Ugrađeni računalni sustav za meteorološku stanicu

Zidar, Josip

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:042720

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

UGRAĐENI RAČUNALNI SUSTAV ZA METEOROLOŠKU STANICU

Završni rad

Josip Zidar

Osijek, 2016.

Sadržaj

1. UVOD 1

1.1. Zadatak završnog rada 1

1.2. Postojeći uređaji na tržištu 2

1.3. Poveznica na izrađenu stranicu 3

2. PRIMIJENJENE TEHNOLOGIJE I ALATI 4

2.1. Arduino razvojno okruženje 4

2.1.1. WiFi modul ESP8266 6

2.1.2. Prikaz senzora 8

2.2. CadSoft EAGLE 10

3. REALIZACIJA SUSTAVA 11

3.1. Povezivanje na Arduino 11

3.1.1. Senzori 12

3.1.2. WiFi modul 17

3.2. Stvaranje baze podataka 18

3.3. Stvaranje web stranice 20

4. ZAKLJUČAK 21

LITERATURA 22

SAŽETAK 23

ABSTRACT 24

ŽIVOTOPIS 25

PRILOZI 26

1. UVOD

Cilj ovog završnog rada je napraviti uređaj koji mjeri sve osnovne meteorološke parametre (temperaturu i vlagu zraka, atmosferski tlak, kišu, brzinu vjetra). Takav uređaj mora obavljati sve automatski, bez ikakvog prisustva čovjeka, i mora sadržavati sve potrebne sklopove za mjerenje meteoroloških parametara zajedno sa sklopom koji obavlja povezivanje na internet. Takav bi sustav mjerio podatke u stvarnom vremenu i slao ih na web stranicu. Web stranica bi zatim koristila prikupljene podatke za prikazivanje. Takav uređaj mora imati i odgovarajuće kučište koje može podnijeti sve uvjete (vlagu, promjenu temperature, kišu, led, snijeg itd.) i mora imati svoje napajanje (baterije ili DC adapter). Kroz rad će se pojasniti koji su moduli korišteni, zatim izrada stranice i stvaranje baze podataka, te u konačnici, sama realizacija sustava.

1.1. Zadatak završnog rada

U ovom završnom radu potrebno je napraviti maketu meteorološke stanice koja na internetu prikazuje izmjerene vrijednosti vjetra, temperature, vlage, magle, kiše te prikazuje sliku sa mjernog mjesta. Na internet stranici potrebno je ispisivati izmjerene vrijednosti tablično i grafički.

1.2. Postojeći uređaji na tržištu

Automatske meteorološke stanice mogu se koristiti za razne potrebe. Najprije, one su tu da pruže meteorološke podatke čijom se daljom obradom može postići mnogo toga, kao na primjer:

- klimatološke analize
- prognoza vremena
- agro-prognoze
- prognoze pojave biljnih štetočina
- procjena povoljnosti za održavanje nekog sportskog događaja ili koncerta na otvorenom

Sl. 1.1 Automatska meteorološka stanica

Meteorološke stanice najpotrebnije su samoj meteorologiji**,** a potom i poljoprivredi**,** sportu **i** turizmu**.** Pojedine (skuplje) stanice imaju opciju postavljanja info-table sa najnovijim podacima, čime se postiže njihova potpuna iskorištenost u turizmu ili na nekom sportskom događaju.

1.3. Poveznica na izrađenu stranicu

Izrađenoj stranici može se pristupiti na: www.meteoroloska-stanica.tk

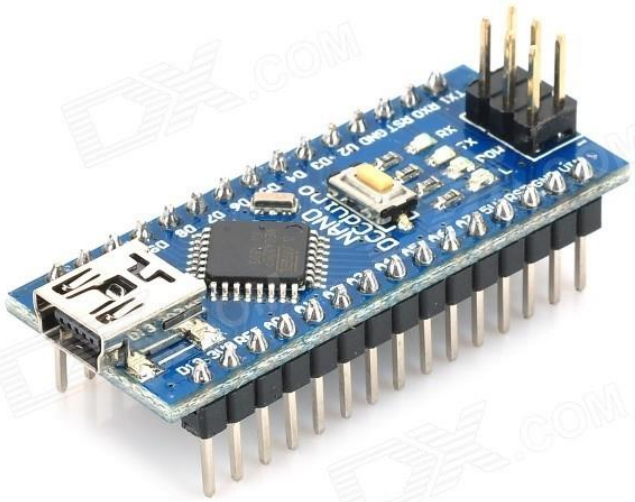
2. PRIMIJENJENE TEHNOLOGIJE I ALATI

2.1. Arduino razvojno okruženje

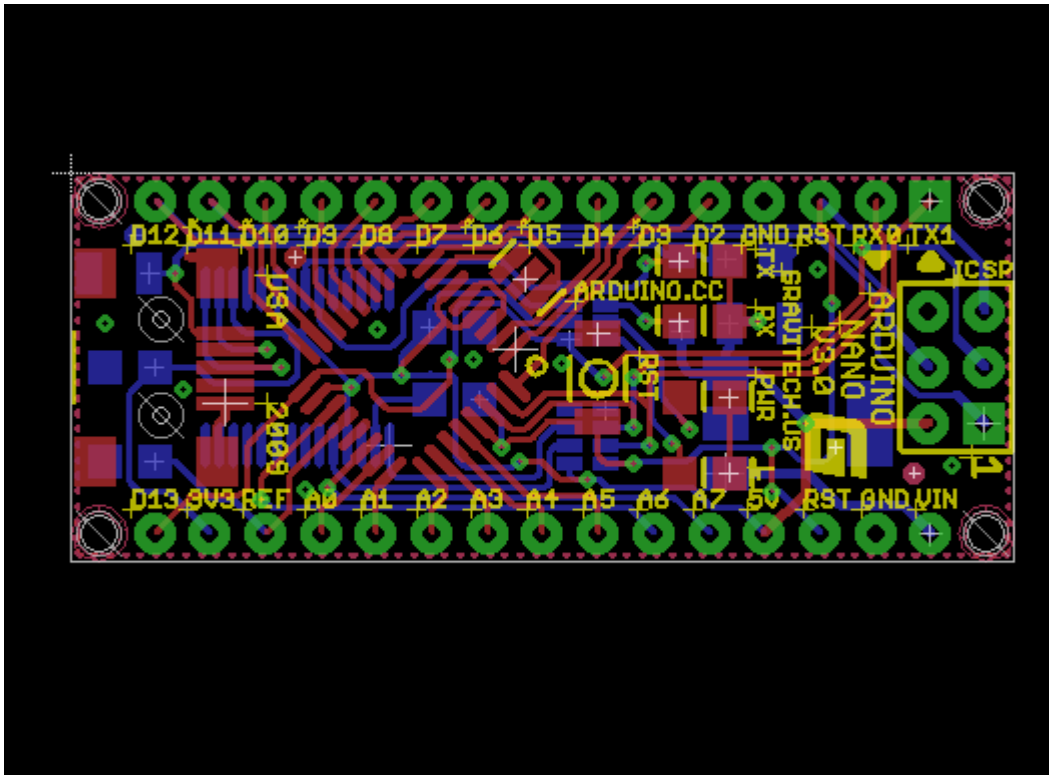
Pri odabiru mikrokontrolera kao najbolji odabir bio je Arduino Nano (ATmega328), zbog svojih malih dimenzija i odgovarajuće funkcionalnosti.

Tab 2.1. Arduino Nano specifikacije

Microcontroller	ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"
Length	45 mm
Width	18 mm
Weigth	5 g



Sl. 2.1. Arduino Nano

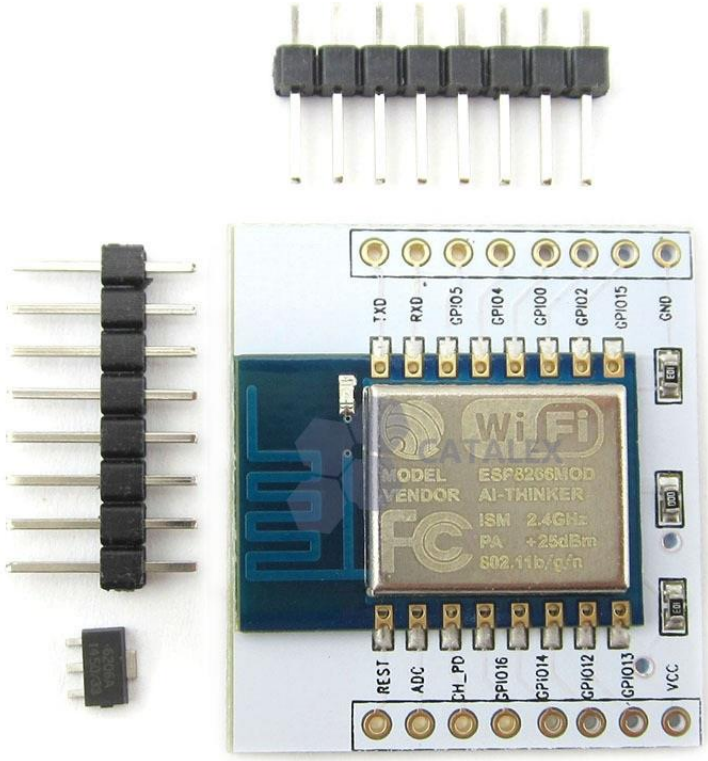


Sl. 2.2. Tiskana pločica Arduino Nano izrađena pomoću programskog paketa EAGLE

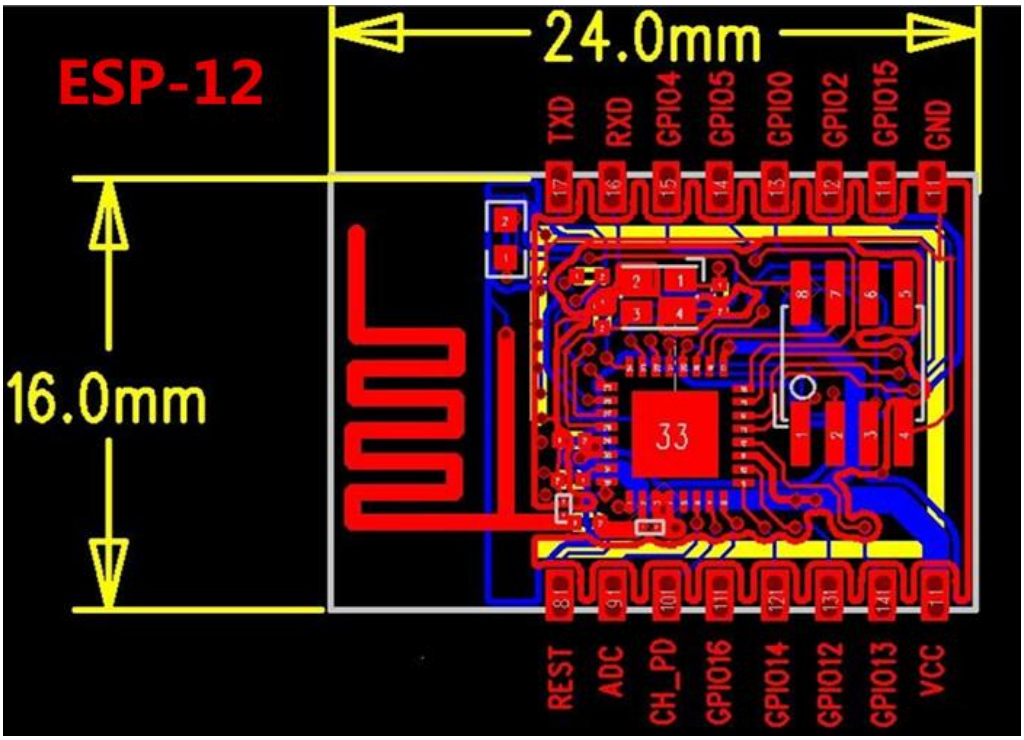
2.1.1. WiFi modul ESP8266

Za slanje podataka odabran je ESP8266 ESP-12 WiFi modul koji podržava 802.11 b/*g*/n standard, i u sebi ima ugrađene načine rada WiFi(AP, STA, AP+STA). To su sljedeća tri načina rada: AP (Access Point), STA (Station) i AP+STA (kombinacija oba).

ESP8266 je kompletan 32-bitni SoC(System on Chip) s ugrađenom flash memorijom, RAMom i EEPROMom koji se može direktno programirati iz Arduino sučelja, bez potrebe za dodatnim mirkokontrolerom.



Sl. 2.3. ESP-12 WiFi modul



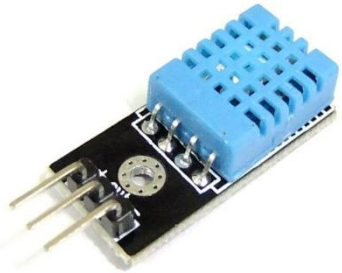
Sl. 2.4. ESP-12 tiskana pločica

2.1.2. Prikaz senzora

Za prikupljanje podataka koriste se senzori prikazani u tablici 2.2.

Tab 2.2. Karakteristike korištenih senzora

Naziv	**Opis**	**Dimenzije**	**Cijena**
DHT11	mjerenje vlage i temperature	15.5mm (D) x 12mm (Š) x 5.5mm (V)	$1.24
DS3231	real time clock, mjerenje vremena	38mm (D) x 22mm (Š) x 14mm (V)	$0.99
BMP180	mjerenje tlaka	3.6mm (D) x 3.8mm (Š) x 0.93mm (V)	$1.99
Tachometer speed module	mjerenje brzine vjetra ili mjerenje broja okretaja motora	32mm (D) x 14mm (Š)	$2.27
Rain module	detekcija kiše	32mm (D) x 15mm (Š)	$0.99

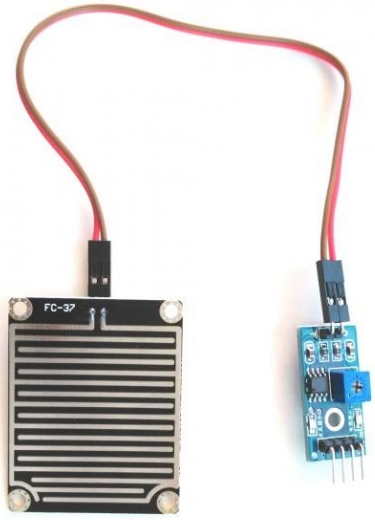


Sl. 2.5. DHT11

Sl. 2.6. DS3231

Sl. 2.7. BMP180

Sl. 2.8. Tachometer speed module



Sl. 2.9. Rain module

2.2. CadSoft EAGLE

CadSoft EAGLE je komercijalni programski paket za uređivanje i izradu električnih shema. Velik broj korisnika, velika baza komponenti i mnogo opcija pri dizajniranju električnih shema razlozi su zbog kojih je ovaj program veoma popularan, a također postoje probna i besplatna verzija (za nekomercijalnu upotrebu). Programski paket CadSoft EAGLE sadrži veliku bazu elektroničkih komponenti, pa tako osim električnih shema omogućuje i jednostavnu izradu tiskanih pločica. Postoji i ručno dodavanje komponenti koje se dodatno mogu naći na internetu (ukoliko komponente nema u bazi). Korištenje besplatne verzije donosi nekoliko ograničenja. Neka od najvažnijih ograničenja su širina tiskane pločice (max. 100mm x 80mm) i ograničen broj slojeva (max. 2). Korištenje plaćene verzije uklanja sva ograničenja besplatne verzije.

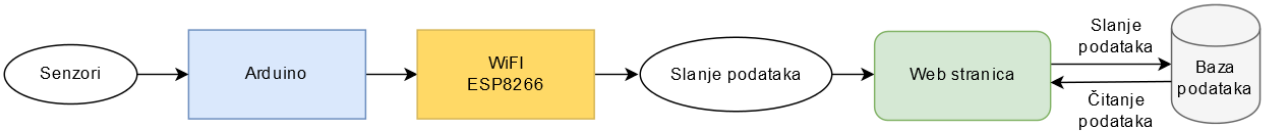
Slika 2.2. prikazuje izgled Arduino Nano tiskane pločice u programskom paketu CadSoft EAGLE. Na slikama u 3. poglavlju nalaze se električne sheme spajanja senzora koje su napravljene korištenjem CadSoft EAGLE programskog paketa.

Sl. 2.10. CadSoft EAGLE 7.6.0

3. REALIZACIJA SUSTAVA

Sustav je izrađen tako da obuhvaća prikupljanje podataka (meteorološki uvjeti) koji se zatim preko Arduino mikrokontrolera obrađuju i šalju preko WiFi modula ESP8266 na web stranicu. Web stranica zaprima zahtjev za slanje podataka putem TCP-a (Transfer Control Protocol). Zaprimljeni podaci se preko stranice zatim šalju u bazu podataka. U bazi podataka se sakupljaju svi podaci koji se koriste za prikaz na stranici. Prilikom posjete na stranicu, učitava se zadnjih pet mjerenja iz baze podataka.

Prilikom izrađivanja stranica, da bi one bile dostupne na Internetu, potrebno je da se smjeste na neki web server. Web server ima funkciju obrađivanja svih zahtjeva korisnika. U ovom slučaju obavljaju se zahtjevi za prikaz stranice (čitanje iz baze podataka), i slanje podataka u bazu podataka.



Sl. 3.1. Arhitektura sustava

3.1. Povezivanje na Arduino

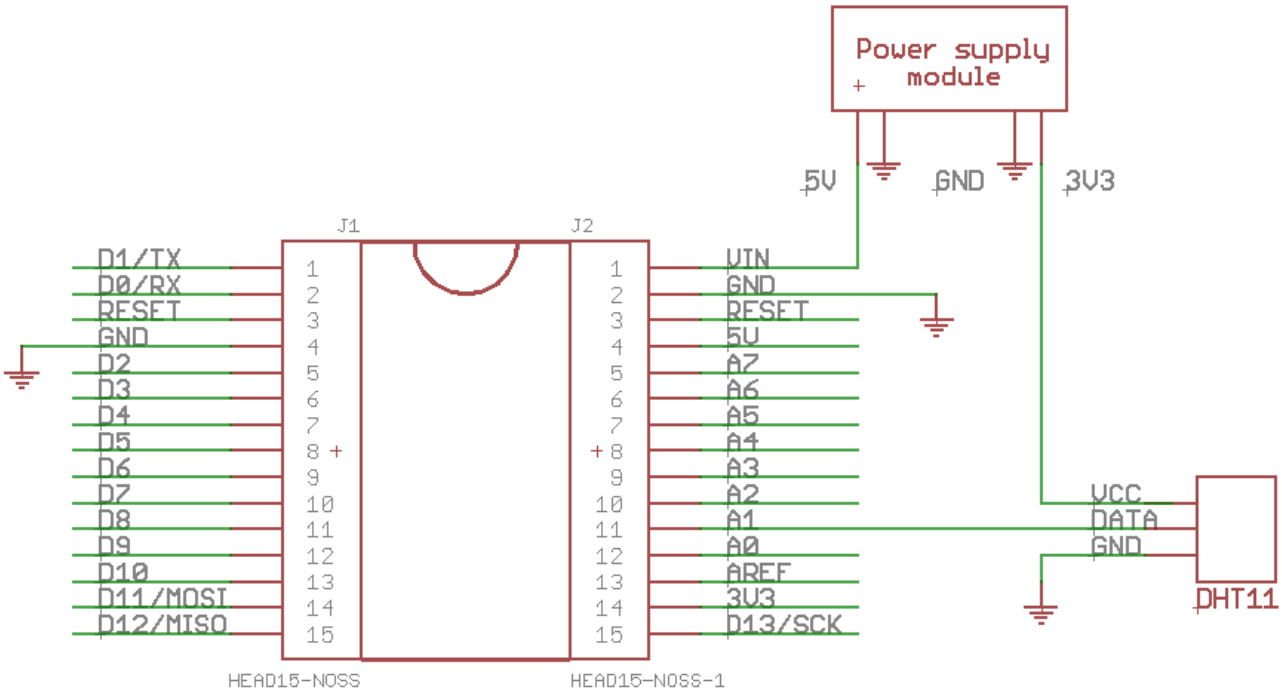
U ovoj cjelini detaljno se prikazuju spojevi pinova Arduino mikrokontrolera sa pinovima senzora i pinovima WiFi modula. Spajanje se obavlja na bredboardu na kojem je priključen i dodatni izvor napajanja, zbog toga što Arduino Nano nema dovoljnu izlaznu snagu. Prikazano je zasebno spajanje svakog modula na Arduino i na izvor napajanja.

3.1.1. Senzori

DHT11 VCC senzora => 3.3V napajanja

OUT senzora => A1 pin Arduino

GND senzora => GND napajanja



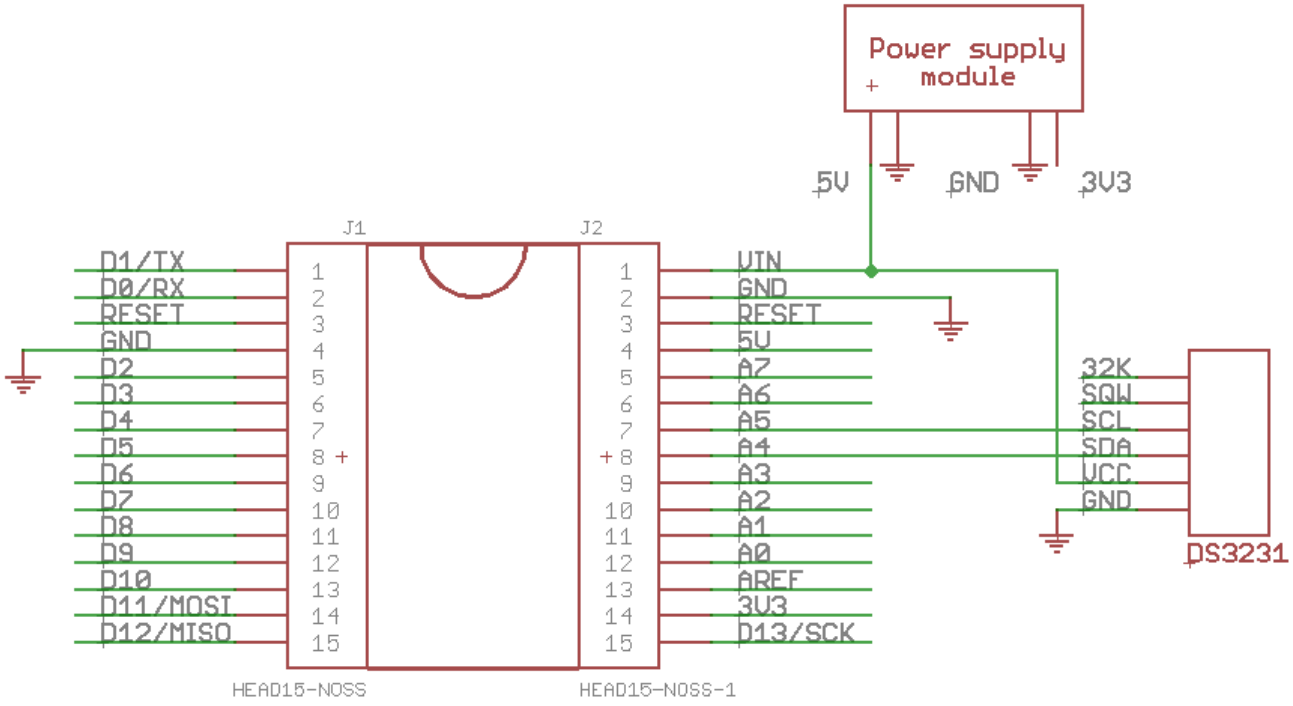
Sl. 3.2. Električna shema spajanja DHT11 modula

DS3231 VCC senzora => 5V napajanja

SDA senzora => A4 pin Arduino

SCL senzora => A5 pin Arduino

GND senzora => GND napajanja



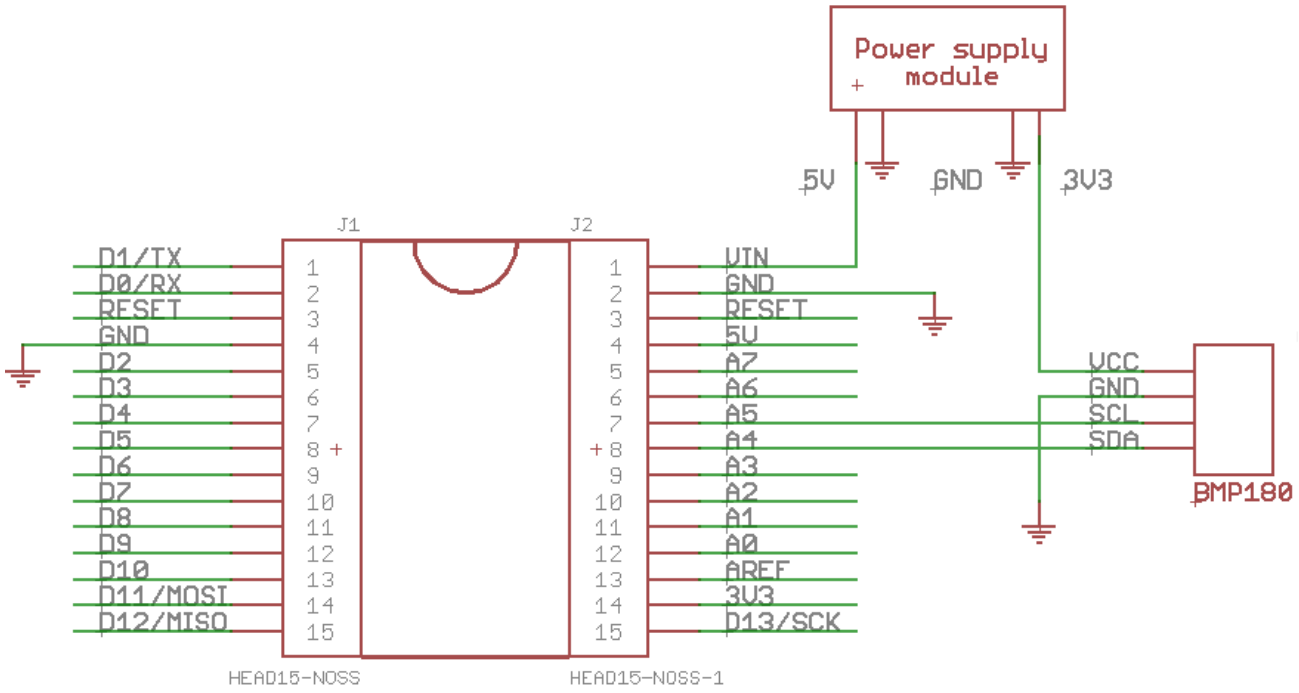
Sl. 3.3. Električna shema spajanja DS3231 modula

BMP180 VCC senzora => 3.3V napajanja

SDA senzora => A4 pin Arduino

SCL senzora => A5 pin Arduino

GND senzora => GND napajanja

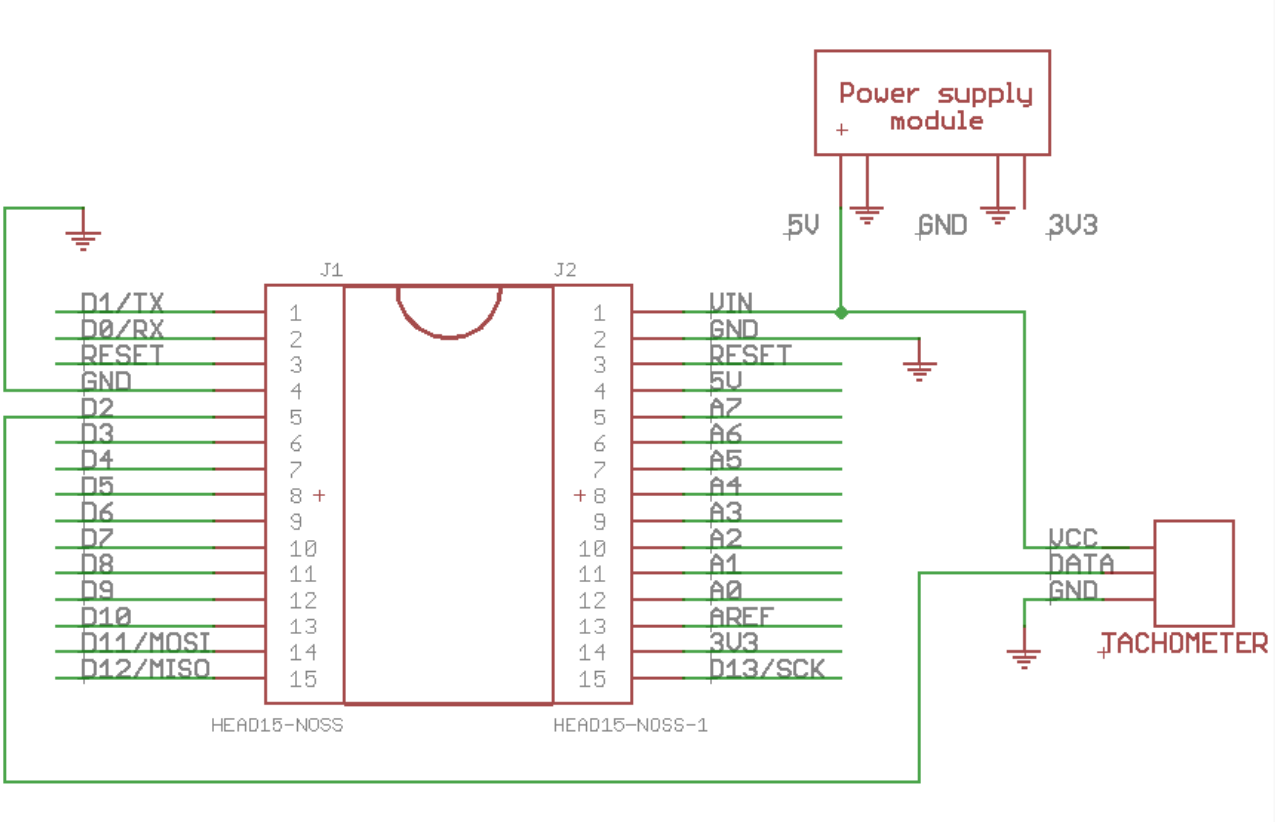


Sl. 3.4. Električna shema spajanja BMP180 modula

Tachometer VCC senzora => 5V napajanja

OUT senzora => D2 pin Arduino (interrupt)

GND senzora => GND napajanja

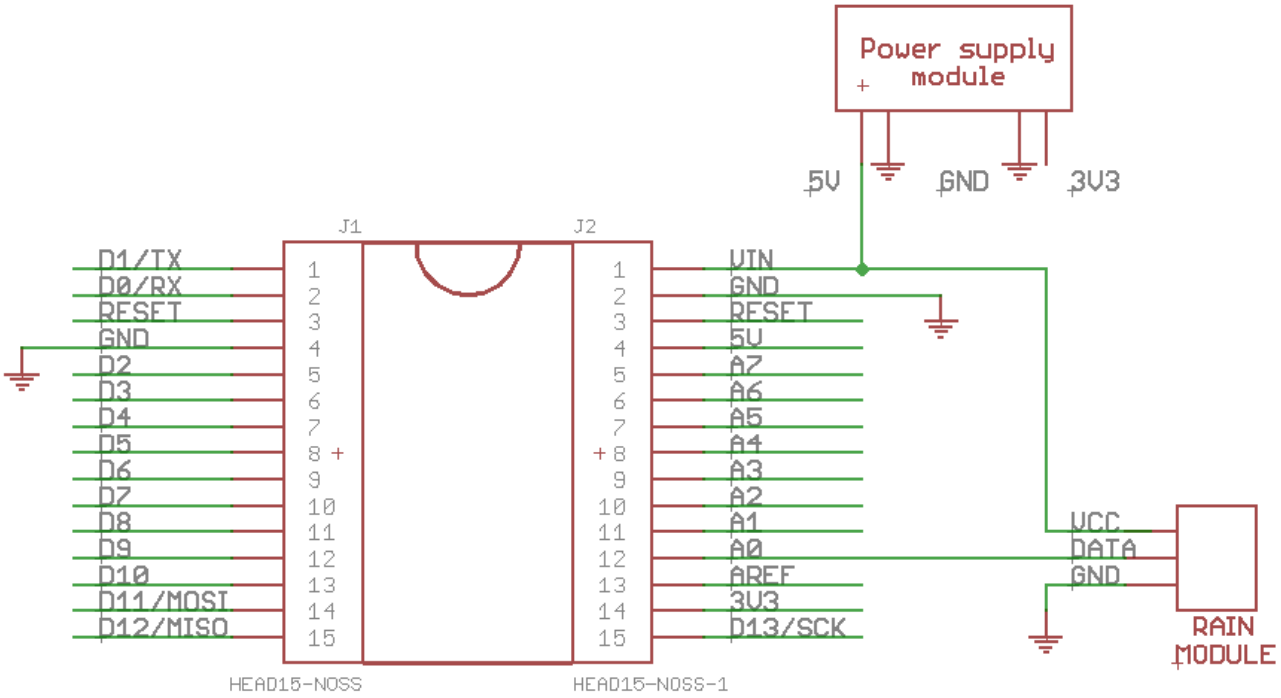


Sl. 3.5. Električna shema spajanja tahometar modula

Rain module VCC senzora => 5V napajanja

A0 senzora => A0 pin Arduino

GND senzora => GND napajanja



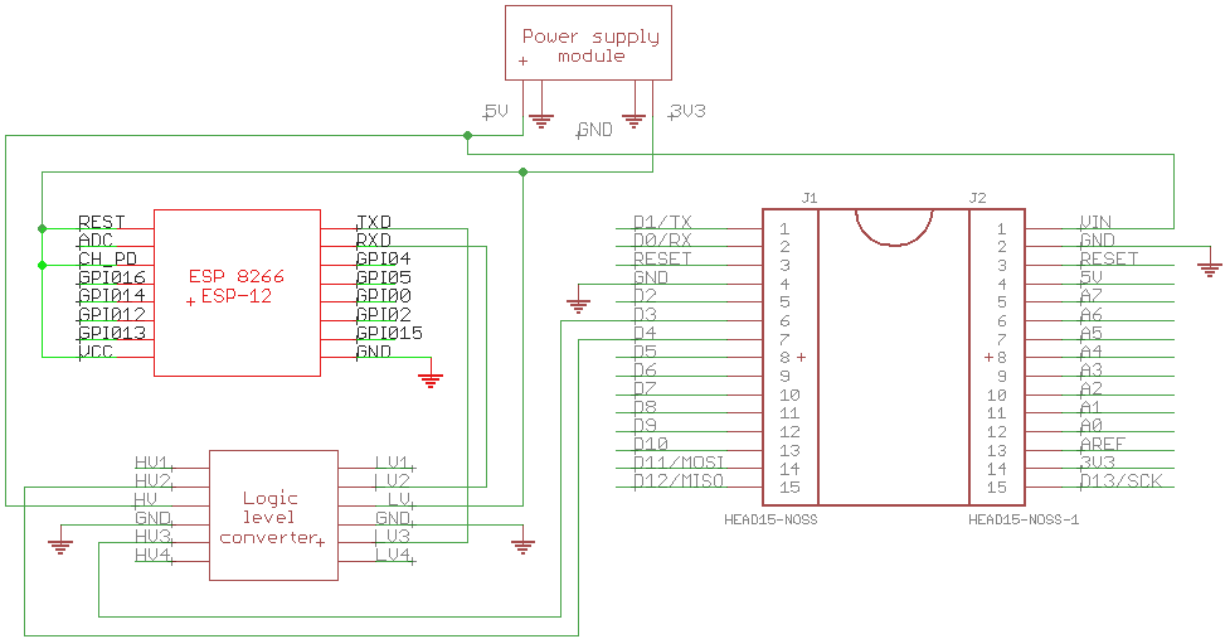
Sl. 3.6. Električna shema spajanja modula za detekciju kiše

3.1.2. WiFi modul

ESP8266 ESP-12	VCC modula => 3.3V napajanja
	REST modula => 3.3V napajanja
	CH_PD modula => 3.3V napajanja
	GND modula => GND napajanja

Za komunikaciju između WiFi modula i Arduina potreban je logic level converter u oba smjera (Bi-Directional).

Logic level converter	HV convertera => 5V napajanja
	LV convertera => 3.3V napajanja
	GND convertera => GND napajanja
	HV2 convertera => D4 pin Arduino
	LV2 convertera => RXD WiFi modula
	HV3 convertera => D3 pin Arduino
	LV3 convertera => TXD WiFi modula



Sl. 3.7. Električna shema spajanja WiFi modula i logic level convertera sa Arduinom

Kod komunikacije između Arduina i WiFi modula, pin D3 predstavlja RX Arduina, a pin D4 predstavlja TX Arduina.

RX Arduina (D3) => logic level converter => TXD WiFi modula

TX Arduina (D4) => logic level converter => RXD WiFi modula

3.2. Stvaranje baze podataka

Za pohranu mjerenih podataka potrebna je baza podataka. Odabrana je stranica www.db4free.net koja omogućava baze podataka u svrhe testiranja i razvoja aplikacija.

Putem phpMyAdmin izrađena je baza podataka pod imenom meteoroloska, sa slijedećim parametrima:

- timeStamp (TIMESTAMP)
- temperatura (INT)
- vlaga (INT)
- tlak (INT)
- vjetar (FLOAT)
- kisa (INT)

PhpMyAdmin je besplatan softverski alat napisan u PHPu, namijenjen za rukovanje MySQL baze podataka preko weba. Često korištene operacije (upravljanje bazama podataka, tablice, stupci, odnosi, indeksi, korisničke dozvole, itd.) mogu se izvesti putem korisničkog sučelja, dok i dalje ostaje mogućnost direktnog izvršavanja SQL koda.

3.2. Stvaranje web stranice

Za izradu nove web stranice potrebni su domena (naziv ne smije biti zauzet) i web hosting. Prvi korak u izradi web stranice je registracija domene. Na stranici www.dot.tk se obavlja provjera dostupnosti naziva stranice i registracija besplatne domene. Potrebno je registrirati se, i potom se odabire naziv stranice koji je slobodan. Moraju se navesti i DNS serveri hostinga.

DNS serveri za 000webhost su: ns01.000webhost.com i ns02.000webhost.com

Nakon registracije domene, potrebna je izrada računa na www.000webhost.com. 000webhost je stranica koja nudi besplatni web hosting te omogućuje bogato korisničko sučelje i Cpanel sa PHPom i MySQLom. Veoma je pouzdan, te među najpopularnijim servisima za besplatan hosting.

Zadnji korak je stvaranje stranice na 000webhost-u, tako da se unese naziv stranice, i odabere lozinka za pristup Cpanelu. Na korisnički račun se zatim dodjeljuje naziv stranice i nakon toga se obično čeka određeno vrijeme da se stranica podigne (može biti 5 minuta, a može biti i dan ili dva), ali to obično bude veoma kratko.

Izrađena stranica: www.meteoroloska-stanica.tk

4. ZAKLJUČAK

U ovom završnom radu napravljen je sustav za mjerenje meteoroloških parametara. Sustav se može podijeliti na: sklop za mjerenje podataka (Arduino i pripadajući senzori), sklop za slanje podataka (WiFi modul) te prikazivanje podataka (web stranica). Sustav je uspješno realiziran i uspješno dostavlja podatke. Prednosti ovakvog sustava su veoma ažurno dostavljanje podataka (svakih 5 minuta), te da podatke možemo gledati bilo gdje (dovoljan je samo pristup internetu). Nedostaci ovog sustava su udaljenost do WiFi mreže, pa se stanica mora postaviti unutar dometa WiFi signala, te blizina napajanja (potreban je DC adapter). Problem blizine napajanja može se riješiti ugradnjom baterija ili kakvim sustavom solarnog napajanja.

LITERATURA

[1] Arduino razvojno okruženje, https://www.arduino.cc/, lipanj 2016.

[2] Tutorial o slanju podataka na web stranicu, http://www.instructables.com/id/PART-1-Send-Arduino-data-to-the-Web-PHP-MySQL-D3js/step2/PHP-MySQL-Application/, lipanj 2016.

[3] Automatske meteorološke stanice, http://meteoplaneta.rs/automatske-meteoroloske-stanice/, srpanj 2016.

[4] ESP8266 modul opis, https://e-radionica.com/hr/esp8266-esp-12-wifi-modul.html, srpanj 2016.

[5] Arduino Nano mikrokontroler, https://www.arduino.cc/en/Main/ArduinoBoardNano, rujan 2016.

[6] CadSoft EAGLE programski paket, https://cadsoft.io/, rujan 2016.

SAŽETAK

Naslov: Ugrađeni računalni sustav za meteorološku stanicu

U ovom radu izrađen je sustav koji obavlja prikupljanje meteoroloških podataka iz okoline te ih putem WiFi mreže šalje na web stranicu. Sustav se sastoji od: senzorskih jedinica povezanih na mikrokontroler (prikupljanje podataka), WiFi modula (slanje podataka), i internet stranice (prikaz podataka). Mjerenje podataka iz okoline obavlja se svakih 5 minuta, a zatim se mjereni podaci spremaju u bazu podataka. Na Web stranici se prikazuju najnoviji podaci iz baze podataka. Kroz rad prikazani su svi korišteni senzori i moduli, a objašnjena je i sama realizacija sustava. Posljednje poglavlje završava sa pojašnjenjima izrade web stranice i baze podataka.

Ključne riječi: Arduino, elektronika, internet stvari, meteorološka stanica

ABSTRACT

Title: Embedded computer system for weather station

In this thesis a system that carries out the collection of meteorological data from the environment was developed, it uses WiFi network to send data to the website. The system consists of: sensor unit connected to the microcontroller (data collection), WiFi module (sending data), and the website (data display). Data measurement from the environment is preformed every 5 minutes, and then the measured data are stored in the database. The website lists the latest data from the database. Through thesis all the used sensors and modules are displayed, and realization of the system is explained. The last chapter ends with explanations of making website and database.

Keywords: Arduino, electronics, internet of things, weather station

ŽIVOTOPIS

Josip Zidar rođen je 27.09.1994. g. u Virovitici. Od svog rođenja živi u Virovitici. Prvi put se s programiranjem susreće u sedmom razredu osnovne škole. Nakon osnovne škole I.B. Mažuranić u Virovitici, upisuje Tehničku školu u Virovitici, smjer Elektrotehnika te maturira 2013. g. Iste godine upisuje sveučilišni studij na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija, smjer elektrotehnika, a zatim se godinu kasnije premješta na smjer računarstvo. Cilj mu je u budućnosti razviti svoja znanja i vještine u računarstvu, sa naglaskom na područje internet stvari (IoT).

PRILOZI

Svi kodovi (uključujući i zavrsni.ino), kao i svi dodatni materijali (električne sheme, HTML kodovi, slike itd.) nalaze se na CD-u priloženom uz ovaj rad. U nastavku slijedi kod važan za praćenje ovog rada.

Prilog A. zavrsni.ino

```
// Meteoroloska stanica

// rtc
byte second = 0;
byte minute = 0;
byte hour = 0;
byte weekday = 0;
byte monthday = 0;
byte month = 0;
byte year = 0;
long int t1; // brojac vremena
long int t2;
#define RTC_senzor 104

// humidity
#include <DHT.h>
#define DHTPIN A1
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// tachometer
unsigned int counter=0;
void blink() {
counter++; // okretaj
}

//pressure
#include <Wire.h>
#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp;

// WIFI
#include <SoftwareSerial.h>
SoftwareSerial ESPserial(3, 4); // RX | TX
String server = "www.meteoroloska-stanica.tk";
String uri = "/add.php";

String t;  // temperatura
String h;  //  vlaga
String p; // tlak
String n; // vjetar
String k; // kisa
String data;
```

```
void setup() {
    Serial.begin(115200);
    Wire.begin();
    ESPserial.begin(115200); // wifi
    reset();
    ESPserial.println("AT+CWMODE=1"); //postavljanje wifi moda
    delay(2000);
    String cmd="AT+CWJAP=\"";
    cmd+="ime_mreze";
    cmd+="\",\"";
    cmd+="lozinka_mreze";
    cmd+="\"";
    ESPserial.println(cmd); //spajanje na wifi mreĹľu
    delay(5000);
    attachInterrupt(0, blink, RISING); // tachometer d2 pin
    dht.begin(); // dht11
    t1 = getUnixTime(); //rtc
    bmp.begin(); //pressure
    data = "";
}

void loop() {
  t2 = getUnixTime();
  if(t2 >= t1+300) // 5 minuta pomak
  {
    t1 = t2;
    // rain
    Serial.print("\t(Rain) Kisa: ");
    k = String(map(analogRead(A0), 0, 1023, 100, 0));
    Serial.println(k);
    // dht11
    h = String(dht.readHumidity());
    Serial.print("\t(Hum) Vlaga: ");
    Serial.println(h);
    t = String(dht.readTemperature());
    Serial.print("\t(Hum) Temp: ");
    Serial.println(t);
    // tachometer
    n = String((float)(0.13*3.14159*counter/30.0) , 2);
    Serial.print("\t(Tacho) Broj okretaja / s: ");
    Serial.println(n);
    //pressure
    p = String(bmp.readPressure());
    Serial.print("\t(Pressure) Tlak: ");
    Serial.print(p);
    Serial.println(" Pa");
    // slanje podataka
    data = "temp=" + t + "&vlaga=" + h +"&tlak=" + p
    + "&vjetar=" + n + "&kisa=" + k;
    httppost();
    counter = 0;
    Serial.println("\n");
  }
  delay(1000);
}
```

```
void httppost () {
    ESPserial.println("AT+CIPSTART=\"TCP\",\"" + server + "\",80"); // TCP
    if(ESPserial.find( (char*)"OK")) {
        Serial.println("TCP uspostavljen");
    }
    delay(1000);

  String postRequest = "POST /add.php HTTP/1.0\r\nHost: " + server + "\r\n" +
  "Accept: *" + "/" + "*\r\n" +
  "Content-Length: " + data.length() + "\r\n" +
  "Content-Type: application/x-www-form-urlencoded\r\n" +
  "\r\n" + data;

  ESPserial.print("AT+CIPSEND=");
  ESPserial.println(postRequest.length() );
  delay(500);
  ESPserial.print(postRequest);
  Serial.println("Slanje..");
  ESPserial.println("AT+CIPCLOSE");
}

void reset() {
ESPserial.println("AT+RST");
delay(1000);
if(ESPserial.find( (char*)"OK") ) Serial.println("Module Reset");
}

//rtc funkcije
void readTime() {
  Wire.beginTransmission(RTC_senzor);
  Wire.write(byte(0));
  Wire.endTransmission();
  Wire.requestFrom(RTC_senzor, 7);
  second = bcdToDec(Wire.read());
  minute = bcdToDec(Wire.read());
  hour = bcdToDec(Wire.read());
  weekday = bcdToDec(Wire.read());
  monthday = bcdToDec(Wire.read());
  month = bcdToDec(Wire.read());
  year = bcdToDec(Wire.read());
}

byte decToBcd(byte val) {
  return ((val/10*16) + (val%10));
}
byte bcdToDec(byte val) {
  return ((val/16*10) + (val%16));
}


#define SEC_1970_TO_2000 946684800
static const uint8_t dim[] = { 31,28,31,30,31,30,31,31,30,31,30,31 };
```

```
long getUnixTime()
{
  uint16_t  dc;
  readTime();
  dc = monthday;
  for (uint8_t i = 0; i<(month-1); i++)
    dc += dim[i];
  if ((month > 2) && (((year-2000) % 4) == 0))
    ++dc;
  dc = dc + (365 * (year-2000)) + (((year-2000) + 3) / 4) - 1;

  return ((((((dc * 24L) + hour) * 60) + minute) * 60) + second) +
SEC_1970_TO_2000;

}
```