

Mladi za napredek Maribora 2017

34. srečanje

NADZOR IN UPRAVLJANJE PROCESOV NA DALJAVO

RAČUNALNIŠTVO
Inovacijski predlog

PROSTOR ZA NALEPKO

Avtor:	LUKA RADOVIĆ, GAŠPER KUNEJ, JERNEJ ORNIK
Mentor:	BRANKO POTISK
Šola:	SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

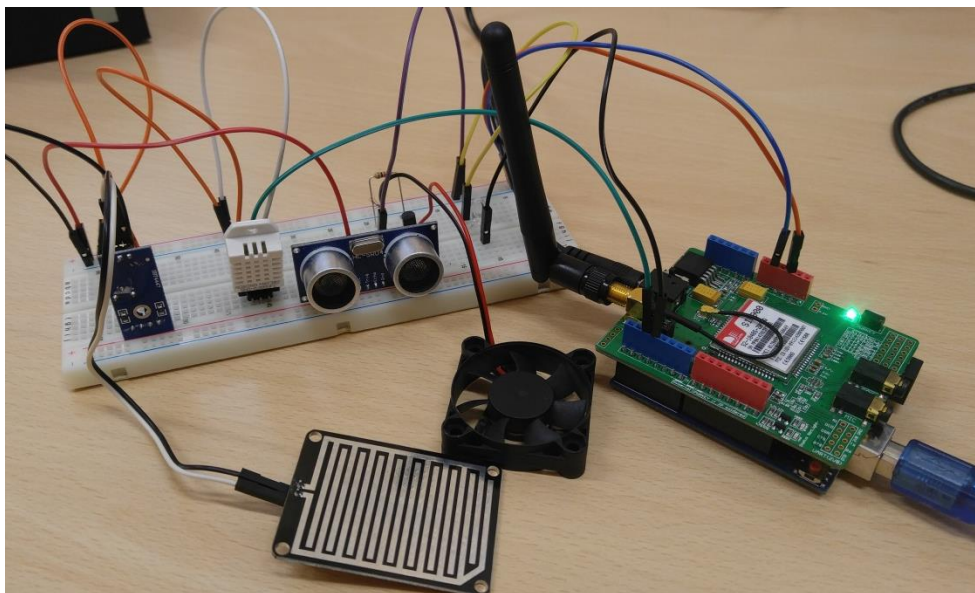
Maribor, februar 2017

Mladi za napredek Maribora 2017

34. srečanje

NADZOR IN UPRAVLJANJE PROCESOV NA DALJAVO

RAČUNALNIŠTVO
Inovacijski predlog



Maribor, februar 2017

KAZALO VSEBINE

POVZETEK	4
ZAHVALA.....	5
1 UVOD	6
1.1 Sorodni izdelki.....	6
2 IDEJA.....	7
3 PRIMERJAVA KRMILNIKOV	8
3.1 Arduino ali Raspberry Pi.....	8
3.2 Razvojna plošča Arduino	8
3.3 Razvojna plošča Raspberry Pi.....	8
3.4 Izbira mikrokrmilnika: Arduino in Raspberry Pi	9
4 PRIMERJAVA MREŽNIH KRMILNIKOV	10
4.1 Izbira mrežnega krmilnika: GPRS, Wi-Fi ali Ethernet	10
4.2 Mrežni krmilnik GPRS (General Packet Radio Service)	10
4.3 Mrežni krmilnik Wi-Fi (Wireless Fidelity)	10
4.4 Mrežni krmilnik Ethernet	11
4.5 Primerjava mrežnih krmilnikov: GPRS, Wi-Fi, Ethernet	11
5 GRADNIKI SISTEMA	12
5.1 Razvojna plošča Arduino/Genuino UNO rev3.....	12
5.2 Mrežni krmilnik GPRS: Simcom SIM900 ščit	13
5.3 Senzor vlage in temperature DHT22.....	14
5.4 Pc ventilator.....	15
5.5 LCD I2C.....	15
6 NAČRT IN IZDELAVA SISTEMA.....	16
6.1 Arduino.....	17
6.2 Spletna stran	21
6.3 Podatkovna baza.....	25
7 DRUŽBENE ODGOVORNOSTI.....	26
8 ZAKLJUČEK.....	27
9 VIRI.....	28
9.1 Pisni viri	28
9.2 Elektronski viri.....	28

KAZALO SLIK

Slika 1: Razvojna plošča Arduino	12
Slika 2: Mrežni krmilnik SIM900	13
Slika 3: Senzor za vlago in temperaturo.....	14
Slika 4: Pc ventilator	15
Slika 5: LCD I2C zaslon	15
Slika 6: Shema povezave.....	16
Slika 7: Shema grafičnega vmesnika na spletni strani	16
Slika 8: Program ki bere temperaturo in vlago	17
Slika 9: inicializacija senzorja za temp. in vlago	18
Slika 10: Inicializacija omrežja	18
Slika 11: Meritev	19
Slika 12: Prošnja tipa GET	19
Slika 13: Vrednost senzorja.....	20
Slika 14 Prkaz trenutnih podatkov na SIM	20
Slika 15: Povezovanje na podatkovno bazo	23
Slika 16: Preverjanje prijave	23
Slika 17: Spletna stran.....	24
Slika 18: Prijavna stran	24
Slika 19: Baza za shranjevanje stanj komponent	25
Slika 20: Baza ki se posodablja.....	25
Slika 21: baza ki uporablja SHA1 kodiranje	25

POVZETEK

V dobi, ko ima skoraj vsak od nas mobilni telefon in dostop do spleta, se nam je pojavila ideja upravljanja in nadzorovanja naprav od kjerkoli. Poznamo že veliko sistemov z katerimi lahko na daljavo s pomočjo spleta upravljamo rastlinjak, hišo ali katerikoli drugi objekt, ampak ti so večinoma namenjeni industriji in se ne prilagajajo uporabniku ali pa cenovno presegajo današnje standarde uporabnikov. Torej veliki del naše ideje obsega modularnost oziroma prilagajanje celotnega sistema po uporabnikovih željah.

ZAHVALA

Kot prvo se zahvaljujemo mentorju, ki nas je spodbujal pri tej inovaciji in nam svetoval pri delu, nato pa bi se še zahvalili prijateljem za izposojlo elementov s katerimi smo ta sistem prikazali.

1 UVOD

Namen naloge je ustvariti cenovno ugoden, enostaven, uporabniku prijazen ter varen sistem upravljanja in spremljanja naprav na daljavo.

Kaj je oddaljen dostop? Je zmožnost dostopa do računalnika ali omrežja nekje v oddaljenosti.

Kot omenjeno modularnost v našem izdelku igra veliko vlogo, zato se spletna stran prilagaja uporabnikovi izbiri nadzorovanih naprav (npr.: prikaz temperature, vlage, upravljanje ventilatorja ter luči na vrtu (napajanje preko akumulatorja ali vtičnice) in prikaz temperature v kuhinji). V sistem je lahko vključena avtomatizacija, ki bo delovala samodejno ali po uporabnikovih nastavitvah.

1.1 Sorodni izdelki

Na trgu je kar nekaj izdelkov ki jih je mogoče upravljati na daljavo kot so klimatska naprava, razni kavni aparat in celo hiše ter različni objekti. Ampak takšni sistemi imajo slabosti saj večinoma upravljajo samo eno stvar in so predragi.

SCADA¹ sistemi so dober primer industrijskega oddaljenega upravljanja. Ti sistemi so velikokrat težko razumljivi, dragi in predvsem niso namenjeni za vsakdanjega človeka.

SmartHome² pokriva večino naših ciljev ampak ima eno bistveno napako poleg visoke cene, to je da ga ne moremo uporabljati v okolju kjer nimamo dostopa do internetnega omrežja (Wi-Fi).

¹ (Supervisory Control And Data Acquisition) je skupno ime za sisteme, ki so namenjeni nadzоровanju in krmiljenju različnih tehnoloških procesov z računalnikom.

² Je podjetje, ki se ukvarja z avtomatizacijo stanovanj in hiš.

2 IDEJA

Kot smo že omenili v uvodu želimo poiskati rešitev za različne primere upravljanja naprav v hiši, okolici, vikendih ter oddaljenih posestvih na najenostavnejši in najcenejši možni način. V mislih imamo nadziranje, ter upravljanje enostavnih komponent, kot so luči, okna, gretje, ventilacija, v oklici tudi opravljanje vrta, ter bolj zahtevnejših opravil, npr.: varovanje objekta oz. posesti.

Upravljavski del smo si zamislili centraliziran sistem, ki mora biti enostavno dostopen z različnejših mobilnih naprav oz. računalnikov. Kot izvedbo se nam ponuja spletni vmesnik, preko katerega upravljamo mikrokrmilnike.

Na trgu je več različnih proizvajalcev mikrokrmilnikov z zelo različnimi zmogljivostmi. Cenovno ugodni in poznan je platforma Arduino.

3 PRIMERJAVA KRMILNIKOV

3.1 Arduino ali Raspberry Pi

Arduino ali Raspberry Pi povzroča kar precej dileme. Ti dve napravi sta zelo poceni in sta približno enako veliki, vendar se uporabljata za drugačne namene. Raspberry Pi je majhen računalnik, na katerem je primarno naložen operacijski sistem Linux, Arduino pa je mikrokrmilniški sistem brez tipičnega operacijskega sistema.

3.2 Razvojna plošča Arduino

Arduino je mikrokrmilnik, ki ga lahko programiramo po želji. Namenjen je branju in obdelavi podatkov iz različnih senzorjev ali dodatkov (senzor vlage, temperature, različni motorčki, stikala...). Podatke lahko prikaže tudi prek LCD monitorja ali LED diod.

Arduino se zaradi svoje modularnosti uporablja v veliko različnih projektih, od 3d tiskalnikov, robotike do alarmov in igralniških dodatkov. Obstaja veliko različic Arduinota, ki se razlikujejo po velikosti, zmogljivosti mikroprocesorja, vhodno-izhodnih pinih, spominu...

3.3 Razvojna plošča Raspberry Pi

Raspberry Pi je v primerjavi z Arduiotom predvsem zmogljivejši, ampak to še ne pomeni da je slabši. Ima operacijski sistem Linux (lahko tudi katerikoli drugi kompatibilen operacijski sistem), ki je naložen na SD kartici. Pri tej napravi imamo že osnovne vhode in izhode kot jih najdemo na računalniku (HDMI, USB, priključek za ethernet kabel, bluetooth...). Vse to omogoča uporabo sistema Raspberry Pi kot namiznega računalnika, potrebno je samo priklopiti monitor, miško in tipkovnico.

Projekti za Raspberry Pi so bolj programsko usmerjeni. Ker sistem poganja operacijski sistem, je večina projektov bolj usmerjenih na programskih in multimedijskih centrih. Nanj pa lahko tudi priključimo strojno opremo, ki jo upravljamo preko GPIO pinov.

3.4 Izbira mikrokrmilnika: Arduino in Raspberry Pi

Izbira plošče je seveda odvisna od tipa projekta, ki ga imamo v mislih, in izkušnje programiranja. Arduino je bolj usmerjen v strojno opremo oziroma senzorje, Raspberry Pi pa bolj v programski del, saj je že sam po sebi majhen računalnik.

Arduino sicer nima priključkov za internet, zvok in sliko, ima pa veliko različnih ščitov (ethernet, WIFI, GSM shield) katere dodamo na osnovno ploščo, z malo več dela je pa tudi možno spraviti zvok na zvočnike in sliko na monitor.

Za naš projekt smo se odločili za Arduinov mikrokrmilnik. Arduino nam ponuja veliko število različnih modelov ki so primerni za naš izdelek (Uno, Mega, Mini...). Izbrati osnovne plošče ni lahko, zato smo se najprej morali vprašati kaj zares potrebujemo. Namreč vsi ti modeli se razlikujejo v število vhodov (oz. izhodov), velikosti spomina, porabi energije, vgrajenimi moduli (Wi-Fi, Ethernet...), velikosti itd. Odločili smo se za Arduino UNO saj izpolnjuje vse naše potrebe, ter zelo dobro sodeluje z analognimi in digitalnimi gradniki iz katerih lahko pridobimo vse podatke ki jih potrebujemo za svoj sistem.

4 PRIMERJAVA MREŽNIH KRMILNIKOV

4.1 Izbira mrežnega krmilnika: GPRS, Wi-Fi ali Ethernet

Z Arduinovim mikrokrmilnikom se na internetno omrežje lahko povežemo na več načinov. Pri izbiri načina povezave smo naleteli na težave, saj v različnih razmerah lahko uporabljamo različne načine povezave. Zaradi potrebe povezanega računalnika preko USB vhoda na Arduino smo takoj izločili povezavo preko serijskega vhoda.

4.2 Mrežni krmilnik GPRS (General Packet Radio Service)

Je mobilna podatkovna storitev v okviru standarda GSM. GPRS povezava ima dobre in slabe lastnosti, saj je lahko zelo koristna v primerih izpada internetnega omrežja ter ko v bližini ni ethernet kabla ali Wi-Fi točke, ampak je potrebno dodatno vzpostaviti naročniško razmerje pri ponudniku telekomunikacije kar zviša stroške celotnega sistema. Za vzpostavitev povezave med SIM kartico in Arduino je potreben SIM800 ali SIM900 ščit (SIM800 in SIM900 se razlikujeta po številu funkcij in ceni) kateri se enostavno priključi na Arduino mikrokrmilnik. Vzpostavitev povezave do strežnika in podatkovne baze pa je bistveno zahtevnejša.

4.3 Mrežni krmilnik Wi-Fi (Wireless Fidelity)

Je brezžična tehnologija, ki omogoča, da se lahko naprava poveže v omrežje. Skoraj vsako gospodinjstvo dandanes ima že brezžično dostopno točko, ampak ker je doseg Wi-Fi točke omejen do nekje 20m se pojavijo težave pri zunanji uporabi sistema oziroma v notranjih prostorih kjer je pokritost omrežja slabša. Pri Arduino UNO mikrokontrolerju je za povezavo na Wi-Fi omrežje potrebno dodati Wi-Fi ščit oziroma modul. Preko ščita lahko vzpostavimo strežnik kar na Arduino, z modulom je sicer tudi to možno vendar ni priporočljivo za večji prenos podatkov. Povezavo na zunanji strežnik kot ga uporabljamo mi pa je zelo enostavno vzpostaviti.

4.4 Mrežni krmilnik Ethernet

Je enak princip povezave kot Wi-Fi vendar preko ethernet kabla. Uporaba etherneteta je najbolj primerna, ko je modem oziroma usmerjevalnik v bližini mikrokrmilnika Arduino. Mikrokontrolerju Arduino UNO je potrebno dodati Ethernet ščit, za katerega veljajo pri povezavi veljajo skoraj enaki pogoji kot pri Wi-Fi ščitu.

4.5 Primerjava mrežnih krmilnikov: GPRS, Wi-Fi, Ethernet

Isto kot pri izbiri mikrokrmilnika je izbira odvisna od tipa projekta. Pri vseh načinih je za naš projekt potreben eden izmed ščitov, kateri se po ceni, velikosti in porabi energije ne razlikujejo veliko. Največja razlika med krmilniki je predvsem v dosegu omrežja. Kljub višjih stroških sistema in višje težavnosti pri povezavi z strežnikom smo se odločili za SIM900 GPRS ščit (pri SIM800 ne rabimo bluetooth in FM povezave), zaradi modularnosti našega sistema ter neodvisnosti od internetnega omrežja v območju.

5 GRADNIKI SISTEMA

5.1 Razvojna plošča Arduino/Genuino UNO rev3



Slika 1: Razvojna plošča Arduino

Vir: <https://goo.gl/6u7oCP>

- Frekvenca procesorja : 16MHz
- Mikroprocesor : ATmega328
- Delovna napetost : 5V
- Digitalni vhodno/izhodni pini : 14 (od tega 6 PWM izhodi)
- Analogni vhodni pini : 6
- Spomin : Flash (32kB), SRAM (2kB) in EEPROM 1kB
- Enosmerni tok na vhodno/izhodni pinih: 40 mA
- Enosmerni tok za pine 3,3V: 50 mA
- Priključki: USB, ICSP priključek

5.2 Mrežni krmilnik GPRS: Simcom SIM900 ščit



Slika 2: Mrežni krmilnik SIM900

Vir: <https://goo.gl/3Slbkp>

- Frekvenčni razpon: Quad-Band 850 / 900/ 1800 / 1900 MHz
- Delovna napetost : 5V
- Upravljanje z AT komandami
- Nizka poraba energije- 1.5mA (spanje)
- GPRS class 10: max. 85.6 kbps
- RTC backup

5.3 Senzor vlage in temperature DHT22



Slika 3: Senzor za vlago in temperaturo

Vir: <https://goo.gl/lq4JBJ>

ki meri temperaturo od -40°C do $+80^{\circ}\text{C}$ in vlago od 0% do 100%. Natančnost senzorja je $\pm 0.1^{\circ}\text{C}$, vrednost pa izmeri vsaki 2 sekundi pri delov.

- Razpon merjenje temperature: od -40°C do $+80^{\circ}\text{C}$
- Razpon merjenje vlage: od 0% do 100%
- Natančnost senzorja : $\pm 0.1^{\circ}\text{C}$
- Razmak merjenja vrednosti: 2s
- Delovna napetost: 5V

5.4 Pc ventilator



Slika 4: Pc ventilator

Vir: <https://goo.gl/PXMFGP>

- Delovna napetost: 5V
- Premer: 60mm
- Obrati na minuro: 3300rpm
- Pretok zraka: 25 m³/h

5.5 LCD I2C



Slika 5: LCD I2C zaslon

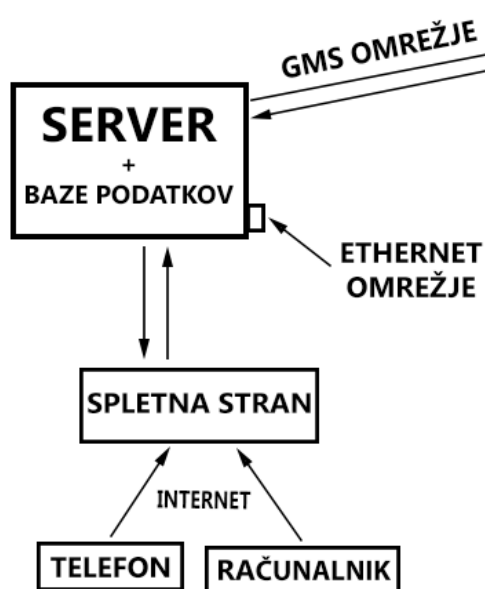
Vir: lasten

- Delovna napetost: 5V
- Velikost: 16x2

6 NAČRT IN IZDELAVA SISTEMA

Da bi se tudi sami prepričali v praktičnost naše ideje, želimo postaviti en del tega sistema.

Najprej smo se pogovori o realizaciji ideje in naredili enostaven in pregleden načrt po katerem bomo naš sistem gradili. Načrt se je sicer spreminjal vendar cilji so ostali enaki. (Slika 6) prikazuje načrt komunikacije med strežnikom, spletno stranjo ter mikrokrmilnikom Arduino.



Slika 6: Shema povezave

Vir: lasten



Slika 7: Shema grafičnega vmesnika na spletni strani

Vir: lasten

6.1 Arduino

Prvi preizkusi z Arduinom so bili enostavni npr.: vklop in izklop LED diode, branje temperature in vlage ter izpis na LCD itd.

```
//knjižnice potrebne za delovanje
#include <TimerOne.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <dht.h>

dht DHT;
#define DHT22_PIN 7 //določanje pina za senzor temperature in vlage

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // nastavitve lcd zaslona

void setup()
{
    lcd.begin(20,4);          // vklop backlighta, inicializacija lcd zaslona
}

//branje in izpis
void loop()
{
    int chk = DHT.read22(DHT22_PIN);
    lcd.setCursor(0,0);
    lcd.print("Temp: ");
    lcd.print(DHT.temperature);
    lcd.print((char)223);
    lcd.print("C");
    lcd.setCursor(0,1);
    lcd.print("Vlaga: ");
    lcd.print(DHT.humidity);
    lcd.print("%");
    delay(1000);
}
```

Slika 8: Program ki bere temperaturo in vlago

Vir: lasten

Kasneje ko smo se naučili osnove programiranja Arduina, smo pričeli z povezavo na strežnik. Kot navedeno smo uporabili GPRS (SIM900) ščit, v njega vstavili sim kartico ki nam omogoča sprejemanje in pošiljanje podatkov. Za programiranje smo uporabili program Arduino IDE (integrirano razvojno okolje) 1.6.0, ki je namenjen za naš mikrokrmilnik. Potrebno je bilo napisati kodo, ki bo omogočila GPRS ščitu, da bo oddajal in prejemal podatke na strežnik.

Ob zagonu krmilnika najprej inicializiramo PIN-e, odpremo serijske vmesnike, ter kličemo funkcijo za inicializacijo omrežja.

```
#include <SoftwareSerial.h>
#include "DHT.h"
#define DHTTYPE DHT22          // DHT 22  (oznaka na senzorju: AM2302).
#define DHTPIN 2               //Podatkovni izhod senzorja priključimo na pin 2
DHT dht(DHTPIN, DHTTYPE);      //Inicijalizacija DHT senzorja, tip, 16 MHz (Arduino Uno)

SoftwareSerial SIM900(7, 8); // inicializacija sim-portov
int Led= 3;

void setup() {
    pinMode(Led, OUTPUT);
    pinMode(3, OUTPUT); //
    SIM900.begin(19200); //odpremo SIM900 serial

    Serial.begin(9600); //odpremo serial za racunalnik

    dht.begin(); //zazeni dht

    Serial.print("Zagon naprave..... \n" );
    delay(10000);
    Serial.print("Inicijalizacija omrezja : \n");
    inicializacija_omrezja(); //klicanje funkcije za inicializacijo omrezja
```

Slika 9: inicializacija senzorja za temp. in vlago

Vir: lasten

Funkcija vsebuje vse podatke, ki jih krmilnik rabi za komunikacijo z internetom.

```
void inicializacija_omrezja() //funkcija inicializacija_omrezja
{
    SIM900.println("AT+CSQ"); // Izpise jakost siganala
    delay(100);
    ShowSerialData(); // prikazi podatke v serialu.

    SIM900.println("AT+CGATT?"); //
    delay(100);
    ShowSerialData();

    SIM900.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"); //nastavljanje tipa povezave
    delay(1000);
    ShowSerialData();

    SIM900.println("AT+SAPBR=3,1,\"APN\",\"internet\"); //nastavljanje apn tocke
    delay(4000);
    ShowSerialData();

    SIM900.println("AT+SAPBR=1,1"); //omogoci omrezje
    delay(2000);
    ShowSerialData();

    SIM900.println("AT+HTTPIPINIT"); //nakaze HTTP request
    delay(2000);
    ShowSerialData();
}
```

Slika 10: Inicijalizacija omrežja

Vir: lasten

Ko se izvrši zagonski del (`setup()`), v glavnem delu programa oz. v zanki `loop()` kličemo funkcije.

Status_LED in meritve.

```
void loop() {  
  
    status_led();  
    meritve();  
    delay(10000);  
  
}
```

Slika 11: Meritev

Vir: lasten

Funkcija `status_led` pošlje HTTP prošnjo tipa GET na URL, ter izpiše in shrani povratne podatke spletnih baz.

```
void status_led() //funkcija status led  
{  
    SIM900.println("AT+HTTPPARA=\"URL\", \"rak.nestnoot.org/jornik/stanje.php?all\""); // določanje URL strani  
    delay(1000); // iz katere preberemo stanje  
    ShowSerialData();  
  
    SIM900.println("AT+HTTPACTION=0"); //pošle prošnjo (HTTPACTION=0 za metodo GET, HTTPACTION=1 za metodo POST)  
    delay(5000); //vec podatkov, vec delaya  
    ShowSerialData();  
  
    SIM900.println("AT+HTTPEXEC"); // preberí kaj vsebuje HTTP request  
    delay(300);  
  
    Serial.print("\n.. content __>... \n");  
    String content = "";  
    while(SIM900.available() != 0)  
    {  
        content = content + String(char (SIM900.read())); //izpiše respond HTTP prošnje  
    }  
  
    Serial.println(content);  
    Serial.print("\n... content <__... \n");  
  
    /* WHILE zanke za izpis substringov  
    Serial.println("Substringi se začnejo : \n");  
    int i=0, j=1;  
    while(i<40){  
        Serial.println(i);  
        Serial.println(content.substring(i,j));  
        i++; j++;  
    }  
    */  
}
```

Slika 12: Prošnja tipa GET

Vir: lasten

Funkcija meritve pa preveri vrednosti senzorja DH22, ter pošlje HTTP prošnjo tipa POST, ki pošlje vrednost senzorjev na URL.

```
void meritve(){
    int temperaturaC=0;
    int vlaznost=0;
    String data="";
    Serial.println("Meritve: \n");
    temperaturaC = (int)dht.readTemperature(); //Branje temperature v °C,
    vlaznost = (int) dht.readHumidity(); //Branje vlaznosti.
    //Inicijalizacija spremenljivk za URL metodo POST
    String output_temp="templ=";
    String output_hum="huml=";
    data = output_temp + temperaturaC +output_hum + vlaznost;
    Serial.println(data);
    SIM900.println("AT+HTTPPARA=\\"URL\\",\\"http://rak.nootnoot.org/jornik/add_moj.php?"+data+"\\ " "); // nastavimo URL
    delay(1000);
    showSerialData();

    SIM900.println("AT+HTTPACTION=1"); //posle posnje
    delay(5000); //vec podatkov, vec delaja
    ShowSerialData();

    SIM900.println("AT+HTTPREAD"); // preberi kaj vsebuje HTTP request
    delay(300);
    ShowSerialData();
}
```

Slika 13: Vrednost senzorja

Vir: lasten

Funkcija ShowSerialData izpiše trenutne podatke na SIM serijalu.

```
void ShowSerialData() //prikaz podatkov ki so v Serialu
{
    while(SIM900.available() !=0)
        Serial.write(char (SIM900.read()));
}
```

Slika 14 Prkaz trenutnih podatkov na SIM

Vir: lasten

6.2 Spletna stran:

Spletno stran smo naredili uporabniku prijazno ter enosravno. Pri izdelavi smo si pomagali s spletnim urejevalnikom Codiad (<http://codiad.com/>) ter še z dvema spletnima mestoma (<http://www.w3schools.com/>) in (<http://getbootstrap.com/>). Spletno stran smo ustvarili z uporabo naslednjih programskih jezikov:

PHP

»Slovensko »orodja za osebno spletno stran« je razširjen odprtokodni programski jezik, ki se uporablja za strežniške uporabe oziroma za razvoj dinamičnih spletnih vsebin. Podoben je običajno strukturiranim programskim jezikom, najbolj jeziku C, in najbolj izkušenim programerjem dovoljuje razvijanje zapletenih uporab brez dolgega učenja. PHP pa primarno teče na spletnem strežniku, kjer jemlje PHP izvorno kodo za vhod in generira spletno stran kot izhod. Kot del PHP-ja sta tudi možnost zaganjanja skript v ukaznem načinu in kreiranje grafičnih aplikacij. «³

HTML

»Hyper Text Markup Language (slovensko jezik za označevanje besedila), HTML je jezik ki je za izdelavo spletnih strani. Predstavlja osnovo spletnega dokumenta. Poleg prikaza dokumenta v spletnem brskalniku se z njim hkrati določi tudi zgradba in semantični pomen delov dokumenta. Izdela se ga lahko v vsakem urejevalniku besedil npr. beležnica itd., saj je zapisan v obliki elementov HTML, ki so sestavljeni iz značk, zapisanih v špičastih oklepajih (<, >) znotraj vsebine spletne strani.«⁴

CSS

»so podloge, predstavljene v obliki preprostega slogovnega jezika, ki skrbi za prezentacijo spletnih strani. Z njimi definiramo stil HTML oz. XHTML elementov v smislu pravil, kako se naj ti prikažejo na strani. Določamo lahko barve, velikosti, odmike, poravnave, obrobe, pozicije in vrsto drugih atributov, prav tako pa lahko nadziramo aktivnosti, ki jih uporabnik nad elementi strani izvaja (npr. prekritje povezave z miško). Podloge so bile razvite z namenom konsistentnega načina podajanja informacij o stilu spletnim dokumentom. «⁵

³ Vir: Wikipedija (Januar, 2017)

⁴ Vir: Wikipedija (Januar, 2017)

⁵ Vir: Wikipedija (Januar, 2017)

JavaScript

»je objektni skriptni programski jezik, ki ga je razvil Netscape, da bi spletnim programerjem pomagal pri ustvarjanju interaktivnih spletnih strani. Jezik je bil razvit neodvisno od Jave, vendar si z njo deli številne lastnosti in strukture. JavaScript lahko sodeluje s HTML-kodo in s tem poživi stran z dinamičnim izvajanjem. JavaScript podpirajo velika programska podjetja in kot odprt jezik ga lahko uporablja vsakdo, ne da bi pri tem potreboval licenco. Podpirajo ga vsi novejši spletni brskalniki.«⁶

⁶ Vir: Wikipedija (Januar, 2017)

1. Povezovanje na podatkovno bazo.

```
AK/jornik/index_bot.php x RAK/index.php x nik/bootstrap/index.php x /jornik/bootstrap/db.php x RAK/ORIG/db.php
1 <?php
2 $config = array(
3     "db_user" => "*****",
4     "db_pass" => "*****",
5     "db_host" => "*****",
6     "db_db" => "*****"
7 );
8
9 //////////////////////////////////////////////////
10
11 $db = new mysqli($config["db_host"], $config["db_user"], $config["db_pass"], $config["db_db"]);
12
13 if ($db->connect_errno > 0) {
14     die("Unable to connect to database [" . $db->connect_error . "]);
15 }
16
17 ?>
```

Slika 15: Povezovanje na podatkovno bazo

Vir: lasten

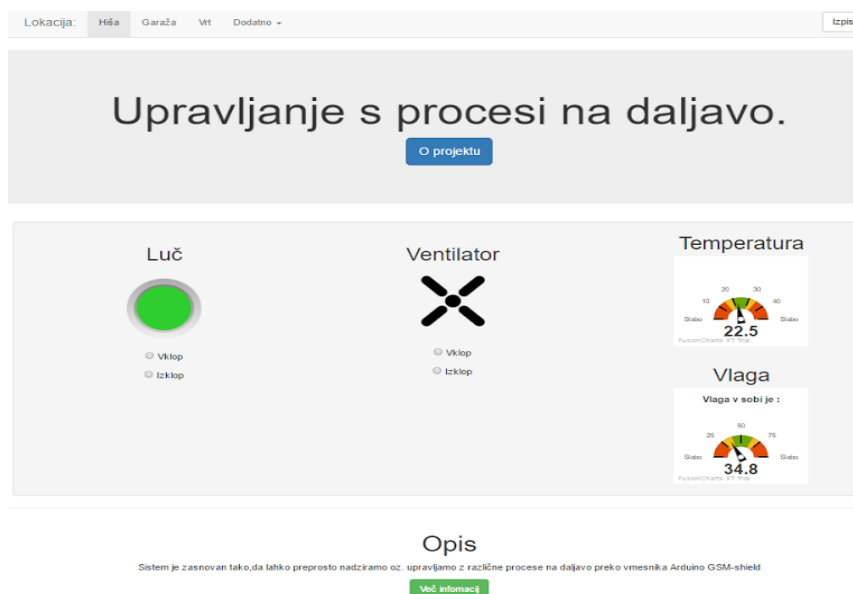
```
6 if ($_SERVER["REQUEST_METHOD"] === "POST") {
7     if (isset($_POST["logout"])) {
8         session_destroy();
9     }
10     if (isset($_POST["user"]) || empty($_POST["user"])) die("Napačno uporabniško ime ali geslo!");
11     else if (isset($_POST["geslo"]) || empty($_POST["geslo"])) die("Napačno uporabniško ime ali geslo!");
12     else {
13         require($back . "/db.php");
14
15         $user = $db->escape_string($_POST["user"]);
16         $geslo = $db->escape_string($_POST["geslo"]);
17
18         $query = "
19             SELECT * FROM `users`
20             WHERE `ime` = '" . $user . "'
21             AND `geslo` = '" . sha1($geslo) . "'";
22
23         if (!$result = $db->query($query)) {
24             die("There was an error running the query [" . $db->error . "]);
25         }
26
27         if ($result->num_rows == 0) die("Napačno uporabniško ime ali geslo!");
28
29         $_SESSION["user"] = $user;
30     }
31     header("Location: " . $back . "/");
32 }
33
34 ?>
35
36 <main>
37 <head>
38 <link rel="stylesheet" href="<?php echo $back ?>/CSS/login.css">
39 <meta name="viewport" content="width=device-width, initial-scale=1">
40 </head>
41 <div class="login-page">
42 <div class="form">
43
44 <form method="POST" action="login.php" >
45 <label>
46 <input type="text" name="user" placeholder="uporabniško ime">
47 </label>
48
49 <div>
50 <label>
51 <input type="password" name="geslo" placeholder="geslo">
52 </label>
53 </div>
54 <input type="submit" value="Prijava" id="prijava">
55 </form>
56
```

Slika 16: Preverjanje prijave

Vir: lasten

Ker je še kode veliko, in je zaenkrat v obdelavi lahko po želji preostanek te kode najdete na (<http://kor.nootnoot.org/ORIG/>)

Trenutno stanje spletne strani (v obdelavi). Testiranje različnih kazalcev, stikal in načinov komuniciranja z podatkovno bazo.



Slika 17: Spletna stran

Vir: lasten

Prijavna stran (eng. login page).

uporabniško ime

geslo

PRIJAVA

Slika 18: Prijavna stran

Vir: lasten

6.3 Podatkovna baza

Trenutno uporabljamo tri podatkovne baze katere smo ustvarili z pomočjo programa phpMyAdmin (<https://www.phpmyadmin.net/>).

Baza (status), kjer se shranjuje stanje komponent (0- pomeni ugasnjeno, 1- vklopljeno).

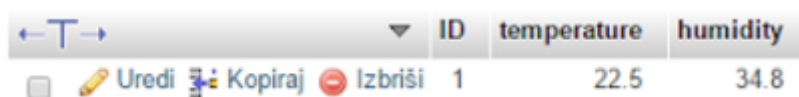


	ID	komponenta	stanje
<input type="checkbox"/> Uredi Kopiraj Izbrisi	1	led	1
<input type="checkbox"/> Uredi Kopiraj Izbrisi	2	ventilator	0

Slika 19: Baza za shranjevanje stanj komponent

Vir: lasten

Baza (log), kjer se vsaki dve sekundi posodablja vrednost temperature in vlage.



	ID	temperature	humidity
<input type="checkbox"/> Uredi Kopiraj Izbrisi	1	22.5	34.8

Slika 20: Baza ki se posodablja

Vir: lasten

Ter baza (users) v kateri se hranijo uporabniška imena in gesla uporabnikov. Gesla se hranijo v podatkovni bazi po algoritmu SHA1.



	ime	geslo
<input type="checkbox"/> Uredi Kopiraj Izbrisi	kuney	6e0ece719a48d5334369bd881b4324
<input type="checkbox"/> Uredi Kopiraj Izbrisi	ornik	d3c0ecab2b0b14eae7bcc7c57947c9
<input type="checkbox"/> Uredi Kopiraj Izbrisi	rado	170be63d6f9233f16aee5be58ec142e

Slika 21: baza ki uporablja SHA1 kodiranje

Vir: lasten

7 DRUŽBENE ODGOVORNOSTI

Z našo idejo želimo različne daljinske nadzore in upravljanja poenostaviti na eno platformo z modularno sestavo, in s tem razširiti možnosti uporabe. Kot sam cilj pa je pomembna cenovno ugodnejša postavitve daljinsko nadzornega in krmilnega sistema glede na uporabnikove želje.

8 ZAKLJUČEK

S tem kar smo naredili v tej nalogi smo zadovoljni. Največ zapletov se je pojavilo pri ustvarjanju povezav med podatkovno bazo in mikrokrmilnikom ter pri izdelavi modularne spletne strani. V nalogi si je vsak od nas pridobil veliko uporabnega znanja pri programiranju mikrokrmilnika in na področju oblikovanju spletnih strani.

9 VIRI

9.1 Pisni viri

- Jon Duckett; HTML and CSS: Design and Build Websites, 2011, New Jersey, John Wiley & Sons

9.2 Elektronski viri

- Raspberry Pi, <https://www.raspberrypi.org>, (8.12.2016)
- SCADA http://www.modan.si/index.php?option=com_content&view=article&id=6 (4.12.2016)
- Arduino <https://www.arduino.cc/>, (4.12.2016)
- GSM in GPRS, <https://www.arduino.cc/en/Tutorial/GSMToolsTestGPRS> (3.12.2016)
- SQL, <http://www.w3schools.com/sql/>, (18.12.2016)