HRVATSKA ZAJEDNICA TEHNIČKE KULTURE

Croatian Society for Educational Technology

ARDUINO KROZ JEDNOSTAVNE PRIMJERE

- pripreme za natjecanja -

PRIPREMA 6
SERIJSKA KOMUNIKACIJA S RAČUNALOM I ANALOGNI ULAZI

Paolo Zenzerović, mag. ing. el.

Zagreb, 2014.

SERIJSKA KOMUNIKACIJA S RAČUNALOM

Serijska komunikacija je tip komunikacije gdje prijemnik i predajnik razmjenjuju podatke putem jednog para signala. Mikrokontroleri na Arduino i Croduino pločici spremni su za serijsku komunikaciju s računalom. Komunikacija se odvija putem istog USB kabla koji se koristi za programiranje mikrokontrolera. Podatci između računala i mikrokontrolera mogu teći u oba smjera – mikrokontroler može slati podatke računalu i računalo može slati podatke mikrokontroleru. Mi ćemo se orijentirati na ovaj prvi slučaj.

Sljedeći sketch prikazuje način slanja podataka na računalo:

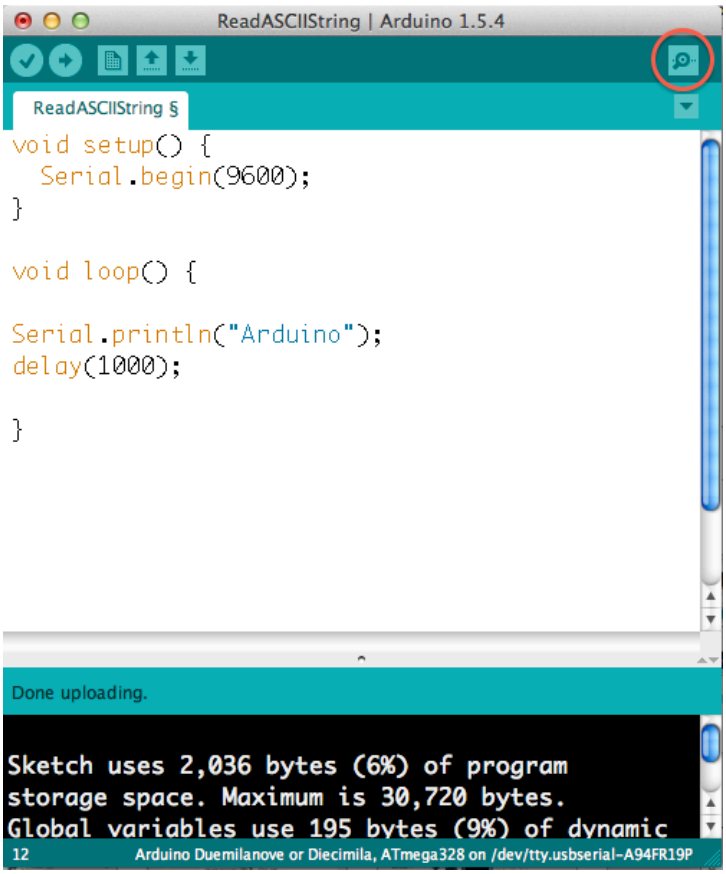
```
void setup() {
  Serial.begin(9600);
}

void loop() {
Serial.println("Arduino");
delay(1000);
}
```

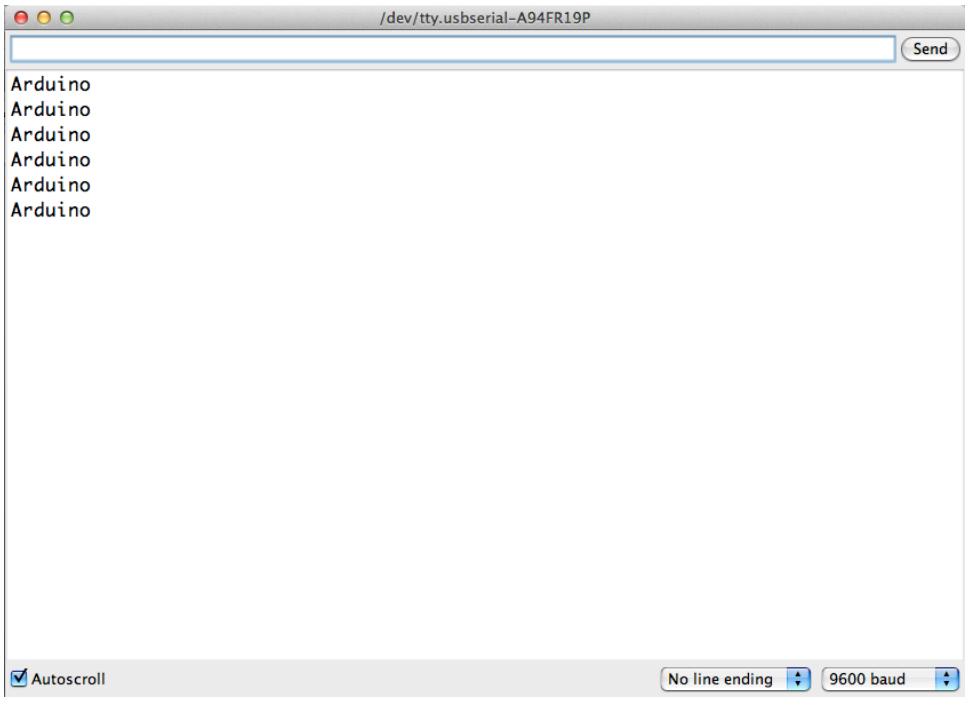
Kako bi započeli serijsku komunikaciju i postavili njezinu brzinu iskoristili smo naredbu Serial.begin(9600). Broj 9600 označava brzinu serijske komunikacije odnosno govori o tome koliko će bitova informacije biti poslano unutar jedne sekunde (bps – bits per second). Za naše potrebe uvijek možete koristiti brzinu od 9600 bps.

Unutar glavne petlje programa koristili smo Serial.println naredbu kako bismo računalu poslali neke podatke. U našem slučaju poslali smo string “Arduino”. Taj string će se svake sekunde poslati računalu od mikrokontrolera.

Kako bi na računalu mogli pogledati podatke koji stižu u Arduino programsko okruženje ugrađen je alat za pregled podataka serijske komunikacije. Kako bi ga pokrenuli potrebno je kliknuti na ikonicu povećala u gornjem desnom uglu Arduino programskog okruženja, kako je naznačeno na sljedećoj slici.



Nakon pritiska na gornju ikonu otvara se novi prozor u kojem možemo vidjeti podatke koji dolaze od mikrokontrolera. Prozor s podatcima je prikazan na sljedećoj slici:



Korištenjem funkcije Serial.println nakon svakog poslanog podatka ispisuje se znak za novi red pa će sljedeći poslani podatak biti prikazan u retku ispod prethodnog. Ako želimo više podataka prikazati u istom retku možemo iskoristiti naredbu Serial.print koja ne šalje znak za novi red.

Osim statičkih podataka kao što je bila riječ "Arduino" u prethodnom primjeru možemo na računalo slat ii podatke koji se mijenjaju recimo iznos neke varijable, očitanje nekog senzora itd.

Pogledajmo jednostavni primjer:

```
int brojac;

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

void loop() {

  for (brojac=0; brojac<10; brojac++){
     Serial.print(brojac);
     delay(500);
    }
  Serial.println();
}
```

Cilj primjera bio je prikazati kako računalu možemo poslati promjenjive podatke. Na početku programa stvorili smo jednu varijablu koju sm nazvali brojac. Varijablu brojač iskoristiti ćemo u for petlji kako bi mogli brojiti od 0 do 9. For petlja nam služi kako bismo dio koda u petlji mogli izvršiti određeni broj puta. For petlja prima tri argumenta koje je potrebno upisati u zagrade nakon ključne riječi for a to su:

- početno stanje brojača: brojac=0
- uvijet do kada će se petlja izvršavati: brojac<10
- što je potrebno učiniti na kraju svake petlje: brojac++

Da pojasnimo, na početku petlje brojač smo stavili na vrijednost nula (brojac=0). U drugom argumentu rekli smo petlji da se izvršava sve dok je brojač manji od 10 (brojac<10), a u tećem argumentu zadali smo da se na kraju izvršenja svakog ciklusa petlje na prethodnu vrijednost brojača nadoda jedan (brojac++) kako bi postigli efekt brojenja.

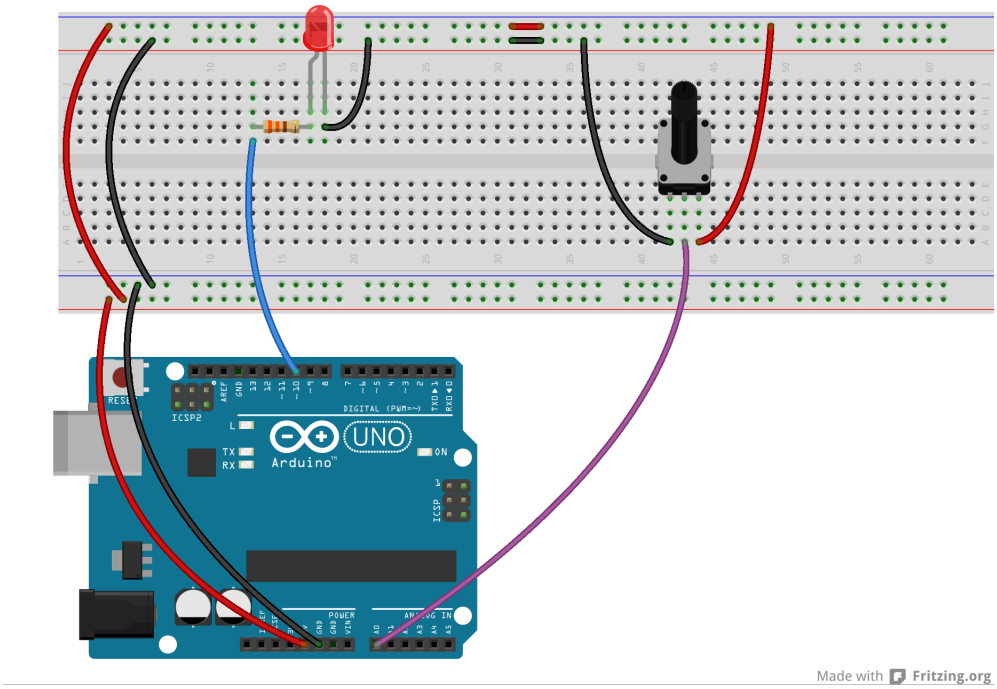
Naredbom Serial.print(brojac) unutar petlje poslali smo računalu brojeve 0, 1, 2, 3 itd do 9. Naredbom Serial.println() koja nema nikakvog argumenta poslali smo računalu samo znak za novi red.

Rezultat ispisa je:

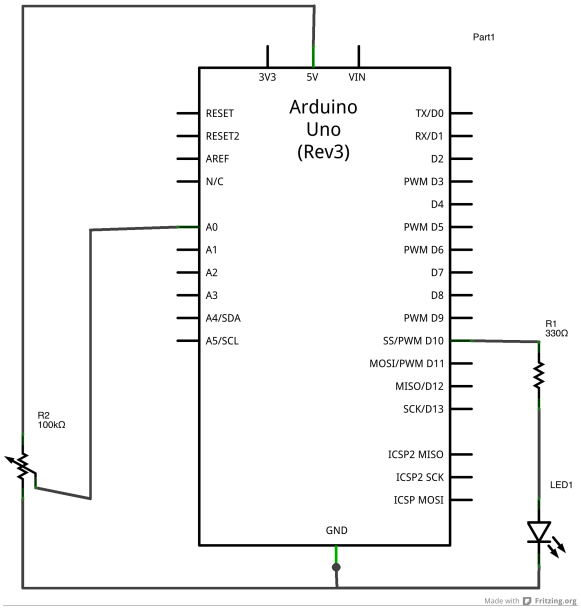
0123456789
0123456789
0123 itd.

Serijska veza nam može biti izuzetno korisna za prikaz očitanja stanja različitih senzora ili očitanja analognih ulaza. Spojimo primjer kao iz pripreme 4 kako bi pokazali ovu funkcionalnost:

Grafički prikaz spajanja na prototipnoj pločici:



Električna shema:



Zadatak 1: Očitajte stanje analognog ulaza na koji je spojen potenciometar i serijskim putem ga pošaljite na računalo kako biste mogli vidjeti kolika je očitana vrijednost.

```
int potenciometar = A0;
int analogna_vrijednost;

void setup() {
  Serial.begin(9600);
  pinMode(potenciometar, INPUT);
}

void loop() {
  analogna_vrijednost=analogRead(potenciometar);
  Serial.println(analogna_vrijednost);
  delay(500);
}
```

Gornji program očitava analognu vrijednost koju dobivamo sa analognog ulaza A0 na koji je spojen potenciometar kao naponsko djelilo. Nakon očitanja te vrijednosti istu šaljemo na računalo putem naredbe Serial.println(analogna_vrijednost);

Ovo nam omogućava da na vrlo jednostavan način očitamo stanja raznih senzora.

Kod korištenja analognih ulaza za očitanje senzora koji nam daju neku vrijednost od 0 do 5V često želimo uključiti ili isključiti uređaj samo onda kada očitana vrijednost prijeđe ispod ili iznad određene razine. Naprimjer kada mjerimo količinu svijetlosti pomoću fotootpornika želimo uključiti rasvjetno tijelo onda kada količina svijetlosti padne ispod određena razine, ili kada mjerimo temperaturu pa želimo uključiti grijanje samo onda kada je temperatura ispod određene vrijednosti.

Zadatak 2: Spojite sklop kao na prethodnoj stranici. Napravite program koji će uključiti LED diodu kada je potenciometar zakrenut više od pola u desnu stranu. Kada se potenciometar vrati u lijevu stranu i prijeđe graničnu vrijednost LED dioda se isključuje.

Za početak iskoristite prethodni primjer da odredite graničnu vrijednost. Spojite potenciometar i zakrećite ga ten a ekranu očitajte vrijednost koju poprima na sredini. To bi trebala biti vrijednost oko 512.

Ja sam za svoj sklop odredio graničnu vrijednost od 515.

Potom zakrenite potenciometar u desnu stranu. U ovisnisti o tome kako ste spojili potenciometar vrijednosti se mogu povećavati ili smanjivati.

U mom slučaju vrijednosti se smanjuju. To znači da u rješenju zadatka moram provjeravati je li trenutna vrijednost analognog ulaza **manja** od 515 te u tom slučaju uključiti diodu.

Ako se vrijednosti za vaš spoj povećavaju onda u rješenju zadatka morate provjeravati je li trenutna vrijednost analognog ulaza **veća** od 515 te u tom slučaju uključiti diodu.

Rješenje za prvi slučaj:

```
int potenciometar = A0;
int analogna_vrijednost;
int led = 10;

void setup() {
  Serial.begin(9600);
  pinMode(potenciometar, INPUT);
  pinMode(led, OUTPUT);
}

void loop() {
  analogna_vrijednost=analogRead(potenciometar);
  if (analogna_vrijednost<515) {
  digitalWrite(led, HIGH);
  } else {
  digitalWrite(led, LOW);
  }
  delay(100);
}
```

Rješenje za drugi slučaj:

```
int potenciometar = A0;
int analogna_vrijednost;
int led = 10;

void setup() {
  Serial.begin(9600);
  pinMode(potenciometar, INPUT);
  pinMode(led, OUTPUT);
}

void loop() {
  analogna_vrijednost=analogRead(potenciometar);
  if (analogna_vrijednost>515) {
  digitalWrite(led, HIGH);
  } else {
  digitalWrite(led, LOW);
  }
  delay(100);
}
```

Dodatni zadatak: Umjesto potenciometra za prethodni zadatak iskoristite fotootpornik spojen u spoj naponskog djelila zajedno s još jednim otpornikom. Kada fotootpornik zamračite neka se LED diode uključi a kada je fotootpornik osvjetljen neka se LED dioda isključi. Fotootpornik možete spojiti kako je prikazano na sljedećoj slici: