

VELEUČILIŠTE U RIJECI
ODJEL TELEMATIKA

Meteorološka stanica na Arduino platformi

projektni rad

Kolegij: Projekt u Telematici
Mentor: Marino Franušić
Student: Petar Faltin

Rijeka, prosinac 2013.

Sadržaj

1. Teorijska pozadina projekta meteorološke postaje.....	3
1.1 Meteorološke postaje	3
1.2 Korišteni senzori.....	4
1.3 Arduino platforma.....	7
2. Detaljan opis korištene Arduino Mega2560 platforme.....	8
3. Pregled postojećih projekata koji koriste Arduino platformu za izradu meteorološke postaje.....	12
4. Opis cjelokupnog projekta meteopostaje (dio kojeg je ovaj projekt).....	13
5. Opis tehničkog rješenja Arduino meteorološke stanice.....	14
6. Terminski plan izrade projekta.....	17
7. Opis izrade programa za dohvaćanje podataka sa senzora i ispis koda.....	19
8. Opis serverske aplikacije za prihvatanje podataka.....	26
9. Analiza rada u usporedbi s planiranim terminima i zamišljenim rješenjem.....	26
10. Troškovnik izrade projekta.....	27
11. Električna shema projekta.....	28
12. Ispis cjelokupnog koda programa.....	29
Literatura	33

1. Teorijska pozadina projekta meteorološke postaje

1.1 Meteorološke postaje

Meteorološke postaje

Meteorološke postaje služe za kontinuirano praćenje i bilježenje stanja atmosfere.

Stanje atmosfere se u osnovi opisuje mjerenjem slijedećih veličina:

- temperatura zraka
- vlažnost zraka
- smjer vjetra
- brzina vjetra
- količina padavina
- tlak zraka

Ovo su osnovni parametri koji se bilježe i kod najjednostavnijih amaterskih postaja.

Profesionalne postaje sa promatračima bilježe veći broj meteoroloških parametara:

- trajanje sijanja Sunca
- globalno zračenje
- difuzno zračenje
- temperatura zraka na 5 cm iznad tla
- temperature tla na standardnim dubinama
- temperatura vode
- ostale komponente Sunčevog zračenja
- UV-B zračenje
- Visina baze oblaka
- Vrsta oblaka
- Visina snijega
-

Današnje profesionalne postaje su uglavnom automatske ali određen broj parametara se mora procijeniti od strane obučenog promatrača kao što je tip oblaka, vidljivost...

Da bi mjerenja bila usporediva svjetska meteorološka organizacija je propisala standarde načina mjerenja i njihovog bilježenja koji su obrađeni u dokumentu "*Guide to Meteorological Instruments and Methods of Observation*", zadnje izdanje 2008 god revidirano 2010 godine.

Današnje amaterske postaje su uglavnom automatske zasnovane na elektroničkom mjerenju i stalno su spojene na Internet radi prikaza podataka. Set veličina koje bilježe je uglavnom osnovni.

Podaci su dijele preko web stranica meteoroloških udruga (istramet.com, pljusak.hr) ili globalnih

servisa (www.wunderground.com)

Amaterizam je dosta raširen i u Hrvatskoj postoji oko 350 amaterskih postaja okupljenih u više udruga.

1.2 Korišteni senzori u meteorološkim postajama

Senzori su naprave koje neke veličine prevode u elektronskim putem mjerljive veličine kao napon, struja, otpor, kapacitet...

Vjetrokaz

Smjer vjetra se mjeri određivanjem položaja vjetrokaza, krilca koje se slobodno usmjerava u struju vjetra njegovom snagom.

Najčešći tipovi su

- kontaktni senzori

- Hall - senzori magnetnog polja

- potenciometri rotacioni

Kontaktni senzori koriste otporničku mrežu koja spaja kontakt na odgovarajućem otporniku.

AD konverter dekodira napon u digitalni podatak a mikrokontroler stanica ga prevodi u podatak smjera vjetra.

Najčešće su preciznosti 1/8 kruga i informacija se prikazuje u obliku S,SI,I,JI,J,JZ,Z,SZ.

Izvedbe sa Hall senzorima koriste par senzora na prstenu sa razlikom položaja od 90 stupnjeva. Na taj način se dekodiranjem napona iz dva senzora može precizno odrediti položaj vjetrokaza.

Hall senzori se u koriste u projektu i dekodiranje položaja se vrši mjerenjem napona u rasponu od 1,44V do 3,6V.

Jedan napon se koristi za pozicioniranje +/- 180° dok drugim se računa precizni položaj. Jedan stupanj odgovara promjeni napona za 0,013V.

AD konverter dekodira napon u digitalni podatak a mikrokontroler stanice ga prevodi u podatak smjera vjetra izražen u stupnjevima.

U projektu se koristi Vjetrokaz sa Hall senzorima.

Anemometar

Brzina predstavlja prijeđeni put u jedinici vremena. Izražava se u m/s, čvorovima, Boforima....

Brzina vjetra se mjeri na dva načina:

- Mehanički

Preko senzora vjetrenjače koja se sastoji od tri polu lopte na osovini koja se vodoravno okreće. Brzina vjetra se mjeri preko mjerenja brzine rotacije vjetrenjače.

U projektu se koristi vjetrenjača koja pri svakoj punoj rotaciji šalje dva pravokutan impulsa.

- Ultrazvučno

Tri ili četiri para ultrazvučnih senzora raspoređenih po krugu. Brzina i smjer vjetera se izračunavaju iz razlike očitanih brzina zvuka na sensorima.

Za anemometar korišten u projektu nema dokumentacije. Pretpostavlja se da koristi magnetni senzor čiji izlaz pobuđuje okidački sklop koji na izlazu daje pravokutne impulse koji se broje da bi se dobila brzina.

Termometar

Temperatura se izražava u stupnjevima ljestvice. Kelinove, Ceilzusove, Farenheitove....

Senzori najčešće koriste pojavu da se kod nekih vodiča otpor mijenja sa temperaturom.

AD konverter dekodira napon na senzoru u digitalni podatak a mikrokontroler stanice ga prevodi u podatak o temperaturi.

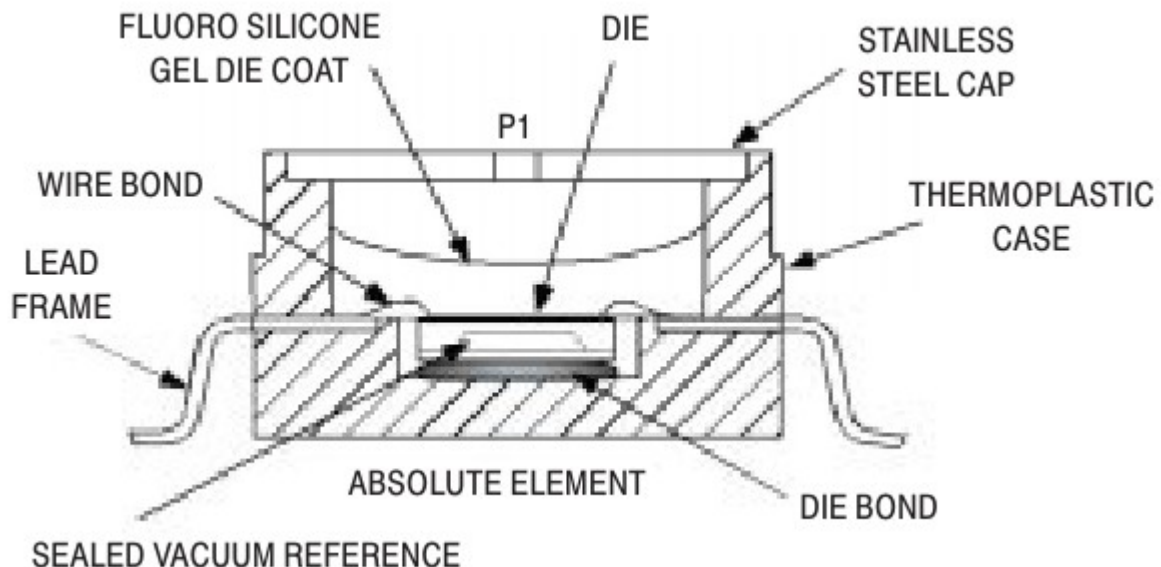
Korišteni su senzori DS18S20 i RHT-03 ali proizvođači u dokumentaciji ne navode točan tip samog senzora ugrađenog u čip.

Barometar

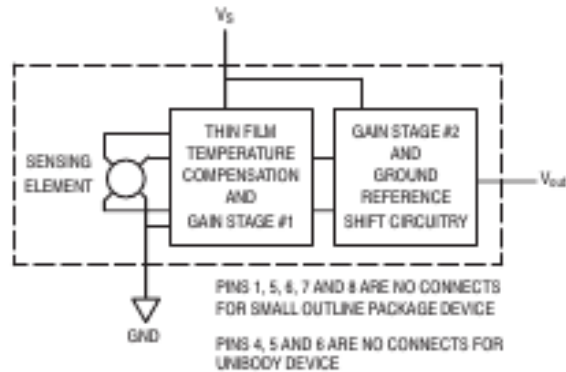
Tlak predstavlja sliju kojom s djeluje na neku površ. Izražava se u milimetrima živinog stupa, paskalima, barima, atmosferama....

Kod elektronskih meteoroloških postaja tlak se mjeri silicijskim sensorima i zasniva se na piezo efektu, uslijed kojeg se mijenja otpor u ovisnosti od sile koja djeluje na materijal. AD konverter dekodira napon na senzoru u digitalni podatak a mikrokontroler stanica ga prevodi u podatak o tlaku.

U projektu je korišten Motorola MPX4100 analogni integrirani senzor tlaka [4]. Za mjerenje tlaka se koristi piezo efekt na silicijskom kristalu.

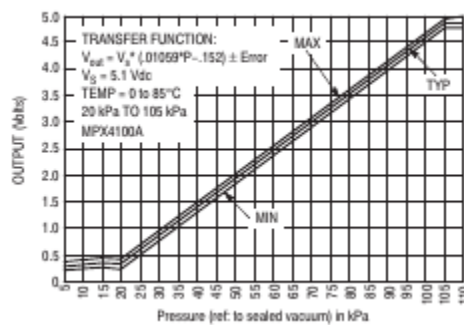


Slika 1, Presjek senzora MPX4100A [4].



Slika 2, Blok shema senzora [4].

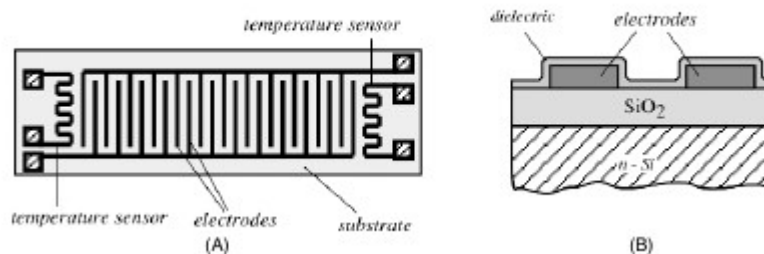
Pored senzorskog elementa u kućište su ugrađeni i elektronička kola za kompenzaciju temperature, pojačalo i kolo za referentno uzemljenje.



Dijagram 1, Izlazna funkcija MPX4100 je linearna za radni opseg od 20kPa do 105kPa [4].

Higromjer

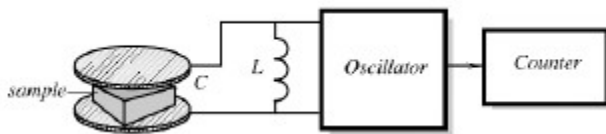
Količina vlage se izražava u relativnoj vlažnosti koja predstavlja postotak vlage u zraku od one koju zrak može sadržavati a da ne dođe do kondenzacije. Direktno je ovisna o temperaturi pa je uvijek u kombinaciji sa termometrom.



Slika 3, Prikaz jednostavnog senzora vlage [6].

Senzor se koristi efektom da se dielektrička konstanta materijala mijenja sa vlažnošću [6]. Sklop mjeri

kapacitet preko oscilatora sa poznatom vrijednosti induktivnog člana. Brojač učestalosti šalje vrijednost mikrokontroleru koristeći i podatak o temperaturi izračunava relativnu vlažnost zraka.



Slika 4 ,Prikaz principa očitavanja senzora vlage [6].

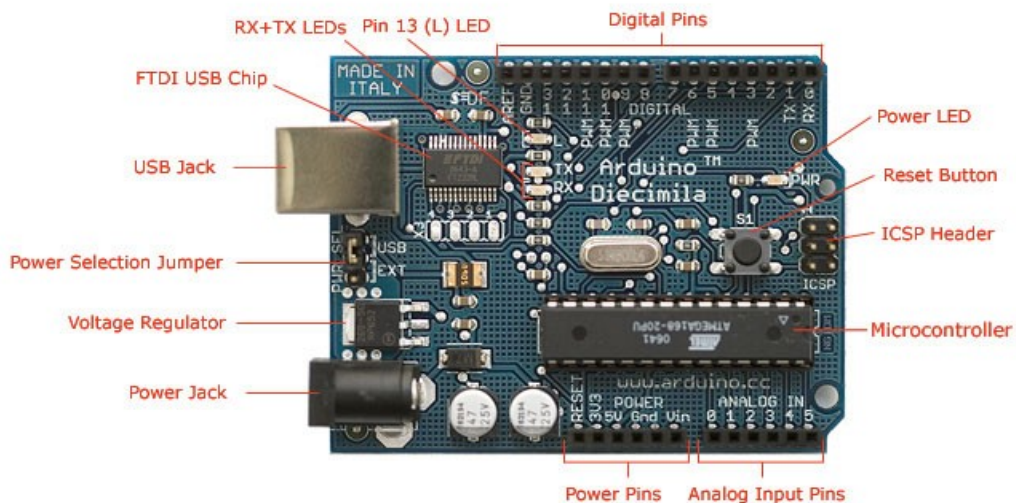
U projektu je korišten senzor tipa RHT03 u kojem je osjetni element je polimer kondenzator osjetljiv na vlažnost[5].

1.3 Arduino platforma

Arduino je mikrokontrolerska razvojna platforma otvorenog koda zasnovana na ATMEGA-mikrokontrolerima namijenjena učenju programiranja i povezivanja mikroracunala sa svijetom oko nas.

U projektiranju s vodilo računa da bude što jednostavnija za korištenje i savladavanje programiranja. Cjelokupan dizajn je u potpunosti otvoren što je omogućilo nisku cijenu i time veliku raširenost kako proizvođača koji koriste originalni dizajn tako i derivata prilagođenih određenim namjenama.

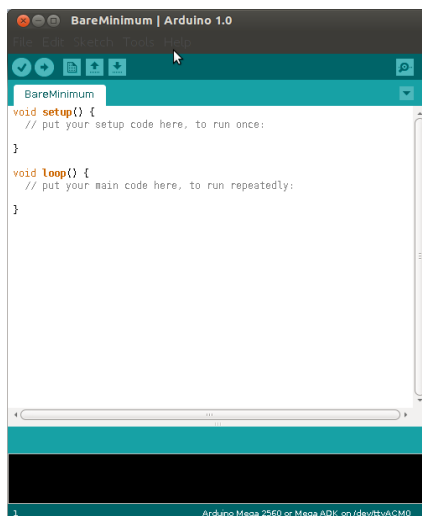
Pored same platforme sa mikrokontrolerom važan dio cjelokupne priče su i razne dodatne ploče za proširenja. Na primjer mrežu, upravljanje motorima, XBEE komunikaciju, BT, SD karticu...



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

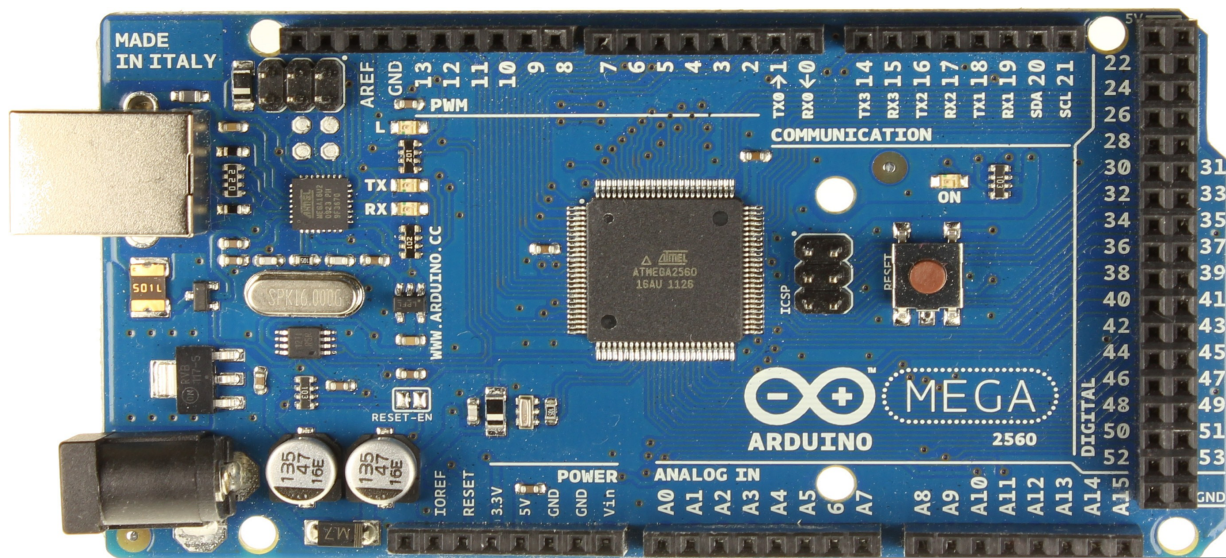
Slika 5, Arduino Diecimila [9].

Cjelokupna platforma sastoji se od razvojnog sučelja, programskog jezika, koji je prilagođena verzija jezika C i samog sklopa koje je u biti pločica sa mikrokontrolerom i minimalnim dodatnim dijelovima da bi se omogućila komunikacija sa PC-om i napajanje sklopa.



Slika 6, Arduino sučelje za programiranje.

2. Detaljan opis korištene Arduino Mega2560 platforme



Slika 7, ArduinoMega2560, prednja strana [7].

Arduino Mega2560 je mikrokontrolerska platforma bazirana na Atmel ATmega 2560 mikrokontroleru [7].

Pored mikrokontrolera sadržava potrebno okruženje za njegov rad, komunikaciju i programiranje sa PC-om putem USB sučelja preko IC ATmega16U2 .

Kompatibilna je sa velikim brojem dodatnih ploča Arduino Duemilanove ili Diecimila platformi.

Tehnički podaci [7]

Mikrokontroler	ATmega2560
Radni napon	5V
Napajanje	7-12V
Ulazni napon (limiti)	6-20V
Digitalni I/O pinovi	54 (14 sa PWM izlazom)
Analogni ulazi	16
Flash Memorija	256 KB / 8 KB za bootloader
SRAM	8 KB
EEPROM	4 KB
Takt	16 MHz

Napajanje

Mega2560 se može napajati preko USB porta ili vanjskog izvora. Odabir je automatski. Vanjsko napajanje se spaja na 2,1mm utičnicu ili na pinove. Preporučeno je napajanje od 7-12V. Može izdržati od 5 do 20 V ali ako je napajanje manje od 7V Arduino može postati nestabilan. Za napajanje od 12 do 20V može doći do pregrijavanja i kvara stabilizatora.

Memorija

Mikrokontroler ATmega2560 ima 256 KB flash memorije za spremanje programskog koda od čega je 8KB za bootloader, 8 KB SRAM i 4 KB EEPROM. EEPROM-u se pristupa preko EEPROM.h biblioteka.

Ulazi i Izlazi

Svaki od 54 digitalna pina može se koristiti kao ulaz i kao izlaz koristeći funkcije `pinMode()`, `digitalWrite()`, and `digitalRead()`. Oni rade na 5V i mogu dati/primiti 40 mA.

Neki pinovi imaju specijalne funkcije:

Serijska komunikacija: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).

Koriste se za primanje (RX) i slanje (TX) serijske komunikacije TTL razina.

Pin 0 i 1 su isto spojeni na odgovarajuće pinove ATmega16U2 USB-to-TTL serijskog kola.

Vanjski prekidi: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).

Ovi pinovi mogu biti podešeni na okidanje na nisku razinu, rastuću ili padajuću rub ili promjenu napona.

Za rad sa prekidima koristi se `attachInterrupt()` funkcija.

PWM: Pulse Width Modulation Pinovi 0 - 13. Daju 8-bit PWM pomoću `analogWrite()` funkcije.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ovi pinovi imaju podršku za SPI komunikaciju preko SPI biblioteka.

LED: 13. Mega ima ugrađeni pin za LED diodu na pinu 13. Kada je na HIGH vrijednosti LED dioda svijetli a na LOW je ugašena.

TWI: 20 (SDA) i 21 (SCL). Podrška ta TWI komunikaciju uz korištenje Wire biblioteke.

Analogni ulazi: Mega2560 ima 16 analognih ulaza koji se očitavaju 10 bitnom rezolucijom (moguće 1024 različite vrijednosti).

Zadano je mjerenje od 0 do 5V ali je moguće promijeniti gornji nivo preko pina AREF i funkcije `analogReference()`.

Reset. Treba na njega dati LOW razinu za reset. Tipično se koristi za dodatne ploče.

Komunikacija

Arduino Mega2560 posjeduje brojne resurse za komunikaciju sa vanjskim svijetom. Na primjer PC, drugi Arduino, neki drugi mikrokontroler

ATmega2560 posjeduje četiri sučelja za serijsku komunikaciju na TTL razinama.

USB kontroler ATmega16U2 daje virtualni serijski port preko USB komunikacije. Na taj način može se ostvariti jednostavna tekstualna komunikacija.

SoftwareSerial biblioteka omogućava serijsku komunikaciju preko bilo kojeg digitalnog pina. TWI komunikacija se radi preko Wire biblioteka a SPI preko SPI biblioteke.

Programiranje

Arduino Mega2560 se programira putem standardnog IDE sučelja. Kompajliranje se odvija po naredbi za prebacivanje koda i putem USB komunikacije se pohranjuje na mikrokontroler.

Moguće je i direktno programiranje putem ICSP (In-Circuit Serial Programming) zaglavlja.

Programski jezik je prilagođena inačica C-jezika. Time je omogućeno da se koristi velika količina razvijena programske logike uz minimalne izmjene.

Osnovna programska struktura je jednostavna,

```
void setup ()
{
    // podešavanje radne okoline programa
}
void loop ()
{
    // radni kod koji se odvija u beskonačnoj petlji
}
```

i sastoji se od podešavanja i beskonačne petlje u kojoj se odvija rad .

3. Pregled postojećih projekata koji koriste Arduino platformu za izradu meteorološke postaje

Linkovi na postojeće projekte slične tematike pronađeni na Internetu:

"Arduino readout of Davis wind sensor"

http://www.qsl.net/on7eq/projects/arduino_davis.htm

Projekt je ograničen samo na čitanje brzine i smjera vjetra i prikaz na LCD. Razlika je da se koristi senzor za pravac na principu potencijometra. Očitavanje brzine je slično.

"Arduino weather shield"

<http://www.zipfelmaus.com/blog/arduino-weather-shield-schematics-layout-code-everything-you-need/>

Web stranica opisuje izgradnju dosta kompleksne stanice napajanje solarnom energijom i bezžičnim slanjem podataka.

"Open source Weather Station"

<http://code.google.com/p/openws/wiki/Home?tm=6>

Projekt sličnih ciljeva. Razlika je u korištenim senzorima.

"How I made fully functional arduino weather station for 300\$"

<http://www.toptal.com/c/how-i-made-a-fully-functional-arduino-weather-station-for-300>

Meteo stanica sa objavom podatka na Internetu. Projekt koristi istu platformu Arduino Mega2560 i temperaturni senzor.

"ManyLabs:Weather Station"

<https://www.manylabs.org/project/weatherStation/>

Projekt koristi senzore koji se prodaju u kompletu za samogradnju i Arduino UNO platformu.

Prvenstvena namjena mu je edukativna, tako da nisu opisani praktični aspekti korištenja meteostanice.

4. Opis cjelokupnog projekta meteopostaje (dio kojeg je ovaj projekt)

Cilj ovog projekta je zamjena neadekvatne "gotove" meteorološke stanice WS2300. Realizacijom ovog projekta će se poboljšati preciznost i pouzdanost mjerenje smjera i brzine vjetera.

Meteorološka stanica je postavljena na polijetištu "Tribalj" iznad istoimenoga mjesta u Crikveničkom zaleđu.

Meteorološka stanica u cijelosti sastoji se od slijedećih dijelova:

- betonski temelj
- čelični cinčani stup visine 7m učvršćen čeličnim sajlama
- kućište od višeslojne plastike, toplinski izolirano za zimske hladnoće i reguliranim ventilacijskim otvorima za ljetne vrućine
- solarni kolektor snage 90W
- olovni akumulatori kapaciteta 150Ah
- regulator punjenja akumulatora i napajanja priključenih uređaja
- router (sa USB portom i WiFi AP), prerađen zamjenskim LINUX firmware-om
- 200MB WiFi link za komunikaciju prema Internetu
- web kamera

Projekt je realiziran u tvrtci "ELMAS" d.o.o..

Solarno napajanje je napravljeno od elemenata

- 90W ploče
- 12V-150Ah akumulatora
- kontrolera punjenja

Osigurava napajanje cjelokupnog sklopa u trajanju od tri dana bez dodatnog punjenja. Ploča i akumulatori su nabavljeni kao odbačena oprema.

Kontroler punjenja preko solarnih ćelija je napravljen za ovaj projekt u tvrtci "Elmas".

5. Opis tehničkog rješenja Arduino meteorološke stanice

Rješenje Arduino meteorološke stanice se temelji na korištenju kvalitetnog anemometra (namijenjenoga za nautičku upotrebu) i arduino Mega2560 platforme.

Točan proizvođač anemometra ne piše na njemu.

Rješenje će mjeriti slijedeće ulazne veličine:

- vjetar brzina i smjer

- temperatura

- vlaga

- tlak

U kasnijoj fazi i padavine.

Za njihovo mjerenje koristiti će senzore:

temperatura: DS18B20

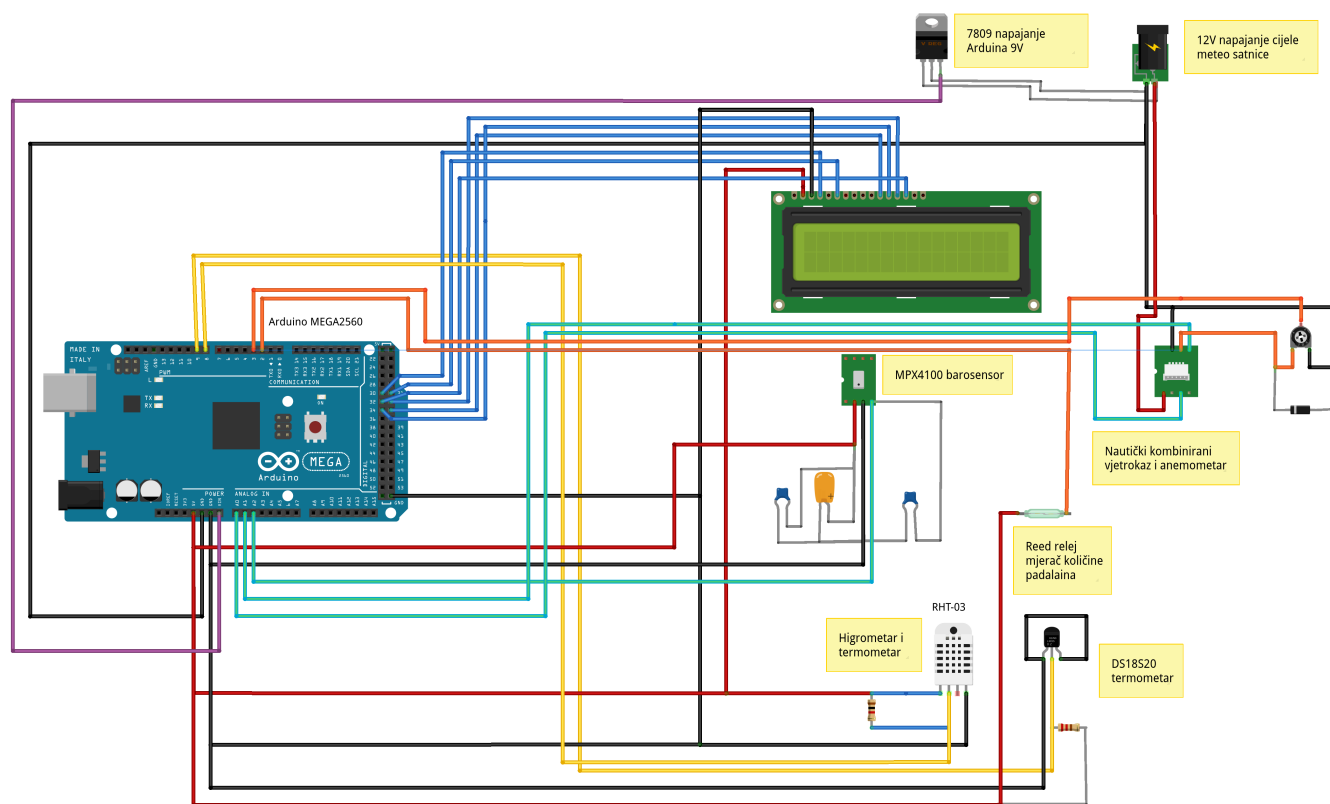
tlak: MPX4100A

vlažnost: RHT-03

padavine: mjerač od tvorničke postaje WS2300

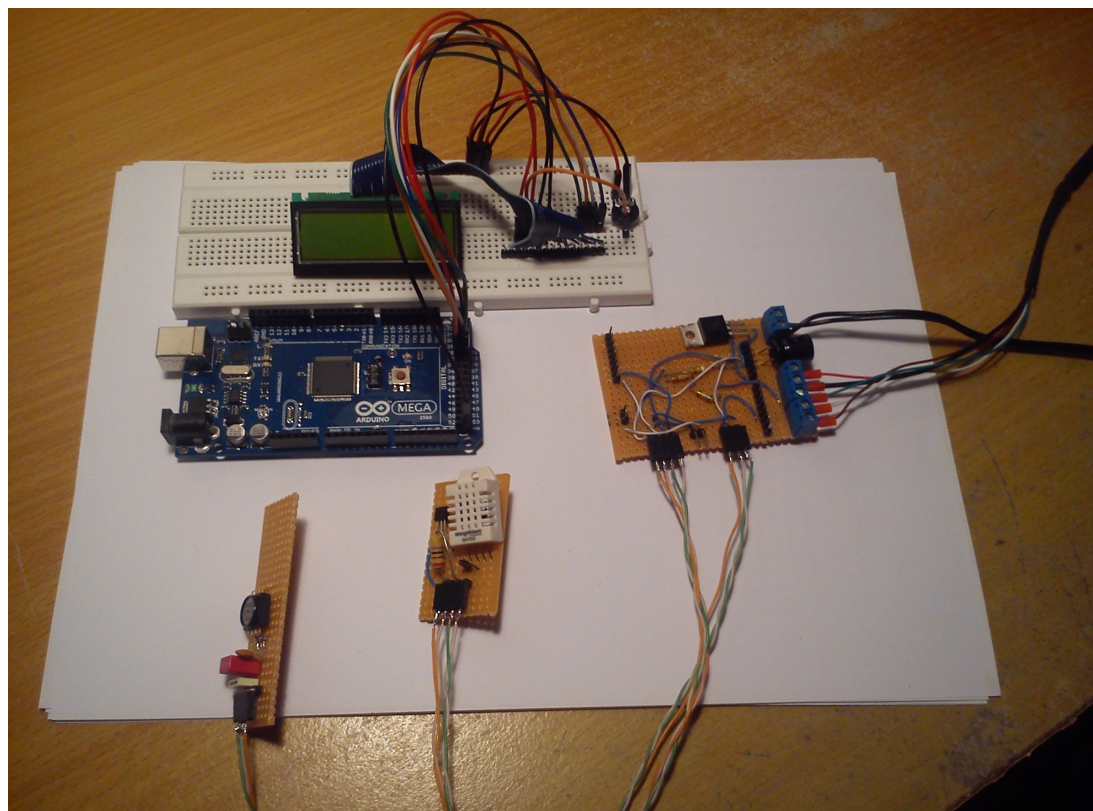
brzina vjetra: nautički, nepoznatog tipa i proizvođača

smjer vjetra: nautički, nepoznatog tipa i proizvođača

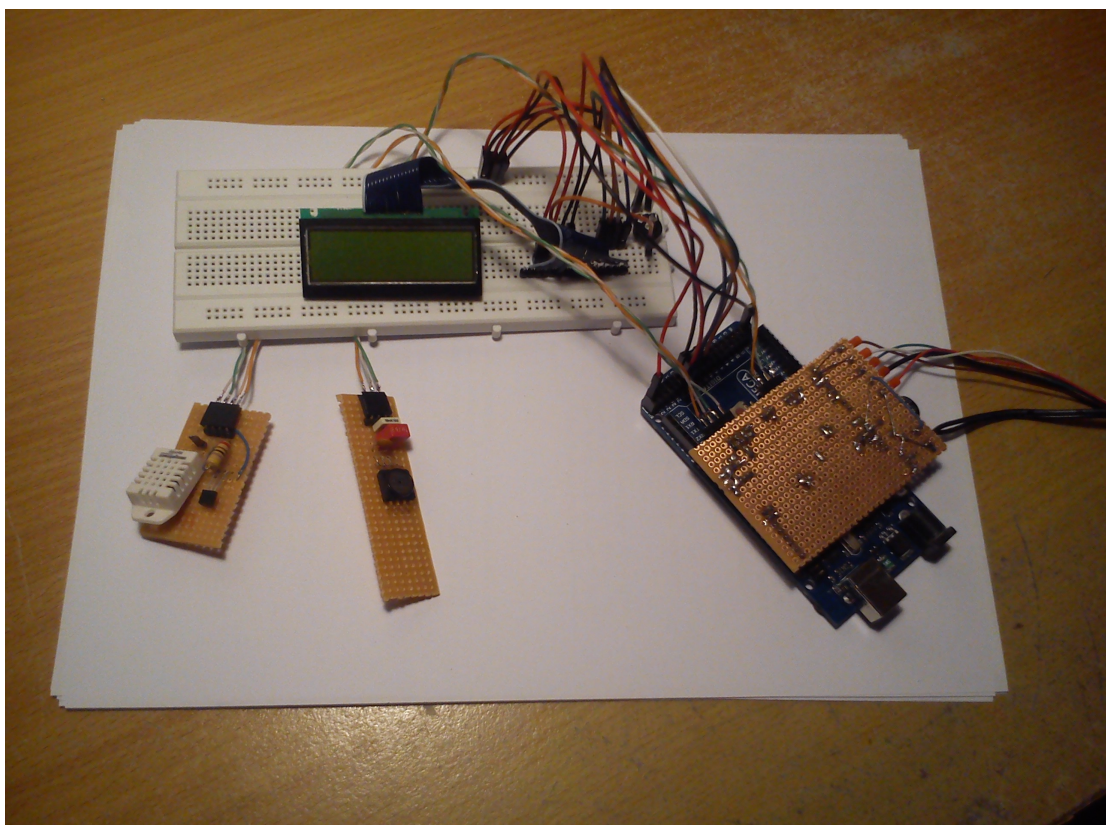


Made with Fritzing.org

Slika 9, Konstrukcija Arduino meteo stanice sa elementima izrađena u programu Fritzing



Slika 10, Elementi konstrukcije.



Slika 11 , Pločica postavljena na Arduino.

Konstrukcija sa dodatnim elementima je napravljena kao Arduino nadgradnja, tj pločica koja se priključi na Arduino. Opremljena je stezaljkama i konektorima za spajanje, elektronikom za regulaciju napajanja za

Arduino i vjetrokaza i dijelom elektronike za rad senzora. Spajanje elementa je lemljenjem.

Korišteni su samo pinovi koji su kompatibilni sa drugim Arduino kontrolerima.

Nadgradnja je isprobana. Utvrđeno je da je konstrukcija ispravna za napajanje i rad anemometra.

Za očitavanje 12V impulsa anemometra na Arduino DI napon je spušten djeliteljem napona sa 12V na 4,7V.

Izlaz iz mikrokontrolera je serijski formatiran prema specifikaciji za datoteke tipa easyweather.dat I prilagođeno za prihvatanje podataka u programu "Cumulus".

Komunikacija će se odvijati preko "serial over IP" pretvornika koji serijsku komunikaciju šalje preko IP komunikacije.

Informacije će se objavljivati na Internetu preko web stranica udruga "Istra Meteo" i PJK "Flumen".

Veličine koje će se moći vidjeti su smjer i brzina vjetrova, temperatura, vlaga, tlak i oborine sa trenutnim i ekstremnim vrijednostima.

Proces rada automatske stanice se sastoji od tri faze

I. - očitavanje električnih vrijednosti senzora ili primanje očitanih vrijednosti

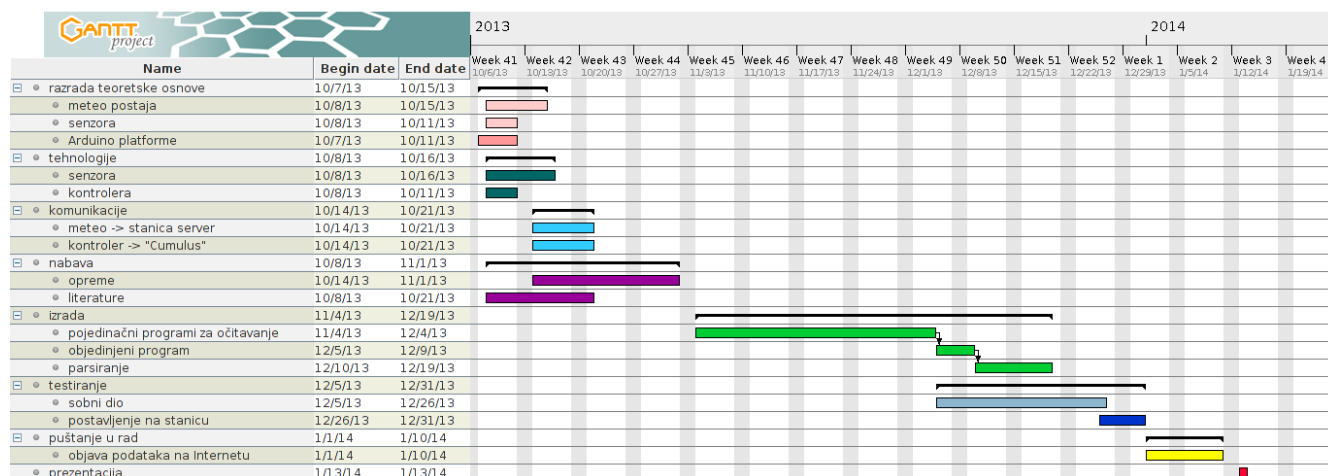
II. - izračun željenih veličina

III. - slanje rezultata

I. faza se u slučajevima temperature i vlage događa na samom senzoru, jer su oni digitalni te posjeduju sklopove za očitavanje i slanje podataka.

Druge dvije faze se odvijaju u beskonačnoj petlji glavnog programa na Arduinu.

6. Terminski plan izrade projekta



Slika 12, Terminski plan izrade projekta prikazan "Gantt" dijagramom

Plan izrade projekta sastoji se od sljedećih etapa :

razrada teoretske osnove

razrada teoretske osnove meteo postaja

razrada teoretske osnove senzora

razrada teoretske osnove Arduino platforme

razrada tehnologije

razrada tehnologije senzora

razrada tehnologije kontrolera

razvoj komunikacije

razvoj komunikacije meteo -> stanica server

razvoj komunikacije kontroler -> "Cumulus"

nabava

nabava opreme

nabava literature

izrada

izrada pojedinačnih Arduino programa za očitavanje senzora

slaganje pojedinačnih Arduino programa za očitavanje senzora

programiranje parsiranja za komunikaciju

testiranje

testiranje u "laboratoriju"

postavljenje na stanicu i testiranje

izvedba potrebnih korekcija

puštanje u rad

objava podataka na Internetu

prezentacija

Termini faza izrade projekta:

Naziv	Početak	Završetak	Trajanje
razrada teoretske osnove	07.10.2013	16.10.2013	7
meteo postaja	08.10.2013	16.10.2013	6
senzora	08.10.2013	12.10.2013	4
Arduino platforme	07.10.2013	12.10.2013	5
tehnologije	08.10.2013	17.10.2013	7
senzora	08.10.2013	17.10.2013	7
kontrolera	08.10.2013	12.10.2013	4
kommunikacije	14.10.2013	22.10.2013	6
meteo -> stanica server	14.10.2013	22.10.2013	6
kontroler -> "Cumulus	14.10.2013	22.10.2013	6
nabava	08.10.2013	02.11.2013	19
opreme	14.10.2013	02.11.2013	15
literature	08.10.2013	22.10.2013	10
izrada	04.11.2013	20.12.2013	34
pojedinačni programi za očitavanje	04.11.2013	05.12.2013	23
objedinjeni program	05.12.2013	10.12.2013	3
parsiranje	10.12.2013	20.12.2013	8
testiranje	05.12.2013	01.01.2014	19
sobni dio	05.12.2013	27.12.2013	16
postavljenje na stanicu	26.12.2013	01.01.2014	4
puštanje u rad	01.01.2014	11.01.2014	8
objava podataka na Internetu	01.01.2014	11.01.2014	8
prezentacija	13.01.2014	14.01.2014	1

7. Opis izrade programa za dohvaćanje podataka sa senzora i ispis koda

Izrada programa se odvijala u etapama. Etape su bile izrada programa po pojedinom senzoru.

Po završetku izrade samostalnog programa za pojedini senzor njegovo rješenje se ugrađivalo u glavni program i onda isprobavalo u skupnom radu.

Za kontrolu bez upotrebe računala korišten je 16x2 text LCD ekran na kome su se ispisivale izmjerene vrijednosti.

Anemometar

Dobivanje vrijednosti sa senzora

Za osnovu očitavanja impulsa anemometra je korišten postojeći kod za "DAVIS Vantage Pro & Vantage Pro 2 " anemometar [1]. Zasniva se na korištenju hardverskih prekida. Njihovim korištenjem se osigurava da brojanje događaja ne bude ometano nekim drugim procesom. Kod je procesom izrade smanjen na mjeru dovoljnu za rad projekta.

Očitavanje se dobiva linijom "attachInterrupt(1, ISR_anemoMetar, RISING)" koja poziva funkciju koja povećava vrijednost za 1. Ta se vrijednost očitava svake sekunde (1000 ms) korištenjem funkcije millis() za precizno pokretanje.

Dobivanje fizičke veličine

Broj dobiven očitavanjem predstavlja broj impulsa koje je anemometar poslao u jednoj sekundi. Šalje se po jedan impuls na pola okreta. Kako za korišteni senzor nema dokumentacije morati će se napraviti umjeravanje pomoću drugog anemometra. Pri tome će se na rezultat staviti neki faktor koji će iz broja impulsa dati brzinu vjetrova.

Kod programa za očitavanje anemometra:

```
// anemometar
#include <avr/pgmspace.h>
#include <avr/io.h>
#include <avr/interrupt.h>
int anemoMetarPIN = 3;
int anemoMetarVrijednost = 0;
setup()
//anemoMetar
pinMode(anemoMetarPIN, INPUT);
attachInterrupt(1, ISR_anemoMetar, RISING); //inicijacija funkcije za hardverski brojac
brzinaIndex=0;
loop()
// - - - anemoMetar - - -
//static variables for wind sensor
static int anemoMetarVrijednost_OBRT;
static long cekajDo;
if (millis() >= cekajDo) {
    anemoMetarVrijednost_OBRT=anemoMetarVrijednost;
    //reset the value
    anemoMetarVrijednost = 0;
    cekajDo = millis() + 1000;    // Make sure we wait for 1 minute.
}
if (brzinaIndex < sizeof(brzina)){
    brzinaIndex++;}
else{
    brzinaIndex=0;
}
```

```
void ISR_anemoMetar()
{ anmeoMetarVrijednost++;}
```

Vjetrokaz

Dobivanje vrijednosti sa senzora

Vjetrokaz se sastoji od strelice koja je na osovini. Na istoj osovini su dva magnetna prstena nad kojima su dva Hall senzora. Senzori daju napone koji su zakrenuti u fazi za 90°. Naponi su dovedeni direktno na analogne ulaze A6 i A7 sa kojih se očitava funkcijom analogRead() .

Napon izlaza se kreće od 1,4 do 3.6 V, i time se dobiva brojna vrijednost na AI od oko 744 za maksimum i oko 280 za minimum.

Dobivanje fizičke veličine

Tražena fizička veličina je smjer vjetra kojeg pokazuje vjetrokaz. Očitavanjem analognih ulaza dobivamo vrijednosti u ovisnosti od položaja osovine. Položaj je kodiran razlikom u napomnima. Napon na A6 ima maksimum na 180° (ako uzmemo da će nosač vjetrokaz biti usmjeren na jug(180°)) dok je na A7 maksimum na 270°.

Po uključenju potrebno je kalibrirati sustav okretanjem vjetrokaza za par krugova. Od izmjerenih vrijednosti uzimaju se ekstremne vrijednosti. Zatim se računa srednja vrijednost po kanalu i dijeli raspon na 180 da bi se dobila vrijednost jednog stupnja.

U ovisnosti o vrijednosti oba napona određuje se kvadrant smjera vjetra i prema njemu formula kojom se računa:

```
// izracun smjera vjetrokaza prema razlici u ocitanju senzora
if (a>a_nul && b>b_nul) b_k=(180-(aMax-a)/stupanjA);
if (a<a_nul && b>b_nul) b_k=(180-(aMax-a)/stupanjA);
if (a>a_nul && b<b_nul) b_k=(180+(aMax-a)/stupanjA);
if (a<a_nul && b<b_nul) b_k=(180+(aMax-a)/stupanjA);
```

Nakon dobivanja smjera u stupnjevima ona se konvertira u slovni prikaz koji se šalje za prikaz na webu.

```
/ prikaz vrijednosti smjera u slovnom obliku
if ( 337 > smjer && smjer < 22 )char smjerTxt[] = "N";
if ( 21 > smjer && smjer < 78 )char smjerTxt[] = "NE";
if ( 77 > smjer && smjer < 112 )char smjerTxt[] = "E";
if ( 113 > smjer && smjer < 158 )char smjerTxt[] = "SE";
if ( 157 > smjer && smjer < 203 )char smjerTxt[] = "S";
if ( 202 > smjer && smjer < 247 )char smjerTxt[] = "SW";
if ( 246 > smjer && smjer < 293 )char smjerTxt[] = "W";
if ( 292 > smjer && smjer < 338 )char smjerTxt[] = "NW";
```

Kod programa za očitavanje vjetrokaza:

```
//smjer

int analogInA = A6; // Analog input pin a hall izlaz
int analogInB = A7; // Analog input pin b hall izlaz
int a, b, smjer, aMax, aMin, bMax, bMin;
float a_nul,b_nul, b_k, stupanjA, stupanjB;
char smjerTxt[3];

//citanje hall senzora
a=analogRead(A6);
```

```

b=analogRead(A7);
//kalibracija max/min vrijednosti
if (a>aMax) aMax=a;
if (b>bMax) bMax=b;
if (a<aMin) aMin=a;
if (b<bMin) bMin=b;
//odredjivanje srednjih vrijednosti
a_nul = aMax-(aMax-aMin)/2;
b_nul = bMax-(bMax-bMin)/2;
//izracun vrijednosti kuta za max-min vrijednosti na AI, 180 stupnjeva jer pojedni senzor ne pokazuje
puni krug
stupanjA=float(aMax-aMin)/180;
// izracun smjera vjetrokaza prema razlici u ocitanju senzora
if (a>a_nul && b>b_nul) b_k=(180-(aMax-a)/stupanjA);
if (a<a_nul && b>b_nul) b_k=(180-(aMax-a)/stupanjA);
if (a>a_nul && b<b_nul) b_k=(180+(aMax-a)/stupanjA);
if (a<a_nul && b<b_nul) b_k=(180+(aMax-a)/stupanjA);
smjer=int(b_k);
// prikaz vrijednosti smjera u slovnom obliku
if ( 337 > smjer && smjer < 22 )char smjerTxt[] = "N";
if ( 21 > smjer && smjer < 78 )char smjerTxt[] = "NE";
if ( 77 > smjer && smjer < 112 )char smjerTxt[] = "E";
if ( 113 > smjer && smjer < 158 )char smjerTxt[] = "SE";
if ( 157 > smjer && smjer < 203 )char smjerTxt[] = "S";
if ( 202 > smjer && smjer < 247 )char smjerTxt[] = "SW";
if ( 246 > smjer && smjer < 293 )char smjerTxt[] = "W";
if ( 292 > smjer && smjer < 338 )char smjerTxt[] = "NW";

```

Barometar.

Dobivanje vrijednosti sa senzora

Barometar je analogni senzor. Njegov izlaz je spojen direktno na analogni A5 sa kojih se očitava funkcijom analogRead() .

Dobivanje fizičke veličine

Prijenosna karakteristika baro-senzora MPX4100A izražena je jednadžbom.

$$V_{out} = V_s(P \times 0.01059 - 0.1518)$$

Kako je napon napajana 5V izraz je:

$$V_{out} = 5(P \times 0.01059 - 0.1518)$$

Rezolucija AI je 10bita što daje 1024 vrijednost. Kada 5V podjelimo sa 1024 dobivamo da je osjetljivost 0,004883 V.

Razlika napona V_{out} za 0,1kPa = 0,005295 V što nam daje da je digitalna rezolucija +/- 0.1 kPa.

Za 1013hPa, $V_{out}=5(101.3 \times 0.01059 - 0.1518)=4,604835$ što nam daje da bi za taj tlak trebali dobiti na Arduino AI očitavanje broja 943.

Kod programa za očitavanje barometra:

```

// barometar MPX4100

int baroSenzor = A5; // odabir pina za o?itanje senzora Motorola MPX4100A
int tlakIz = 0;      // varijabla za izracunati tlak

loop

```

```

tlakIz = int(tlak(analogRead(baroSenzor))); //poziv funkcije

float tlak(float senzorTlak){
    float tlakIzracun;
    //senzorTlak=(analogRead(baroSenzor));
    //senzorTlak=943;
    tlakIzracun = (((senzorTlak/1024)+0.1518)/0.01059)*10;
    return tlakIzracun;
}

```

Temperatura.

Temperatura se registrira pomoću čipa DS18S20. Čip komunicira "1Wire" protokolom te se za njegovo očitavanje koriste gotove biblioteke za taj protokol. Uz samu biblioteku dolazi primjer u kojem je obrađena komunikacija sa korištenim čipom te je ona korištena uz male izmjene.

Dobivanje fizičke veličine.

Prema proizvođačkoj specifikaciji izlaz je 9 bitne rezolucije. To daje rezoluciju od 0.5°C pri čitanju vrijednosti.

Kod programa za očitavanje temperature:

```

temperatura = temperaturaDS(); //poziv funkcije
/ tremometar DS1820
float temperaturaDS(){
    byte i;
    byte present = 0;
    byte type_s;
    byte data[12];
    byte addr[8];
    float tempDS;

    if ( !ds.search(addr)) {
        ds.reset_search();
        delay(250);
    }
    if (OneWire::crc8(addr, 7) != addr[7]) {
        Serial.println("CRC nije dobar!");
    }
    // ROM byte indicira tip senzora
    switch (addr[0]) {
        case 0x10:
            type_s = 1;
            break;
        case 0x28:
            type_s = 0;
            break;
        case 0x22:
            type_s = 0;
            break;
    }
    ds.reset();
    ds.select(addr);
    ds.write(0x44, 1);
    delay(1000);
    present = ds.reset();
    ds.select(addr);
    ds.write(0xBE);
    // citanje podataka
    for ( i = 0; i < 9; i++) {
        data[i] = ds.read();
    }
}

```



```

}
// konverzija binarnog podatka u temperaturu
int16_t raw = (data[1] << 8) | data[0];
if (type_s) {
    raw = raw << 3;
    if (data[7] == 0x10) {
        raw = (raw & 0xFFF0) + 12 - data[6];
    }
} else {
    byte cfg = (data[4] & 0x60);
    if (cfg == 0x00) raw = raw & ~7;
    else if (cfg == 0x20) raw = raw & ~3;
    else if (cfg == 0x40) raw = raw & ~1;
}
tempDS = (float)raw / 16.0;
return tempDS;
}

```

Vlaga i temperatura

Korišten je senzor vlage je RHT-03. To je kombinirani senzor vlage i temperature.

Korišten je pokazni kod koji dolazi uz biblioteku namijenjenu za čitanje vrijednosti, izmijenjen za potrebe projekta.

Kod programa za očitavanje vlažnosti:

```

#define DHT22_PIN 9

// podešavanje DHT22 instance
DHT22 senzorRT03(DHT22_PIN);
DHT22_ERROR_t kodGreske;

loop
    delay(2000);
    kodGreske = senzorRT03.readData();
    switch(kodGreske)
    {
        case DHT_ERROR_NONE:
            Serial.print("podaci: ");
            Serial.print(senzorRT03.getTemperatureC());
            Serial.print("C ");
            Serial.print(senzorRT03.getHumidity());
            Serial.println("%");
            lcd.setCursor(0, 0);
            lcd.print(senzorRT03.getTemperatureC());
            lcd.print("C");
            lcd.setCursor(0, 1);
            lcd.print(senzorRT03.getHumidity());
            lcd.print("%");
            break;
        case DHT_ERROR_CHECKSUM:
            Serial.print("check sum error ");
            Serial.print(senzorRT03.getTemperatureC());
            Serial.print("C ");
            Serial.print(senzorRT03.getHumidity());
            Serial.println("%");
            break;
    }

```

Opći dio programa

U programu se pojedini procesi pokreću ciklično.

Očitavanje senzora temperature :

```
// Svake sekunde:
if (millis() > vrijeme+1000){
  // - - - temperatura - - -
  temperatura = temperaturaDS(); //poziv funkcije
  // - - - kraj temperatura - - -

  brzina[brzinaIndex]=anemoMetarVrijednost_OBRT; // upis vrijednosti brzine u polje sa 600 vrijednosti
  radi dobivanja prosjeka i max udara

  lcd.clear();
  vrijeme = millis();
}
```

Izračun srednjih vrijednosti brzine vjetra:

Srednja brzina vjetra se, prema standardu WMO[2], računa na period od deset minuta a udar u trajanju od tri sekunde. Proces sakuplja i sprema u niz vrijednosti brzine svake sekunde. U memoriji se čuva 600 vrijednosti i svakih trideset sekundi se računa srednja vrijednost i najjači udar od tri sekunde.

```
// svakih 30 sec : izracuni srednjih i max vrijednosti vjetra i slanje svih podataka
if (millis() > vrijeme1+30000){
  vrijeme1 = millis();
  int i, brZb;
  for (i=0; i<= sizeof(brzina); i++){
    brzinaUdar=max(brzinaUdar,(brzina[i] + brzina[i-1] + brzina[i-2])/3);
    brZb=brZb+brzina[i];
  }
  brzinaSred=brZb/600;
```

Slanje podataka:

Podaci se šalju u easyweather.dat formatu [3] svakih trideset sekundi u ruter preko serijskog porta 1.

To je primarna funkcija. Pored toga se podaci šalju i na LCD i USB serijski port radi kontrole rada i prikupljenih podataka.

```
// slanje podataka u easyweather.dat
// izvj,0,0,2013-00-00,12:00:00
AM,0,0,0,0,temp,0,0,0,tlak,brzinaSred,0,brzinaUdar,0,0,smjerTxt,0,0,0,0,0,0,0
izvj++;
Serial.println();
Serial.print(izvj,DEC);
Serial.print(",0,0,2013-00-00,12:00:00 AM,0,0,0,0,");
Serial.print(temperatura);
Serial.print(",0,0,0,");
Serial.print(tlakIz, 1);
Serial.print(",");
Serial.print(brzinaSred,1);
Serial.print(",");
Serial.print(brzinaUdar);
Serial.print(",0,0,");
Serial.print(smjerTxt);
Serial.print(",0,0,0,0,0,0,0,0");

// slanje na serijski port Serial1
Serial1.println();
Serial1.print(izvj,DEC);
Serial1.print(",0,0,2013-00-00,12:00:00 AM,0,0,0,0,");
```

```

    Serial1.print(temperatura);
    Serial1.print(",0,0,0,");
    Serial1.print(tlakIz, 1);
    Serial1.print(",");
    Serial1.print(brzinaSred,1);
    Serial1.print(",");
    Serial1.print(brzinaUdar);
    Serial1.print(",0,0,");
    Serial1.print(smjerTxt);
    Serial1.print(",0,0,0,0,0,0,0,0");

// - - - USB serial KONTROLNI IZLAZI za potrebe odrzavanja - - -

    Serial.println("\n-----");

    //kontrolni izlaz anemo
    Serial.print("w: ");
    Serial.print(anemoMetarVrijednost_OBRT, DEC);
    Serial.print("; ");

    //kontrolni izlaz smjera
    Serial.print("\tSmjer=");
    Serial.print(smjer);

    //kontrolni baro
    Serial.print("\t P = ");
    Serial.print(tlakIz);
    Serial.print(" hPa");
    Serial.print("\t");
    Serial.print(analogRead(baroSenzor));

    //kontrolni temp
    Serial.print("\t T= ");
    Serial.print(temperatura);
    Serial.print(" C");

// - - - LCD KONTROLNI IZLAZI za potrebe odrzavanja - - -

    // smjer
    lcd.setCursor(0,1);
    lcd.print("s=");
    lcd.print(smjer);
    lcd.print(" ");
    // brzina
    lcd.setCursor(0,0);
    lcd.print("w=");
    lcd.setCursor(3,0);
    lcd.print(anemoMetarVrijednost_OBRT, DEC);
    // baro
    lcd.setCursor(7,0);
    lcd.print("p=");
    lcd.print(tlakIz);
    lcd.print("hPa");
    //temp
    lcd.setCursor(7, 1);
    lcd.print("t=");
    lcd.print(temperatura);
    lcd.print("C");

```

8. Opis serverske aplikacije za prihvrat podataka

Prihvrat podataka vrši aplikacija "Open2300". Detaljna objašnjenja za rad i preuzimanje su na ovoj Internet poveznici: <http://www.lavrsen.dk/foswiki/bin/view/Open2300/WebHome>

Aplikacija je instalirana na mrežni ruter kojem je promijenjen operativni sustav na "OpenWRT". To je Linux prilagođen radu na ruterima koji najčešće služe kućnoj upotrebi. Detaljna objašnjenja za rad i preuzimanje su na ovoj Internet poveznici: <http://wiki.openwrt.org/toh/start>

Aplikacija komunicira sa meteorološkom stanicom preko serijske veze, koja je na ruteru izvučena sa konektora matične ploče i odvija se na TTL nivoima.

Aplikacija podatke prima i zapisuje u datoteku a može i direktno u SQL bazu.

Sama aplikacija koristi isključivo komandnu liniju i nije namijenjena normalnom korištenju za očitavanje podataka nego za distribuciju podataka u ovom slučaju na web stranice "IstraMet".

9. Analiza rada u usporedbi s planiranim terminima i zamišljenim rješenjem

Rad na izgradnji meteorološke stanice protekao je u okvirima planiranih termina. Konkretni rad na etapama je bio uvjetovan drugima obavezama, što se kompeziralo većim angažmanom u slobodnim vremenskim razdobljima.

Krajnji rezultat u većem dijelu odgovara zamišljenom rješenju. Glavni cilj je bio kvalitetnije mjerenje vjetra što je i dobiveno korištenjem nautičkog senzora.

Od zadanih ciljeva nije se uspjelo postaviti stanicu u "živi" rad na lokaciji polijetišta "Tribalj" i pokrenuti objavu podataka na meteo-portalu "IstraMet". Pored lošeg vremena razlog je potreba angažiranja djelatnika i tehničke opreme tvrtke "Elmas" što se nije moglo napraviti.

Time je i razvoj očitavanja količine padavina izostavljen jer je to predviđeno korištenjem postojećeg senzora koji je monitoran na postojećoj meteo stanici, stup visine 5m na nepristupačnom terenu.

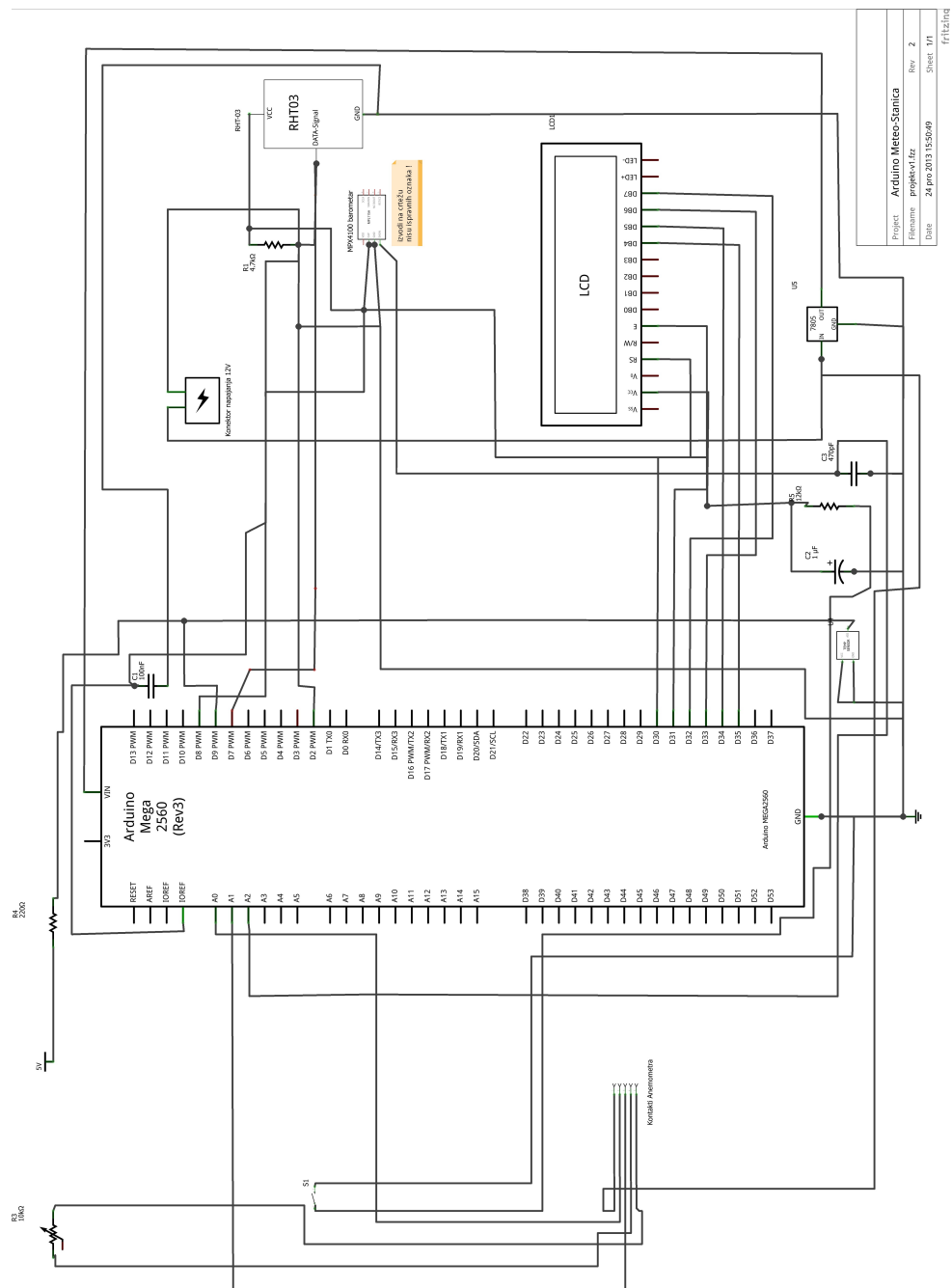
10. Troškovnik izrade projekta

Troškovnik elemenata za izradu Arduino meteo stanice:

<i>Naziv komponente</i>	<i>tip-oznaka</i>	<i>komada</i>	<i>cijena-kom</i>	<i>ukupno</i>
Nautički senzor vjetra	nepoznato	1	1.051,00 kn	1.051,00 kn
Barosenzor	MPX4100	1	220,00 kn	220,00 kn
Higrosenzor	RHT-03	1	75,00 kn	75,00 kn
Termosenzor	DS18S20	1	34,00 kn	34,00 kn
Otpornik	4.7kOhm	2	1,00 kn	2,00 kn
Otpornik	12kOhm	1	1,00 kn	1,00 kn
Potenciometar	10kOhm	1	5,00 kn	5,00 kn
Kondenzator	1uF	1	1,30 kn	1,30 kn
Kondenzator	0.1uF	2	1,00 kn	2,00 kn
Stabilizator 9V	TS7809	1	8,00 kn	8,00 kn
Pločica bušena proto pertinaks	-	1	10,00 kn	10,00 kn
Arduino Mega	Mega2560	1	305,60 kn	305,60 kn
Konektor slip muški	-	1	2,50 kn	2,50 kn
Konektor slip ženski	-	1	5,60 kn	5,60 kn
Konektor slip kutni muški	-	1	3,00 kn	3,00 kn
			Ukupni trošak	1.726,00 kn

11. Električna shema projekta

Električna shema izrađena je u programu Fritzing:



Project	Arduino Mero-Sanica
File name	projekat.fzz
Date	24. pros 2013 15:52:49
Sheet	1/1
Rev	2

Fritzing

12. Ispis cjelokupnog koda programa

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(30, 31, 35, 34, 33, 32);

// anemo
#include <avr/pgmspace.h>
#include <avr/io.h>
#include <avr/interrupt.h>
int anmeoMetarPIN = 3;
int anmeoMetarVrijednost = 0;
//smjer

int analogInA = A6; // Analog input pin a hall izlaz
int analogInB = A7; // Analog input pin b hall izlaz
int a, b, smjer, aMax, aMin, bMax, bMin;
float a_nul,b_nul, b_k, stupanjA, stupanjB;
char smjerTxt[3];

// baro

int baroSenzor = A5; // odabir pina za očitavanje senzora Motorola MPX4100A
int tlakIz = 0; // varijabla za izračunati tlak

// - - - tremometar DS1820
#include <OneWire.h>
OneWire ds(8); // pin 8 , 4.7K otpornik između D i Vcc
float temperatura;

// slanje podataka u easyweather.dat
unsigned long izvj=0;//broj poslanog izvjestaja
int brzina[600], brzinaIndex,brzinaSred,brzinaUdar; //niz za spremanje vrijednosti brzina,

// cekalica
unsigned long vrijeme, vrijeme1, vrijeme2;

void setup(){
  lcd.begin(16, 2);
  Serial.begin(9600);
  // smjer
  aMax=600;
  aMin=400;
  bMax=600;
  bMin=400;

  //anemoMetar
  pinMode(anmeoMetarPIN, INPUT);
  attachInterrupt(1, ISR_anemoMetar, RISING);
  brzinaIndex=0;
}

void loop()
{
  // - - - anemoMetar - - -
  //static variables for wind sensor
  static int anmeoMetarVrijednost_OBRT;
  static long cekajDo;

  if (millis() >= cekajDo) {
    anmeoMetarVrijednost_OBRT=anmeoMetarVrijednost;
    //reset the value
    anmeoMetarVrijednost = 0;
    cekajDo = millis() + 1000; // Make sure we wait for 1 minute.
  }
  if (brzinaIndex < sizeof(brzina)){
    brzinaIndex++;}
  else{
    brzinaIndex=0;
  }

  // - - - kraj anemoMetar - - -

  //- - - smjer - -

  //citanje hall senzora
  a=analogRead(A6);
```

```

b=analogRead(A7);
//kalibracija max/min vrijednosti
if (a>aMax) aMax=a;
if (b>bMax) bMax=b;
if (a<aMin) aMin=a;
if (b<bMin) bMin=b;
//određivanje srednjih
a_nul = aMax-(aMax-aMin)/2;
b_nul = bMax-(bMax-bMin)/2;
//izračun vrijednosti kuta za nivo na AI
stupanjA=float(aMax-aMin)/180;
//podešavanje izračuna smjera prema razlici u očitavanju senzora
if (a>a_nul && b>b_nul) b_k=(180-(aMax-a)/stupanjA);
if (a<a_nul && b>b_nul) b_k=(180-(aMax-a)/stupanjA);
if (a>a_nul && b<b_nul) b_k=(180+(aMax-a)/stupanjA);
if (a<a_nul && b<b_nul) b_k=(180+(aMax-a)/stupanjA);

smjer=int(b_k);
if ( 337 > smjer && smjer < 22 )char smjerTxt[] = "N";
if ( 21 > smjer && smjer < 78 )char smjerTxt[] = "NE";
if ( 77 > smjer && smjer < 112 )char smjerTxt[] = "E";
if ( 113 > smjer && smjer < 158 )char smjerTxt[] = "SE";
if ( 157 > smjer && smjer < 203 )char smjerTxt[] = "S";
if ( 202 > smjer && smjer < 247 )char smjerTxt[] = "SW";
if ( 246 > smjer && smjer < 293 )char smjerTxt[] = "W";
if ( 292 > smjer && smjer < 338 )char smjerTxt[] = "NW";

// - - - kraj smjer - - -

// - - - baro - - -

tlakIz = int(tlak(analogRead(baroSenzor)));

// - - - kraj baro - - -

// Svake sekunde:
if (millis() > vrijeme+1000){
// - - - temperatura - - -
temperatura = temperaturaDS();
// - - - kraj temperatura - - -
brzina[brzinaIndex]=anemoMetarVrijednost_OBRT;

  lcd.clear();
  vrijeme = millis();
}

// svakih 30 sec :
if (millis() > vrijeme1+30000){
  vrijeme1 = millis();
  int i, brZb;
  for (i=0; i<= sizeof(brzina); i++){
    brzinaUdar=max(brzinaUdar,(brzina[i] + brzina[i-1] + brzina[i-2])/3);
    brZb=brZb+brzina[i];
  }
  brzinaSred=brZb/600;
  // slanje podataka u easyweather.dat
  // izvj,0,0,2013-00-00,12:00:00 AM,0,0,0,0,temp,0,0,0,tlak,brzinaSred,0,brzinaUdar,0,0,smjerTxt,0,0,0,0,0,0,0
  izvj++;
  Serial.println();
  Serial.print(izvj,DEC);
  Serial.print(",0,0,2013-00-00,12:00:00 AM,0,0,0,0,");
  Serial.print(temperatura);
  Serial.print(",0,0,0,0,");
  Serial.print(tlakIz, 1);
  Serial.print(",");
  Serial.print(brzinaSred,1);
  Serial.print(",");
  Serial.print(brzinaUdar);
  Serial.print(",0,0,0,");
  Serial.print(smjerTxt);
  Serial.print(",0,0,0,0,0,0,0,0");

  // slanje na serijski port Serial1
  Serial1.println();
  Serial1.print(izvj,DEC);
  Serial1.print(",0,0,2013-00-00,12:00:00 AM,0,0,0,0,");
  Serial1.print(temperatura);
  Serial1.print(",0,0,0,0,");
  Serial1.print(tlakIz, 1);
  Serial1.print(",");
  Serial1.print(brzinaSred,1);
  Serial1.print(",");
  Serial1.print(brzinaUdar);
  Serial1.print(",0,0,0,");
  Serial1.print(smjerTxt);
  Serial1.print(",0,0,0,0,0,0,0,0");
}

```



```

    }

// - - - KONTROLNI IZLAZI - - -

Serial.println("\n-----");

//kontrolni izlaz anemo
Serial.print("w: ");
Serial.print(anemoMetarVrijednost_OBRT, DEC);
Serial.print(" ");

//kontrolni izlaz smjera
Serial.print("\tSmjer=");
Serial.print(smjer);

//kontrolni baro
Serial.print("\t P = ");
Serial.print(tlakIz);
Serial.print(" hPa");
Serial.print("\t");
Serial.print(analogRead(baroSenzor));

//kontrolni temp
Serial.print("\t T= ");
Serial.print(temperatura);
Serial.print(" C");

// smjer
lcd.setCursor(0,1);
lcd.print("s=");
lcd.print(smjer);
lcd.print(" ");
// brzina
lcd.setCursor(0,0);
lcd.print("w=");
lcd.setCursor(3,0);
lcd.print(anemoMetarVrijednost_OBRT, DEC);
// baro
lcd.setCursor(7,0);
lcd.print("p=");
lcd.print(tlakIz);
lcd.print("hPa");
//temp
lcd.setCursor(7, 1);
lcd.print("t=");
lcd.print(temperatura);
lcd.print("C");

} //----- kraj loop-a -----

//-----FUNKCIJE-----
// anemoMetar
void ISR_anemoMetar()
{
    // check the value again - since it takes some time to
    // activate the interrupt routine, we get a clear signal.
    anemoMetarVrijednost++;
}
// kraj anemoMetar

// baro

float tlak(float senzorTlak){
    float tlakIzracun;
    //senzorTlak=(analogRead(baroSenzor));
    //senzorTlak=943;
    tlakIzracun = (((senzorTlak/1024)+0.1518)/0.01059)*10;
    return tlakIzracun;

    /* Motorola MPX Data Sheet :Transfer Function
       Vout = VS (P x 0.01059 â 0.1518) +/- â (Pressure Error x Temp. Factor x 0.01059 x VS)
       VS = 5.1 V Â± 0.25 Vdc
    */
}

// kraj baro

// tremometar DS1820
float temperaturaDS(){
    byte i;
    byte present = 0;
    byte type_s;

```

```

byte data[12];
byte addr[8];
float tempDS;

if ( !ds.search(addr)) {
    ds.reset_search();
    delay(250);
}
if (OneWire::crc8(addr, 7) != addr[7]) {
    Serial.println("CRC nije dobar!");
}
// ROM byte indicira tip senzora
switch (addr[0]) {
    case 0x10:
        type_s = 1;
        break;
    case 0x28:
        type_s = 0;
        break;
    case 0x22:
        type_s = 0;
        break;
}
ds.reset();
ds.select(addr);
ds.write(0x44, 1);
delay(1000);
present = ds.reset();
ds.select(addr);
ds.write(0xBE);
// citanje podataka
for ( i = 0; i < 9; i++) {
    data[i] = ds.read();
}

// konverzija binarnog podatka u temperaturu
int16_t raw = (data[1] << 8) | data[0];
if (type_s) {
    raw = raw << 3;
    if (data[7] == 0x10) {
        raw = (raw & 0xFFF0) + 12 - data[6];
    }
} else {
    byte cfg = (data[4] & 0x60);
    if (cfg == 0x00) raw = raw & ~7;
    else if (cfg == 0x20) raw = raw & ~3;
    else if (cfg == 0x40) raw = raw & ~1;
}
tempDS = (float)raw / 16.0;
return tempDS;
} // kraj tremometar DS1820

```

Literatura:

- [1] http://www.qsl.net/on7eq/projects/arduino_davis.htm 24.12.2013
- [2] B. A. Harper 1 , J. D. Kepert 2 and J. D. Ginger, GUIDELINES FOR CONVERTING BETWEEN VARIOUS WIND AVERAGING PERIODS IN TROPICAL CYCLONE CONDITIONS, World Meteorological Organization, 2008
- [3] http://wiki.sandaysoft.com/a/EasyWeather_Format 24.12.2013
- [4] Integrated Silicon Pressure Sensor for Manifold Absolute Pressure Applications On-Chip Signal Conditioned, MPX4100A MPXA4100A SERIES, Motorola, Inc. 2001
- [5] Digital relative humidity & temperature sensor RHT03, MaxDetect Technology Co., Ltd.
- [6] F.Jacob, Handbook of modern sensors : physics, designs, and applications, Springer-Verlag New York, Inc.2004
- [7] <http://arduino.cc/en/Main/ArduinoBoardMega2560> 24.12.2013
- [8] www.arduino.cc, 24.12.2013