



ARDUINO KROZ JEDNOSTAVNE PRIMJERE

- pripreme za natjecanja -

PRIPREMA 5 - 2015 DVOSMJERNA SERIJSKA KOMUNIKACIJA

Paolo Zenzerović, mag. ing. el.

Zagreb, 2015.

DVOSMJERNA SERIJSKA KOMUNIKACIJA

Već smo u pripremama prošle godine naučili koristiti serijsku komunikaciju između mikrokontrolera i računala, ali smo ju dosad koristili samo jednosmjerno – mikrokontroler je slao podatke računalu. Sada ćemo pogledati kako možemo poslati podatke s računala na mikrokontroler.

Zadatak 1: Napišite program koji će uključivati i isključivati svjetleću diodu spojenu na izvod broj 13 na Arduino pločici pomoću naredbi poslanih serijskim putem s računala. Kada s računala pošaljemo slovo “H” neka se dioda uključi, a kada pošaljemo slovo “L” neka se dioda isključi. Također, neka se na računalo povratno javi informacija je li dioda uključena ili isključena. Ukoliko mikrokontroler primi neku drugu poruku neka javi serijskim putem na računalo da je ta naredba nepoznata.

```
int primljeno = 0;

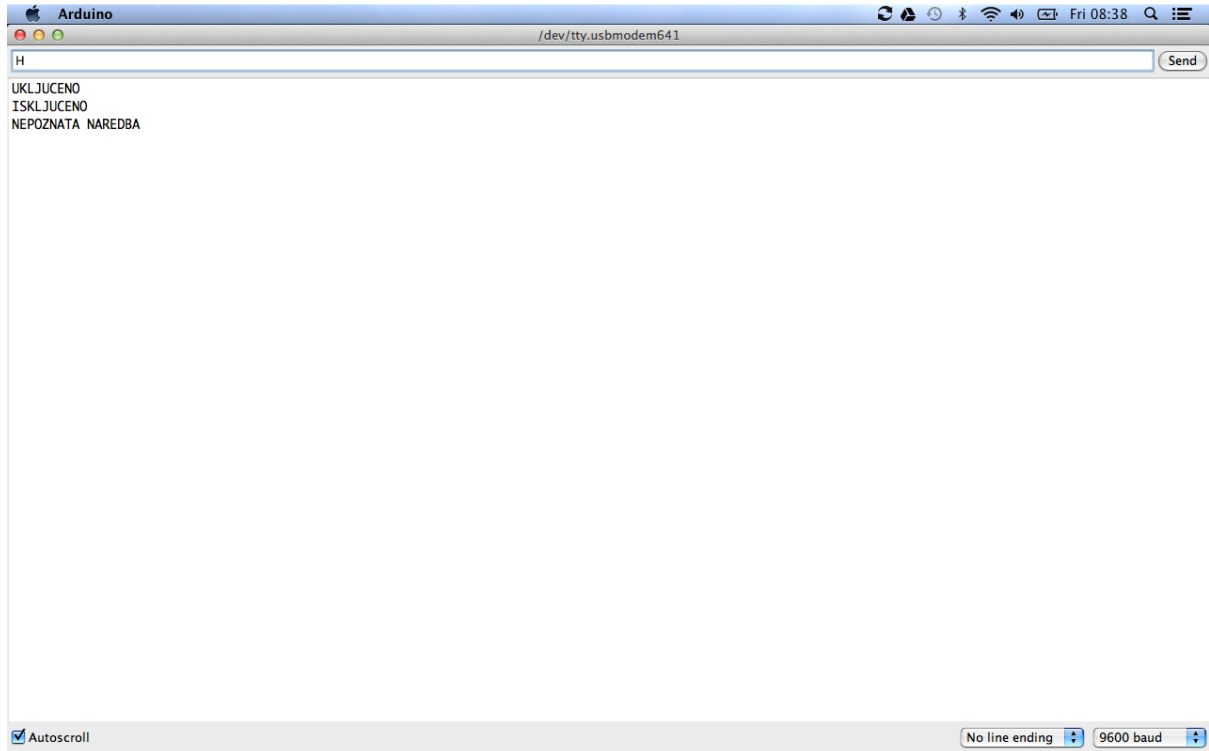
void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);
}

void loop() {
  while (Serial.available() > 0) {
    primljeno = Serial.read();

    if (primljeno == 'H') {
      digitalWrite(13, HIGH);
      Serial.println("UKLJUCENO");
    }
    else if (primljeno == 'L') {
      digitalWrite(13, LOW);
      Serial.println("ISKLJUCENO");
    }
    else {
      Serial.println("NEPOZNATA NAREDBA");
    }
  }
}
```

Kao i do sada pri korištenju serijske komunikacije moramo ju inicijalizirati naredbom `Serial.begin`. Unutar `loop` dijela programa dodajemo naredbu `Serial.available` koja nam govori je li mikrokontroler primio podatak putem serijske veze. Naime, mikrokontroler prima podatke neovisno o našem programu i pohranjuje ih u jednu specijaliziranu unutarnju memoriju. Kako bismo pročitali što je mikrokontroler primio koristimo naredbu `Serial.read`. Ova nam naredba daje podatak koji je spremljen u toj memoriji serijske komunikacije. U našem smo programu tu vrijednost spremili u varijablu podatak. Nakon toga sa nizom `if-else if-else` naredbi provjeravamo je li primljena poruka “H”, “L” ili neka druga te prema tome uključujemo, isključujemo svjetleću diodu i šaljemo povratne poruke na računalo.

Prebacite program u mikrokontroler i pokrenite Serial monitor alat. Kako biste poslali poruku na mikrokontroler u gornje polje za upis upišite poruku i stisnite enter na tipkovnici. Tada će vaša poruka biti poslana. Pokušajte poslati poruke “H”, “L” i neku drugu poruku kako biste isprobali sve mogućnosti programa.



VAŽNO: Kada računalo šalje poruku mikrokontroleru naizgled ono šalje tekst odnosno slovo – recimo H. Međutim ono što se zapravo zbiva je da računalo šalje ASCII kod od tog slova na mikrokontroler. Iz tog smo razloga varijablu primljeno definirali kao integer. Kako nebi morali znati ASCII vrijednosti pojedinih znakova kod privjere odgovara li poslana poruka nekom znaku taj znak stavljamo pod jednostruke navodnike i program za nas sam pretvara slova u njihov ASCII kod kod kompajliranja programa.

Ovaj pristup dobro funkcionira dok god se naredba koju smo poslali sastoji od samo jednog znaka. Pokušajte u naredbu upisati 2 znaka i poslati to mikrokontroleru. Što se zbiva?

Zadatak 2: Napravite isto što i u prvo zadatku ali neka naredbe budu "UK" za uključivanje i "ISK" za isključivanje.

```
String primljeno;

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);
}

void loop() {
  while (Serial.available() > 0) {
    primljeno = Serial.readStringUntil('\n');

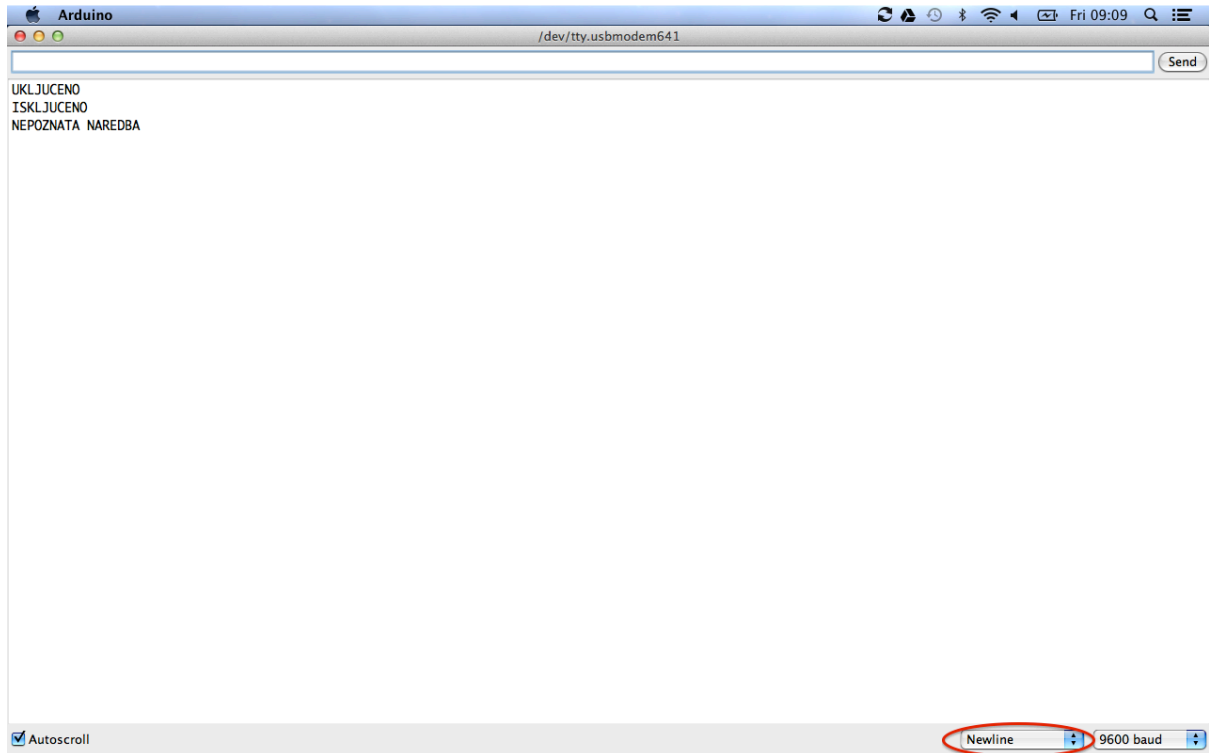
    if (primljeno == "UK") {
      digitalWrite(13, HIGH);
      Serial.println("UKLJUCENO");
    }
    else if (primljeno == "ISK") {
      digitalWrite(13, LOW);
      Serial.println("ISKLJUCENO");
    }
    else {
      Serial.println("NEPOZNATA NAREDBA");
    }
  }
}
```

Varijabla primljeno više nije integer tipa nego string tipa što znači da ćemo primiti tekst a ne broj.

Umjesto Serial.read naredbe iskoristili smo Serial.readStringUntil naredbu koja čita dolazni niz znakova sve dok ne naiđe na znak koji je definiran u zagradi. Mi smo za to iskoristili posebni znak '\n' koji označava novi red.

Važno je da kod slanja u prozoru Serial monitor alata postavimo u donjem desnom kutu da se nakon pritiska na tipku enter za slanje pošalje i znak za novi red – Newline. Primjena je zaokružena na donjoj slici.

Obratite pažnju gdje koristimo jednostruke a gdje dvostruke navodnike!



Zadatak 3: Napišite program koji će s računala na mikrokontroler primiti neki broj a na računalo će vratiti taj isti broj uvećan za 2.

```
String primljeno;

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);
}

void loop() {
  while (Serial.available() > 0) {
    primljeno = Serial.readStringUntil('\n');
    Serial.println(primljeno.toInt()+2);
  }
}
```

Ovaj primjer pokazuje nam kako na jednostavan način možemo primiti neku brojčanu vrijednost s računala. Kao i u prethodnom zadatku s računala smo pročitali string a potom smo ga korištenjem `.toInt` naredbe pretvorili u cijelobrojni podatak, uvećali za 2 i poslali natrag na računalo.

Dodatni zadatak: Na izvod mikrokontrolera s mogućnošću pulsno-širinske modulacije spojite jednu svjetleću diodu. Napišite program kojim ćete pomoću računala upravljati svjetlošću diode. S računala šaljite brojeve od 0 do 255 i primite ih na mikrokontroler. Dolazne podatke iskoristite kako biste pojačavali odnosno smanjivali intenzitet svijetlosti diode.