

3

Osnove programskog jezika C

Za programiranje Arduina koristi se programski jezik C. U ovom poglavlju upoznaćete njegove osnove. To što ovde naučite koristićete u svakom programu koji ćete napisati za Arduino. Da biste izvukli najviše od Arduina, morate razumeti ove osnove.

Programiranje

Nije neuobičajeno da ljudi govore više od jednog jezika. U stvari, što više govornih jezika učite, to vam učenje postaje sve lakše pošto počinjete da pronalazite zajedničke šablone u gramatici i vokabularu. Isto važi i za programske jezike. Prema tome, ako ste već koristili neki programski jezik, brzo ćete shvatiti i C.

Prednost je to što je fond reči programskog jezika mnogo manji od fonda reči govornog jezika, a pošto ga pišete a ne govorite, uvek možete pri ruci imati rečnik kada treba nešto da potražite. Uz to, gramatika i sintaksa programskog jezika krajnje su pravilni, pa kada shvatite nekoliko osnovnih koncepata, brže učenje će vam postati sasvim prirodno.

O programu (u svetu Arduina on se naziva *sketch* – skica) najbolje je razmišljati kao o listi instrukcija koje treba izvršiti po redosledu kojim su napisane. Na primer, pretpostavite da treba da napišete sledeće:

```
digitalWrite(13, HIGH);  
delay(500);  
digitalWrite(13, LOW);
```

Svaki od ova tri reda obavio bi neki posao. Prvi red bi podesio izlaz iz pina 13 na HIGH. To je pin na koji je povezana LED dioda ugrađena na Arduino pločicu, što znači da bi se – u ovom trenutku – LED dioda uključila. Drugi red bi samo čekao 500 milisekundi (pola sekunde) a zatim bi treći red isključio LED diodu. Prema tome, pomoću ova tri reda postigao bi se cilj da LED dioda trepne jedanput.

Već ste videli zbunjajući niz čudno upotrebljenih znakova interpunkcije i engleskih reči između kojih nema razmaka. Frustracija mnogih novih programera mogla bi se izraziti sledećim rečima: „Znam šta hoću da uradim, ali ne znam šta treba da napišem!“ Ne brinite, sve ćemo to uskoro objasniti.

Pozabavimo se prvo interpunkcijom i načinom formiranja reči. I jedno i drugo spadaju u sintaksu jezika. U većini jezika morate biti krajnje precizni sa sintaksom, a jedno od glavnih pravila je da imena stvari moraju biti jedna reč – to jest, ona ne mogu sadržati razmake. Znači, **digitalWrite** je ime nečega. To je ime ugrađene funkcije, engl. *built-in function* (o funkcijama će biti reči kasnije) koja će obaviti posao podešavanja izlaznog pina na Arduino pločici. Ne samo što morate izbegavati razmake u imenima, već morate voditi računa i o velikim i malim slovima. Prema tome, morate napisati **digitalWrite**, a ne **DigitalWrite** niti **Digitalwrite**.

Funkcija **digitalWrite** mora da zna koji pin da podesi i da li da ga postavi na HIGH ili na LOW. Ta dva delića informacije zovu se *argumenti*, za koje se kaže da se *prosleđuju* funkciji kada se ona *pozove*. Argumenti funkcije moraju biti navedeni u zagradama i razdvojeni zarezima.

Po konvenciji, otvorena zagrada se piše odmah iza poslednjeg slova imena funkcije, a iza zareza koji razdvaja sledeći parametar umeće se razmak. Međutim, unutar zagrada možete dodavati razmake po želji.

Ako funkcija ima samo jedan argument, zarez nije potreban.

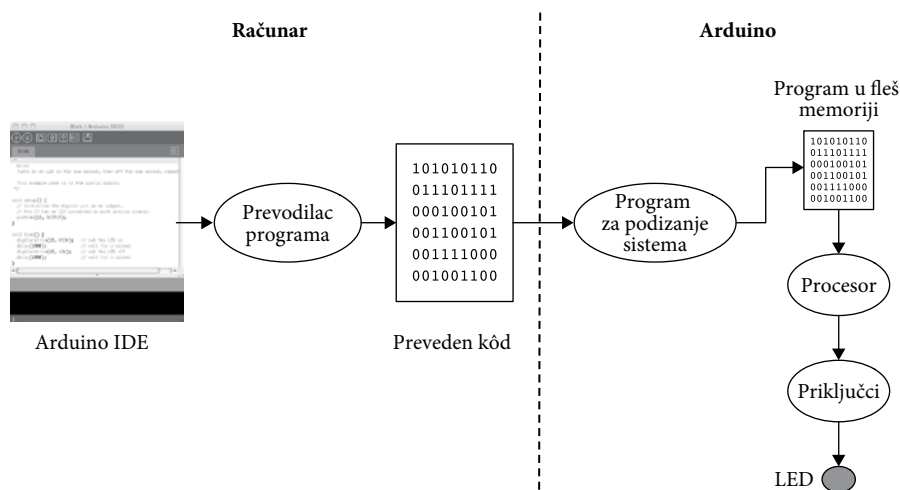
Zapazite da se svaki red završava znakom tačka i zarez. Bilo bi logičnije da su na tim mestima tačke, zato što znak tačka i zarez označava kraj jedne komande – što je slično kraju rečenice.

U narednom odeljku saznaćete više o tome šta se dešava kada pritisnete dugme Upload u Arduinovom integrisanom razvojnom okruženju (engl. *integrated development environment*, IDE). Tada ćete moći da isprobate i nekoliko primera.

Šta je programski jezik?

Možda malo iznenađuje to što smo stigli do poglavlja 3 u knjizi o programiranju a da nismo tačno definisali šta je programski jezik. Možemo da prepoznamo program za Arduino i verovatno imamo grubu predstavu o tome šta on radi, ali moramo malo dublje sagledati kako se programski kôd pretvara od nekih reči napisanih na strani u nešto što obavlja konkretan posao kao što je uključivanje i isključivanje LED diode.

Na slici 3-1 sažeto je prikazan proces koji se odvija od upisivanja koda u Arduinovo integrisano razvojno okruženje do izvršavanja programa na pločici.



Slika 3-1 Od koda do pločice.

Kada pritisnete dugme Upload u Arduinovom razvojnom okruženju, pokreće se niz događaja čiji je rezultat instaliranje vašeg programa na Arduino i njegovo izvršavanje. Znači, cela operacija se ne svodi samo na precacivanje teksta koji ste otkucali u editoru na Arduino pločicu.

Prvi korak je *kompiliranje*, tj. *prevođenje* koda koji ste napisali u mašinski kôd – binarni jezik koji Arduino razume. Ako pritisnete dugme Verify u Arduinovom razvojnom okruženju, kompajler će pokušati da prevedu kôd ali ga neće tada poslati u Arduino IDE. Istovremeno s prevođenjem, proverava se i da li je kôd napisan u skladu s pravilima jezika C.