

Arduino meteorološka stanica

Upute za slaganje i programiranje



Djelo Upute za slaganje i programiranje Arduino meteorološke stanice, čiji je autor [Ivan Novosel](#), ustupljeno je pod licencom [CC Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima](#).

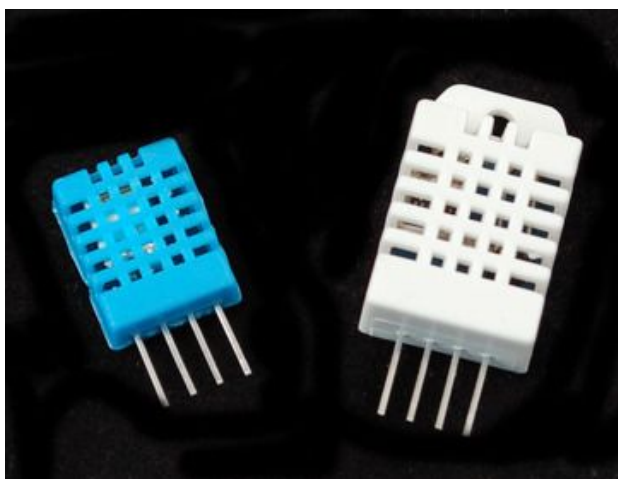
TLDR: Po ovim uputama moći ćete napraviti samostojeći mjerni uređaj za temperaturu i relativnu vlažnost zraka. Koriste se Arduino pločica, DHT senzor za temperaturu i vlažnost i LCD ekran. Upute su složene po koracima tako da možete provjeriti da svaki dio uređaja radi jednom kad je sklopljen, ali ako ste iskusniji s Arduinoom možete direktno skočiti na podnaslove za spajanje ekrana, tamo se nalazi konačna shema i program. Za dodatna pitanja javite se na ivan.novosel@tvornica-znanosti.org, a sve popratne datoteke dostupne su [ovdje](#).

DHT senzori

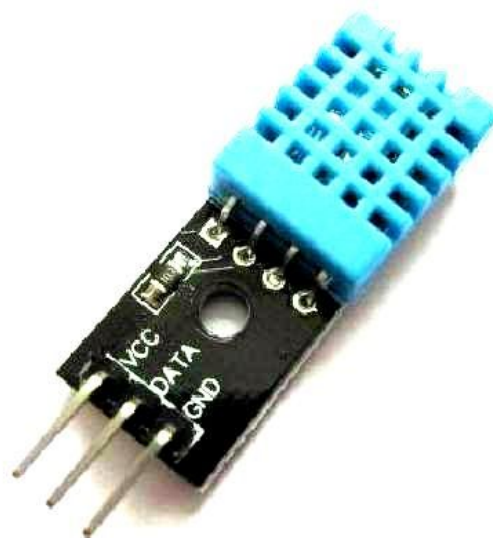
Za mjerenje relativne vlažnosti zraka i temperature koristi se DHT senzor koji ima više verzija i ovisno o tome ima malo drugačije karakteristike, no svi DHT senzori se koriste jednako. Sastoji se od dvije komponente koje su stavljene u zajedničko kućište:

- [termistor](#) - otpornik kojem se otpor mijenja ovisno o temperaturi
- [kapacitivni higrometar](#) - uređaj kojem se mijenja kapacitivnost ovisno o vlazi zraka

Senzor automatski očitava obje vrijednosti i pakira ih u jedan digitalni signal koji se može pročitati. Uobičajene verzije su DHT11, koji dolazi u vrlo malenom plavom kućištu i nešto je manje precizan, te DHT 22, koji dolazi u nešto većem bijelom kućištu i malo je veće preciznosti. Objе verzije standardno dolaze sa 4 izvoda na kućištu (s lijeva na desno): (1) VCC (+ napon), (2) Signal (podaci), (3) Ne radi ništa i (4) GND (- napon). Razni proizvođači rade i dodatne pločice koje imaju samo 3 izvoda, ali onda na pločici piše točno što koji izvod radi (ovo varira pa uvijek provjerite).

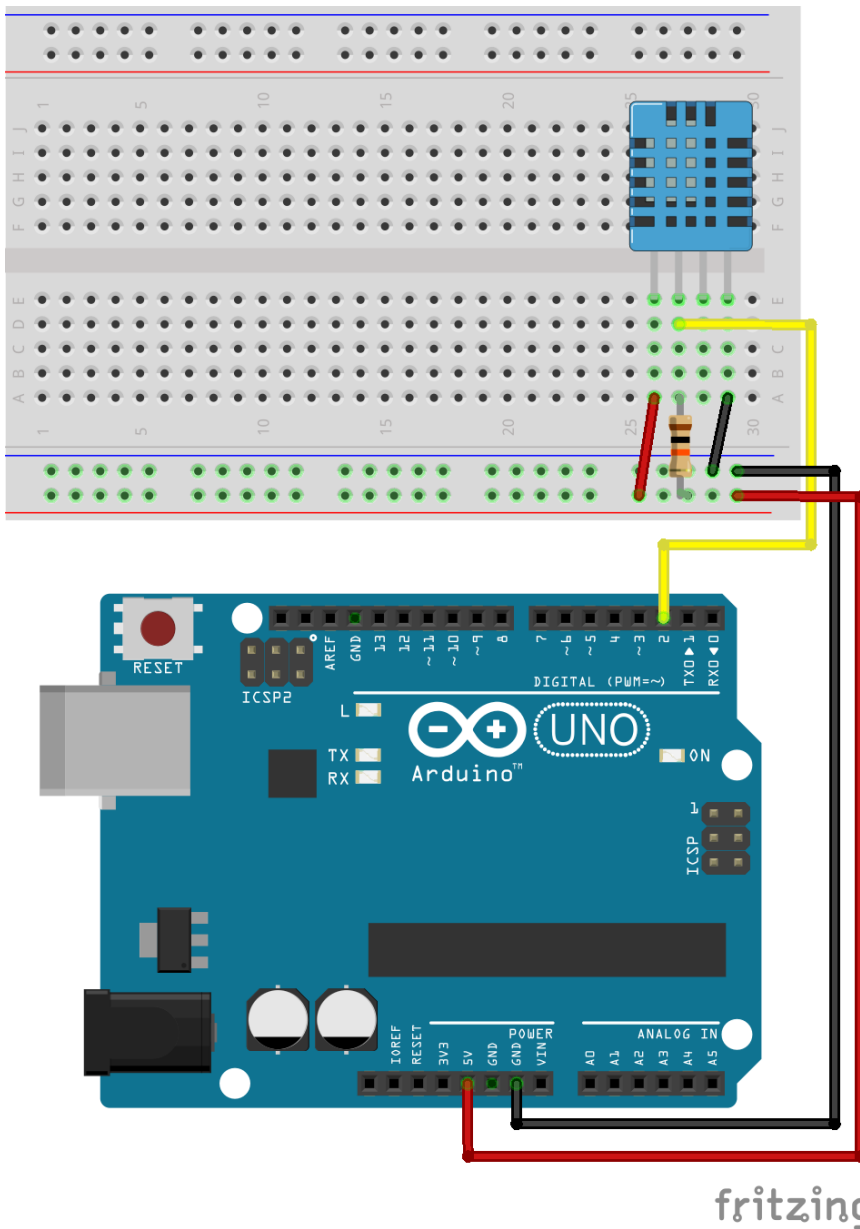


Izgled DHT11 (plavi) i DHT22 (bijeli) senzora.
Izvor: [Adafruit](#)



DHT11 senzor sa prepravljenim izvodima, ovisno o proizvođaču redoslijed može biti drugačiji, ali uvijek piše koji izvod je koji: VCC(+), Data (D,S ili Signal), GND(-)

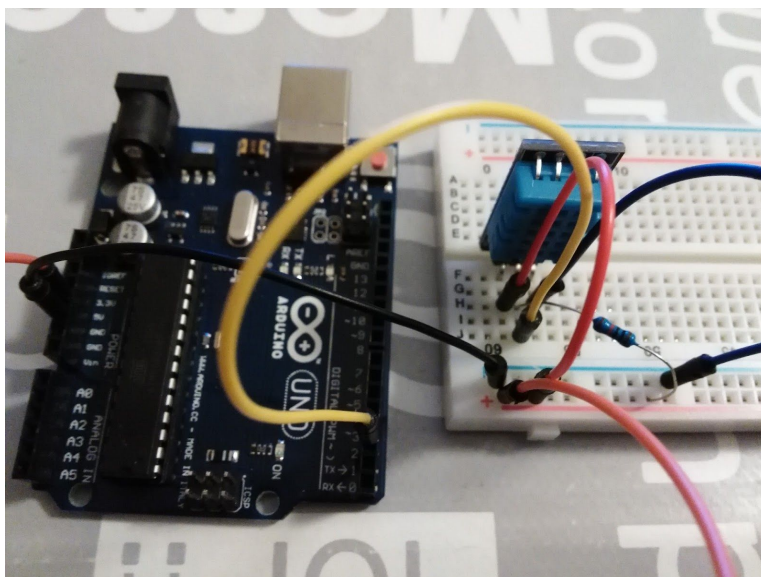
Spajanje DHT senzora na Arduino



S lijeve strane je prikaz spajanja DHT11 senzora na Arduino. Ako posjedujete verziju s drugačijim izvodima pripazite na sljedeće:

- (1) **VCC (+)** izvod senzora spaja se na 5V utor na Arduino pločici
- (2) **GND (-)** izvod senzora spaja se na GND utor na Arduino pločici
- (3) **Data** izvod senzora spaja se na jedan od digitalnih utora na Arduino (u ovim uputama to će nadalje biti utor 2)
- (4) Otpornik od 10kΩ spaja se s jednim krajem na Data izvod senzora, a drugim krajem na VCC senzora

fritzing

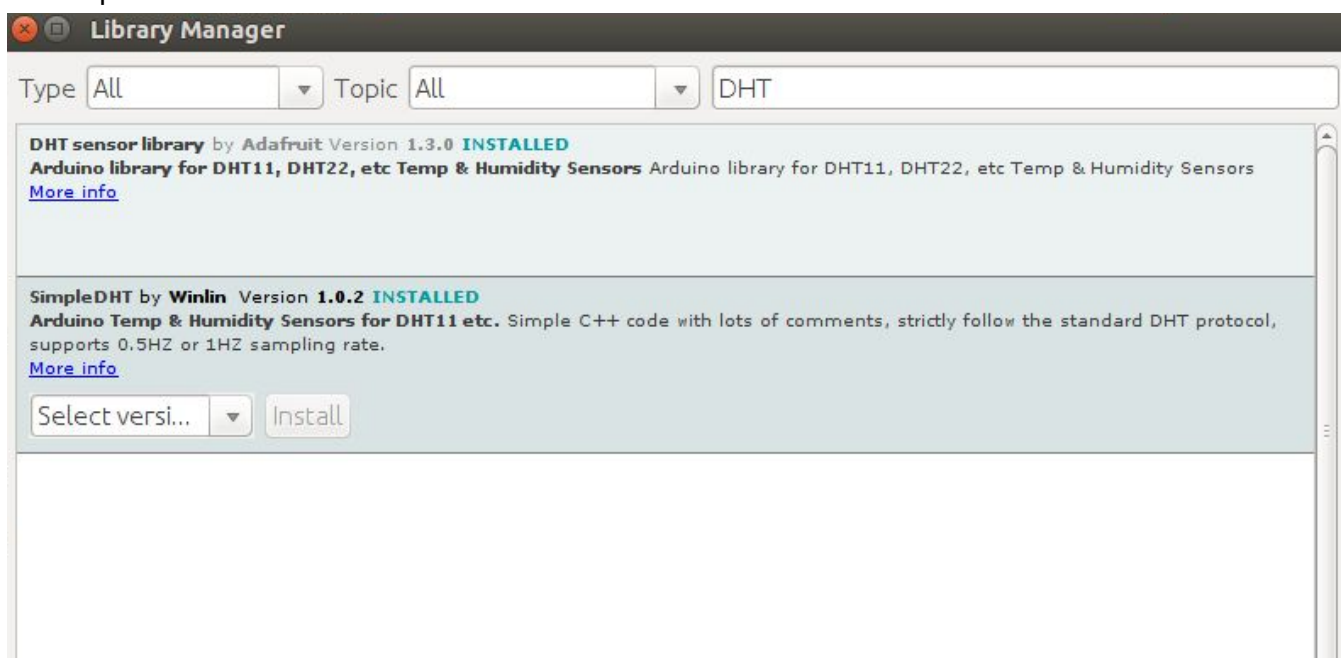


DHT11 senzor spojen na Arduino ovako izgleda uživo. Provjerite što piše na vašem senzoru kako bi bili sigurni da su izvodi dobro spojeni.

Programiranje DHT senzora

Kako bi mogli koristiti DHT senzor potrebno je dodati DHT library u Arduino IDE. Ako ste spojeni na internet najlakši način je da ga preuzmete sa Arduino repozitorija:

1. U prozoru Arduino IDE odaberite izbornik *Sketch -> Include Library -> Manage Libraries...*
2. Otvorit će vam se novi prozor u kojem će se prikazati sve moguće dodatne biblioteke koje se mogu dodati. U polje u kojem piše "*Filter your search...*" upišite DHT i prikazat će vam se dvije opcije.
3. Odaberite **DHT sensor library** i pritisnite gumb Install. Pričekajte dok se biblioteka ne preuzme.
4. Nakon toga ponovite proceduru, ali potražite **Adafruit Unified Sensor** biblioteku. U prozoru će biti prikazano više biblioteka, odaberite i instalirajte samo biblioteku s tim imenom.
5. Prije nego što pokrećete bilo kakav program koji uključuje biblioteku zatvorite i ponovno pokrenite Arduino IDE.



Izgled Library Manager prozora nakon što je biblioteka preuzeta i instalirana. Pored instalirane biblioteke treba se pojaviti plavi natpis INSTALLED.

Ako iz nekog razloga nemate pristup internetu biblioteka se može dodati odvojeno u obliku ZIP arhive:

1. Preuzmite biblioteku [ovdje](#). Pritisnite gumb Preuzmi (Download) u gornjem desnom uglu.
2. ZIP datoteku prebacite na željeno računalo.
3. U Arduino IDE programu odaberite *Sketch -> Include Library -> Add ZIP. Library...* i u prozoru koji se otvori pronađite odgovarajuću ZIP datoteku.

U ovim uputama koristit ćemo **DHT Sensor Library**. Ova biblioteka definira pristup raznim verzijama DHT senzora i sadrži nekoliko metoda koje olakšavaju rad sa podacima (prebacivanje između °C i °F, izračun prividne temperature (eng. heat index)). Za pregled svih opcija možete pogledati primjere koda koji dolaze sa bibliotekom (Izbornik: *File -> Examples -> DHT Sensor Library*), a u nastavku je priložen primjer koda s komentarima koji su nam potrebni da bi napravili meteo postaju. U ovom primjeru svi podaci se šalju na računalo tako da upalite Serial Monitor nakon što učitate kod u Arduino.

```

#include "DHT.h" //dodaje biblioteku koja nam treba za senzor

#define DHTPIN 2 // Definiramo pin 2 kao ulazni

// Ovaj dio koda definira koji tip senzora koristimo
// Trenutno je odabran DHT11, u slučaju da imate neki drugi tip
// zakomentirajte neželjeni red sa //, a odkomentirajte potrebni
#define DHTTYPE DHT11 // za DHT 11
//#define DHTTYPE DHT22 // za DHT 22 (AM2302), AM2321
//#define DHTTYPE DHT21 // za DHT 21 (AM2301)

// Ova linija koda definira objekt tipa DHT koji koristimo za pristup senzoru
DHT senzor(DHTPIN, DHTTYPE);

void setup() {
    // Program podatke šalje preko serijske veze
    // otvorite Serial Monitor kako bi vidjeli ispis na računalu
    Serial.begin(9600);
    senzor.begin(); // Ova linija pokreće praćenje senzora
}

void loop() {
    // DHT senzor treba neko vrijeme da bi napravio mjerenje, tako da se između
    svakog
    // mjerenja čeka 2 sekunde. Ova vrijednost ne smije biti manja od 1000 ms!
    delay(2000);
    Serial.println("-----");

    // readHumidity() metoda vraća relativnu vlažnost u postotcima
    float vlaga = senzor.readHumidity();
    // readTemperature() metoda vraća temperaturu u °C
    float temperatura = senzor.readTemperature();

    // ako neka vrijednost nije pročitana ovaj dio koda će uhvatiti grešku
    // i vratiti se na početak glavne petlje
    if (isnan(vlaga) || isnan(temperatura)) {
        Serial.println("GREŠKA: nema očitavanja sa senzora");
        return;
    }

    // ispis temperature na računalu
    Serial.print("Temperatura: ");
    Serial.print(temperatura);
    Serial.print(" *C\n");
    // ispis relativne vlažnosti na računalu
    Serial.print("Relativna vlaznost: ");
    Serial.print(vlaga);
    Serial.print(" %\n");
}

```


}

Korištenje LCD ekrana

Sklop koji je napravljen do sada mora biti spojen na računalo da bi vidjeli rezultate mjerenja. To je praktično za testiranje toga je li senzor dobro spojen, no u konačnici meteorološka postaja bi trebala biti samostalan uređaj. Postoji jako puno različitih LCD ekrana koji se mogu koristiti sa Arduinoom, no većina početničkih paketa dolazi sa ekranima baziranim na Hitachi HD44780 upravljaču, a Arduino IDE standardno uključuje biblioteku za ispis na takve ekrane.

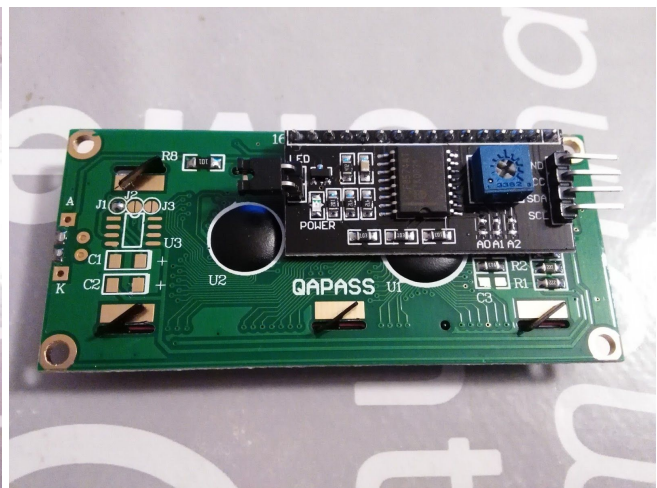
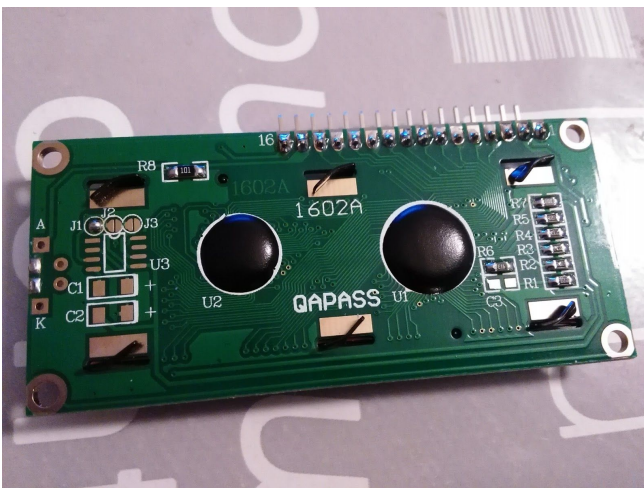
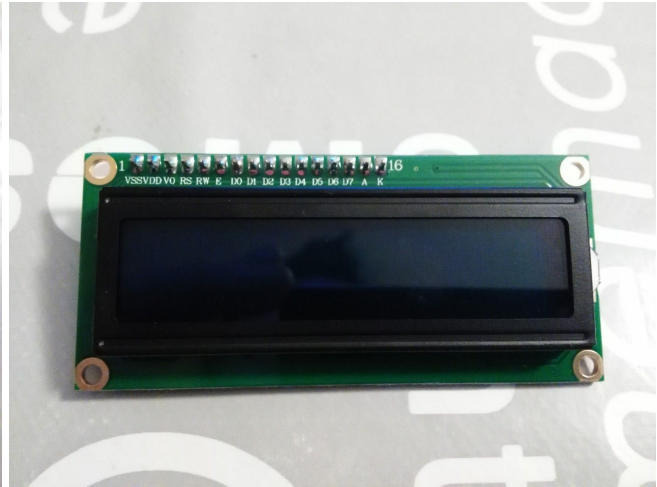
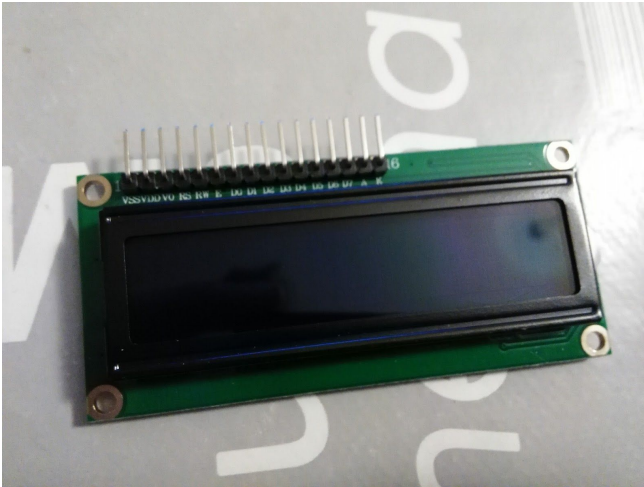
Takvi ekrani tipično imaju mogućnost prikazivanja 16x2 znakova, a ovisno o tipu ekrana koji ste dobili u paketu s Arduinoom postoji velika razlika kako ih se koristi:

- **Obična verzija** dolazi sa 16 izvoda i prima podatke paralelno preko digitalnih izlaza na Arduinou.
Prednost: programira se sa LiquidCrystal bibliotekom koja standardno dolazi uz Arduino
Mana: zahtjeva više žica i dodatni potencijometar (promjenjivi otpornik)

- **I2C verzija** na sebi ima dodatnu pločicu koja omogućava ispis preko I2C protokola, za koji su potrebne samo dvije žice za komunikaciju, uz napajanje.
Prednost: vrlo jednostavan za spajanje

Mana: potrebna instalacija dodatne biblioteke za LCD sa I2C protokolom, koja je nepouzdanija

U nastavku su upute za obje verzije ekrana, prema slikama ispod možete odrediti verziju koju imate.



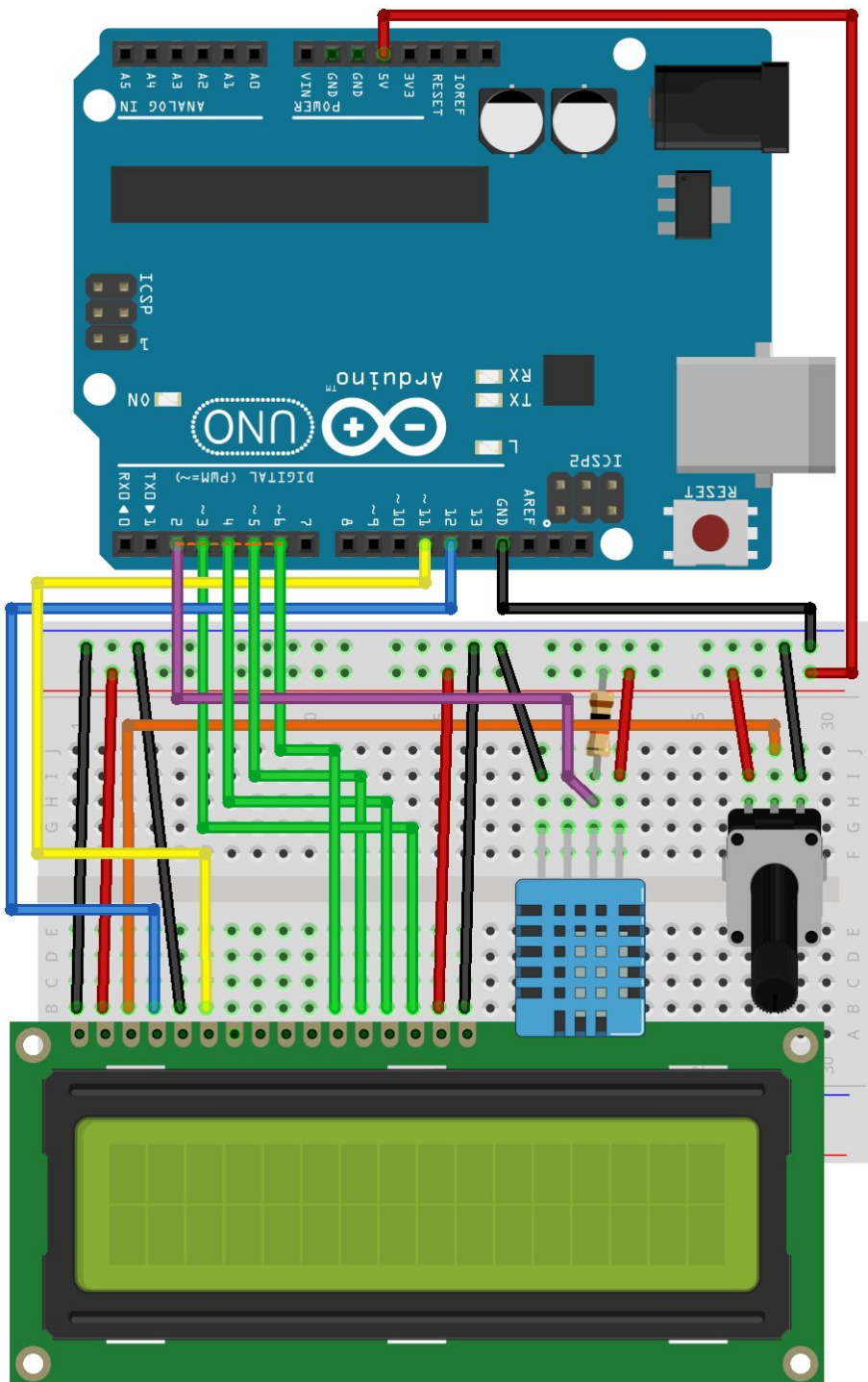
Obična verzija ima 16 izvoda koji se mogu direktno staviti u eksperimentalnu pločicu,

I2C verzija srijeda izgleda jednako, ali sa stražnje strane se nalaze 4 izvoda koji se žicama

stražnja strana je prazna.

povezuju sa ostatkom sklopa.

Spajanje obične verzije ekrana



Obična verzija LCD ekrana treba više žica za spajanje tako da budete oprezni kod spajanja. DHT senzor može ostati spojen kao i ranije, ovdje na slici je pomaknut kako bi potenciometar bio uz rub pločice (tako da ne nestane u gomili žica).

Jednom kada LCD umetnete na željeno mjesto na eksp. pločici ako pogledate LCD vidjet ćete da su nožice s lijeva na desno (1 do 16) označene i spajaju se:

- VSS** (lcd1) ide na **GND** (-)
- VDD** (lcd2) ide na **5V** (+)
- V0** (lcd3) ide na **srednju** nožicu potenciometra
- RS** (lcd4) ide na Arduino **pin 12**
- RW** (lcd5) ide na **GND** (-)
- E** (lcd6) ide na Arduino **pin 11**
- D0-D3 ostaju prazni
- D4** (lcd11) ide na Arduino **pin 6**
- D5** (lcd12) ide na Arduino **pin 5**
- D6** (lcd13) ide na Arduino **pin 4**
- D7** (lcd14) ide na Arduino **pin 3**
- A** (lcd15) ide na **5V** (+)
- K** (lcd16) ide na **GND** (-)

Potenciometar služi za namještanje kontrasta na LCDu. Srednja nožica ide na V0 pin LCD-a, lijeva na + napon, a druga na - napon.

Svi izvodi koji idu na GND pin Arduina su spojeni na zajedničku (-) liniju i sve te žice su crne boje na slici. Svi izvodi koji idu na 5V pin Arduina su spojeni na zajedničku (+) liniju i sve te žice su crvene boje.

Programiranje obične verzije ekrana

Obična verzija ekrana je kompliciranija za spajanje, ali programiranje se radi pomoću **LiquidCrystal** biblioteke koja standardno dolazi uz Arduino IDE i na službenim Arduino stranicama je [dobro dokumentirana](#). Za početak ćemo napraviti program koji će ispisivati kratku poruku na ekranu, kako bi provjerili da li radi.

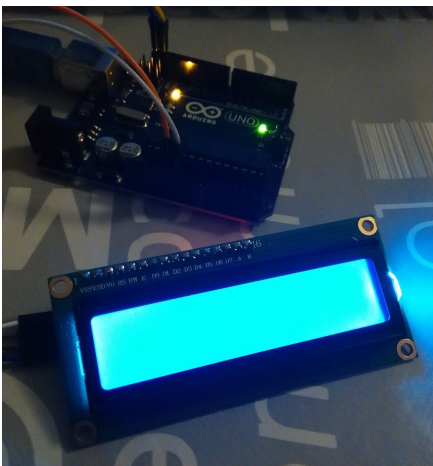
```
#include <LiquidCrystal.h> // daje funkcije za rad s LCDom

// sljedeći red postavlja objekat za rukovanje ekranom. funkciji dajemo brojeve
// pinova koji služe za kontrolu LCD-a: lcd(RS, E, D4, D5, D6, D7)
// brojevi moraju odgovarati tome kako je LCD spojen!
LiquidCrystal lcd(12, 11, 6, 5, 4, 3);

void setup() {
  // pokretanje ekrana i podešavanje točnog broja stupaca i redova
  lcd.begin(16, 2);
  // naredba print() ispisuje tekst na ekranu na trenutnoj poziciji kurzora
  lcd.print("Radim :)");
}

void loop() {
  // postavljanje kurzora na početak drugog reda
  // setCursor(stupac,red) uzima željenu poziciju
  // npr. (2,0) je treći znak prvog reda (brojanje počinje od nule!)
  lcd.setCursor(0, 1);
  // ispisuje broj sekundi od početka pokretanja
  lcd.print(millis() / 1000);
}
```

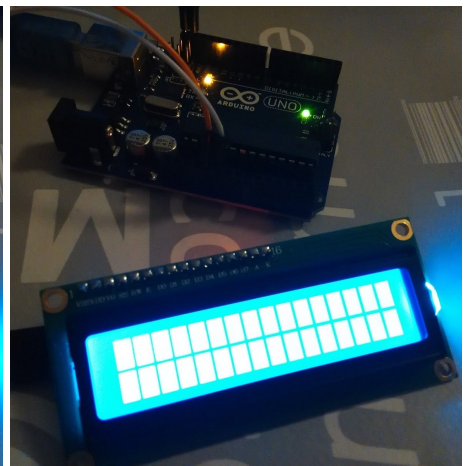
Ako je sve u redu ekran će ispisati poruku. Ako na ekranu vidite bijele kocke ili je sve u pozadinskoj boji (npr. plava) onda je kontrast prevelik ili premali. Okrećite potencijometar dok ne vidite poruku. (Slika ispod pokazuje razliku, no prikazana je verzija za I2C)



Kontrast je premali, slova se uopće ne vide.



Optimalan kontrast, slova se jasno vide, a pozadina se ne ističe.



Preistaknuta pozadina, ako gledate sa strane vidjet ćete poruku, ali jedva.

Kad je ekran namješten možemo složiti konačni program za meteorološku postaju. Koristit ćemo već postojeći kod za DHT senzor, ali ćemo naredbe za ispis u Serial monitor zamijeniti s onima za ispis na LCD ekran.

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal.h>

#define DHTPIN 2
#define DHTTYPE DHT11

DHT senzor(DHTPIN, DHTTYPE);
LiquidCrystal lcd(12, 11, 6, 5, 4, 3);

void setup() {
  lcd.begin(16, 2);
  senzor.begin();
}

void loop() {
  delay(2000);
  lcd.clear(); // na početku svake petlje počistimo prethodni sadržaj ekrana
  float vlaga = senzor.readHumidity();
  float temperatura = senzor.readTemperature();

  if (isnan(vlaga) || isnan(temperatura)) {
    lcd.setCursor(0,0); // namješta ispis na početak 1. reda
    // LCD podržava samo ASCII znakove (engleska abeceda), tako da puno
    // dodatnih znakova ne prepozna je osim ako ih ne napravite sa createChar()
    lcd.print("GRESKA");
    lcd.setCursor(0,1); // namješta ispis na početak 2. reda
    lcd.print("nema podataka");
    return;
  }

  lcd.setCursor(0,0);
  lcd.print("Temp: ");
  lcd.setCursor(6,0);
  lcd.print(temperatura);
  lcd.setCursor(11,0);
  lcd.print(" *C");

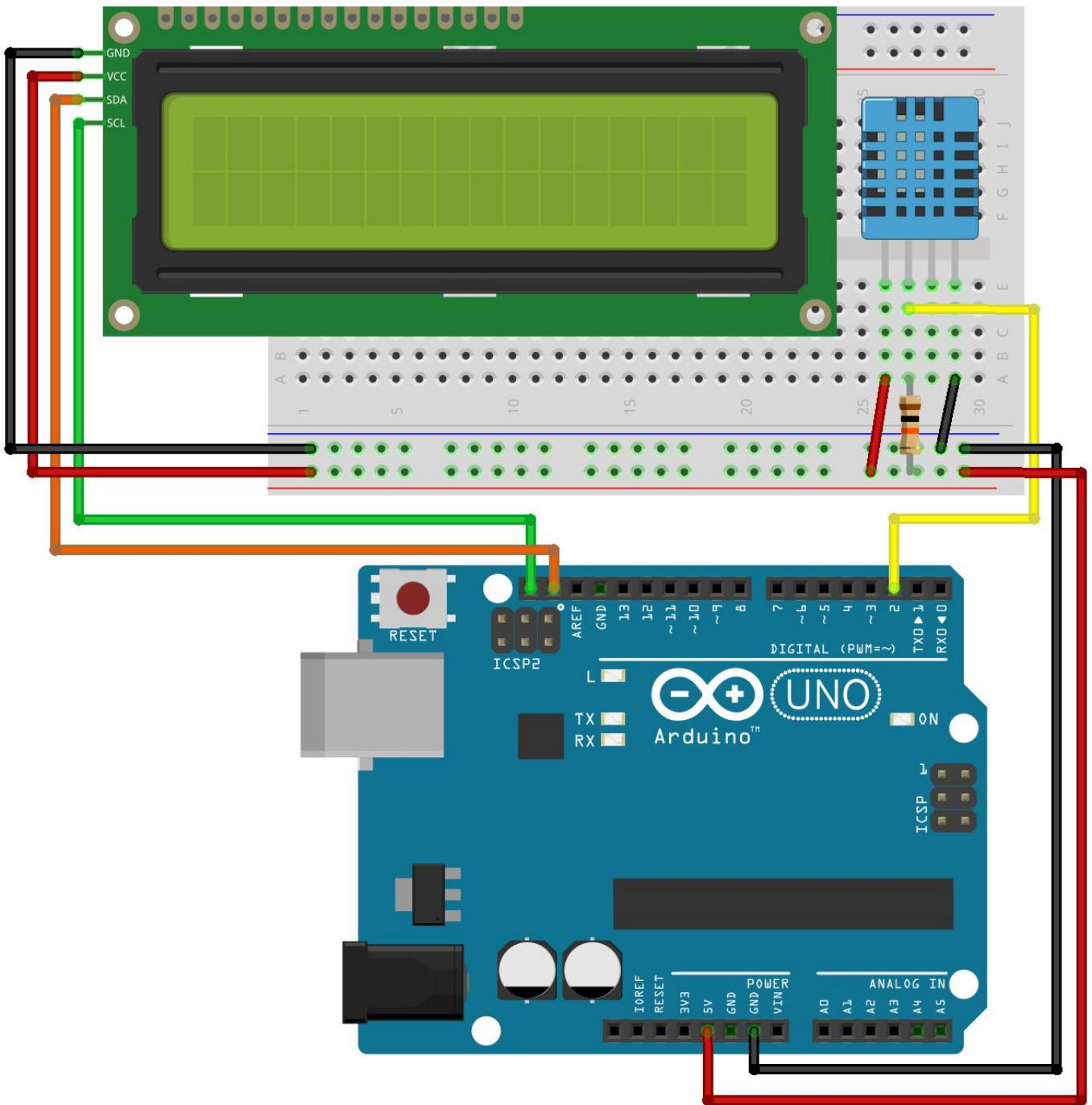
  lcd.setCursor(0,1);
  lcd.print("Vlaga: ");
  lcd.setCursor(7,1);
  lcd.print(temperatura);
  lcd.setCursor(12,1);
  lcd.print(" %");
}
```

Spajanje I2C verzije ekrana

I2C je protokol za serijsku komunikaciju koji koristi jako puno uređaja. Svi takvi uređaji koriste dva specijalna pina na Arduinu: **SDA** (Serial Data) i **SCL** (Serial Clock). Ti pinovi se na većini Arduino pločica nalaze iznad AREF pina, a standardne UNO pločice natpis za te pinove imaju s donje strane, tako da ako ih gledate odozgo nećete vidjeti koje su. Na slici ispod nacrtan je način spajanja LCD ekrana sa I2C sučeljem. LCD ekran će sa donje strane imati natpis koji izvod je koji. Redom, izvodi se spajaju:

1. GND (- napon) spojite sa GND pinom Arduinu, na slici senzor i ekran zajedno dijele GND
2. VCC (+ napon) spojite sa 5V pinom Arduina, na slici senzor i ekran zajedno dijele VCC
3. SDA izvod ekrana spojite na SDA pin Arduina
4. SCL izvod ekrana spojite na SCL pin Arduina

Plavi potencijometar sa stražnje strane ekrana služi za namještanje kontrasta slova na ekranu, pomoću malog odvijača namjestite ga po volji (slova će se prikazati tek kad ga isprogramirate).



Programiranje I2C verzije ekrana

Spajanje I2C verzije ekrana je jednostavno, ali programiranje I2C verzije zahtjeva dodatni korak kako bi uopće bili sigurni da ispis radi dobro. Prvi korak je da postavite **LiquidCrystal I2C** biblioteku¹ na vaše računalo. Biblioteka je dostupna online preko *Sketch -> Include Library -> Manage Libraries...* ili ju možete preuzeti [ovdje](#) pa dodati preko *Sketch -> Include Library -> Add ZIP. Library...*

Nakon toga moramo saznati adresu našeg ekrana. Naime I2C protokol dopušta spajanje više uređaja istovremeno. Kako bi se to uopće izvelo pomoću ukupno dvije žice (SDA i SCL) svaki od uređaja uz podatke šalje i svoju adresu pa Arduino može razlikovati više uređaja koji s njim žele pričati istovremeno. Kako različiti proizvođači koriste različite verzije čipova za I2C koje dodaju na LCD ekran

¹ Ako ste radili vježbu s ekranom na prethodnim radionicama moguće je da već imate instaliranu ovu biblioteku.

(obično je to neka verzija tzv. PCF8574) koristit ćemo program koji će doznati adresu našeg ekrana. **Bez ovog koraka I2C verzija LCD ekrana vrlo vjerojatno neće raditi.**

```
#include <Wire.h>

void setup() {
  Serial.begin(9600);

  while (!Serial)
    {
    }

  Serial.println ();
  Serial.println ("I2C scanner. Scanning ...");
  byte count = 0;

  Wire.begin();
  // ova petlja prolazi kroz moguće adrese i ako dobije neki odgovor
  // ispisuje na kojim adresama se nalaze uređaji
  for (byte i = 8; i < 120; i++)
  {
    Wire.beginTransmission (i);
    if (Wire.endTransmission () == 0)
    {
      Serial.print("Found address: ");
      Serial.print(i, DEC);
      Serial.print(" (0x");
      Serial.print(i, HEX);
      Serial.println(")");
      count++;
      delay (1);
    }
  }
  Serial.println ("Done.");
  Serial.print ("Found ");
  Serial.print (count, DEC);
  Serial.println (" device(s).");
}

void loop() {}
```

Ako je LCD ekran ispravno spojen nakon što učitate gornji program u Arduino trebali bi dobiti nešto poput ove poruke u Serial Monitoru:

```
I2C scanner. Scanning ...
Found address: 63 (0x3F)
Done.
Found 1 device(s).
```

Naglašena heksadecimalna brojka je adresa našeg ekrana, zabilježite ju za sljedeći korak. Kako bi vidjeli neka slova na ekranu napraviti ćemo program koji će ispisati kratku poruku:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // dodaje biblioteku za lcd preko I2C

// postavlja adresu na 0x3F, širinu ekrana na 16 znakova, a visinu na 2 znaka
// ako vaš uređaj ima drugu adresu promijenite ju tako da odgovara vašem ekranu
```



```

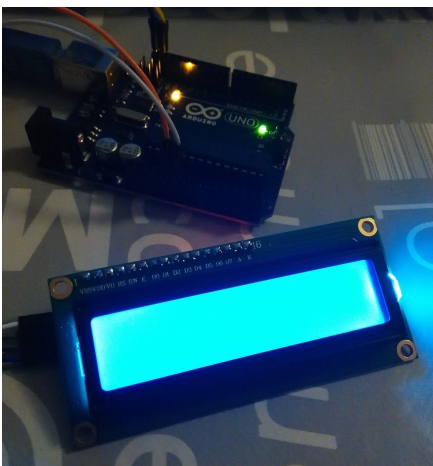
LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup()
{
  lcd.init(); // pokretanje ekrana
  lcd.backlight(); // paljenje pozadinskog svjetla
}

void loop() {
  lcd.print("Radim :)"); // ispis poruke na ekranu
}

```

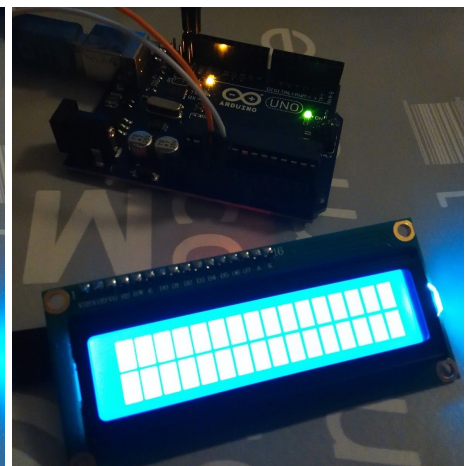
Ako je sve u redu ekran će ispisati poruku. Ako na ekranu vidite bijele kocke ili je sve u pozadinskoj boji (npr. plava) onda je kontrast prevelik ili premali. Uzmite mali odvijač i na stražnoj strani okrećite potencijometar dok ne namjestite slova da budu čitka.



Kontrast je premali, slova se uopće ne vide.



Optimalan kontrast, slova se jasno vide, a pozadina se ne ističe.



Preistaknuta pozadina, ako gledate sa strane vidjet ćete poruku, ali jedva.

Kad je ekran namješten možemo složiti konačni program za meteorološku postaju. Koristit ćemo već postojeći kod za DHT senzor, ali ćemo naredbe za ispis u Serial monitor zamijeniti s onima za ispis na LCD ekran. Biblioteka `LiquidCrystal_I2C.h` ima veći broj funkcija koje u ovom primjeru nećemo koristiti, ali ako vas interesiraju možete pogledati [upute](#) za obični Liquid Crystal. Popis dostupnih naredbi i način korištenja je za većinu funkcija identičan.

```

#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define DHTPIN 2
#define DHTTYPE DHT11

DHT senzor(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup() {
  lcd.init();

```

```

    lcd.backlight();
    senzor.begin();
}

void loop() {
    delay(2000);
    lcd.clear(); // na početku svake petlje počistimo prethodni sadržaj ekrana

    float vlaga = senzor.readHumidity();
    float temperatura = senzor.readTemperature();

    if (isnan(vlaga) || isnan(temperatura)) {
        // setCursor(stupac,red) namješta ispis na točno mjesto, s time da brojanje
        // počinje sa 0,0 - prvi stupac, prvi red
        lcd.setCursor(0,0); // namješta ispis na početak 1. reda
        lcd.print("GRESKA");
        lcd.setCursor(0,1); // namješta ispis na početak 2. reda
        lcd.print("nema podataka");
        return;
    }

    lcd.setCursor(0,0);
    lcd.print("Temp: ");
    lcd.setCursor(6,0);
    lcd.print(temperatura);
    lcd.setCursor(11,0);
    lcd.print(" *C");

    lcd.setCursor(0,1);
    lcd.print("Vlaga: ");
    lcd.setCursor(7,1);
    lcd.print(temperatura);
    lcd.setCursor(12,1);
    lcd.print(" %");
}

```

Priprema za odlazak van

Jednom kad je program spremljen na Arduinu on se može izvršavati bez toga da je uključen u računalo, ali mu obavezno treba izvor energije. Sve Arduino UNO pločice dolaze sa dodatnim utorom za vanjski izvor napajanja, pa u njega možete uključiti bateriju.

Eksperimentalne pločice, su kao što im i ime kaže, napravljene za testiranje ideja, tj. toga kako bi konačni uređaj trebao izgledati. **Računajte da se u nošenju ovako spojene meteorološke stanice uokolo lako može dogoditi da se neka od žica odspoji.** Obavezno napravite kućište za vašu meteo postaju. Za to možete žrtvovati neku manju plastičnu posudu s poklopcem. Potrebno je izrezati otvor za senzor ako ostatak elemenata želimo držati zaštićen, a ako posuda nije prozirna onda vam treba i otvor za LCD ekran.

Zgodan dodatak je i prekidač za paljenje/gašenje. Nažalost većina početničkih setova za Arduino ih ne uključuje, ali nisu skupi, a pritom pomažu u tome da lakše radite s postajom. Bez ovog prekidača paljenje i gašenje se radi uključivanjem i isključivanjem baterije.

Dodati slike svega spojenog na kraju...