úvod

Ako záverečné zadanie k predmetu POIT som si vybral ovládanie svietivosti diódy a následne meranie intenzity svetla fotorezistorom pomocou vytvorenej web stránky.

Hardwarovú realizáciu som urobil pomocou arduina, ktoré bolo pripojené cez USB port do raspberry pi. Na raspberry pi bol spustený server, ktorý pomocou socketov komunikoval s arduinom. Táto komunikácia medzi serverom a arduinom mala za úlohu posielat' a prijímať dáta na ovládanie diódy a meranie intenzity svetla. Na server je možné sa pripojiť ako klient pomocou webového prehliadača. Pri realizácii boli použité reálne zariadenia pripojené do testovacej lokálnej siete.



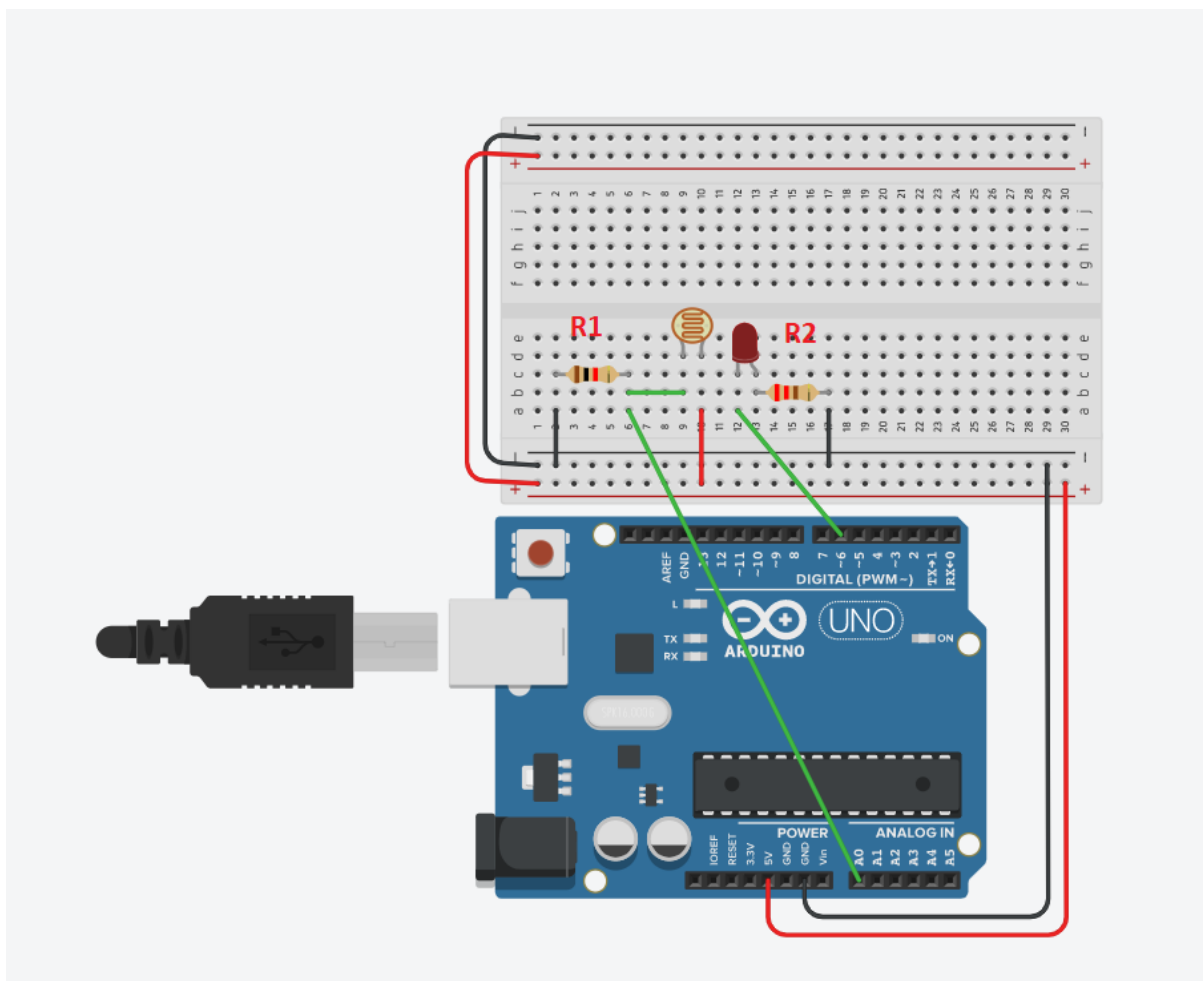
Obrázok 1Bloková schéma zapojenia

Arduino

Na záverečné zadanie bolo použité Arduino UNO.

Zoznam použitých súčiastok:

- 1x Arduino UNO
- 1x USB 2.0 kábel typ A/B
- 1x Svetelný senzor
- 1x nepájiva doska
- 1x dióda
- 1x 220 ohm rezistor – R2
- 1x 1000 ohm rezistor – R1
- 6x spojovacie káblíky



Obrázok 2 Schéma zapojenia arduino

Dióda s odporom R2 je zapojená do výstupného PWM digitálneho pinu D6 a na zem. Pri digitálnych výstupných pinoch v mode PWM sa na pin posieľa celé číslo z rozsahu 0 až 255. Veľmi zjednodušene si to môžeme predstaviť ako namapovaný rozsah el. napätia 0V až 5V do číselného intervalu 0 až 255 a zariadenie pripojené na takýto výstup sa chová, akoby sme zmenou číselnej hodnoty 0-255 na pine zmenili el. napätie na zariadení.

Fotosenzor (fotorezistor) je napájaný napätím 5 voltov s rezistorom do zeme a do analógového pinu A0 na čítanie hodnoty. Analógové piny pracujú ako A/D prevodník a nadobúdajú hodnoty od 0 do 1023 v závislosti od elektrického napätia na vstupnom analógovom pine.

Dióda a fotosenzor boli osadene vo vzdialenosti do 5 mm pre zachytávanie hlavne svetla z diódy. Pri realizácii musíme brať ohľad na to, že na fotosenzor vplýva aj okolité svetlo. Silu

svietenia diódy budeme ovládať pomocou hodnôt posielaných zo servera do arduina v rozsahu od 0-255 kde hodnota 0 je vypnutá dióda a hodnota 255 je najvyššia svietivosť diódy.

```
int x;
int bright;

void setup() {
  Serial.begin(9600);
  Serial.setTimeout(1);
  pinMode(6, OUTPUT);
  bright = 10;
}

void loop() {

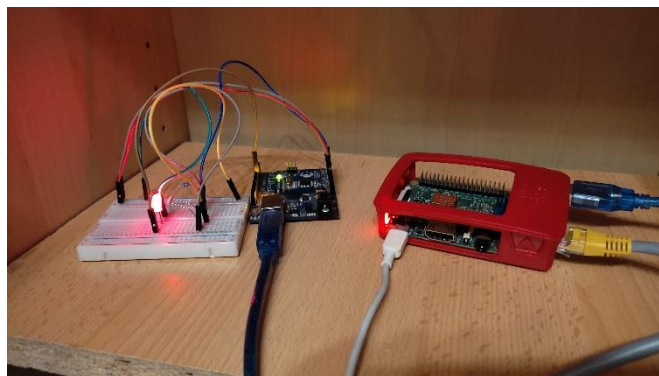
  int analogSensor = analogRead(A0);
  analogWrite(6, bright);

  //Serial.println("Sensor:");
  Serial.println(analogSensor);

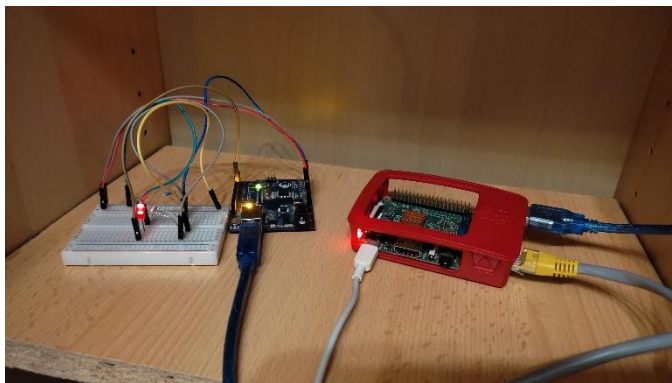
  if (Serial.available() > 0) {
    x = Serial.readString().toInt();
    Serial.println(x);
    bright = x;
  }
  delay(1000);
}
```

Obrázok 3 Kód arduino

Na začiatku si zadefinujem premenné. Vo funkcii *void setup* zadefinujeme, na akom pine je pripojená dióda. Vo funkcii *void loop* čítame hodnotu z analógového pinu v rozsahu 0 – 1023, priradíme do premennej a zapíšeme na seriálový port. Hodnota zapísaná na seriálový port bude čítať server. Podmienka *if* sa spustí vtedy, keď zo serveru príde nová hodnota na nastavenie sily svietenia diódy. Tato hodnota sa zapíše do premennej *bright* a dióda bude svietiť takou intenzitou.



Obrázok 4 Svietenie diódy s hodnotou 255



Obrázok 5Svietenie diódy s hodnotou 10

Raspberry Pi

Na raspberry pi je spustený server. Server je napísaný v jazyku Python a použitý web framework Flask, na vytváranie webu a webových aplikácií. Flask umožňuje vytvoriť localhost, na ktorom môže byť umiestnená stránka, ktorou budeme ovládať silu svietenia diódy. Taktiež budeme na stránke vypisovať hodnoty zo svetelného senzora.

```
from threading import Lock
from flask import Flask, render_template, session, request, jsonify, url_for
from flask_socketio import SocketIO, emit, disconnect
import time
import random
import math
import serial

arduino = serial.Serial(port='/dev/ttyUSB0', baudrate=9600, timeout=.1)
async_mode = None

app = Flask(__name__)

app.config['SECRET_KEY'] = 'secret!'
socketio = SocketIO(app, async_mode=async_mode)
thread = None
thread_lock = Lock()
```

Na začiatku kódu si importujeme knižnice. Aby server komunikoval s arduinom, je potrebné zadať správny sériálový port, na ktorom je arduino pripojené.

```

## vlakno ktore sa spusti nazaciatku
def background_thread(args):
    time.sleep(3)
    count = 0
    y = "0"
    B = 1 #pomocna premena, porovna sa s A
    print("args")
    print(args)
    def write_read(x):
        if B!=A:
            print("poslana hodnota do arduina")
            arduino.write(bytes(x, 'utf-8'))
            time.sleep(1)
            hodnota = arduino.readline().strip()
            return hodnota
        else:
            #print("vypisuje predosle")
            hodnota = arduino.readline().decode().strip('\r\n')
            hodnota = format(hodnota)
            return hodnota

```

Spustí sa funkcia *background_thread*. Priradia sa premenne a zadefinuje sa funkcia *write_read*.

```

while True:
    if args:
        A = dict(args).get('A')#vstup od uzivatela
    else:
        A = 1

    y = dict(args).get('start')#vstup od uzivatela
    A=str(A)

    socketio.sleep(1)#ako casto chceme refreshovat
    count += 1 #zvysujeme pocitadlo pri receive
    print("A=",A)

    #num="200"
    #num = input("Enter a number: ") # Taking input from user
    outa = write_read(A)
    outa = str(outa)
    #print(outa) # printing the value
    B = A
    #print("B",B)
    print("Tlacitko Start",y)#na zaciatku tlacitko start je NONE
    if y == "1":
        socketio.emit('my_response',
                      {'data': outa, 'count': count},
                      namespace='/test')

```

Táto časť kódu sa nachádza stále vo funkcii *background_thread*. Na začiatku sa do premennej *A* sa priradí hodnota sily led diódy ,zadanej od klienta. Táto hodnota sily svietenia diódy sa

pošle cez komunikáciu do arduina. Nastaví sa hodnota sily na 1. Po pripojení klienta na server sa začnú merať hodnoty zo svetelného senzora až po stlačení tlačítka štart. Ak server prijme novu hodnotu sily svietenia diody, funkcia *write_read* skontroluje novu hodnotu a pošle ju do arduina. Stav diody sa zmení. Ak ale užívateľ nezadal novú hodnotu svietenia, funkcia vráti hodnotu zo senzora a v poslednej podmienke ich pošle na stranu klienta.

```
@app.route('/')
def index():
    return render_template('index.html', async_mode=socketio.async_mode)

##ak dojde od klienta udalost my_event tak sa urobi funkcia
@socketio.on('my_event', namespace='/test')
def test_message(message):
    session['receive_count'] = session.get('receive_count', 0) + 1
    session['A'] = message['value']
    print("message['value']")
    print(message['value'])
    emit('my_response',
         {'data': ('Nova hodnota je:', message['value']), 'count': session['receive_count']})

@socketio.on('disconnect_request', namespace='/test')
def disconnect_request():
    session['receive_count'] = session.get('receive_count', 0) + 1
    emit('my_response',
         {'data': 'Disconnected!', 'count': session['receive_count']})
    disconnect()

@socketio.on('start', namespace='/test')
def db_message(message):
    session['start'] = message['value']

@socketio.on('stop', namespace='/test')
def db_message(message):
    session['start'] = message['value']

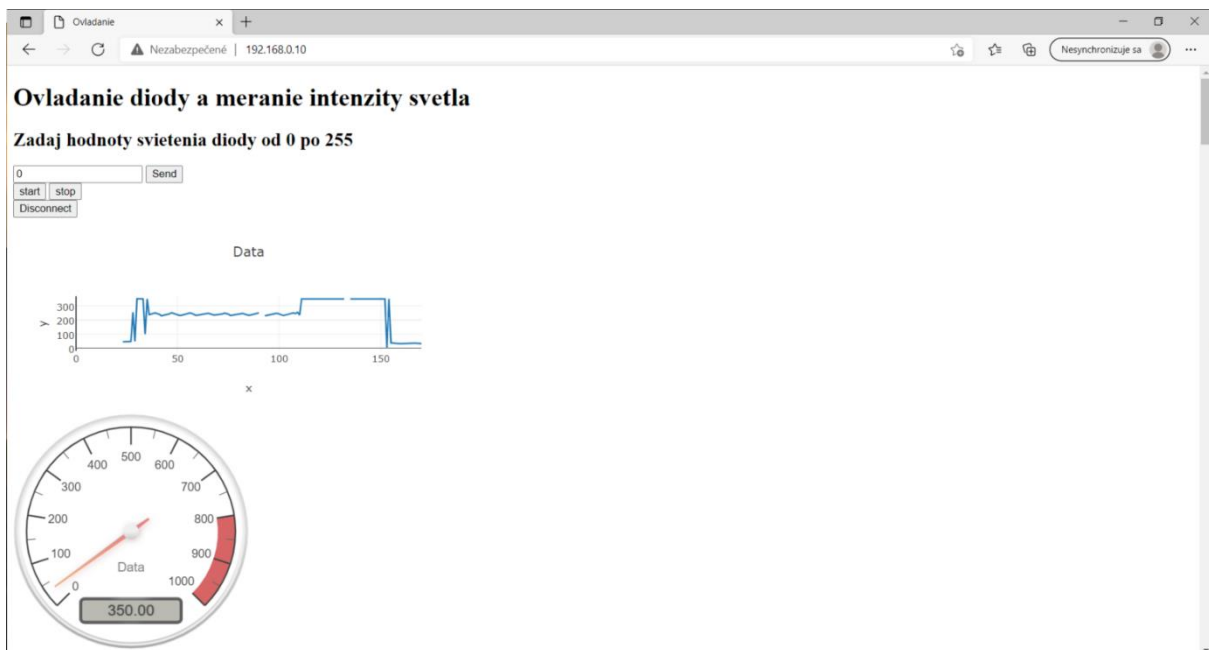
@socketio.on('connect', namespace='/test')
def test_connect():
```

V tejto časti sme zadefinovali funkcie, ktoré sa spustia ak príde požiadavka zo strany klienta.

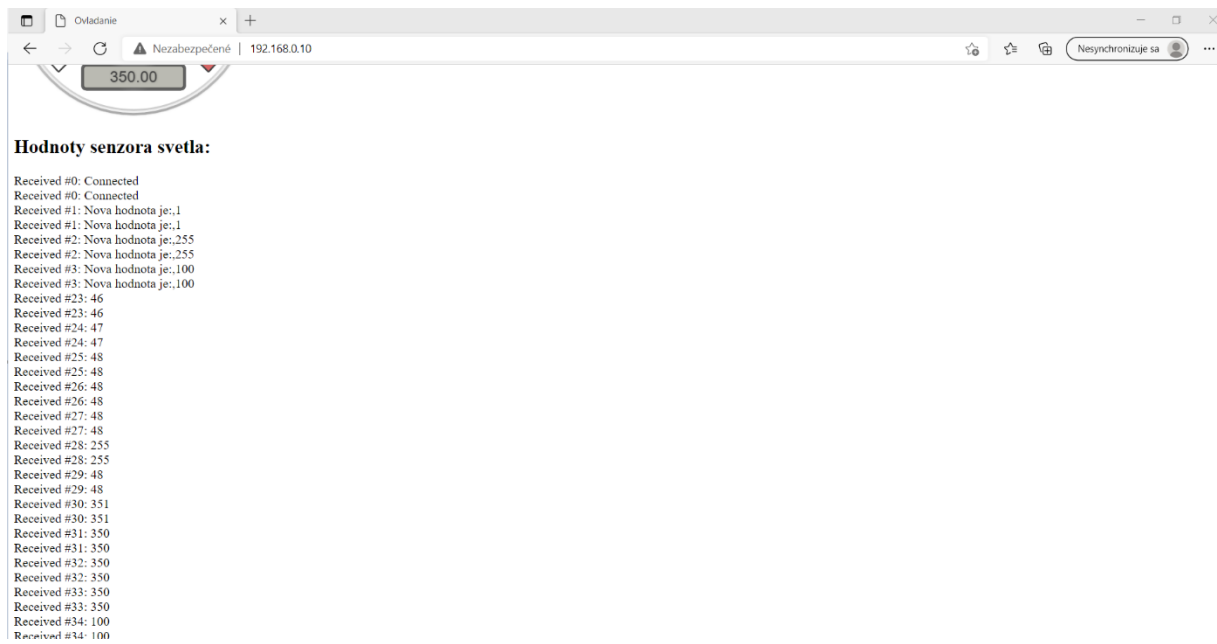
Klient

Vytvoril som webstránku, ktorá ovláda diódu a meria senzor. Stránka je napísaná v html jazyku. Využívané skripty sú použité na komunikáciu so serverom a vytvárajú dynamické prvky stránky.

Po prvom pripojení sa na server, klient nebude dostávať žiadne hodnoty zo senzora. Po zadání želanej sily svietenia sa po stlačení tlačítka štart pošle požiadavka na server a server zapíše túto hodnotu do premennej A. Tlačítkom start spustíme výpis meraných hodnôt a tlačítkom stop prerušíme. Disconnect je na odpojenie od servera. Vo webovom rozhraní sú 3 spôsoby výpisu hodnôt zo senzora- graf, ciferník a výpis v texte.



Obrázok 6 Webové rozhranie



Obrázok 7 Webové rozhranie

Zaver

Pri vypnutej dióde (hodnota sily 0) svetelný senzor nameria hodnotu 64. Pri najväčšej sile svietenia(255) senzor nameria hodnotu 333. Výsledne hodnoty sa môžu líšiť pri zmene okolitých svetelných podmienok.