



# **ARDUINO KROZ JEDNOSTAVNE PRIMJERE**

*- pripreme za natjecanja -*

## **PRIPREMA 8 KORIŠTENJE TEKSTUALNOG LCD EKRANA**

Paolo Zenzerović, mag. ing. el.

Zagreb, 2014.

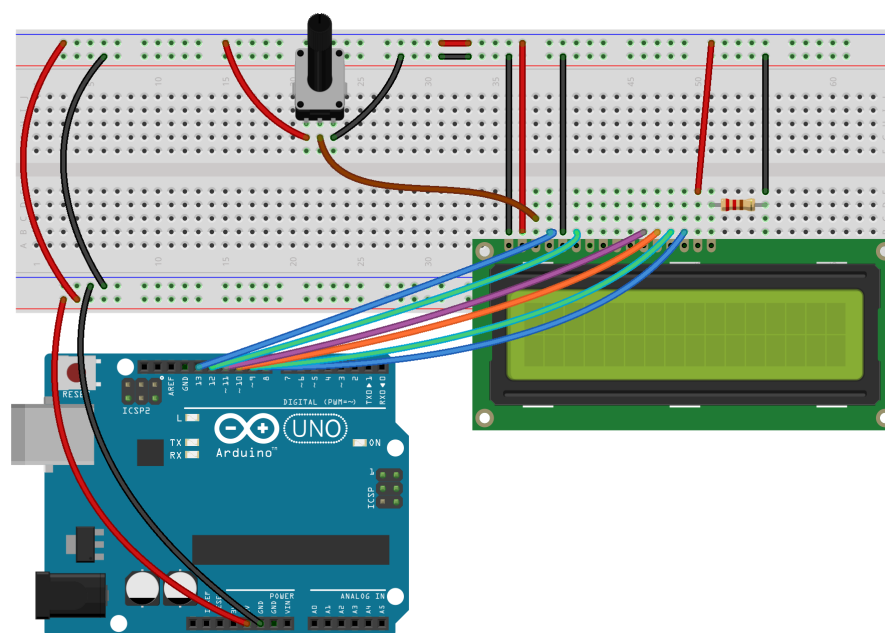
## TEKSTUALNI LCD EKRAN

Pri korištenju mikrokontrolera za upravljanje raznim procesima često postoji potreba za prikazivanjem nekih podataka korisniku. Do sada smo naučili kako možemo neko stanje prikazivati uključivanjem i isključivanjem LED dioda te ispisom podataka na računalo putem serijske veze. Međutim, ponekad postoji potreba da korisnik dobije neke podatke a da u našem sustavu ne postoji računalo. Tada možemo iskoristiti tekstualni LCD ekran kako bi ispisali neke kratke poruke.

Tekstualni LCD ekrani su vrlo jednostavni za korištenje. Postoji dvije vrste takvih ekrana; ekrani s paralelnom komunikacijom i ekrani sa serijskom komunikacijom. Mi ćemo koristiti one s paralelnom komunikacijom jer su najrašireniji i najdostupniji.

Tekstualni LCD ekrani se proizvode u nekoliko veličina. Obično imaju jedan, dva ili četiri retka. U svaki redak obično stanje osam, šesnaest, dvadeset ili četrdeset znakova. Kod obilježavanja veličine ekrana prvo se navodi broj znakova u retku a onda broj redaka – npr 16x2 ekran ima mogućnost ispisa šesnaest znakova u dva retka.

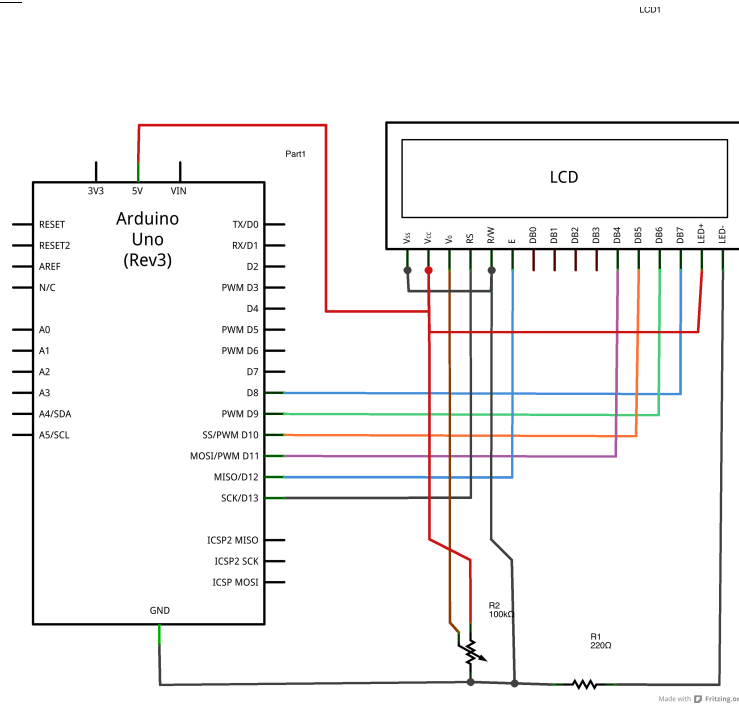
Pogledajmo kako se ekran spaja na mikrokontroler:



Made with  Fritzing.org

Potenciometar u spoju na slici služi za reguliranje kontrasta LCD ekrana, dok otpornik služi za ograničavanje struje LED diode za pozadinsko osvetljenje.

## Električna shema:



Zadatak 1: Pokušajmo ispisati neki pozdravni tekst na LCD ekranu – npr. “Dobar dan!”

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Dobar dan!");
}

void loop() {
}
```

Na samom početku programa vidimo novu vrstu naredbe `#include`. Ona govori programskom jeziku da ćemo koristiti neke naredbe koje nisu odmah ugrađene u programski jezik nego su definirane u datoteci `LiquidCrystal.h`. Takva datoteka naziva se knjižnica.

Naredba `LiquidCrystal lcd(13, 12, 11, 10, 9, 8)` definira kako je spojen sam LCD modul sa mikrokotrolerom. Potrebno je kao argumente funkcije nabrojiti redom sljedeće izvode: RS, E, D4, D5, D6, D7 tj. brojeve izvoda mikrokotrolera na koje su oni spojeni.

Naredba `lcd.begin` definira broj redaka i stupaca ekrana, dok naredba `lcd.print` ispisuje poruku na ekranu.

Zadatak 2: Mjerimo vrijeme – U prvi redak ispišimo poruku “Vrijeme:” a u drugi redak ispišimo koliko je vremena proteklo od pokretanja mikrokontrolera. Prisjetite se posljednje pripreme i funkcije `millis`!

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Vrijeme:");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

Prvi dio programa je gotovo isti kao i u prethodnom primjeru. U drugom dijelu imamo jednu novu naredbu – `lcd.setCursor(0, 1)`. Ona nam služi za pozicioniranje kursora za ispis teksta na LCD ekranu. Prva znamenka (0) označava da ćemo se pozicionirati na prvo mjesto unutar retka, dok druga znamenka (1) označava da ćemo se pozicionirati u drugi redak.

Zadatci: Modificirajte posljednja dva primjera iz prethodne pripreme tako da se podatci ne ispisuju na računalu nego na LCD ekranu.

Dodatni zadatak: Napravite štopericu.