

---

# Kolekcija projekata sa senzorima

40+ Projekata za Arduino, Raspberry Pi i ESP32



Dogan Ibrahim

Agencija Eho  
[www.infoelektronika.net](http://www.infoelektronika.net)

---

- Sva prava zadržana. Nijedan deo ove knjige ne sme biti reprodukovan u bilo kom materijalnom obliku, uključujući fotokopiranje ili slučajno ili nenamerno smeštanje na bilo koji elektronski medijum sa ili uz pomoć bilo kog elektronskog sredstva, bez pismenog odobrenja nosioca autorskih prava osim u skladu sa odredbama zakona o autorskim pravima, dizajnu i patentima iz 1988. godine ili pod uslovima izdatim od Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Prijave za pismene dozvole radi štampanja bilo kog dela ove publikacije upućuje se izdavaču ove knjige.
- Izjava: Autor i izdavač su uložili najveće napore da bi se obezbedila tačnost informacija sadržanih u ovoj knjizi. Autor i izdavač ne mogu da pretpostave neprijatnosti i ovom izjavom isključuju bilo kakvu odgovornost za bilo koju stranku koja bi imala gubitke ili štetu uzrokovanu greškama ili propustima u ovoj knjizi, bez obzira da li su greške ili propusti nastali usled nemara, nezgode ili bilo kog drugog razloga.

**ISBN 978-86-80134-27-7**

**Kolekcija projekata sa senzorima**

Naslov originala: The Ultimate Compendium of Sensor Projects

Autor: Dogan Ibrahim

Prevod: Biljana Tešić

Izdaje i štampa: Agencija Eho, Niš

e-mail: redakcija@infoelektronika.net

Tiraž: 300

Godina izdanja: 2020

CIP - Каталогизација у публикацији  
Народна библиотека Србије, Београд

004.383/.384  
681.586

ИБРАХИМ, Доган, 1954-

Колекција пројеката са сензорима : 40+ Пројеката за Arduino, Raspberry Pi i ESP32

/ Dogan Ibrahim ; [prevod Biljana Tešić]. - Niš : Agencija Eho, 2020  
(Niš : Agencija Eho). - 328 str. : ilustr. ; 24 cm

Prevod dela: The Ultimate Compendium of Sensor Projects. - Tiraž 300.

ISBN 978-86-80134-27-7

а) Микроконтролери -- Примена б) Мерни инструменти, сензорски -- Примена

COBISS.SR-ID 281391372

<b>Sadržaj .....</b>	<b>3</b>
Predgovor .....	11
Poglavlje 1 • Arduino Uno hardverski interfejs i razvoj projekta .....	13
1.1 Pregled.....	13
1.2 Arduino Uno ploča.....	13
1.3 Razvoj programa Arduino Uno .....	14
1.4 Projekat - Treperenje dvobojnog LED-a.....	15
1.5 Rezime .....	19
<b>Poglavlje 2 • Raspberry Pi hardverski interfejs i razvoj projekta .....</b>	<b>20</b>
2.1 Pregled.....	20
2.2 Ploča Raspberry Pi 3 .....	20
2.3 GPIO definicije pinova na Raspberry Pi 3 .....	21
2.4 Postavljanje Wi-Fi-ja i udaljenog pristupa.....	22
2.5 Isključivanje ili ponovno pokretanje u GUI režimu.....	27
2.6 Daljinski pristup radnoj površini.....	27
2.7 Kreiranje i pokretanje programa Python .....	29
2.8 GPIO biblioteka .....	32
2.9 Ciklus razvoja projekta Raspberry Pi .....	35
2.10 Projekat – Treperenje dvobojnog LED-a.....	36
2.11 Rezime .....	39
<b>Poglavlje 3 • ESP32 hardverski interfejs i razvoj projekta .....</b>	<b>40</b>
3.1 Pregled.....	40
3.2 ESP32 DevKitC hardver .....	40
3.4 Projekat - Treperenje dvobojnog LED-a.....	47
3.5 Rezime .....	50

<b>Poglavlje 4 • Osnovni projekti senzora: Arduino - Raspberry Pi - ESP32 .....</b>	<b>51</b>
4.1 Pregled.....	51
4.2 Projekti za svetlo .....	51
4.3 Rezime .....	114
<b>Poglavlje 5 • Projekti za infracrveni prijemnik/predajnik .....</b>	<b>115</b>
5.1 Pregled.....	115
5.2 Projekat 1 - Prijem i dekodiranje kodova komercijalnih IR uređaja.....	115
5.3 Projekat 2 - Kontrolisanje 2 – LED-ovi u boji sa komercijalnim IR uređajem.....	132
5.4 Projekat 3 - Infracrveni predajnik - Skenirajte TV kanale, koristeći komercijalni IC uređaj.....	137
5.5 Projekat 4 - Dva Arduino Uno-a koja komuniciraju pomoću IR-a .....	142
5.6 Rezime .....	146
<b>Poglavlje 6 • Projekti vibracija i šoka .....</b>	<b>147</b>
6.1 Pregled.....	147
6.2 Projekat 1 - Detektor pucanja u metu.....	147
6.3 Projekat 2 - Preklopnik zasnovan na vibraciji .....	151
6.4 Projekat 3 - Bez merenja trajanja vremena šoka .....	154
6.5 Rezime .....	156
Poglavlje 7 • Projekti ultrazvučnih senzora.....	157
7.1 Pregled.....	157
7.2 Projekat 1 - Ultrazvučno parkiranje unazad sa zujalicom .....	157
7.3 Rezime .....	168
<b>Poglavlje 8 • Projekti zvučnih senzora.....</b>	<b>169</b>
8.1 Pregled.....	169
8.2 Projekat 1 • Uključivanje/isključivanje svetla na pljesak ruku .....	169
8.3 Rezime .....	172

<b>Poglavlje 9 • Projekti senzora pasivnih piezo zujalica.....</b>	<b>173</b>
9.1 Pregled.....	173
9.2 Projekat 1 - Reprodukovanje melodije .....	173
9.3 Rezime .....	180
<b>Poglavlje 10 • Projekti magnetnih senzora.....</b>	<b>181</b>
10.1 Pregled.....	181
10.2 Projekat 1 - Merenje jačine magnetnog polja.....	181
10.3 Projekat 2 - Magnetni alarm na vratima.....	184
10.4 Projekat 3 - Magnetni muzički instrument .....	186
10.5 Rezime .....	188
<b>Poglavlje 11 • Projekti senzora plamena .....</b>	<b>189</b>
11.1 Pregled.....	189
11.2 Projekat 1 - Senzor plamena sa zvukom .....	189
11.3 Rezime .....	192
<b>Poglavlje 12 • Projekti modula džojstika .....</b>	<b>193</b>
12.1 Pregled.....	193
12.2 Projekat 1 - Korišćenje džojstika.....	193
12.3 Projekat 2 - Muzički instrument zasnovan na džojstiku .....	199
12.4 Rezime .....	201
<b>Poglavlje 13 • Projekti senzora za prepreke.....</b>	<b>202</b>
13.1 Pregled.....	202
13.2 Projekat 1 - Pomoć pri parkiranju .....	202
13.3 Projekat 2 - Metalni senzor osetljiv na dodir .....	205
13.4 Rezime .....	207
<b>Poglavlje 14 • Projekti modula senzora za praćenje .....</b>	<b>208</b>
14.1 Pregled.....	208
14.2 Projekat 1 - Praćenje linija.....	208

14.3 Projekat 2 - Tajna šifra za zaključavanje .....	212
14.4 Rezime .....	217
<b>Poglavlje 15 • Projekti modula rotacionog enkodera .....</b>	<b>218</b>
15.1 Pregled.....	218
15.2 Projekat 1 - Procena rotacionog enkodera.....	218
15.3 Projekat 2 - Smer i položaj rotacionog enkodera .....	221
15.4 Rezime .....	225
<b>Poglavlje 16 • Projekti modula senzora za merenje otkucaje srca.....</b>	<b>226</b>
16.1 Pregled.....	226
16.2 Projekt 1 - Prikaz otkucaja srca.....	226
16.3 Rezime .....	228
<b>Poglavlje 17 • Projekti senzora za temperaturu, vlagu i pritisak .....</b>	<b>229</b>
17.1 Pregled.....	229
17.2 Projekat 1 - Prikazivanje i iscrtavanje temperature okoline na monitoru .....	229
17.3 Projekat 2 - Senzor temperature sa zujalicom .....	237
17.4 Projekat 3 - Prikazivanje temperature na LCD-u - Arduino Uno .....	240
17.5 Projekat 4 - Čuvanje temperature u CSV datoteci na računaru sa vremenskom oznakom - Arduino Uno .....	247
17.6 Projekat 5 - Prikazivanje temperature na LCD-u - ESP32 DevKitC .....	253
17.7 Projekat 6 - Prikazivanje temperature na LCD-u - Raspberry Pi .....	255
17.8 Projekat 7 - Čuvanje temperature kao CSV datoteke na računaru sa vremenskom oznakom za Raspberry Pi .....	259
17.9 Projekat 8 - ON/OFF kontrola temperature - Arduino Uno .....	261
17.10. Projekat 9 - ON/OFF kontrola temperature - ESP32 DevKitC .....	266
17.11. Projekat 10 - ON/OFF kontrola temperature - Raspberry Pi .....	267
17.12 Rezime .....	270

<b>Poglavlje 18 • Wi-Fi i Bluetooth projekti zasnovani na sensorima - ESP32 DevKitC .....</b>	<b>271</b>
18.1 Pregled.....	271
18.2 Projekat 1 - Prikazivanje temperature i vlažnosti na mobilnom telefonu putem Wi-Fi veze .....	271
18.3 Projekat 2 - Daljinsko upravljanje sa mobilnog telefona putem Wi-Fi veze .....	280
18.4 Projekat 3 - Slanje temperature i vlažnosti na mobilni telefon putem klasične Bluetooth veze .....	285
<b>Poglavlje 19 • Wi-Fi i Bluetooth projekti zasnovani na sensorima - Raspberry Pi .....</b>	<b>289</b>
19.1 Pregled.....	289
19.2 Projekat 1 - Prikazivanje temperature i vlažnosti na mobilnom telefonu putem Wi-Fi veze .....	289
19.3 Projekat 2 - Slanje podataka o temperaturi i vlažnosti u „oblak“ putem Wi-Fi veze .....	292
19.4 Projekat 3 - Bluetooth daljinski upravljač sa mobilnog telefona.....	298
<b>Poglavlje 20 • Wi-Fi i Bluetooth projekti zasnovani na sensorima - Arduino Uno .....</b>	<b>308</b>
20.1 Pregled.....	308
20.2 Projekat 1 - Kontrola releja sa mobilnog telefona putem Wi-Fi veze.....	308
20.3 Projekat 2 - Prikazivanje temperature i vlažnosti na mobilnom telefonu putem Wi-Fi veze .....	314
20.4 Projekat 3 - Kontrola releja sa mobilnog telefona putem Bluetooth veze.....	318
<b>Dodatak A • Sadržaj kompleta senzora (JOY-iT Sensor-Kit X40) .....</b>	<b>324</b>
<b>Dodatak B • Projekti i korišćeni moduli senzora .....</b>	<b>325</b>
<b>Dodatak C • Moduli senzora koji se koriste u projektima.....</b>	<b>327</b>



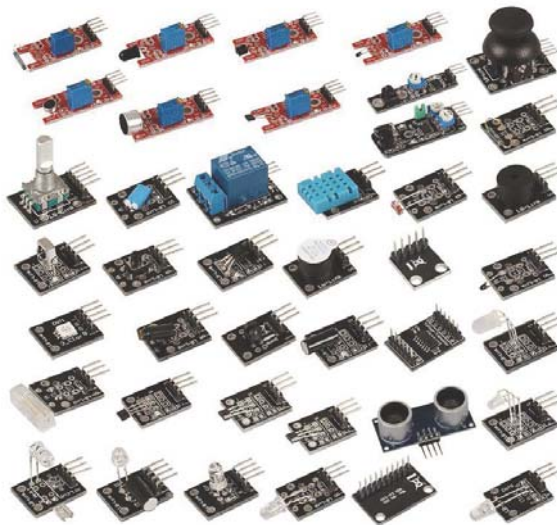


## Predgovor

Senzori su uređaji ili komponente koji detektuju događaje ili promene u svom okruženju i šalju informacije drugim elektronskim uređajima, najčešće sistemima mikrokontrolera. Senzori se koriste u svakodnevnom životu za merenje stavki, kao što su temperatura, vlaga, pritisak, vetar i kiša, dodir, nivo svetlosti, nivo tečnosti, nadmorska visina, sila i još mnogo toga.

Iako neki senzori imaju digitalne izlaze, većina senzora koji se koriste u svakodnevnom životu imaju analogne izlaze, obično u obliku napona koji su proporcionalni izmerenim količinama. Ovaj izlazni napon se obično dovodi do ulaznog porta mikrokontrolera, radi obrade. Na primer, izlaz analognog temperaturnog senzora je povezan sa analognim ulaznim portom (analogno-digitalnim pretvaračem) mikrokontrolera. Mikrokontroler očitava temperaturu kao digitalnu vrednost i pretvara je u stvarnu fizičku temperaturu, koja se zatim prikazuje ili koristi za kontrolu temperature mašine ili prostorije.

Ova knjiga govori o korišćenju senzora koji se nalaze u kompletu senzora. U kompletu ima ukupno 40 senzora. Jedni senzori imaju analogne izlaze, dok drugi imaju digitalne izlaze, a treći imaju i analogne i digitalne izlaze. Komplet uključuje senzore za merenje temperature, vlažnosti, atmosferskog pritiska, jačine svetlosti i zvuka. Tu su takođe 2 i 3 LED indikatora u boji, nagibni prekidači, magnetni prekidači, releji, reed prekidači, piezo zujalica, taster, džojstik, detektor prepreka, senzor otkucaja srca, analogno-digitalni pretvarač, pretvarač napona, prekidač za vibracije itd.



40 senzora u kompletu All-in-1

Knjiga je namenjena za učenje kako se koriste senzori sa popularnim sistemima za razvoj mikrokontrolera: Arduino Uno, ESP32 DevKitC i Raspberry Pi. Dati su primeri projekata u kojima je prikazano kako se koriste senzori sa ovim mikrokontrolerima. Programi se

mogu modifikovati za ostale mikrokontrolere, kao što su PIC, STM32, Banana Pi, Cubie-Board, Beaglebone itd.

Svi projekti navedeni u knjizi su izrađeni korišćenjem prototipske ploče standardne veličine i potpuno su testirani i funkcionalni. Projekti su opisani sa sledećim podnaslovima:

- Opis projekta
- Cilj projekta
- Blok dijagram
- Električna šema
- Listing programa

Rad svakog senzora i svaki listing programa detaljno su opisani tako da čitaoci neće imati poteškoća u konstruisanju ili proširivanju datog projekta. U nekim projektima se koristi više senzora iz kompleta. Čitaocima se preporučuje da prate projekte navedenim redosledom, jer neke od softverskih alatki koje se koriste u nekim projektima zavise od instalacije ovih alatki u prethodnom projektu.

Kompletni listing programa projekata sa mnogo komentara dostupan je na veb sajtu Elektor knjige, a čitaoci bi trebalo da kopiraju i koriste ove programe, bez potrebe za izmenama.

Arduino Uno i ESP32 DevKitC programi zasnivaju se na korišćenju Arduino IDE-a sa C jezikom. U Raspberry Pi projektima koristi se programski jezik Python. Iako se u projektima zasnovanim na Arduinu navodi Arduino Uno kao razvojna ploča, ovi bi projekti trebalo takođe da funkcionišu sa drugim Arduino razvojnim pločama, kao što su Arduino Mega, Arduino Nano itd.

Nadam se da će čitaocima knjiga biti od koristi i da će uživati u eksperimentisanju sa različitim sensorima.

*Dogan Ibrahim*

*jun, 2019.*

## Poglavlje 1 • Arduino Uno hardverski interfejs i razvoj projekta

### 1.1 Pregled

U ovoj knjizi ćemo koristiti Arduino Uno ploču u senzorskim projektima koji se zasnivaju na Arduinu. Ovo poglavlje prikazuje lokaciju različitih komponenti na ploči Arduino Uno i takođe opisuje hardverski interfejs za spoljni svet. U ovom poglavlju dat je jednostavan projekat koji će čitaoca upoznati sa razvojem programa pomoću ploče Arduino Uno. U prvom programu dvobojni LED treperi naizmenično svake sekunde.

### 1.2 Arduino Uno ploča

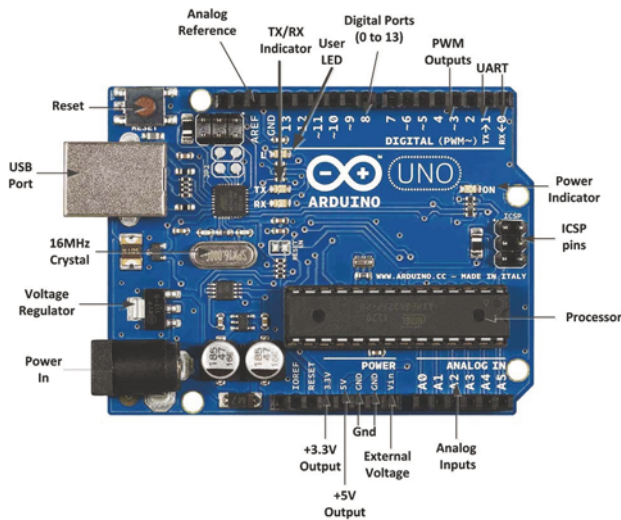
Na slici 1.1 je detaljno prikazana ploča Arduino Uno sa označenim glavnim komponentama.

Definicije pinova su sledeće (pogledajte 1.2).

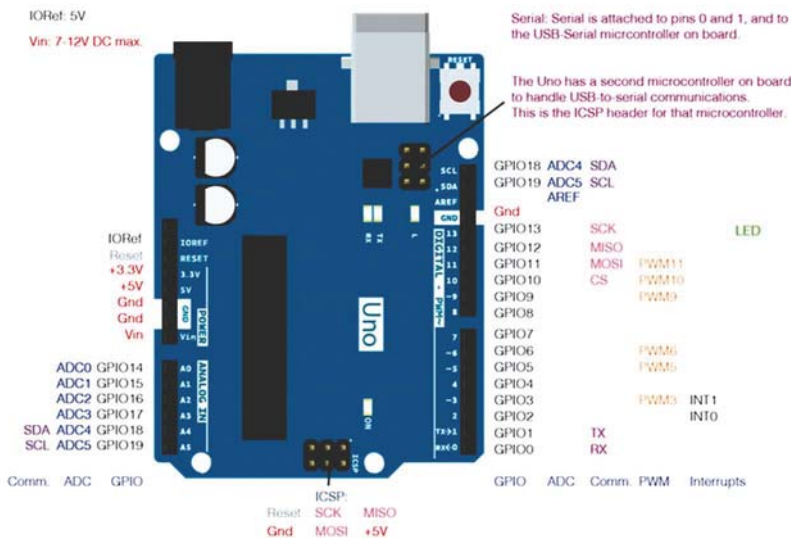
A0 – A5	Analogni ulazni portovi
0 – 13	Portovi za digitalni ulaz/izlaz
~3,~5,~6,~9,~10,~11:	PWM izlazni portovi
0,1:	UART RX/TX pinovi. LED-ovi sa oznakom TX, RX će treperiti kada se podaci šalju ili primaju, tim redosledom.
GND:	Pin za uzemljenje napajanja
5 V:	Regulisani izlaz od +5 V
3,3 V	Regulisani izlaz od 3,3 V
Vin:	Ulaz napona (umesto upotrebe Power In-a ili USB-a). Napon mora biti u opsegu od 7 V do 12 V. Interno je regulisan na +5 V. Ovaj pin se takođe može koristiti kao izlaz napona (ako se napaja pomoću Power In-a ili USB priključka). Napon je kopija napona koji se dobija pomoću Power In-a ili USB priključka.
I0Ref:	Koriste ga spoljni moduli da bi znali da li treba da rade kao +3,3 V ili kao +5 V uređaji.
Power In:	Barrel Jack pin za napajanje (od 6 V do 12 V).
USB port:	Priključak za napajanje i podatke (povezuje se sa računarem).
User LED:	Ugrađeni LED koji je povezan sa izlaznim portom 13 (može se koristiti za svrhe testiranja).

Primetićete da kada Arduino Uno napaja USB priključak (npr. sa računara), maksimalni kapacitet struje je oko 400 mA za +5 V pin i 150 mA za + 3,3 V pin. Kada se napaja iz spoljnog izvora, maksimalna struja za 5 V pin je oko 900 mA i 150 mA za + 3,3 V pin. Svaka struja izvučena iz + 3,3 V prolazi kroz pin + 5 V. Stoga, ovo morate uzeti u obzir prilikom napajanja spoljnih uređaja.

Apsolutna maksimalna struja za bilo koji I/O pin je određena kao 40 mA (ipak se preporučuje da ne premaši 20 mA). Apsolutna ukupna struja iz svih I/O pinova je 200 mA.



Slika 1.1 Arduino Uno ploča



Slika 1.2 Arduino Uno raspored pinova

### 1.3 Razvoj programa Arduino Uno

Lepa karakteristika svih Arduino ploča je ta što se sve one mogu programirati pomoću Arduino IDE-a. Najnovija verzija Arduino IDE-a može se preuzeti sa veb sajta koji je dat ispod. U vreme pisanja ove knjige najnovija verzija je bila 1.8.8:

<https://www.arduino.cc/en/Main/Software>

Ovo su koraci za pisanje i učitavanje programa na Arduino Uno:

Povežite Arduino Uno sa USB priključkom na računaru.

Pokrenite Arduino IDE na računaru.

- Kliknite na **Tools** -> **Board** i izaberite tip ploče **Arduino/Gerduino** Uno.
- Kliknite na **Tools** -> **Port** i izaberite serijski port sa kojim je povezan vaš Arduino Uno.
- Napišite i sačuvajte vaš program.
- Kliknite na **Sketch** -> **Verify/Compile** da biste kompajlirali program. Proverite da li ima grešaka prilikom kompajliranja.
- Kliknite na **Sketch** -> **Upload** da biste poslali izvršivi kod u programsku memoriju Arduino Uno-a.

U narednim odeljcima data su dva primera projekta da bi se čitalac upoznao sa ciklusom razvoja projekata, koristeći Arduino Uno. Pretpostavlja se da čitalac ima neko osnovno znanje o elektronici, a takođe i neko praktično poznavanje pisanja jezika visokog nivoa.

#### 1.4 Projekat - Treperenje dvobojnog LED-a

**Opis** - Ovo je možda najlakši projekat koji možete dizajnirati, koristeći Arduino Uno. U ovom projektu dvobojni LED sa zajedničkom katodom povezan je sa Arduino Uno-m. LED boje trepere naizmenično brzinom od jedne sekunde.

**Cilj** - Cilj ovog projekta je da se prikaže kako se program može napisati, kompajlirati i preneti u Arduino Uno. U projektu je takođe prikazano kako se koriste neki od I/O i funkcija za podešavanje vremena na Arduino.

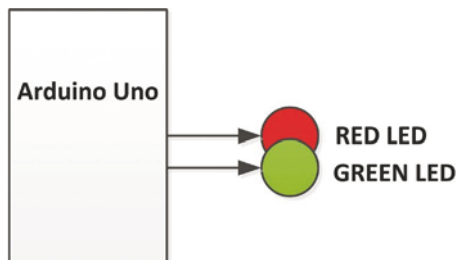
**Senzor koji se koristi** - U ovom projektu se koristi senzor KY-011. Ovaj senzor sastoji se od LED crvene i zelene boje smeštene u paketu sa zajedničkim katodnim terminalom. Senzor i njegovi pinovi prikazani su na slici 1.3. Senzor ima 3 pina - crveni LED pin, zeleni LED pin i GND pin. **Zeleni LED pin označen je slovom S na ploči, radi identifikacije.** LED se uključuje kada se logička 1 primeni na njegove pinove. Osnovne specifikacije ovog senzora su sledeće:

Radni napon:	od 2,0 do 2,5 V
Radna struja:	10 mA (zavisi od potrebne svetline)
Talasna dužina:	zeleni (571 nm), crveni (644 nm)
Jačina svetlosti (MCD):	zeleni (20–40), crveni (40–80)



Slika 1.3 Senzor KY-011

**Blok dijagram** - Blok dijagram projekta prikazan je na slici 1.4.



Slika 1.4 Blok dijagram projekta

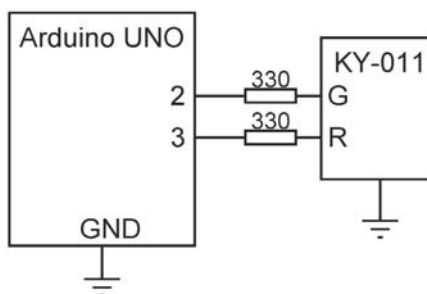
**Električna šema** - Električna šema projekta prikazana je na slici 1.5. Zeleni i crveni LED pinovi su povezani sa pinovima porta 2 i 3 Arduino Uno-a putem otpornika za ograničavanje struje. GND pin je povezan sa Arduino Uno GND pinom. Vrednost otpornika za ograničavanje struje izračunava se na sledeći način:

Visoki napon izlaznog pina je 5 V. Napon propusne polarizacije na LED-u je oko 2,0 V (za crveni LED je 1,8 V, a za zeleni LED 2,8 V). Pod pretpostavkom da će propusna struja za LED biti postavljena na 10 mA (razumna svetlina), tada je vrednost otpornika za ograničavanje struje:

$$R = (5 \text{ V} - 2 \text{ V}) / 10 \text{ mA} = 300 \text{ oma, koristite otpornik od 330 oma}$$

(možete koristiti manju vrednost, npr. 270 oma za zeleni LED)

Na slici 1.5, LED radi u režimu snabdevanja strujom u kojem visoki izlaz iz I/O pina pokreće LED. LED može da radi i u trenutnom režimu odvođenja struje u kojem je drugi kraj LED-a povezan sa napajanjem od +5 V, a ne sa uzemljenjem. U režimu odvođenja struje, LED se uključuje kada je I/O pin logički nizak.



Slika 1.5 Električna šema projekta

**Konstrukcija** - Projekat je izrađen na ploči, kao što je prikazano na slici 1.6. Kratkospojnici se koriste za povezivanje senzora KY-011 sa digitalnim pinovima porta 2 i 3 i GND-om Arduino Uno-a.



Slika 1.6 Projekat konstruisan na prototipskoj ploči

**Listing programa** - Program se zove **TWOCOLOUR**, a listing je prikazan na slici 1.7. Na početku programa pinovi porta 2 i 3 se dodeljuju zelenom LED pinu i crvenom LED pinu, tim redom. Unutar rutine podešavanja pinovi porta 2 i 3 na kojima su priključeni LED pinovi konfigurisani su kao izlazi. Unutar glavne programske petlje dve boje se uključuju i isključuju naizmenično, sa kašnjenjem od jedne sekunde između svakog izlaza. Kliknite na **Verify/Compile** da biste bili sigurni da nema grešaka, a zatim kliknite na **Upload** da biste otpremili programski kod u Arduino Uno. Trebalo bi da vidite kako dve boje trepere brzinom od jedne sekunde.

Autor izričito preporučuje da komentarišete redove u vašem programu i da opišete operaciju koja se izvodi, kao što je prikazano na slici 1.7. Takođe trebalo bi da dodate naslov i da ukratko opišete šta program radi. To olakšava kasnije razumevanje i održavanje programa. To takođe olakšava bilo kome drugom da razume logiku programa kada ga pročita.

```

/*****
 *          FLASHING GREEN AND RED LEDs
 *          =====
 *
 * In this program the KY-011 sensor is used which has a green and
 * a red LED with a common cathode terminal. The green LED and the
 * red LED are connected to I/O pins 2 and 3 of the Arduino Uno
 * respectively. The program flashes the LEDs alternately at a rate
 * of one second.
 *
 * Author: Dogan Ibrahim
 * Date  : April 2019
 * File  : TWOCOLOUR
 *****/
int GreenLED = 2;           // Green LED pin
int RedLED = 3;            // Red LED pin
#define ON HIGH
#define OFF LOW

void setup()
{
  pinMode(GreenLED, OUTPUT); // Set as output
  pinMode(RedLED, OUTPUT);  // Set as output
}

void loop()
{
  digitalWrite(GreenLED, ON); // Turn ON green
  digitalWrite(RedLED, OFF);  // Turn OFF red
  delay(1000);                // Wait 1 sec
  digitalWrite(GreenLED, OFF); // Turn OFF green
  digitalWrite(RedLED, ON);   // Turn ON red
}

```



```
    delay(1000);                // Wait 1 sec
}
```

Slika 1.7 Listing programa projekta

**Šta smo naučili** - U ovom projektu smo naučili kako da koristimo sledeće Arduino funkcije:

<b>int:</b>	deklariše celobrojnu promenljivu
<b>#define:</b>	dodeljuje vrednost ili tekst nizu
<b>pinMode (port pin, mode):</b>	koristi se za konfigurisanje I/O pin porta kao ulaza ili izlaza
<b>digitalWrite(pin port, value):</b>	koristi se za ispis digitalne vrednosti (logičko LOW ili HIGH) na pin portu
<b>delay(n):</b>	kreira kašnjenje od <b>n</b> milisekundi

### 1.5 Rezime

U ovom poglavlju smo videli razne komponente i definicije pinova Arduino Uno mikrokontrolera. Osim toga, dat je jednostavan projekat u kome je prikazano kako se dizajniraju jednostavni projekti i kako se kompajlira i šalje programski kod u programsku memoriju Arduino Uno mikrokontrolera.

U sledećem poglavlju ćemo pogledati kako se koristi Raspberry Pi mikrokontrolera u projektima.

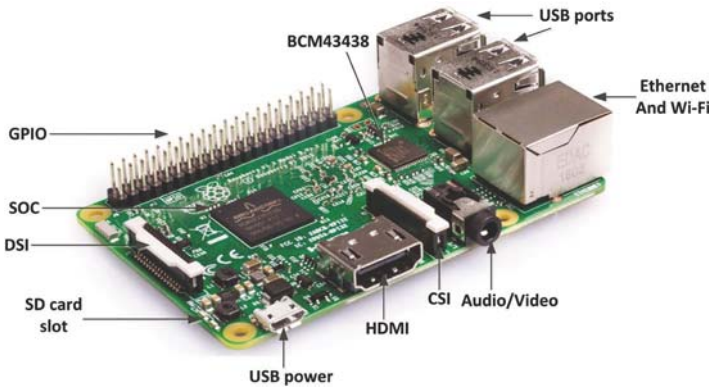
## Poglavlje 2 • Raspberry Pi hardverski interfejs i razvoj projekta

### 2.1 Pregled

U ovoj knjizi ćemo koristiti Raspberry Pi 3 Model B u našim Raspberry Pi projektima. U ovom poglavlju prikazana je lokacija različitih komponenti na ploči Raspberry Pi i opisan je hardverski interfejs za spoljni svet. Podešavanje Wi-Fi mreže i daljinski pristup Raspberry Pi računaru i upoznavanje sa Python programskim okruženjem su takođe ukratko opisani. Prikazan je jednostavan projekat da bi se čitalac upoznao sa koracima za dizajniranje projekta. U ovoj knjizi kada pišemo Raspberry Pi 3 zapravo mislimo na Raspberry Pi 3 Model B.

### 2.2 Ploča Raspberry Pi 3

Na slici 2.1 prikazana je ploča Raspberry Pi 3 sa označenim glavnim komponentama. Neki detalji o svakoj komponenti dati su u ovom odeljku.



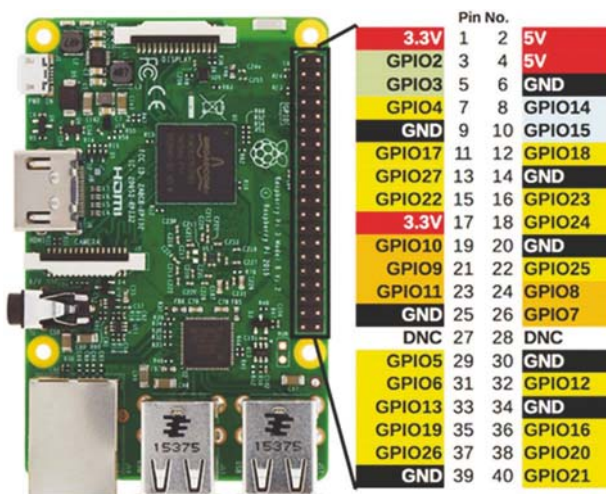
Slika 2.1 Ploča Raspberry Pi 3

- USB priključci:** Raspberry Pi 3 ima 4 USB priključka za povezivanje miša, tastature, veb kamere itd.
- Ethernet i Wi-Fi:** Iako Raspberry Pi 3 ima ugrađen Wi-Fi, on se takođe može direktno povezati sa ruterom preko Ethernet kabla koji je povezan sa ovom utičnicom.
- Audio/video priključak:** Na ovu 3,5 mm utičnicu mogu se priključiti slušalice ili zvučnik. Ova utičnica takođe sadrži kompozitni video interfejs.
- CSI:** Ovo je serijski interfejs kamere na koji se može priključiti kompatibilna Raspberry Pi kamera.
- HDMI:** Na ovaj priključak se može povezati odgovarajući monitor. Priključak prenosi i audio i video signale.
- USB napajanje:** Na ovu USB utičnicu treba priključiti napajanje od + 5 V 2 A da bi se napajao Raspberry Pi.

- Otvor za SD karticu:** Mikro SD kartica koja prenosi operativni sistem mora biti priključena na ovaj otvor.
- DSI:** Sa ovim interfejs ekranom može se povezati odgovarajući Display Interface konektor.
- SOC:** Ovo je Broadcom BCM2837 System On Chip (SOC) koji sadrži 64-bitni četvororojezgarni ARM Cortex-A53 procesor od 1,2 GHz.
- GPIO:** Ulazno-izlazni port za opštu namenu je širok 40 pinova.
- BCM43438:** Ovaj čip obezbeđuje Wi-Fi i Bluetooth za Raspberry Pi 3.

### 2.3 GPIO definicije pinova na Raspberry Pi 3

Raspberry Pi 3 je povezan sa spoljnim digitalnim elektronskom kolima i uređajima, pomoću GPIO (General Purpose Input Output) konektora porta. Ovo je 2,54 mm 40-pinsko zaobljeno za proširenje koje je raspoređeno na traci 2 x 20, kao što je prikazano na slici 2.2.



Slika 2.2 GPIO pinovi na Raspberry Pi 3

Kada je GPIO konektor na drugoj strani ploče, pinovi na dnu, počevši od leve strane konektora, označeni su brojevima 1, 3, 5, 7 i tako dalje, dok su oni na vrhu označeni kao 2, 4, 6, 8 i tako dalje.

GPIO obezbeđuje 26 dvosmernih I/O pinova opšte namene. Neki pinovi imaju više funkcija. Na primer, pinovi 3 i 5 su GPIO2 i GPIO3 pinovi za ulaz i izlaz. Ovi pinovi se takođe mogu koristiti kao I<sup>2</sup>C magistrala I<sup>2</sup>C SDA i I<sup>2</sup>C SCL pinova, tim redom. Slično tome, pinovi 9,10,11,19 se mogu koristiti kao ulaz-izlaz opšte namene ili kao SPI pinovi magistrale. Pinovi 8 i 10 su rezervisani za UART serijsku komunikaciju.

Obezbeđena su dva izlaza za napajanje: +3,3 V i +5,0 V. GPIO pinovi rade na logičkim nivoima od +3,3 V (ne kao kod mnogih računarskih kola koja rade na +5 V). Pin može biti

ulaz ili izlaz. Kada je konfigurisan kao izlaz, napon pina je 0 V (logička 0) ili +3,3 V (logička 1).

Raspberry Pi 3 obično radi pomoću spoljnog napajanja (npr. pomoću mrežnog adaptera) sa +5 V izlazom i minimalnim naponom od 2 A. Izlazni napon od 3,3 V može da napaja do 16 mA struje. Ukupna struja izvučena iz svih izlaznih pinova ne sme preći granicu od 51 mA. Treba da budete oprezni kada priključujete spoljne uređaje na GPIO pinove, jer prevelika potrošnja struje ili kratki spoj na pinu mogu lako oštetiti vaš Pi. Količina struje koja može da napaja 5 V pin zavisi od mnogih faktora, kao što su struja koja je potrebna za sam Pi, struja koju koriste USB periferne jedinice, struja kamere, struja HDMI priključka i tako dalje.

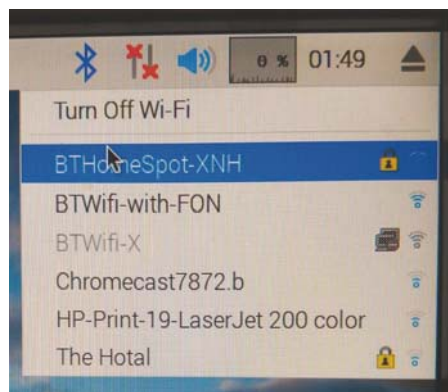
Kada je konfigurisan kao ulaz, napon iznad + 1,7 V se smatra kao logička 1, a napon ispod + 1,7 V se smatra kao logička 0. Treba voditi računa da napon veći od + 3,3 V ne bude isporučan na I/O pinove, jer veliki naponi mogu lako da oštete vaš Pi. Raspberry Pi 3, kao i ostali Pi računari u porodici, nemaju zaštitno kolo od prenapona.

## 2.4 Postavljanje Wi-Fi-ja i udaljenog pristupa

Vrlo je verovatno da želite da pristupite Raspberry Pi-u 3 daljinskim putem sa radne površine ili sa prenosnog računara. Najlakša opcija ovde je da na vašem Pi računaru omogućite Wi-Fi i da mu zatim pristupite sa računara, koristeći SSH protokol klijenta. Ovaj protokol zahteva server i klijent. Server je vaš Pi računar, a klijent je vaš stoni ili prenosni računar. U ovom odeljku ćemo videti kako da na Pi računaru omogućimo Wi-Fi i kako da mu pristupimo daljinskim putem.

### Podešavanje Wi-Fi-ja

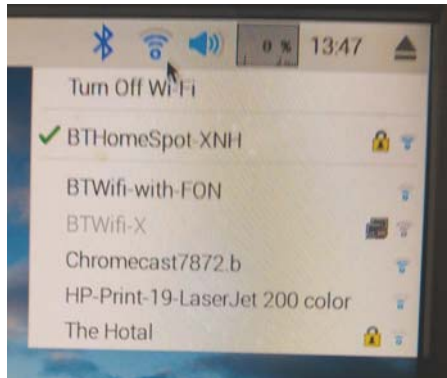
- Da biste omogućili Wi-Fi na vašem Pi-u, pratite sledeće korake:
- Kliknite na ikonu Wi-Fi, tj. na par crvenih krstića u gornjem desnom uglu ekrana.
- Izaberite vaš Wi-Fi ruter sa prikazane liste (pogledajte sliku 2.3).



Slika 2.3 Izaberite vaš Wi-Fi sa liste

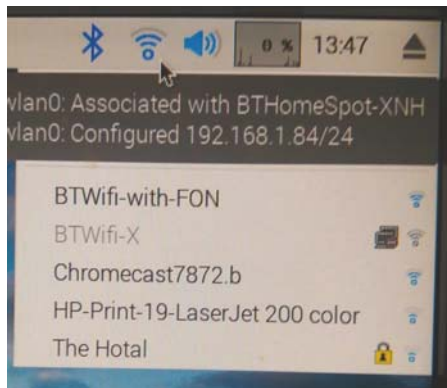
- Unesite lozinku za vaš Wi-Fi ruter.

- Ikona WiFi trebalo bi da postane tipična Wi-Fi slika. Ako sada kliknete na ikonu, trebalo bi da vidite zeleni znak potvrde pored odabranog rutera, kao što je prikazano na slici 2.4.



Slika 2.4 Uspešno povezivanje sa Wi-Fi mrežom

- Da biste videli IP adresu vaše Wi-Fi veze, postavite miš iznad ikone Wi-Fi, kao što je prikazano na slici 2.5. U ovom primeru IP adresa je 192.168.1.84.



Slika 2.5 IP adresa naše veze

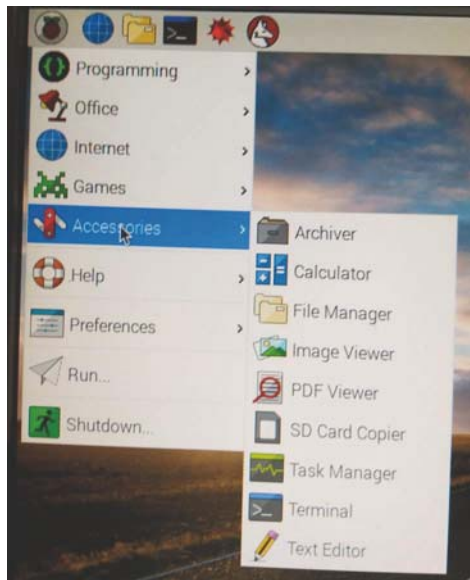
## Daljinski pristup

Program koji ćemo koristiti za pristup našem Raspberry Pi-u 3 zove se **Putty** sa SSH protokolom. Ovo su koraci za preuzimanje i upotrebu aplikacije Putty:

- Preuzmite Putty sa sledećeg linka (ili u pretraživaču Google potražite „Download Putty“).

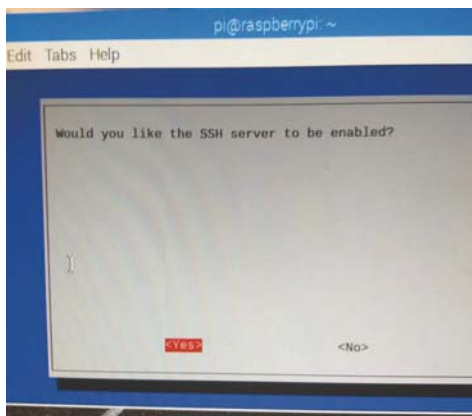
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

- Zbog bezbednosnih razloga SSH protokol je na novom operativnom sistemu podrazumevano onemogućen. Da biste ga omogućili, kliknite na meni **Applications** u gornjem levom delu ekrana, kliknite na **Accessories**, a zatim kliknite na **Terminal** (pogledajte sliku 2.6).



Slika 2.6 Pristupite meniju Terminal

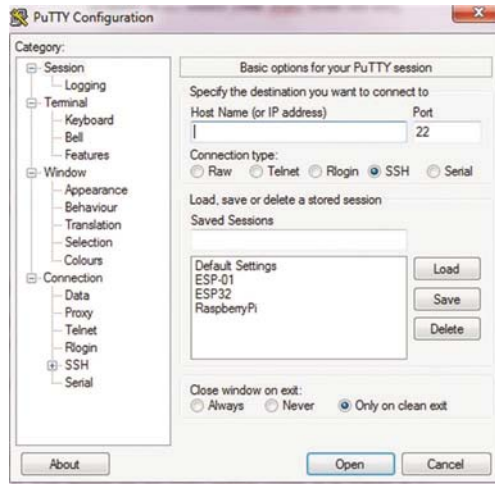
- Sada bi trebalo da budete u komandnom odzivniku Raspberry Pi 3. Tip: `sudo raspi-config`
- Da biste ušli u meni za konfiguraciju i izabrali **Interface Options**. Idite na **P2 SSH** i omogućite SSH, kao što je prikazano na slici 2.7.



Slika 2.7 Omogućite SHH server

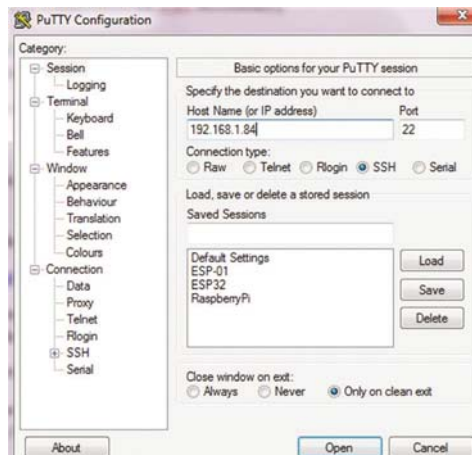
- Kliknite na **<Finish>** da biste izašli iz konfiguracionog menija. Sada bi trebalo da se vratite u komandni režim, koji je identifikovao odzivnik:  
`pi@raspberrypi:~ $`

- Putty je samostalan program i nema potrebe da ga instalirate. Jednostavno kliknite dva puta da biste ga pokrenuli. Trebalo bi da vidite ekran za pokretanje softvera Putty kao na slici 2.8.



Slika 2.8 Putty početni ekran

- Proverite da li je tip veze SSH i unesite IP adresu vašeg Raspberry Pi 3. Kliknite na Open, kao što je prikazano na slici 2.9.



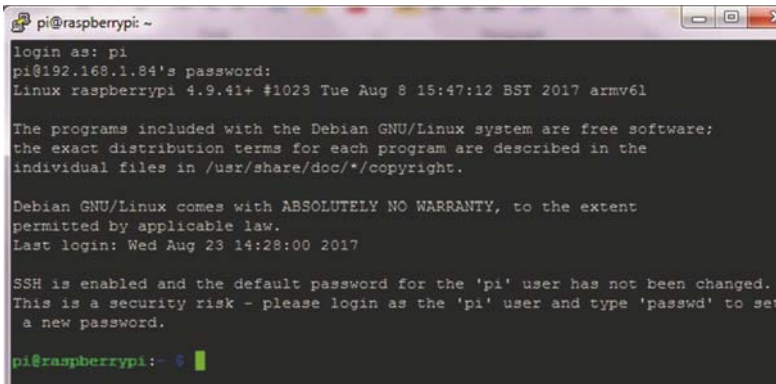
Slika 2.9 Unesite IP adresu

- Poruka koja je prikazana na slici 2.10 će biti prikazana na ekranu računara prvi put kada pristupite Raspberry Pi-u 3. Kliknite na Yes da biste prihvatili ovo bezbedonosno upozorenje.



Slika 2.10 Kliknite na Yes za prihvatanje

- Tada ćete biti upitani za korisničko ime i lozinku. Zadate vrednosti su:  
Korisničko ime: **pi**  
Lozinka: **raspberry**
- Nakon uspešne prijave, trebalo bi da vidite Raspberry Pi komandni odzivnik kao na slici 2.



Slika 2.11 Uspešno prijavljivanje

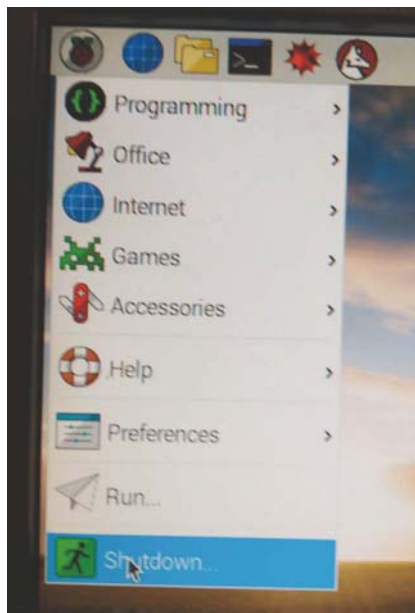
- Da biste promenili lozinku, unesite sledeću komandu:  
`passwd`
- Da biste ponovo pokrenuli Raspberry Pi ZW, unesite sledeću komandu:  
`sudo reboot`
- Da biste isključili Raspberry Pi ZW, unesite sledeću komandu. Nikada ga ne isključujte izvlačenjem kabla za napajanje, jer to može dovesti do oštećenja ili gubitka datoteka:  
`sudo shutdown -h now`



## 2.5 Isključivanje ili ponovno pokretanje u GUI režimu

Uvek morate pravilno isključiti svoj Pi računar. Da biste ga isključili u GUI režimu, sledite korake koji su navedeni ispod:

- Kliknite na meni Applications (gornji levi ugao).
- Kliknite na Shutdown (pogledajte sliku 2.12).
- Kliknite na Shutdown (ili ponovo pokrenite Pi računar po potrebi).



Slika 2.12 Isključite ili ponovno pokrenite Pi računar u GUI režimu

## 2.6 Daljinski pristup radnoj površini

Ako koristite Raspberry Pi 3 sa lokalnom tastaturom, mišem i monitorom, možete da preskočite ovaj odeljak. Ako s druge strane želite da radnoj površini pristupite daljinski putem mreže, ustanovićete da se SSH usluge ne mogu koristiti. Najlakši i najjednostavniji način na koji možete daljinski pristupiti radnoj površini sa računara je instaliranjem VNC (Virtual Network Connection) klijenta i servera. VNC server se pokreće na Pi-u, a VNC klijent radi na računaru. Koraci za instaliranje i upotrebu VNC-a dati su u nastavku:

- Povežite se na svoj Pi računar pomoću SSH-a, kao što je ranije objašnjeno. Zatim, unesite sledeću komandu da biste na svoj Pi računar instalirali program koji se zove **TightVNC** server. Videćete mnogo redova poruka. Proverite da li ima poruka o grešci:

```
sudo apt-get update
```

```
sudo apt-get install tightvncserver
```

- Pokrenite VNC server na Pi računaru unosom sledeće komande:

```
vncserver : 1
```

- Od vas će biti zatraženo da unesete i potvrdite lozinku. Ovo će biti lozinka koju ćete koristiti za daljinski pristup radnoj površini (pogledajte sliku 2.13).

```
update-alternatives: using /usr/bin/tightvncpasswd to provide
(vncpasswd) in auto mode
Processing triggers for man-db (2.7.6.1-2) ...
Processing triggers for shared-mime-info (1.8-1) ...
Processing triggers for gnome-menus (3.13.3-9) ...
Processing triggers for hicolor-icon-theme (0.15-1) ...
Setting up xfonts-base (1:1.0.4+nmu1) ...
Processing triggers for fontconfig (2.11.0-6.7) ...
pi@raspberrypi:~$ vncserver :1

You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? n

New 'X' desktop is raspberrypi:1

Creating default startup script /home/pi/.vnc/xstartup
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log

pi@raspberrypi:~$
```

Slika 2.13 Unesite lozinku za VNC server

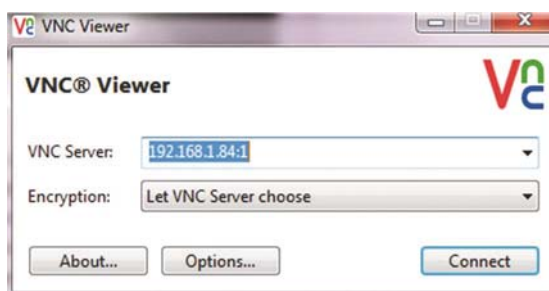
- VNC server sada radi na vašem Pi računaru. Jedina komanda koju je potrebno da unesete na Pi računar da biste pokrenuli VNC server je:

```
vncserver : 1
```

- Sada moramo da podesimo VNC klijent na našem prenosnom računaru (ili stonom računaru). Na raspolaganju je mnogo VNC klijenata, ali preporučeni klijent koji je kompatibilan sa Tight-VNC-om je VNC Viewer koji možete preuzeti sa linka koji je dat u nastavku. Imajte na umu da ovaj program nije besplatan, ali je dostupna 30-dnevna besplatna probna verzija. Treba da se registrujete da biste dobili licencu za probu, a zatim tu licencu primenite na softver koji ćete besplatno koristiti 30 dana:

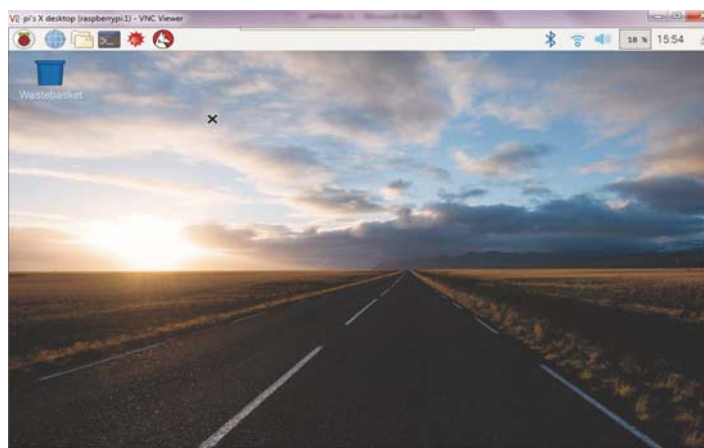
<http://www.realvnc.com>

- Preuzmite program VNC Viewer u odgovarajući direktorijum na računaru.
- Dva puta kliknite da biste ga instalirali i unesite potrebnu licencu. Pokrenite program **VNC Viewer** dvostrukim klikom na njegovu ikonu na radnoj površini. Unesite IP adresu vašeg Raspberry Pi-a 3, a zatim **1**, kao što je prikazano na slici 2.14 i kliknite na **Connect**.



Slika 2.14 Unesite IP adresu

Unesite prethodno izabranu lozinku. Sada bi trebalo da vidite Raspberry Pi 3 radnu površinu prikazanu na vašem prenosnom (ili stonom) računaru kao na slici 2.15 i da pristupite svim aplikacijama radne površine daljinskim putem.



Slika 2.15 Raspberry Pi 3 radna površina prikazana na prenosnom računaru

## 2.7 Kreiranje i pokretanje programa Python

Programiraćemo naš Raspberry Pi 3, koristeći programski jezik Python. Korisno bi bilo da pogledamo kreiranje i pokretanje jednostavnog programa Python na Pi računaru. U ovom odeljku ćemo prikazati poruku **Hello From Raspberry Pi 3** na ekranu računara.

Kao što je opisano u nastavku, postoje 3 metode pomoću kojih možemo kreirati i pokretati Python programe na Raspberry Pi 3:

### Metod 1 - Interaktivno iz komandnog odzivnika

U ovoj metodi ćemo se prijaviti na Raspberry Pi 3 pomoću SSH-a, a zatim ćemo kreirati i pokrenuti naš program interaktivno. Ova metoda je odlična za male programe. Koraci su dati u nastavku:

- Prijavite se na Raspberry Pi 3 pomoću SSH-a.
- U komandnoj liniji unesite **python**. Trebalo bi da vidite Python komandni režim koji identifikuju tri znakova >>>.
- Ukucajte program:  
print ("Hello From Raspberry Pi 3")
- Traženi tekst će se interaktivno prikazivati na ekranu, kao što je prikazano na slici 2.16.

```
pi@raspberrypi:~ $ python
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
[GCC 6.3.0 20170124] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello From Raspberry Pi 3")
Hello From Raspberry Pi 3
>>> █
```

Slika 2.16 Interaktivno pokretanje programa

### Metod 2 - Kreirajte Python datoteku u komandnom režimu

U ovoj metodi ćemo se prijaviti na naš Raspberry Pi 3 pomoću SSH-a kao i ranije, a zatim kreirati Python datoteku. Ova datoteka je jednostavna tekstualna datoteka sa nastavkom **.py**. Možemo koristiti uređivač teksta, kao što je **nano**, da bismo kreirali našu datoteku. U ovom primeru kreirana je datoteka pod nazivom **hello.py**, pomoću **nano** uređivača teksta. Na slici 2.17 prikazan je sadržaj datoteke **hello.py**. Na ovoj slici je takođe prikazano kako se pokreće datoteka u Pythonu. Primetićete da se program pokreće unosom komande:

```
>>> python hello.py
```

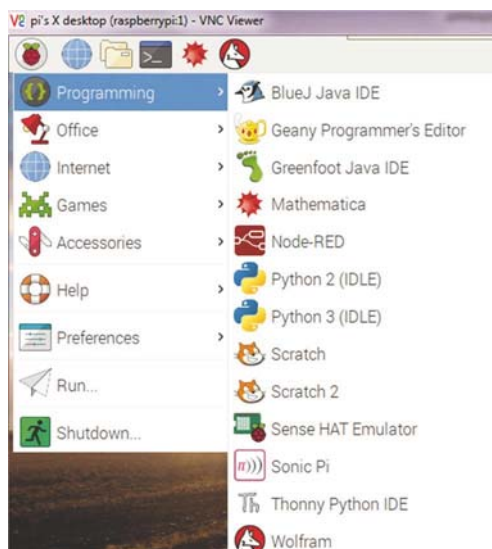
```
pi@raspberrypi:~ $ ls hello.py
hello.py
pi@raspberrypi:~ $ cat hello.py
print ("Hello From Raspberry Pi Zero W")
pi@raspberrypi:~ $ python hello.py
Hello From Raspberry Pi Zero W
pi@raspberrypi:~ $ █
```

Slika 2.17 Kreiranje i pokretanje Python datoteke

### Metod 3 - Kreirajte Python datoteke u GUI režimu

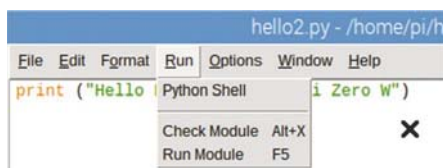
Na ovaj način ćemo se prijaviti u naš Raspberry Pi 3 pomoću VNC-a, a zatim kreirati i pokrenuti naš program u GUI režimu. Koraci su dati u nastavku:

- Kliknite na meni Application.
- Kliknite na Programming i izaberite Python 2 ili Python 3 (pogledajte sliku 2.18).



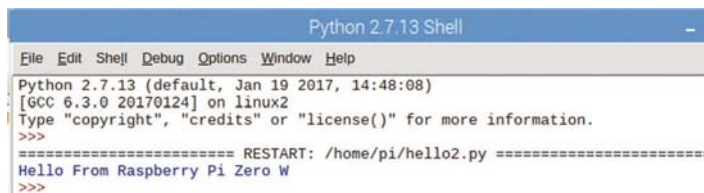
Slika 2.18 Izaberite Python 2 programiranje

- Trebalo bi da vidite režim komande Python koji je identifikovan znakovima >>>.
- Kliknite na **File**, a zatim na **New File** i napišite svoj program.
- Sačuvajte datoteku, tako što ćete joj dati naziv (npr. hello2).
- Pokrenite program klikom na **Run**, a zatim kliknite na **Run Model**, kao što je prikazano na slici 2.19.



Slika 2.19 Pokrenite program

- Biće prikazan novi ekran sa izlazom programa kao na slici 2.20.



Slika 2.20 Izlaz programa

### **Koju metodu izabrati?**

Izbor metode zavisi od veličine i složenosti programa. Mali programi se mogu pokrenuti interaktivno, bez kreiranja programske datoteke. Veći programi se mogu kreirati kao Python datoteke, a zatim se mogu izvršiti u komandnom režimu ili u GUI režimu. U ovoj knjizi će biti kreirane programske datoteke za sve Python programe.

## **2.8 GPIO biblioteka**

GPIO biblioteka se zove RPi.GPIO i već bi trebalo da je instalirana na vašem Raspberry Pi-u 3. Ova biblioteka mora biti dodata na početku vaših Python programa ukoliko želite da koristite GPIO funkcije. Ovo je iskaz za dodavanje ove biblioteke:

```
import RPi.GPIO as GPIO
```

Ako se prikaže greška prilikom pokušaja uvoza GPIO biblioteka, onda biblioteka možda nije instalirana. Unesite sledeće komande dok ste u komandnom režimu (koji je identifikovan odzivnikom **pi@raspberrypi:~ \$**) da biste instalirali GPIO biblioteku (znakovi koje bi trebalo da unesete su podebljani):

```
pi@raspberrypi: ~ $ sudo apt-get update  
pi@raspberrypi: ~ $ sudo apt-get install python-dev  
pi@raspberrypi: ~ $ sudo apt-get install python-rpi.gpio
```

GPIO obezbeđuje brojne korisne funkcije. Dostupne funkcije su date u sledećim odeljcima.

### **2.8.1 Numerisanje pinova**

Postoje dva načina na koja možemo ukazati na GPIO pinove. Prvi način je korišćenje BOARD numerisanja u kome se koriste pinovi na GPIO konektoru Raspberry Pi-a 3. Da biste koristili metodu BOARD, unesite sledeći skaz:

```
GPIO.setmode (GPIO.BOARD)
```

Drugi sistem numerisanja, poznat i kao BCM metoda, je poželjna metoda i koristi brojeve kanala koji su dodeljeni pinovima. Ova metoda zahteva da znate koji se broj kanala odnosi na koji pin na ploči. U ovoj knjizi ćemo koristiti ovu drugu metodu. Da biste koristili BCM metodu, unesite sledeći iskaz:

```
GPIO.setmode (GPIO.BCM)
```

GPIO je 40-pinsko zaglavlje, koje je postavljeno na jednoj strani ploče. U Dodatku A prikazana je konfiguracija GPIO pinova na Raspberry Pi-u 3.

### **2.8.2 Konfiguracija kanala (I/O pinovi porta)**

#### **Konfiguracija ulaza**

Morate da konfigurišete kanale (ili pinove porta) koje koristite, bez obzira da li su oni ulazni ili izlazni kanali. Sledeći iskaz se koristi za konfiguiranje kanala kao ulaza.

Ovde se kanal za razvoj projekata odnosi na broj kanala na osnovu prethodnog iskaza **set-mode**:

**GPIO.setup(channel, GPIO.IN)**

Kada na ulazni pin nije ništa povezano, podaci na ovom ulazu nisu definisani. Dodatnim parametrima možemo odrediti ulazni konfiguracioni iskaz za povezivanje pull-up ili pull-down otpornika, pomoću softvera na ulaznom pinu. Ovo su neophodni iskazi:

Za pull-up otpornike:

**GPIO.setup(channel, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)**

Za pull-down otpornike:

**GPIO.setup(channel, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)**

Na ulaznom pinu možemo otkriti promenu ivice ulaznog signala. Promena ivice je kada se signal menja iz LOW u HIGH (rastuća ivica) ili iz HIGH u LOW (opadajuća ivica). Na primer, pritisak na taster može izazvati promenu ivice na ulazu pina. Iskazi koji su dati u nastavku mogu se koristiti za čekanje ivice ulaznog signala. Ovo su funkcije blokiranja, tj. program će sačekati dok se na ulaznom signalu ne detektuje određena ivica. Na primer, ako je ovo taster, program će sačekati dok se ne pritisne taster:

Za čekanje rastuće ivice:

**GPIO.wait\_for\_edge(channel, GPIO.RISING)**

Za čekanje opadajuće ivice:

**GPIO.wait\_for\_edge(channel, GPIO.FALLING)**

Takođe možemo da sačekamo dok se ne detektuje rastuća ili opadajuća ivica, koristeći sledeći iskaz:

**GPIO.wait\_for\_edge(channel, GPIO.BOTH)**

Možemo koristiti funkciju detekcije događaja sa ulaznim pinom. Na ovaj način možemo izvršiti kod za detektovanje događaja kad god je detektovan. Događaji mogu biti rastuća ivica, opadajuća ivica ili promena na bilo kojoj ivici signala. Detekcija događaja se obično koristi u petljama u kojima možemo da proveravamo događaj dok izvršavamo drugi kod.

Na primer, da biste dodali detekciju rastućih događaja u ulazni pin koristite sledeći iskaz:

**GPIO.add\_event\_detect(kanal, GPIO.RISING)**

Možemo proveriti da li se događaj desio pomoću sledećeg iskaza:

```
If GPIO.event_detected(channel):
```

```
.....  
.....
```

Detekcija događaja se može ukloniti pomoću sledećeg iskaza:

```
GPIO.remove_event_detect(channel)
```

Takođe možemo koristiti objekte prekida (ili povratne pozive) da bismo detektovali događaj. Ovde se događajem upravlja unutar korisničke funkcije. Glavni program izvršava svoje uobičajene zadatke i čim se događaj desi, program prestaje sve što radi i prelazi na upravljanje događajem funkcija. Na primer, iskaz koji je dat u nastavku se može koristiti za dodavanje upravljanja događajem zasnovanim na prekidu u naše programe na rastućoj ivici ulaznog signala. U ovom primeru kod za upravljanje događajem je funkcija koja je nazvana **MyHandler**:

```
GPIO.add_event_detect(channel, GPIO.RISING, callback=MyHandler)
```

```
.....  
.....
```

```
def MyHandler(channel):
```

```
.....  
.....
```

Možemo dodati više prekida, koristeći funkciju `add_event_callback`. Ovde se funkcije povratnog poziva izvršavaju uzastopno:

```
GPIO.add_event_detect(channel, GPIO.RISING)
```

```
GPIO.add_event_callback(channel, MyHandler1)
```

```
GPIO.add_event_callback(channel, MyHandler2)
```

```
.....  
.....
```

```
def MyHandler1(channel):
```

```
.....  
.....
```

```
def MyHandler2(channel):
```

```
.....  
.....
```

Kada u projektima koristimo mehaničke prekidače, dobijamo ono što je poznato kao problem odskakanja prekidača. To se događa kada kontakti prekidača odskoče mnogo puta dok se ne „smire“. Odskakanje prekidača može da generiše nekoliko impulsa pre nego što se prekidač „smiri“. Možemo da izbegnemo probleme u vezi sa odskakanjem prekidača u hardveru ili softveru.



GPIO biblioteka obezbeđuje parametar pod nazivom `bouncetime` koji se može koristiti za uklanjanje problema odskakanja prekidača. Primer upotrebe ovog parametra prikazan je ispod, gde se pretpostavlja da je vreme odskakanja prekidača 10 ms:

```
GPIO.add_event_detect(channel,GPIO=RISING,callback=MyHandler, bouncetime=10)
```

Takođe možemo da upotrebimo iskaz `bouncetime` da bismo odredili vreme odskakanja za prekidač na sledeći način:

```
GPIO.add_event_callback(channel, MyHandler, bouncetime=10)
```

Da bismo pročitali stanje ulaznog pina, možemo upotrebiti sledeći iskaz:

```
GPIO.input(channel)
```

### **Konfiguracija izlaza**

Sledeći iskaz se koristi za konfigurisanje kanala kao izlaza. Ovde se kanal odnosi na broj porta zasnovanim na ranije navedenom iskazu **setmode**:

```
GPIO.setup(channel, GPIO.OUT)
```

Možemo odrediti vrednost za izlazni pin tokom njegovog podešavanja. Na primer, možemo da konfigurišemo kanal kao izlaz i da istovremeno podesimo njegovu vrednost na HIGH (+3,3 V):

```
GPIO.setup(channel, GPIO.OUT, initial=GPIO.HIGH)
```

Za slanje podataka na pin izlaznog porta možemo koristiti sledeći iskaz:

```
GPIO.output(channel, value)
```

Gde vrednost može biti 0 (ili `GPIO.LOW`, odnosno `False`), ili 1 (ili `GPIO.HIGH`, odnosno `True`).

Na kraju programa trebalo bi da vratimo sve iskorišćene resurse u operativni sistem. To se postiže dodavanjem sledećeg iskaza na kraju našeg programa:

```
GPIO.cleanup()
```

## **2.9 Ciklus razvoja projekta Raspberry Pi**

Ciklus razvoja projekata generalno uključuje razvoj hardvera i softvera. U jednostavnim projektima u kojima se ne koristi spoljni hardver, možemo jednostavno koristiti Raspberry Pi ploču onakvu kakva jeste i tada je potreban samo razvoj softvera. Međutim, većina projekata su složeniji i zahtevaju dodatne spoljne komponente, kao što su LED, motori, displeji, tastature itd. Tada programer mora da osigura da hardver bude pravilno podešen i da je u potpunosti funkcionalan pre nego što se kreira bilo koji softver. Ako hardver nije pravilno podešen, izgubićemo vreme pokušavajući da razvijemo softver. Razvoj hardvera može zahtevati dodatne veštine, kao što su poznavanje povezivanja i pravilno korišćenje različitih elektronskih komponenata na sistemima koji se zasnivaju na mikrokontrolerima.

Ciklus razvoja projekta, posebno ciklus razvoja softvera, je iterativni proces u kojem programer mora da se vrati i da nastavi da unosi izmene u kod dok ne dobije traženi odgovor od sistema u razvoju. Postoje razne alatke koje mogu biti korisne tokom razvoja projekta. Na primer, dijagrami toka, jezik za opisivanje programa (PDL - Program Description Program), UML, mašine stanja i mnoge druge alatke koje se mogu koristiti tokom razvojnog ciklusa.

U projektima za Raspberry Pi u ovoj knjizi korišćićemo programski jezik Python. Ovaj jezik je veoma moćan interaktivni programski jezik koji se uči kao prvi programski jezik na mnogim univerzitetima i fakultetima širom sveta. Python je dostupan na računarima Raspberry Pi i podržava veliki broj biblioteka koje znatno pojednostavljaju zadatak programiranja.

U ostatku ovog poglavlja kreiraćemo jednostavan projekat, koristeći dvobojnu LED ploču KY-011, kao i u prethodnom poglavlju.

## 2.10 Projekat – Treperenje dvobojnog LED-a

**Opis** - Ovo je možda najjednostavniji projekat koji možete dizajnirati, koristeći Raspberry Pi. U ovom projektu dvobojni LED sa zajedničkom katodom povezan je sa Raspberry Pi-em. LED boje trepere naizmenično brzinom od jedne sekunde.

**Cilj** - Cilj ovog projekta je da se prikaže kako se program može napisati i pokrenuti na Raspberry Pi-u. Projekat takođe prikazuje kako se koriste neki od I/O i funkcija za podešavanje vremena na Raspberry Pi-u.

**Senzor koji se koristi** – U ovom projektu se koristi senzor KY-011, kao i u prethodnom poglavlju. Senzor i njegovi pinovi prikazani su na slici 1.3. Senzor ima 3 pina - crveni LED pin, zeleni LED pin i GND pin. **Zeleni LED pin označen je slovom S na ploči, radi identifikacije.** LED se uključuje kada se logička 1 primeni na njegove pinove.

**Blok dijagram** – Blok dijagram projekta prikazan je na slici 2.21.

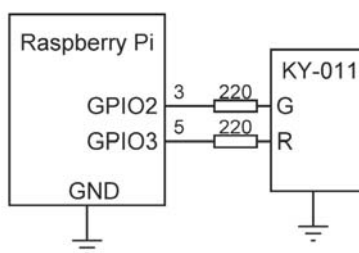


Slika 2.21 Blok dijagram projekta

**Električna šema** - Električna šema projekta prikazana je na slici 2.22. Zeleni i crveni LED pinovi su povezani sa pinovima porta GPIO 2 (pin 3) i GPIO 3 (pin 5) Raspberry Pi-a putem otpornika za ograničavanje struje. GND pin je povezan sa Raspberry Pi GND pinom. Vrednost otpornika za ograničavanje struje izračunava se na sledeći način:

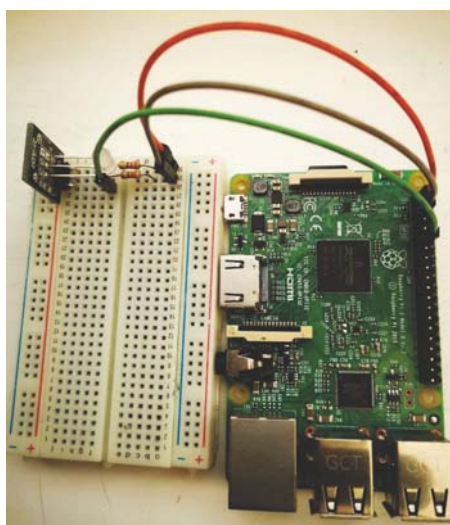
Visoki napon GPIO pina je 3,3 V. Napon na LED-u je oko 2,0 V (za crveni LED je 1,8 V, a za zeleni LED 2,8 V). Propusna struja u LED-u zavisi od tipa LED-a koji se koristi i količine potrebne svetline. Pod pretpostavkom da smo izabrali propusnu struju od 5 mA, tada je vrednost otpornika za ograničavanje struje (svetlina LED-a se može povećavati, tako što će se povećavati propusna struja):

$R = (3,3 \text{ V} - 2 \text{ V}) / 5 \text{ mA} = 260 \text{ oma}$ . Izabrali smo otpornike od 220 oma da bismo omogućili 6 mA propusne struje za svaki LED (za zeleni LED možete koristiti manju vrednost, npr. 150 oma)



Slika 2.22 Električna šema projekta

**Konstrukcija** - Projekat je izrađen na prototipskoj ploči, kao što je prikazano na slici 2.23. Kratkospojnici se koriste za povezivanje senzora KY-011 sa digitalnim pinovima porta 2 i 3 i GND-om Raspberry Pi-a.



Slika 2.23 Projekat koji je konstruisan na prototipskoj ploči

**Listing programa** - Program se zove **rpitwoled.py**, a listing je prikazan na slici 2.24. Na početku programa uvezene su biblioteke **RPI.GPIO** i **time**. GPIO brojevi porta 2 i 3 se dodeljuju zelenom LED pinu i crvenom LED pinu, tim redom. LED-ovi se zatim konfiguriraju kao izlazi. Ostatak programa se izvršava u beskonačnoj petlji. Unutar ove petlje zeleni i crveni LED-ovi trepere naizmenično, sa kašnjenjem od jedne sekunde između svakog izlaza.

Program možete pokrenuti iz komandne linije unosom sledeće komande (znakovi koje je uneo korisnik podebljani su radi jasnoće):

```
pi@raspberrypi:~ $ python rpitwoled.py

#-----
#   TWO COLOUR LED FLASHING
#   =====
#
# In this project the two colour LED KY-011 is used. The
# green and red LED pins are connected to Raspberry Pi port
# pins GPIO2 and GPIO3. The LEDs flash alternately with a
# rate of one second
#
# Author: Dogan Ibrahim
# Date  : April 2019
# File  : rpitwoled.py
#-----

import RPi.GPIO as GPIO      # import Rpi library
import time                  # import time library
GPIO.setwarnings(False)     # disable warnings
GPIO.setmode(GPIO.BCM)     # set BCM mode

GreenLED = 2                 # green LED
RedLED = 3                   # red LED
ON = 1
OFF = 0

GPIO.setup(GreenLED, GPIO.OUT) # Configure as output
GPIO.setup(RedLED, GPIO.OUT)   # configure as output

while True:                  # do forever
    GPIO.output(GreenLED, ON)  # green LED ON
    GPIO.output(RedLED, OFF)   # red LED OFF
    time.sleep(1)              # wait 1 second
    GPIO.output(GreenLED, OFF) # green LED ON
    GPIO.output(RedLED, ON)    # red LED ON
    time.sleep(1)              # wait 1 seconds
```

Slika 2.24 Listing programa projekta

**Napomena:** Možda će vam biti lakše da kreirate i da pokrenete Python programe iz GUI desktop interfejsa (IDLE 2), jer se odgovarajuća uvlačenja automatski stavljaju u vaš kod dok kucate.

**Šta smo naučili** - U ovom projektu smo naučili kako da koristimo sledeće Raspberry Pi funkcije:

<b>import:</b>	uvozi bibliotečki modul u program
<b>GPIO.setmode(mode):</b>	podešava režim upotrebe GPIO pina u programu
<b>GPIO.setwarnings(mode):</b>	onemogućava (False) ili omogućava (True) poruke upozorenja
<b>GPIO.setup(port pin, mode):</b>	postavlja pinski režim rada kao ulaz ili izlaz
<b>GPIO.output(port pin, value):</b>	šalje vrednost (0 ili 1) na navedeni pin porta
<b>time.sleep(n):</b>	kreira kašnjenje od <b>n</b> milisekundi

## 2.11 Rezime

U ovom poglavlju smo pregledali konfiguraciju pinova GPIO konektora i kako se omogućava Wi-Fi modul i pristupa Raspberry Pi-u daljinski putem sa stonog ili prenosnog računara. Osim toga, pregledali smo dizajn jednostavnog projekta koji se zasniva na KY-011 dvobojnom LED-u u kome se koristi programski jezik Python.

U sledećem poglavlju ćete videti kako se instalira firmver na ESP32 mikrokontroleru i kako se koristi u projektima.