UVOD

Pločice Arduino obezbeđuju primenu jeftine i jednostavne tehnologije za realizovanje projekata zasnovanih na mikrokontrolerima. Uz malo elektronike, Arduino vam može obavljati najrazličitije poslove: od regulisanja svetlosnih efekata na umetničkoj instalaciji do upravljanja snagom solarnog energetskog sistema.

Ima mnogo knjiga fokusiranih na projekte koje vam pokazuju kako da povežete razne stvari s pločicom Arduino, uključujući i moju knjigu *30 Arduino Projects for the Evil Genius*. Međutim, knjiga koja je pred vama bavi se – pre svega – programiranjem Arduina.

Iz ove knjige ćete saznati kako da programiranje Arduina učinite jednostavnim i zabavnim, izbegavajući probleme s nekooperativnim kodom koji tako često opterećuju projekte. Kroz proces programiranja Arduina prolazićete korak po korak, počevši od osnova programskog jezika C – jezika koji koriste pločice Arduino.

Šta je Arduino?

Arduino je mala mikrokontrolerska pločica sa univerzalnom serijskom magistralom (engl. *universal serial bus*, USB) na koju povezujete svoj računar, i nizom utičnica u koje se mogu priključiti spoljne elektronske komponente kao što su motori, releji, svetlosni senzori, laserske diode, zvučnici, mikrofoni i mnoge druge. One se mogu napajati na razne načine: s računara – preko USB priključka, pomoću baterije od 9 V ili preko zasebnog strujnog izvora. Svim komponentama može se upravljati preko računara ili tako što se one programiraju pomoću računara a zatim odvoje od njega i puste da rade autonomno.

Arduino je projektovan kao hardver otvorenog koda (engl. *open source*), što znači da svako može praviti pločice kompatibilne sa Arduinom. Takva konkurencija je dovela do niskih cena i najrazličitijih varijanti „standardnih“ pločica.

Uz osnovne Arduino pločice koriste se i dodatne kartice – proširenja ili „štitovi“ (engl. *shield boards*) – koje se mogu montirati povrh Arduino pločice.

Softver za programiranje Arduina lako se koristi i dostupan je besplatno za Windows, Mac i Linux računare.

Šta će mi trebati?

Ova knjiga je namenjena početnicima, ali će biti korisna i onima koji su radili s Arduinom neko vreme i žele da nauče više o programiranju Arduina ili da bolje shvate osnove. Opisana je upotreba pločice Arduino Uno, ali će gotovo sav kôd – neizmenjen – raditi na svim modelima i varijantama Arduina.

Ne morate imati nikakvog programerskog iskustva niti biti tehnički obrazovani, a za vežbe iz knjige nije potrebno lemljenje. Treba vam samo želja da nešto napravite.

Ukoliko želite da izvučete maksimum iz knjige i da izvedete neke od opisanih eksperimenata, korisno je da pri ruci imate sledeće:

- Manju količinu pune žice
- Jeftin digitalni multimetar

I jedno i drugo možete povoljno nabaviti u prodavnicama sa elektronikom za hobiste (npr. Adafruit, Sparkfun i sl.). Naravno, trebaće vam i pločica Arduino Uno.

Ako želite da odete korak dalje i da eksperimentišete sa displejima i mrežnim vezama, moraćete da kupite i dodatne kartice (štitove) u nekoj prodavnici na vebu. Detalje o tome pronaći ćete u poglavljima 9 i 10.

Organizacija knjige

Struktura ove knjige će vam omogućiti da brzo počnete s radom i da kasnije postepeno nadograđujete već stečeno znanje. Međutim, možete i da preskočite ili samo ovlaš pregledate neka od uvodnih poglavlja kako biste pronašli odgovarajući nivo za sebe.

Knjiga sadrži sledeća poglavlja:

- **Poglavlje 1: Ovo je Arduino** Kao uvod u Arduino hardver, ovo poglavlje opisuje njegove mogućnosti i različite vrste dostupnih Arduino pločica.
- **Poglavlje 2: Početak rada** Ovde obavljate svoje prve eksperimente sa Arduino pločicom: instaliranje softvera, napajanje i učitavanje prvog programa (engl. *sketch*).
- **Poglavlje 3: Osnove programskog jezika C** U ovom poglavlju obrađene su osnove jezika C; za one koji nemaju nikakvog iskustva u programiranju, ovo poglavlje predstavlja i uvod u opšte aspekte programiranja.
- **Poglavlje 4: Funkcije** Objašnjava ključni koncept korišćenja i pisanja funkcija u programima za Arduino. Ti programi su predstavljeni detaljno, sa izvršivim primerima koda.
- **Poglavlje 5: Nizovi i znakovni nizovi** Ovde ćete naučiti kako da pravite i koristite strukture podataka složenije od jednostavnih celobrojnih promenljivih. Primer projekta s Morzeovom azbukom razvijamo postupno da bismo ilustrovali koncepte objašnjene u tekstu.
- **Poglavlje 6: Ulaz i izlaz** Saznaćete kako da u svojim programima koristite digitalne i analogne ulaze i izlaze na Arduino pločici. Multimetar će vam pokazati šta se događa na Arduinovim ulazno/izlaznim priključcima.
- **Poglavlje 7: Standardna biblioteka za Arduino** U ovom poglavlju je objašnjeno kako da iskoristite funkcije iz standardne biblioteke za Arduino pločice.
- **Poglavlje 8: Skladištenje podataka** Ovde ćete saznati kako da pišete programe koji mogu da sačuvaju podatke u EEPROM memoriji. EEPROM memorija (engl. *electrically erasable read-only memory*) služi samo za čitanje i može se obrisati elektronskim putem. Naučićete i kako se koristi Arduinova ugrađena fleš memorija.
- **Poglavlje 9: Displeji** U ovom poglavlju naučićete kako da povežete Arduino pločicu sa displejima i napravite jednostavnu USB tablu za poruke (engl. *message board*).

- **Poglavlje 10: Programiranje Arduina za internet stvari** Saznaćete kako se postiže da se Arduino ponaša kao veb server i komunicira sa internetom koristeći servise kao što su dweet i IFTTT.
- **Poglavlje 11: C++ i biblioteke** Ovo poglavlje će vas odvesti dalje od C-a, pokazujući vam kako da primenite objektno orijentisano programiranje i pišete sopstvene biblioteke za Arduino.

Prateći resursi

Ova knjiga ima svoju prateću veb lokaciju:

www.arduinobook.com.

Tu ćete pronaći sav izvorni kôd korišćen u knjizi i druge relevantne informacije – recimo, eventualno uočene greške.

SADRŽAJ

Predgovor ix
Zahvalnica xi
Uvod xiii

1 Ovo je Arduino 1
Mikrokontroleri 1
Razvojne ploče 2
Kratak vodič kroz Arduino pločicu 3
Izvor napajanja 4
Konektori za napajanje 4
Analogni ulazi 4
Digitalne veze 5
Mikrokontroler 5
Ostale komponente 6
Istorijat Arduino pločice 7
Porodica Arduino pločica 8
Uno i Leonardo 8
Mega i Due 9
Micro i druge male Arduino pločice 10
Yun 11
Lilypad 12
Druge „zvanične“ pločice 12
Klonovi i varijante pločice Arduino 13
Zaključak 13

2 Početak rada 15
Napajanje 15
Instaliranje softvera 15
Učitavanje prvog programa 16
Aplikacija Arduino 22
Zaključak 24

3 Osnove programskog jezika C 25
Programiranje 25
Šta je programski jezik? 27

Program Blink – ponovo! 31
Promenljive 33
Eksperimenti u C-u 35
Numeričke promenljive i aritmetičke operacije 36
Komande 39
if 39
for 41
while 44
Konstante 45
Zaključak 45

4 Funkcije 47
Šta je funkcija? 47
Parametri 48
Globalne, lokalne i statičke promenljive 49
Povratne vrednosti 52
Drugi tipovi promenljivih 53
Tip float 54
Tip boolean 54
Drugi tipovi podataka 56
Stil pisanja koda 57
Uvlačenje redova koda 58
Otvorene vitičaste zagrade 58
Beline 59
Komentari 59
Zaključak 61

5 Nizovi i znakovni nizovi 63
Nizovi 63
Morzeov SOS signal dobijen pomoću niza 66
Znakovni nizovi 67
Znakovni literali 68
Znakovne promenljive 69
Prevodilac Morzeove azbuke 69
Podaci 70
Globalne promenljive i funkcija setup 71
Funkcija loop 72
Funkcija flashSequence 74
Funkcija flashDotOrDash 75
Objedinjavanje 76
Zaključak 78

6 Ulaz i izlaz 79
Digitalni izlazi 79
Digitalni ulazi 82
Podizni otpornici 83
Ugrađeni podizni otpornici 85
Resetovanje položaja tastera 87
Analogni izlazi 92
Analogni ulaz 94
Zaključak 95

7 Standardna biblioteka za Arduino 97
Slučajni brojevi 97
Matematičke funkcije 100
Manipulacija bitovima 101
Složenije ulazno/izlazne operacije 103
Generisanje tonova 103
Pomerački registri 104
Prekidi 104
Zaključak 106

8 Skladištenje podataka 107
Konstantne vrednosti 107
Skladištenje podataka u fleš memoriji 108
EEPROM 110
Čuvanje celobrojne vrednosti u EEPROM memoriji 111
Upotreba biblioteke AVR EEPROM 112
Čuvanje broja s pokretnim zarezom u EEPROM memoriji 113
Čuvanje znakovnog niza u EEPROM memoriji 114
Brisanje sadržaja EEPROM memorije 115
Komprimovanje podataka 116
Komprimovanje opsega 116
Zaključak 117

9 Displeji 119
Alfanumerički LCD displeji 119
USB tabla za poruke 120
Korišćenje displeja 123
Druge funkcije iz LCD biblioteke 123
OLED grafički displeji 124
Povezivanje OLED displeja 125
Softver 126
Zaključak 128

10 Programiranje Arduina za internet stvari 129
Komunikacija s veb serverima 130
HTTP 130
HTML 130
Arduino Uno kao veb server 132
Arduino kojim se upravlja preko veba 135
Veb server Node MCU 140
Pločica Node MCU kojom se upravlja preko veba 145
Pozivanje veb servisa 148
Arduino Uno i IFTTT 150
ESP8266 pločica Node MCU i IFTTT 152
Ostale IoT opcije 154
Arduino Yun 154
Particle Photon 155
Zaključak 155

11 C++ i biblioteke 157
Objektno orijentisano programiranje 157
Klase i metode 157
Primer ugrađene biblioteke 158
Pisanje biblioteka 158
Datoteka zaglavlja 159
Datoteka implementacije 160
Dovršavanje biblioteke 161
Zaključak 164

Spisak termina korišćenih u knjizi 167

Indeks 169

1

Ovo je Arduino

Arduino je mikrokontrolerska platforma koja već desetak godina zaokuplja maštu elektroničara entuzijasta. Lakoća kojom se koristi i zasnovanost na otvorenom kodu učinili su je odličnim izborom za sve koji žele da realizuju najrazličitije elektronske projekte.

Ukratko, preko njenih pinova možete povezati elektronske komponente tako da može da upravlja raznim stvarima – na primer, da uključuje i isključuje svetla ili motore, ili da detektuje svetlost i temperaturu. Zbog toga se korišćenje Arduina ponekad opisuje kao *fizičko računarstvo* (engl. *physical computing*). Pošto se Arduino pločice mogu povezati s računarom preko kabla univerzalne serijske magistrale (USB), Arduino možete koristiti i kao interfejs karticu pomoću koje možete upravljati istom tom elektronikom sa svog računara.

Ovo poglavlje sadrži uvod u Arduino, uključujući istoriju i poreklo te platforme, kao i pregled relevantnog hardvera.

Mikrokontroleri

Srce Arduino pločice je mikrokontroler. Skoro sve ostalo što postoji na pločici služi za njeno napajanje i omogućavanje njene komunikacije s vašim stonim računarom.

Mikrokontroler je – u suštini – mali računar na čipu. On ima sve što su imali prvi kućni računari – pa i više od toga. Ima procesor, jedan ili dva kilobajta RAM memorije (memorije s nasumičnim pristupom, engl. *random access memory*) za skladištenje podataka, nekoliko kilobajta EEPROM ili fleš memorije u kojoj se čuvaju vaši programi i ima ulazne i izlazne pinove.

Ti ulazno/izlazni (U/I) pinovi povezuju mikrokontroler sa ostatkom vaše elektronike.

Ulazi mogu da budu digitalni (da li je prekidač uključen ili isključen?) i analogni (koliki je napon na određenom pinu?). To otvara mogućnost povezivanja mnogih vrsta senzora – za svetlost, temperaturu, zvuk itd.

Izlazi takođe mogu biti analogni ili digitalni. Znači, možete podesiti da određeni pin bude uključen ili isključen (5 volti odnosno 0 volti) i tako uključivati ili isključivati LED diode direktno, a izlaz možete koristiti i za upravljanje uređajima veće snage – kao što su motori. Na izlazu možete dobiti i analogni signal. To jest, možete podesiti neki pin na određenu vrednost izlaznog napona, i tako upravljati brzinom motora ili jačinom izvora svetlosti – umesto da ih samo uključujete ili isključujete.

Na pločici Arduino Uno, mikrokontroler je 28-pinski čip postavljen u utičnicu na sredini pločice. Ovaj čip sadrži memoriju, procesor i svu elektroniku za ulazno/izlazne pinove. Proizvodi ga kompanija Atmel – jedan od najvećih proizvođača mikrokontrolera. Svaki proizvođač mikrokontrolera proizvodi na desetine različitih mikrokontrolera grupisanih u porodice. Nisu svi mikrokontroleri namenjeni za nas koji se bavimo elektronikom kao hobijem. Mi smo samo mali deo tog širokog tržišta. Mikrokontroleri su u stvari namenjeni za ugrađivanje u komercijalne proizvode široke potrošnje – uključujući automobile, mašine za pranje veša, DVD plejere, dečije igračke, pa čak i osveživače vazduha.

Odlično je to što Arduino smanjuje te zbunjujuće velike mogućnosti izbora tako što standardno koristi samo jedan mikrokontroler i drži se njega. (Kao što ćemo kasnije videti, ova tvrdnja nije baš potpuno tačna, ali je dovoljno blizu.)

To znači da kada započinjete nov projekat, ne morate prvo procenjivati prednosti i mane različitih vrsta mikrokontrolera.

Razvojne ploče

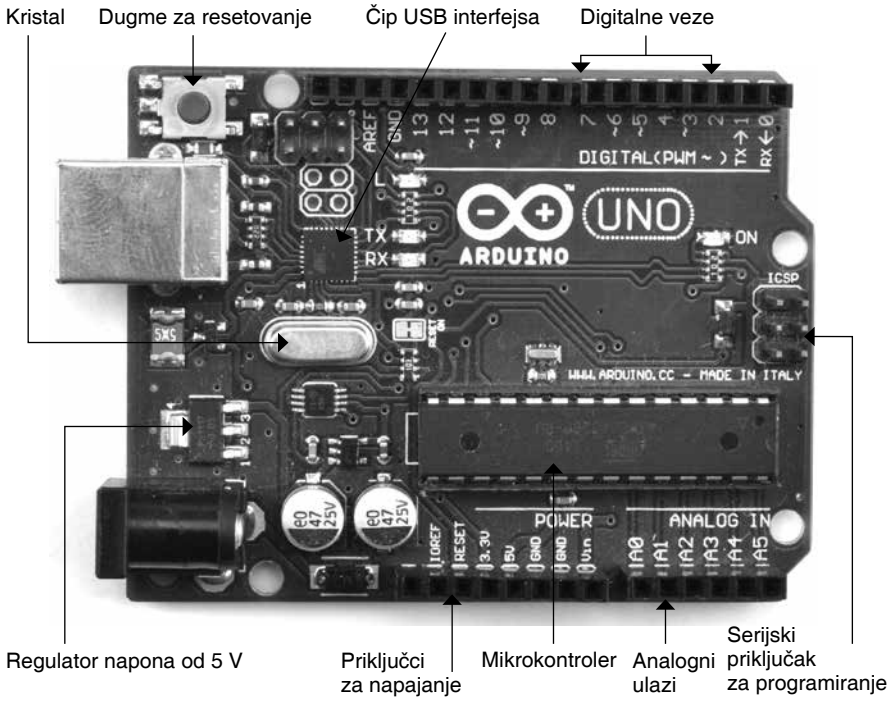
Rekli smo da je mikrokontroler u stvari samo jedan čip. Taj čip neće raditi sam od sebe bez prateće elektronike koja mu obezbeđuje regulisano i precizno napajanje električnom energijom (mikrokontroleri su veoma zahtevni po tom pitanju) kao i način za komunikaciju s računarom preko kojeg će se mikrokontroler programirati.

Tu stupaju na scenu razvojne ploče (engl. *development boards*). Arduino Uno je pločica za razvoj mikrokontrolera, ali istovremeno i nezavisan hardverski uređaj zasnovan na otvorenom kodu. To znači da su sve projektne datoteke za štampanu ploču (engl. *printed circuit board*, PCB) i sve električne šeme javno dostupne, pa svako može slobodno koristiti te resurse za izradu i prodaju sopstvenih Arduino pločica.

Svi proizvođači mikrokontrolera – uključujući Atmel, koji pravi mikrokontroler ATmega328 za Arduino pločice – obezbeđuju i sopstvene razvojne ploče i softver. Mada su one obično relativno jeftine, ipak su pre svega namenjene profesionalcima – inženjerima elektronike – a ne hobistima. To znači da se razvojne ploče i pripadajući softver svakako teže koriste i zahtevaju detaljniju obuku da bi se moglo dobiti nešto korisno od njih.

Kratak vodič kroz Arduino pločicu

Slika 1-1 prikazuje pločicu Arduino Uno. Osvrnimo se ukratko na različite komponente na njoj.



Slika 1-1 *Pločica Arduino Uno.*

Izvor napajanja

Na slici 1-1, neposredno ispod USB priključka nalazi se regulator napona od 5 volti (5V). On pretvara svaki napon (između 7 V i 12 V) koji dolazi iz utičnice izvora DC napajanja u konstantan napon od 5 V.

Čip za regulaciju napona od 5 V u stvari je prilično velik za komponentu koja se površinski montira. To je zato da bi mogao da emituje toplotu potrebnu za regulisanje napona pri razumno velikoj jačini struje – što je korisno kada se napaja spoljna elektronika.

Mada je napajanje Arduina preko utičnice izvora DC napajanja korisno kada Arduino radi na baterije ili povezan pomoću odgovarajućeg kabla s konektorom, Arduino Uno se može napajati i preko USB priključka, koji se koristi i za programiranje Arduina.

Konektori za napajanje

Pogledajmo sada konektore na dnu slike 1-1, pored kojih su i njihove oznake. Namena konektora Reset ista je kao dugmeta Reset na Arduino pločici. Slično kao ponovno pokretanje PC računara, korišćenje konektora Reset resetuje mikrokontroler tako da on započinje svoj program iz početka. Da biste resetovali mikrokontroler preko konektora Reset, postavite ga na nizak naponski nivo (povezujući ga na napon od 0 V).

Ostali konektori u ovom delu samo obezbeđuju različite napone, u skladu s njihovim oznakama (3.3V, 5V, GND i Vin). GND – tj. uzemljenje (engl. *ground*) – znači nula volti. To je referentni napon u odnosu na koji se mere svi ostali naponi na pločici.

Analogni ulazi

Šest pinova sa oznakama od Analog In A0 do A5 mogu se upotrebiti za merenje napona povezanih na njih, tako da se izmerena vrednost može koristiti u programu za Arduino. Imajte na umu da se na njima meri napon a ne struja. Kroz njih može da protiče samo veoma slaba struja – i to ka zemlji – zato što su njihovi unutrašnji otpori vrlo veliki. Drugim rečima, kroz pin s velikim unutrašnjim otporom može da teče samo sasvim slaba struja.

Mada su pomenuti ulazni priključci označeni kao analogni – a podrazumevano i jesu analogni – mogu se koristiti i kao digitalni ulazi ili izlazi.

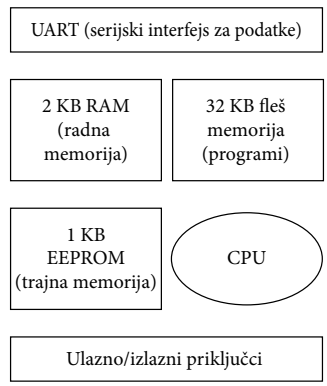
Digitalne veze

Prelazimo na gornji konektor i počinjemo od desne strane na slici 1-1. Tu se nalaze pinovi sa oznakama od Digital 0 do 13 a mogu se koristiti i kao ulazi i kao izlazi. Kada se koriste kao izlazi, ponašaju se slično kao konektori za napajanje opisani ranije u ovom odeljku, osim što su svi na 5 V i mogu se uključivati i isključivati iz programa. Znači, ako ih uključite iz programa biće na 5 V a ukoliko ih isključite biće na 0 V. Kao i u slučaju konektora za napajanje, morate paziti da ne prekoračite maksimalno dozvoljenu struju. Prva dva pina (0 i 1) označena su i sa RX i TX – za prijem (engl. *receive*) odnosno predaju (engl. *transmit*). Rezervisani su za korišćenje u komunikaciji i indirektno predstavljaju prijemni i predajni priključak za USB vezu ka vašem računaru.

Ovi digitalni priključci mogu da obezbede 40 mA (miliampera) pri naponu od 5 V. To je više nego dovoljno za napajanje standardne LED diode, ali nedovoljno za direktno pokretanje električnog motora.

Mikrokontroler

Nastavljamo turu po Arduino pločici. Čip mikrokontrolera je crn pravougaoni uređaj sa 28 pinova. Postavljen je u utičnicu sa dva reda otvora (engl. *dual inline socket*, DIL) tako da se može lako zameniti. 28-pinski čip mikrokontrolera koji se koristi na pločici Arduino Uno jeste ATmega328. Na slici 1-2 je blok-dijagram koji prikazuje glavne mogućnosti ovog uređaja.



Slika 1-2 *Blok-dijagram mikrokontrolera ATmega328.*

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 zbunjujući 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.

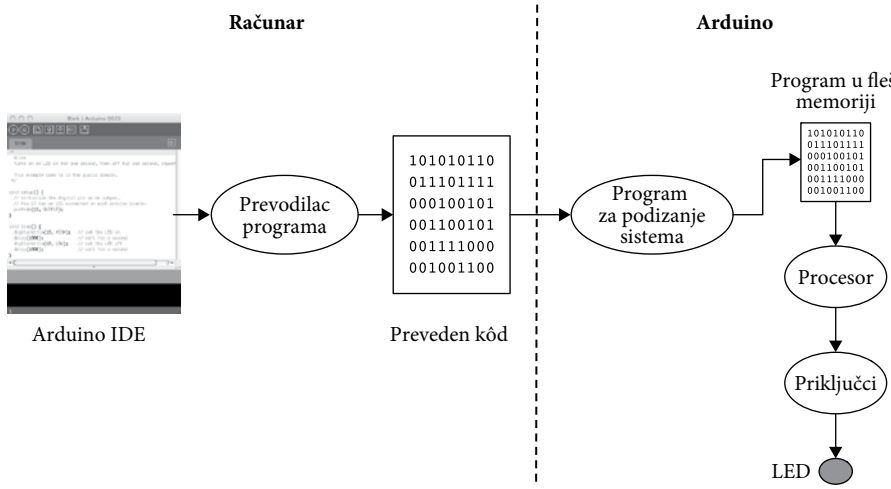
Zapazite da se svaki red završava znakom tačka i zarez. Bilo bi logičnije da su na tim mestima tačke, zato što znak tačka i zarez označava kraj jedne komande – što je slično kraju rečenice.

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 prebacivanje 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 prevede 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.

6

Ulaz i izlaz

Arduino je zasnovan na fizičkom računarstvu, što znači da se elektronske komponente priključuju na Arduino pločicu. Zato treba da znate kako da koristite različite opcije za priključne pinove.

Izlazi mogu biti digitalni – što samo znači da ili imaju napon od 0 V ili od 5 V – ili analogni, što vam omogućava da podesite napon na bilo koju vrednost između 0 i 5 V, mada ipak nije sve tako jednostavno, kao što ćete se uveriti i sami.

Slično tome, i ulazi mogu biti digitalni (recimo, određuju da li je neko dugme pritisnuto ili nije) ili analogni (kao što su oni na senzoru za svetlo).

U knjizi koja se u suštini bavi više softverom nego hardverom, pokušaćemo da ne ulazimo previše u priču o elektronici. Međutim, lakše ćete razumeti materiju u ovom poglavlju ako pri ruci budete imali multimetar i malo parče pune žice.

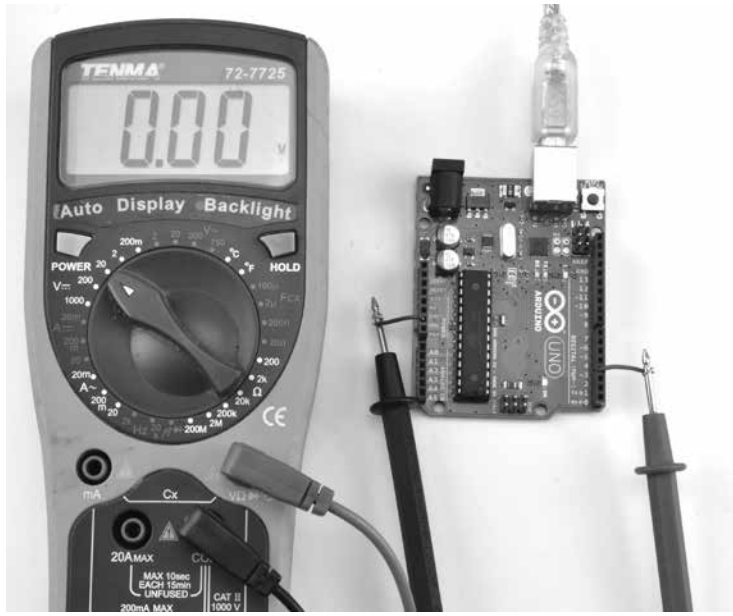
Ukoliko želite da naučite više o elektronici, možda će vas zanimati moja knjiga *Hacking Electronics* (TAB/McGraw-Hill, 2013).

Digitalni izlazi

U ranijim poglavljima koristili ste LED diodu priključenu na digitalni pin 13 Arduino pločice. Primera radi, u poglavlju 5 upotrebili ste je kao signalni uređaj za Morzeovu azbuku. Arduino pločica ima veliki broj digitalnih pinova.

Koristićete digitalni pin 4 i, da biste shvatili šta se dešava, zakačite žicu na merne vezice multimetra i povežite sa Arduinom. Slika 6-1 prikazuje kako to treba da izgleda. Ako vaš multimetar ima krokodilske kleme, skinite izolaciju

s krajeva žice. Jedan kraj spojite s klemom a drugi utaknite u utičnicu na Arduinu. U slučaju da nemate krokodilske kleme, uvijte jedan ogoljen kraj žice oko sonde (slika 6-1).



Slika 6.1 *Merenje izlaza multimetrom.*

Multimetar treba da je podešen na opseg 0–20 V direktne struje (DC). Negativna (crna) vezica treba da je povezana na pin za uzemljenje (GND), a pozitivna na D3. Žica je povezana samo sa sondom i utaknuta u utičnice na Arduino pločici.

Učitajte program 6-01:

```
// program 6-01

const int outPin = 3;

void setup()
{
  pinMode(outPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Enter 1 or 0");
}

void loop()
{
```

```
  if (Serial.available() > 0)
  {
    char ch = Serial.read();
    if (ch == '1')
    {
      digitalWrite(outPin, HIGH);
    }
    else if (ch == '0')
    {
      digitalWrite(outPin, LOW);
    }
  }
}
```

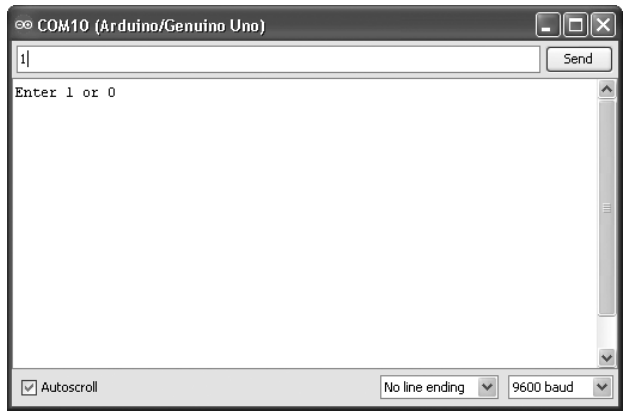
Na početku programa, uočićete komandu **pinMode**. Nju treba da upotrebljavate za svaki pin koji koristite u projektu kako bi elektronske komponente povezane na taj pin Arduino konfigurisao kao ulazne ili izlazne, kao u sledećem primeru:

```
pinMode(outPin, OUTPUT);
```

Kao što ste možda i pretpostavili, **pinMode** je ugrađena funkcija. Njen prvi argument je broj datog pina (**int**), a drugi argument je režim koji mora biti **INPUT**, **INPUT_PULLUP** ili **OUTPUT**. Obratite pažnju na to da naziv režima mora biti ispisan velikim slovima.

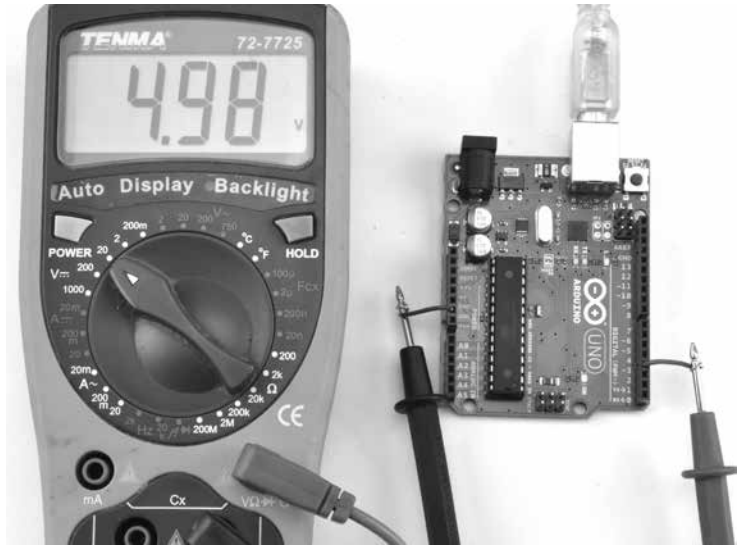
Ova funkcija **loop** čeka da iz Serial Monitora dobije komandu **1** ili **0**. Ako to bude **1**, pin 3 će se uključiti; u suprotnom, on će se isključiti.

Učitajte taj program na Arduino pločicu i otvorite Serial Monitor (slika 6-2).



Slika 6.2 *Serial Monitor.*

Dakle, kada je multimetar uključen i priključen na Arduino, trebalo bi da se očitane vrednosti menjaju od 0 V do otprilike 5 V dok šaljete komande na pločicu iz Serial Monitora tako što ili pritisnete 1 i potom taster ENTER, ili 0 i opet ENTER. Na slici 6-3 prikazana je očitana vrednost na multimetru nakon što je 1 poslato iz prozora Serial Monitor.



Slika 6.3 *Podešavanje izlaza na vrednost High.*

Ukoliko za vaš projekat nema dovoljno pinova sa oznakom „D“, možete koristiti i pinove označene sa „A“ (analogni) kao digitalne izlaze. Samo otkucajte slovo A ispred imena analognog pina – recimo, A0. To možete isprobati tako što ćete izmeniti prvi red u programu 6-01 i zadati da se koristi pin A0, a pozitivnu vezicu multimetra prebacićete na pin A0 na Arduinu.

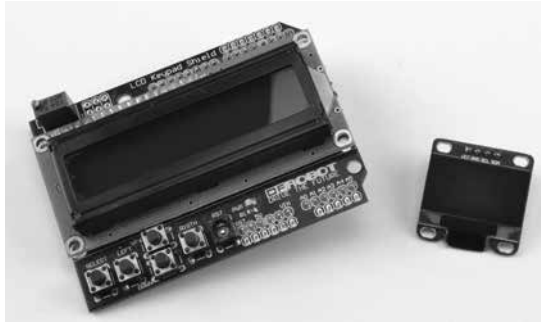
To bi bilo sve o digitalnim izlazima, pa odmah prelazimo na digitalne ulaze.

Digitalni ulazi

Digitalni ulazi najčešće služe da se utvrdi kada je neki prekidač zatvoren. Digitalni ulaz može biti ili uključen ili isključen. Ako je napon na ulazu manji od 2,5 V (polovina od 5 V), on će imati vrednost 0 (isključen), a ukoliko je preko 2,5 V, imaće vrednost 1 (uključen).

9
Displeji

U ovom poglavlju saznaćete kako da pišete softver za upravljanje displejima. Na slici 9-1 prikazana su dva tipa displeja koja ćete koristiti. Prvi je dodatna kartica (štit) za alfanumerički LCD displej. Drugi je OLED (*organic light-emitting diode*) grafički displej rezolucije 128 × 64 piksela. Ta dva tipa displeja veoma su popularni za Arduino.



Slika 9.1 *Alfanumerička LCD kartica (levo) i OLED displej (desno).*

Ovo je knjiga o softveru a ne hardveru, ali ćemo u ovom poglavlju ipak reći ponešto o načinu rada elektronike navedenih displeja da biste razumeli kako da upravljate njima.

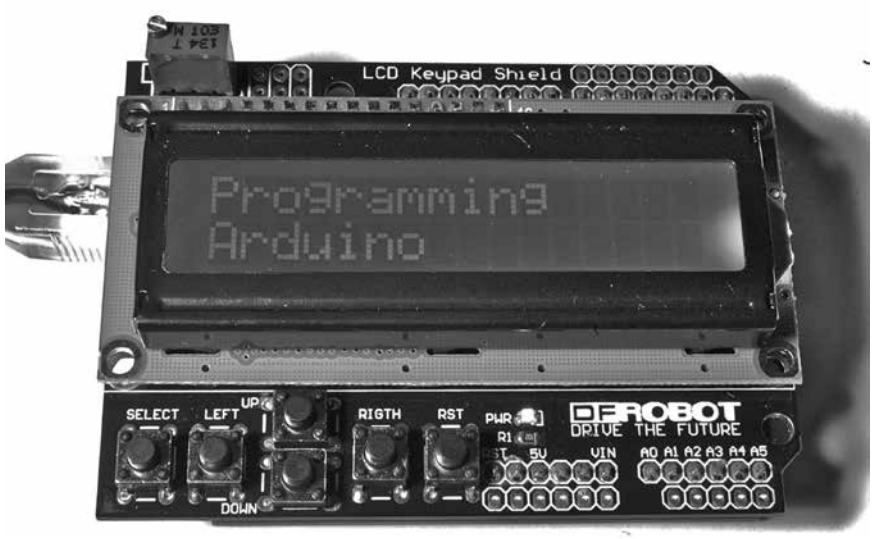
Alfanumerički LCD displeji

Kao LCD modul koristimo dodatnu karticu za Arduino koja se može utaknuti povrh Arduino pločice. Pored displeja, ona sadrži i neku dugmad.

Postoje različite dodatne kartice, ali gotovo sve koriste isti kontrolerski čip za LCD (HD44780), pa potražite takvu karticu.

Ja sam koristio LCD Keypad Shield za Arduino. Taj modul – koji isporučuje kompanija DFRobot (www.robotshop.com) – nije skup i ima LCD displej sa 16 znakova u dva reda i 6 dugmadi na pritiskanje.

Kartica se isporučuje već sklopljena, tako da nikakvo lemljenje nije potrebno; samo je utaknite povrh Arduino pločice (slika 9-2).



Slika 9.2 *LCD kartica priključena na Arduino pločicu.*

LCD kartica koristi sedam pinova na Arduino pločici za upravljanje LCD displejem i jedan analogni pin za dugmad. Znači, te pinove ne možemo koristiti za druge namene.

USB tabla za poruke

Kao ilustraciju jednostavne upotrebe displeja, napravićemo USB tablu za poruke (engl. *message board*). Na njoj će se prikazivati poruke poslate sa Serial Monitora.

Arduino IDE ima LCD biblioteku, što znatno olakšava proces korišćenja LCD displeja. Ta biblioteka sadrži korisne funkcije koje možete pozivati:

- **clear** uklanja sav tekst sa displeja.
- **setCursor** definiše poziciju – tj. red i kolonu – gde će se prikazati ono što sledeće unesete.
- **print** ispisuje znakovni niz na toj poziciji.

Ovaj primer je sadržan u programu 9-01:

```
// program 9-01 USB tabla za poruke

#include <LiquidCrystal.h>

// lcd(RS, E, D4, D5, D6, D7)
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
int numRows = 2;
int numCols = 16;

void setup()
{
  Serial.begin(9600);
  lcd.begin(numRows, numCols);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Arduino");
  lcd.setCursor(0,1);
  lcd.print("Rules");
}

void loop()
{
  if (Serial.available() > 0)
  {
    char ch = Serial.read();
    if (ch == '#')
    {
      lcd.clear();
    }
    else if (ch == '/')
    {
      // novi red
      lcd.setCursor(0, 1);
    }
    else
    {
      lcd.write(ch);
    }
  }
}
```

Kao i za sve Arduino biblioteke, morate početi tako što ćete navesti biblioteku pomoću direktive **#include** kako bi kompajler znao da ćete je koristiti.

Naredni red koda definiše koje pinove na Arduino pločici koristi dodatna kartica i za koje svrhe. Ako imate neku drugu karticu, verovatno će se koristiti neki drugi pinovi, pa proučite dokumentaciju za izabranu karticu.

U ovom slučaju, za upravljanje displejem koristi se sledećih šest pinova: D4, D5, D6, D7, D8 i D9. Namena svakog od njih opisana je u tabeli 9-1.

Parametar funkcije LCD()	Arduinov pin	Namena
RS	8	Register Select; Podešen na 1 ili 0, zavisno od toga da li Arduino šalje podatke za znakove ili neku instrukciju. To bi mogla da bude, na primer, instrukcija za treptanje kursora.
E	9	Enable; Uključuje se kada treba da saopšti kontroleru za LCD da su podaci na sledeća četiri pina spremna za očitavanje.
Data 4	4	Ova četiri pina koriste se za prenos podataka. Kontrolerski čip za LCD koji se nalazi na kartici može da koristi osmobitne ili četvorobitne podatke. Ova kartica radi sa četiri bita tako da se koriste bitovi 4–7 a ne 0–7.
Data 5	5	
Data 6	6	
Data 7	7	

Tabela 9-1 *Pinovi na LCD kartici.*

Funkcija **setup** je jasna. Započinjete serijsku komunikaciju da bi Serial Monitor mogao da šalje komande i inicijalizuje LCD biblioteku u skladu s dimenzijama displeja koji se koristi. Osim toga, prikazujete poruku „Arduino Rules“ u dva reda tako što postavljate kursor gore levo, ispisujete „Arduino,“ zatim pomerate kursor na početak drugog reda i ispisujete „Rules.“

Najveći deo akcije odvija se u funkciji **loop**, koja proverava ima li znakova koji dolaze sa Serial Monitora. Program radi s jednim po jednim znakom.

Pored običnih znakova koje će program prikazati, tu je i nekoliko specijalnih znakova. Ako je primljeni znak #, program briše ceo displej, a ukoliko je to znak /, program prelazi na sledeći red. U svim drugim slučajevima, program samo prikazuje znak na tekućoj poziciji kursora koristeći funkciju **write**. Funkcija **write** je ista kao **print**, ali ispisuje samo jedan znak a ne niz znakova.