



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradu informacija
Sveučilište u Zagrebu

Anemometar



- Δ Dokumentacija za krajnjeg korisnika
- Δ Arduino mikrokontroler
- Δ Senzor za određivanje brzine vjetra
- Δ Senzor za određivanje smjera vjetra

Sažetak

Anemometar je uređaj koji služi za mjerenje brzine vjetra. Dijelimo ih na pretvornike za mjerenje linearne i pretvornike za mjerenje rotacijske brzine. Glavnu primjenu je pronašao u nautici i meteorologiji, ali ćemo mi projektirati anemometar koji će se moći koristiti u malim meteorološkim stanicama ili u kućanstvu. Za mjerenje smjera vjetra potreban nam je dodatni senzor. Naš sklop će mjeriti brzinu i smjer vjetra svakih 20 sekundi i izmjerene podatke slati preko USB sučelja na web server. Prednosti ovakvog sklopa su što možemo lako realizirati anemometar pomoću DC motora iz nekih starih uređaja. Nedostataka ima podosta i svi utječu na točnost mjerenja, a neki od njih su devijacija vjetra, razina na kojoj se nalazi sklop, vrijeme potrebno da se motor počne i zaustavi vrtjeti, nesavršenosti motora koje mogu utjecati na osjetljivost i teško obuhvaćanje svih komponentata vektora vjetra.

Sadržaj

1. UVOD	3
2. OPIS SKLOPOVLJA	4
3. KORIŠTENI SKLOPOVLJE	5
3.1. Arduino Uno R3.....	5
3.2. Arduino Ethernet Shield R3.....	6
3.3. DC motor	7
3.4. Enkoder	8
4. SPAJANJE SKLOPOVLJA	9
5. PROGRAMSKA PODRŠKA.....	11
6. ZAKLJUČAK.....	15
7. LITERATURA.....	16
8. POJMOVNIK	17

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

1. Uvod

Vjetar ne puše u svim dijelovima zemlje jednako. Postoje područja veće prosječne godišnje brzine vjetra i područja manje prosječne godišnje brzine vjetra. Vodeći evidenciju o brzini i smjeru puhanja vjetra možemo stvoriti vlastitu bazu podataka pomoću koje možemo procijeniti brzine vjetra za određeno razdoblje.

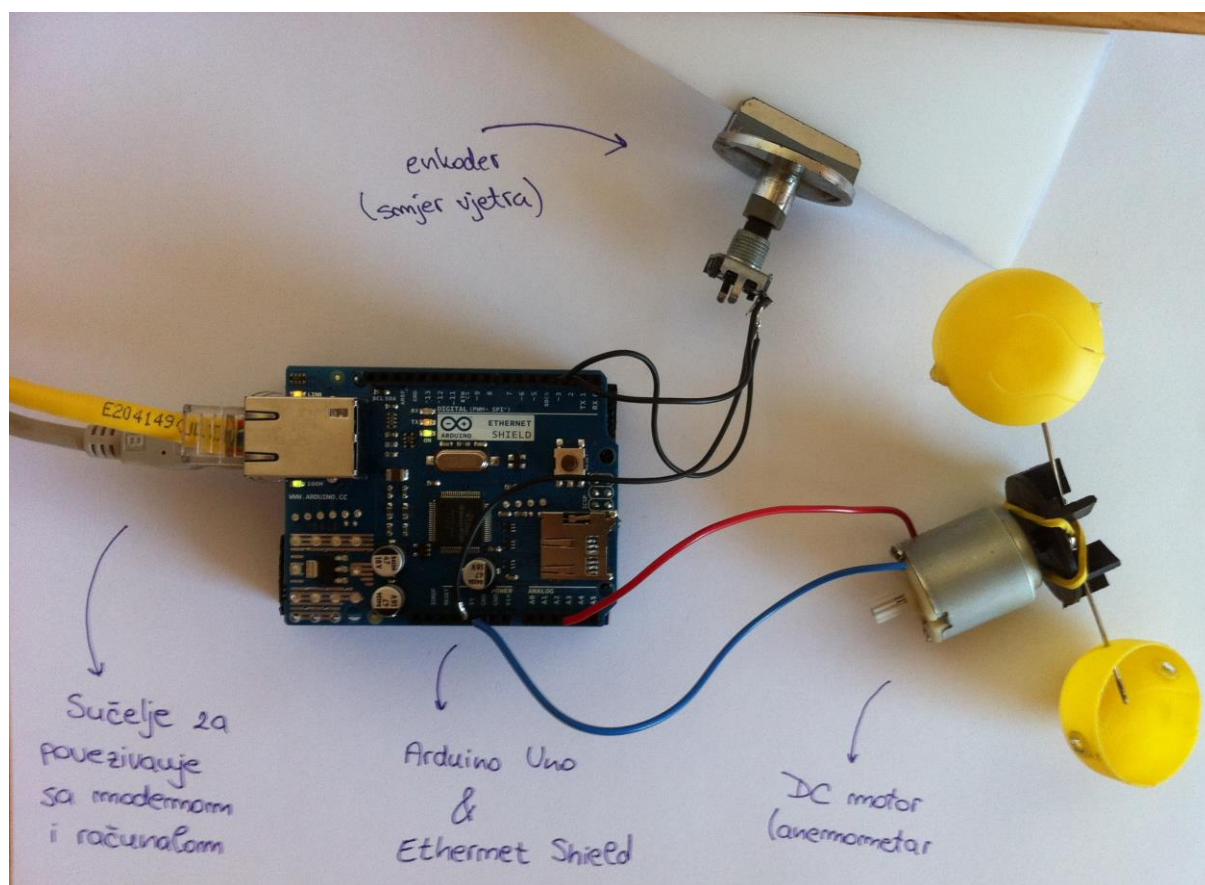
Ovakav sustav je zgodan za ljude koji se planiraju baviti meteorologijom, a uz par dodatnih modifikacija poput dodavanja senzora za padaline, temperaturu i vlagu, moguće je realizirati vlastitu malu meteorološku stanicu.

Sve podatke koje prikupimo možemo slati na dogovoreni web server. Na taj način smo prikupljene podatke dali na korištenje drugim sustavima.

2. Opis sklopovlja

Cijeli sustav se sastoji od Arduino mikrokontrolera, Arduino Ethernet Shielda, DC motora koji će obavljati ulogu anemometra, rotacijskog enkodera koji služi za dobivanje podataka o smjeru vjetra i USB sučelje. Na Arduino mikrokontroler spajamo sve komponente i sami mikrokontroler povezujemo sa računalom preko USB sučelja.

Na sljedećoj slici će biti prikazano kako je potrebno spojiti sklopovlje. Arduino Ethernet Shield ima izvode tako da ga postavljamo direktno na Arduino mikrokontroler koji ga i napaja.



Slika 1. Način spajanja sklopovlja

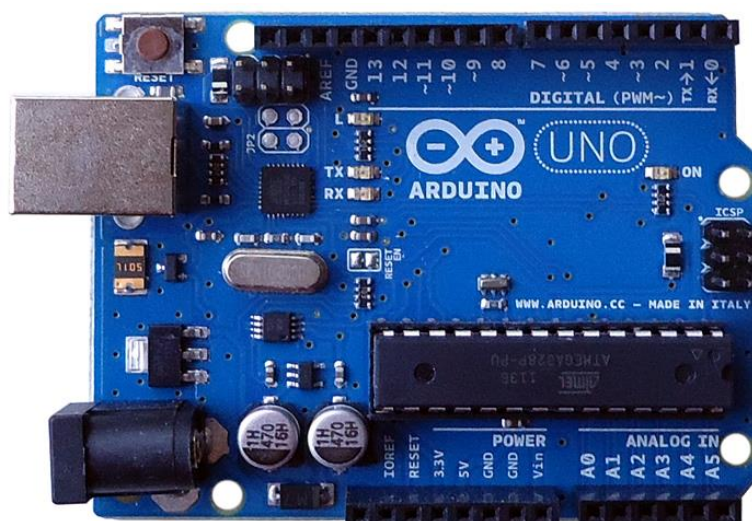
3. Korišteno sklopovlje

Arduino je open-source platforma za kreiranje električnih prototipova bazirana na sklopovlju i programskom paketu koji je fleksibilan i jednostavan za korištenje. Mogu ga koristiti elektroničari, programeri, dizajneri, umjetnici i svi koji su zainteresirani za kreiranje interaktivnih objekata i okruženja.

Arduino je skup elektroničkih i softverskih komponenti koje se mogu jednostavno povezati u složenije cjeline. U našem projektu ćemo koristiti Arduino Uno mikrokontroler i Arduino Ethernet Shield.

3.1. Arduino Uno R3

Za razliku od svoje prethodne generacije, Arduino Uno R3 koristi ATmega16U2 mikrokontroler umjesto 8U2. Novi mikrokontroler je zaslužan za veće brzine prijenosa i veću memoriju. Dodana su dva nova pina (SDA i SDL), koji se na pločici nalaze neposredno nakon pina za AREF. Također su dodana dva dodatna pina uz pin RESET-a. Jedan se zove IOREF koji omogućuje zaštitu prilikom prilagodbe naponu koji je se nalazi na pločici. Drugi nije spojen i rezerviran je za neke buduće upotrebe. Jedna od najkorisnijih značajki nam omogućava da upišemo softvere u naš MCU-u bez da ga vadimo iz pločice. To nam omogućava ICSP.



Slika 2. Arduino Uno R3

Određene značajke koje krasi ovaj mikrokontroler su:

- ATmega328 mikrokontroler
- Ulazni napon od 7V do 12V
- 14 digitalnih I/O pinova (6 PWM izlaza)
- 6 analognih ulaza
- 32k Flash memorije
- 16MHz keramički rezonator
- USB konektor
- Utičnica za napajanje
- Tipka za RESET
- ICSP- In Circuit Serial Programming

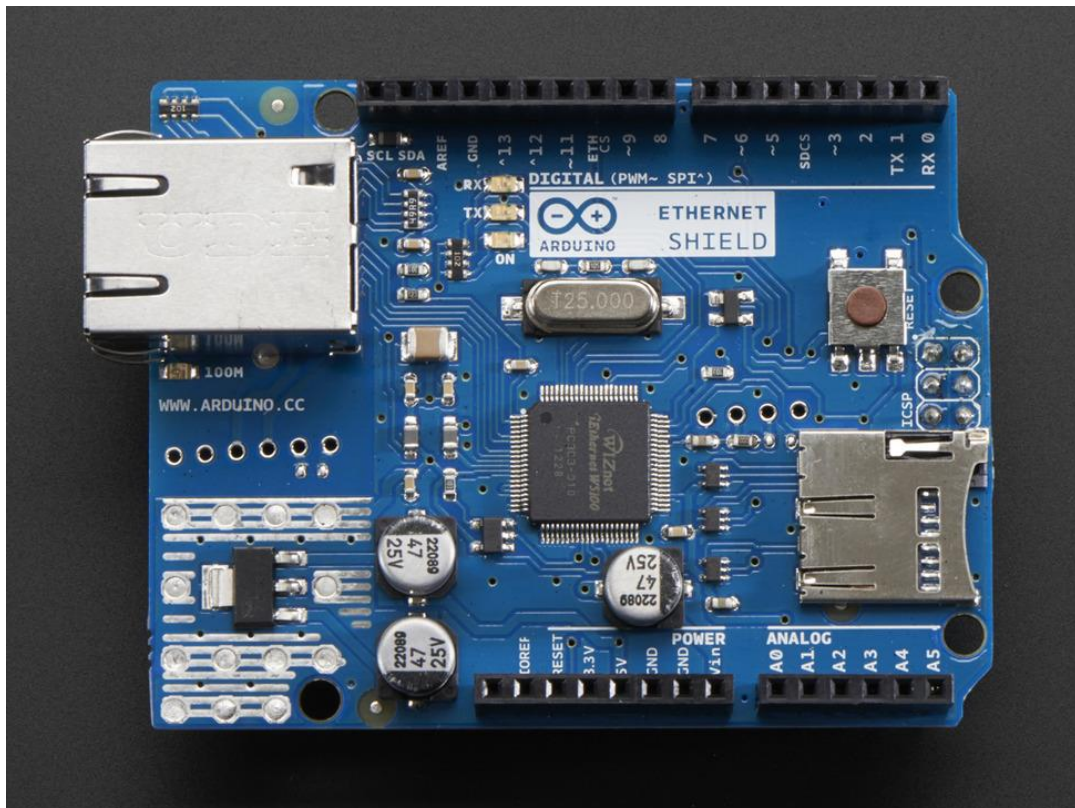
3.2. Arduino Ethernet Shield R3

Ovaj sklop nam služi za spajanje Arduina na Internet u svega nekoliko minuta. Napaja ga Arduino mikrokontroler. Sa mrežom ga možemo spojiti koristeći kabel sa RJ-45 konektorom. Baziran je na Wiznet W5100 Ethernet čipu. Wiznet W5100 nam omogućava rad i sa TCP-om i UDP-om.

Ethernet shield je se jednostavno spaja na Arduino pločicu, jer nožice Ethernet Shielda se savršeno poklapaju sa pinovima pločice. Na taj način su kontakti pinova zaštićeni i ne mogu doći u kontakt s drugom komponentom koju možemo spojiti na Ethernet Shield.

Značajke i određeni dijelovi pločice:

- Brzina prijenosa podataka: 10/100 Mb
- Informativne LED diode:
 - Primamo podatke ili šaljemo podatke
 - Imamo napajanje pločice
 - Postignuta maksimalna brzina prijenosa
 - Full-duplex
 - Kolizija u vezi
- Radni napon iznosi 5V
- Koristi Ethernet kontroler W5100 sa unutrašnjim spremnikom 16k



Slika 3. Arduino Ethernet Shield R3

3.3. DC motor

Istosmjerni motor je elektromehanički uređaj koji istosmjernu struju pretvara u rotacijsko gibanje. Ukoliko se rotor istosmjernog motora mehanički spoji s izvorom rotacijskog gibanja na izvodima će se pojaviti inducirani napon, čime je realiziran istosmjerni generator. Brzina vrtnje motora je proporcionalna naponu.

Motor sam izvadio iz stare igračke, a za rad mu je potrebno 9V.



Slika 4. Istosmjerni motor sa izvodima

3.4. Enkoder

Enkoderi su vrsta senzora koja služi za mjerenje pomaka, a možemo ih podijeliti na inkrementalne i apsolutne. Svaka od ovih podjela može se razgranati još na linearne i rotacijske. Inkrementalni enkoderi generiraju impuls za svaki pojedinačni zakret senzora (za rotacijski enkoder) ili impuls za svaki inkrement linearnog pomaka. Ukupna prijeđena udaljenost mjeri se brojanjem izlaznih impulsa.

Apsolutni enkoder ima određeni broj izlaznih kanala tako da svaki položaj može biti opisan jedinstvenim izlaznim kodom. Što je veća osjetljivost to je broj potrebnih izlaznih kanala veći. Prema smjeru kretanja mogu biti jednokanalni (ne prati smjer kretanja) i dvokanalni (prati smjer kretanja).



Slika 5. Rotacijski enkoder

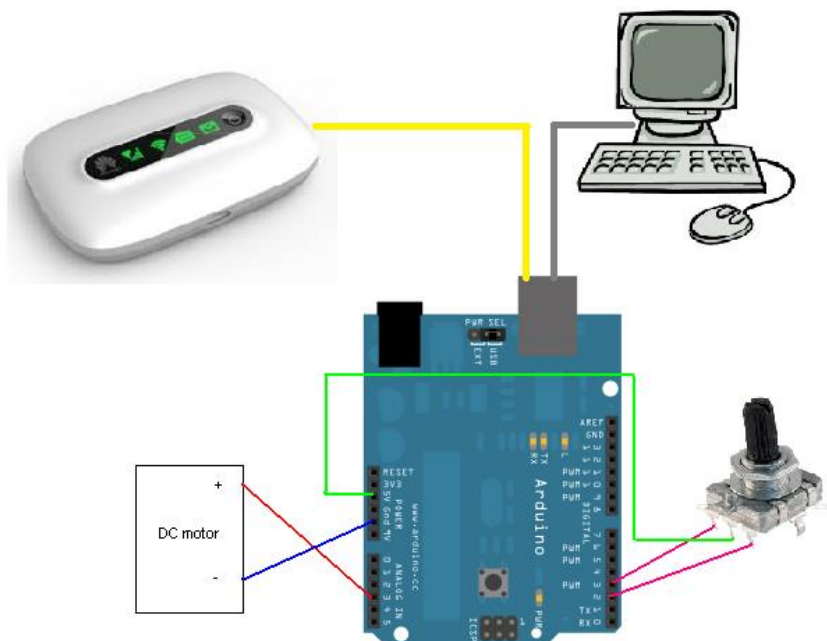
Pomoću rotacijskog enkodera ćemo pokušati implementirati brojač kojeg ćemo povećavati i smanjivati s obzirom na okretanje čvora. Potrebno je odrediti neku polaznu poziciju enkodera. Nakon toga program će znati da li će povećati ili smanjiti stanje brojila, a operacija ovisi o memorijskoj vrijednosti enkodera.

Postoji više metoda kako se mjerenje pomaka može mjeriti efektivno i točno, ali jedna odskače od drugih. To je metoda korištenjem vremenskog prekida.

Učestalost čitanja porta mora biti podešena tako da se svaka promjena stanja na portu registrira. Ako je učestalost čitanja porta u odnosu na brzinu promjene stanja na ulazu u sport preniska, vjerojatno nećemo registrirati svaku promjenu. Tada ćemo imati preskoke stanja i pogrešno očitane pozicije.

4. Spajanje sklopovlja

Na slici 1 možemo vidjeti kako se spajaju komponente. Jedino što nedostaje na slici je mrežni kabel sa RJ-45 konektorom. Mrežni kabel se spaja jednim krajem u Ethernet shield, a drugim krajem u modem.



Slika 6. Konačna shema spajanja

Kako se lopatice DC motora budu okretale, na njegovim izlaznim stezaljkama će se generirati napon. Potrebno je programirati Arduino na taj način da generirani napon bude proporcionalan brzini okretanja njegovih vanjski postavljenih lopatica. Na slici je vidljivo da DC motor spajamo na ulazni analogni pin 3 i na GROUND pin. Enkoder mi je zadao puno veće probleme. Naš enkoder ima 24 moguća položaja. Položaje sam podijelio u osam različitih skupina, a svaka skupina predstavlja određeni smjer puhanja vjetra. Na enkoder je potrebno instalirati određene propelere koji bi okretali sami enkoder, a prilikom stacioniranja u određenom položaju za povratnu informaciju bi nam izbacio smjer vjetra. Problem sa enkoderom je taj što je podosta robustan. Potrebno mu je dovesti napajanje (5 V), a druga dva izvoda enkodera spajamo na digitalne pinove Arduino pločice.

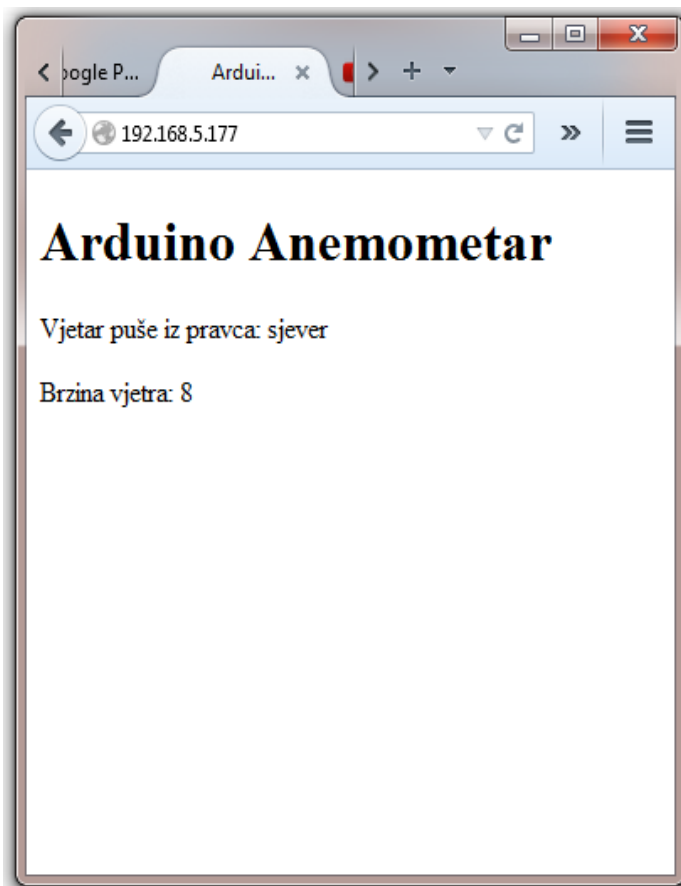
Razlika između analognih i digitalnih pinova je ta što sa digitalnim pinovima možemo generirati prekide. Ako ih koristimo kao ulaze, oni tada predstavljaju početni dio nekog sustava i na njih se spajaju razni senzori koji će prikupljati diskretne informacije. Oznaka PWM uz 3. pin označava

pulsno širinsku modulaciju, a služi nam za dobivanje analognih vrijednosti preko digitalnih.

Nakon što povežemo Arduino Uno pomoću USB sučelja sa računalom (siva linija na slici 6) i Ethernet Shield pomoću mrežnog kabela (žuta linija na slici 6) sa modemom, možemo početi programirati naš Arduino. Za početak je potreban neki server na kojem ćemo prikazivati rezultate. Postoje gotova rješenja koja nam osiguravaju server, ali je potrebno paziti na koji ćemo način zadati IP adresu i MAC adresu.

MAC adresu očitamo sa podnožja Arduino Ethernet kartice i ukucamo na za to predodređeno mjesto. IP adresa mora biti u rasponu korištenih adresa. Našu IP adresu možemo lako doznati. Prvo pokrenemo 'run' na našem računalu i zatim u njega ukucamo 'cmd'. Otvara nam se novi prozor u kojem upišemo naredbu 'ipconfig'. IP adresa servera mora biti jedinstvena.

Ukoliko utipkamo korištenu IP adresu u naš web preglednik, otvoriti će nam se prozor sa naslovom "Arduino Anemometar". Ispod naslova imamo dvije linije, jednu za ispis smjera vjetra, a drugu za ispis brzine strujanja vjetra.



Slika 7. Prikaz web servera

5. Programska podrška

U završnom poglavlju ćemo se baviti programskim kodom. Prije nego prikažemo sami kod, objasniti ćemo zašto smo koristili Bounce biblioteku. Bounce biblioteku koristimo jer nam omogućava pouzdano očitavanje mehaničkih senzora. Proučili smo jedan primjer koji je čitao stanje na pinu što je brže moguće. Svaki puta kada bi pin prelazio iz visokog stanja u nisko, uvećala bi se varijabla za brojanje. Napravili smo usporedbu sa i bez Bouncea. Kada smo koristili program bez Bounce biblioteke i načina programiranja, naš brojač se nije uvećavao ravnomjerno. Ponekad se brojač uvećao za više od jedan. Budući da biblioteka Bounce obrađuje mehaničke odskoke pravilno, brojač se povećava točno za jedan prilikom svakog pritiska gumba.

Često se umjesto Bounce biblioteke koristi jednostavno kašnjenje nakon čitanja mehaničkog prekidača ili tipkala. U drugim programima, kašnjenje od `Serial.print()` ili neke druge funkcije može biti dovoljno za eliminaciju preskoka u brojaču.

Prije samog pokretanja programa je potrebno podesiti korišteno USB sučelje i korištenu pločicu. Ukoliko nismo sigurni na koji nam je port spojen naš Arduino možemo to vrlo jednostavno provjeriti u Device Manageru.

```
#include <SPI.h>
#include <Ethernet.h>
#include <Bounce.h>

int Pin2 = 2; //deklaracija pinova enkodera
int Pin3 = 3;
unsigned int tcnt2;
Bounce bouncer1 = Bounce( Pin2,7 );

int temp_brzvjetra=0;
int temp_smjerbr=0;
volatile int i=0;
volatile long enkoder = 0;
volatile unsigned short smjerbr = 0;

int analogPin = 3; //deklaracija pina iz DC motora
float brzvjetra = 0;
float vrijednost = 0;

// MAC adresa sa dna Ethernet shield pločice
byte mac[] = {0x90, 0xA2, 0xDA, 0x0D, 0xA8, 0x34 };
IPAddress ip(192,168,5,177); // IP adresa u rasponu naše mreže
EthernetServer server(80); // server radimo na 80 portu

String HTTP_req; // skladištenje HTTP zahtjeva
```

```
void setup()
{
  Serial.begin (115200);

  pinMode(Pin2, INPUT);
  pinMode(Pin3, INPUT);
  pinMode(analogPin, INPUT);

  TIMSK2 &= ~(1<<TOIE2);
  TCCR2A &= ~(1<<WGM21 | (1<<WGM20));
  TCCR2B &= ~(1<<WGM22);
  ASSR &= ~(1<<AS2);
  TIMSK2 &= ~(1<<OCIE2A);
  TCCR2B |= (1<<CS22 | (1<<CS20));
  TCCR2B &= ~(1<<CS21);
  tcnt2 = 131;
  TCNT2 = tcnt2;
  TIMSK2 |= (1<<TOIE2);

  Ethernet.begin(mac, ip); // inicijalizacija Ethernet uređaja
  server.begin(); // počinjemo slušati za klijente
  Serial.begin(9600); // za dijagnostiku
}

ISR(TIMER2_OVF_vect) {
  TCNT2 = tcnt2;
  bouncer1.update();
  if(bouncer1.risingEdge()){
    if (digitalRead(Pin3)){
      enkoder--;
      if(enkoder==-1){ //ako je -1 onda je to 23 korak
        enkoder=23;}
    }else{
      enkoder++;}
    }
}

void loop()
{
  smjervjetra();
  brznavjetra();
  delay(1000);
  if(brzvjetra != temp_brzvjetra || smjerbr != temp_smjerbr){
    Serial.print("d");
    Serial.print(1);
    Serial.print("s");
    Serial.print(3);
    Serial.print("bv");
    Serial.print(brzvjetra);
    Serial.print("ju");
    Serial.print(smjerbr);
    temp_brzvjetra=brzvjetra;
    temp_smjerbr=smjerbr;
  }
}
```

```

EthernetClient client = server.available();

if (client) {
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      HTTP_req += c;
      if (c == '\n' && currentLineIsBlank) {
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: keep-alive");
        client.println();

        if (HTTP_req.indexOf("ajax_switch") > -1) {
          /
          GetAjaxData(client);
        }
        else {
          client.println("<!DOCTYPE html>");
          client.println("<html>");
          client.println("<head>");
          client.println("<title>Arduino Web Page</title>");
          client.println("<script>");
          client.println("function GetAnalogData() {}");
          client.println(
            "nocache = \"&nocache=\" + Math.random() * 1000000;");
          client.println("var request = new XMLHttpRequest();");
          client.println("request.onreadystatechange = function(){}");
          client.println("if (this.readyState == 4) {}");
          client.println("if (this.status == 200) {}");
          client.println("if (this.responseText != null) {}");
          client.println("document.getElementById(\"sw_an_data\")\
.innerHTML = this.responseText;");
          client.println("}}}}");
          client.println(
            "request.open(\"GET\", \"ajax_switch\" + nocache, true);");
          client.println("request.send(null);");
          client.println("setTimeout('GetAnalogData()', 1000);");
          client.println("{}");
          client.println("</script>");
          client.println("</head>");
          client.println("<body onload=\"GetAnalogData()\">");
          client.println("<h1>Arduino Anemometer</h1>");
          client.println("<div id=\"sw_an_data\">");
          client.println("</div>");
          client.println("</body>");
          client.println("</html>");
        }

        Serial.print(HTTP_req);
        HTTP_req = "";
        break;
      }
    }
  }
}

```

```

if (c == '\n') {
    currentLineIsBlank = true;
}
else if (c != '\r') {
    currentLineIsBlank = false;
}
}
}
delay(1);
client.stop();
}
}

void smjervjetra() { //funkcija za računanje smjera vjetra
    enkoder=enkoder%24; //ako je veće od 24 napravi modulo
    if((enkoder<=1)|| (enkoder>=23)){
        smjerbr=0;}
    else if(enkoder<=4){
        smjerbr=1;}
    else if(enkoder<=7){
        smjerbr=2;}
    else if(enkoder<=10){
        smjerbr=3;}
    else if(enkoder<=13){
        smjerbr=4;}
    else if(enkoder<=16){
        smjerbr=5;}
    else if(enkoder<=19){
        smjerbr=6;}
    else{ smjerbr=7;}
}

void brznavjetra() { //funkcija za mjerenje brzine vjetra
    vrijednost = analogRead(analogPin); //citanje pina 3
    brzvjetra =vrijednost/4; //racunanje brzine vjetra
}

void GetAjaxData(EthernetClient cl)
{
    int brzina_vjetra;
    int a;

    a=temp_smjerbr;
    if(a==0)
        cl.println("</p>Vjetar puše iz pravca: istok");
    else if(a==1)
        cl.println("</p>Vjetar puše iz pravca: sjevero-istok");
    else if(a==2)
        cl.println("</p>Vjetar puše iz pravca: sjever");
    else if(a==3)
        cl.println("</p>Vjetar puše iz pravca: sjevero-zapad");
    else if(a==4)
        cl.println("</p>Vjetar puše iz pravca: zapad");
    else if(a==5)
        cl.println("</p>Vjetar puše iz pravca:jugo-zapad");
    else if(a==6)
        cl.println("</p>Vjetar puše iz pravca: jug");
    else {a==7;
        cl.println("</p>Vjetar puše iz pravca: jugo-istok");}
    // read analog pin A3
    brzina_vjetra = temp_brzvjetra;
    cl.print("<p>Brzina vjetra: ");
    cl.print(brzina_vjetra);
    cl.println("</p>");
}

```

6. Zaključak

Realizirani sustav može zabilježiti brzinu vjetra i smjer vjetra. Sve rezultate smo prikazivali na IP adresi 192.168.5.177. Sustav je vrlo jednostavan za implementaciju, a može se koristiti u sklopu male kućne meteorološke stanice ili možemo dobivene rezultate preko servera dijeliti sa različitim znanstvenim ustanovama.

Cijeli se projekt može poboljšati dodavanjem većih lopatica na DC motor, kako bi se što lakše obuhvatio vektor puhanja vjetra. Također, potrebno je naći bolje rješenje za određivanje smjera vjetra. Enkoder je previše robustan, tako kada na njega dodamo lopatice, koje će ga vrtjeti prilikom puhanja vjetra, on bi se teško okretao i ne bi mogli točno izmjeriti smjer vjetra. Još jedan problem sam uočio tijekom testiranja, a to je da se lopatica dugo opire okretanju, a kada popusti, premaši možda i previše koraka enkodera. Potrebno je koristiti kvalitetniji enkoder ili prijeći na neku drugu tehnologiju.

7. Literatura

- [1] Petak Deni. Pametan vjetar. Zagreb 2012/2013. URL:
http://pametne-kuce.zesoi.fer.hr/doku.php?id=2013:petak_deni:start
- [2] Starting electronics, 2012. URL:
<http://startingelectronics.com/tutorials/arduino/ethernet-shield-web-server-tutorial/>
- [3] Electronic Projects, 2013. URL:
https://www.pjrc.com/teensy/td_libs_Bounce.html
- [4] Practical Usage, 2014. URL:
<http://practicalusage.com/?p=267>
- [5] Arduino, 2014. URL:
<http://www.arduino.cc/>
- [6] Lučev Vasić Željka. Mjerenje pomaka. Zagreb 2009. URL:
http://www.fer.unizg.hr/_download/repository/ST04-MjerenjePomaka.pdf

8. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
DC motor	Istosmjerni motor	http://en.wikipedia.org/wiki/DC_motor
Inkrementalni enkoder	Koristi se za detekciju smjera vjetra	http://www.etf.ucg.ac.me/materijal/1331303085Opticki_davaci.pdf
Arduino Uno	Razvojna pločica	http://arduino.cc/en/Main/arduinoBoardUno
Ethernet Shield	Pločica za povezivanje s Internetom	http://arduino.cc/en/Main/ArduinoEthernetShield
Anemometar	Uređaj za mjerenje brzine strujanja vjetra	http://hr.wikipedia.org/wiki/Anemometar