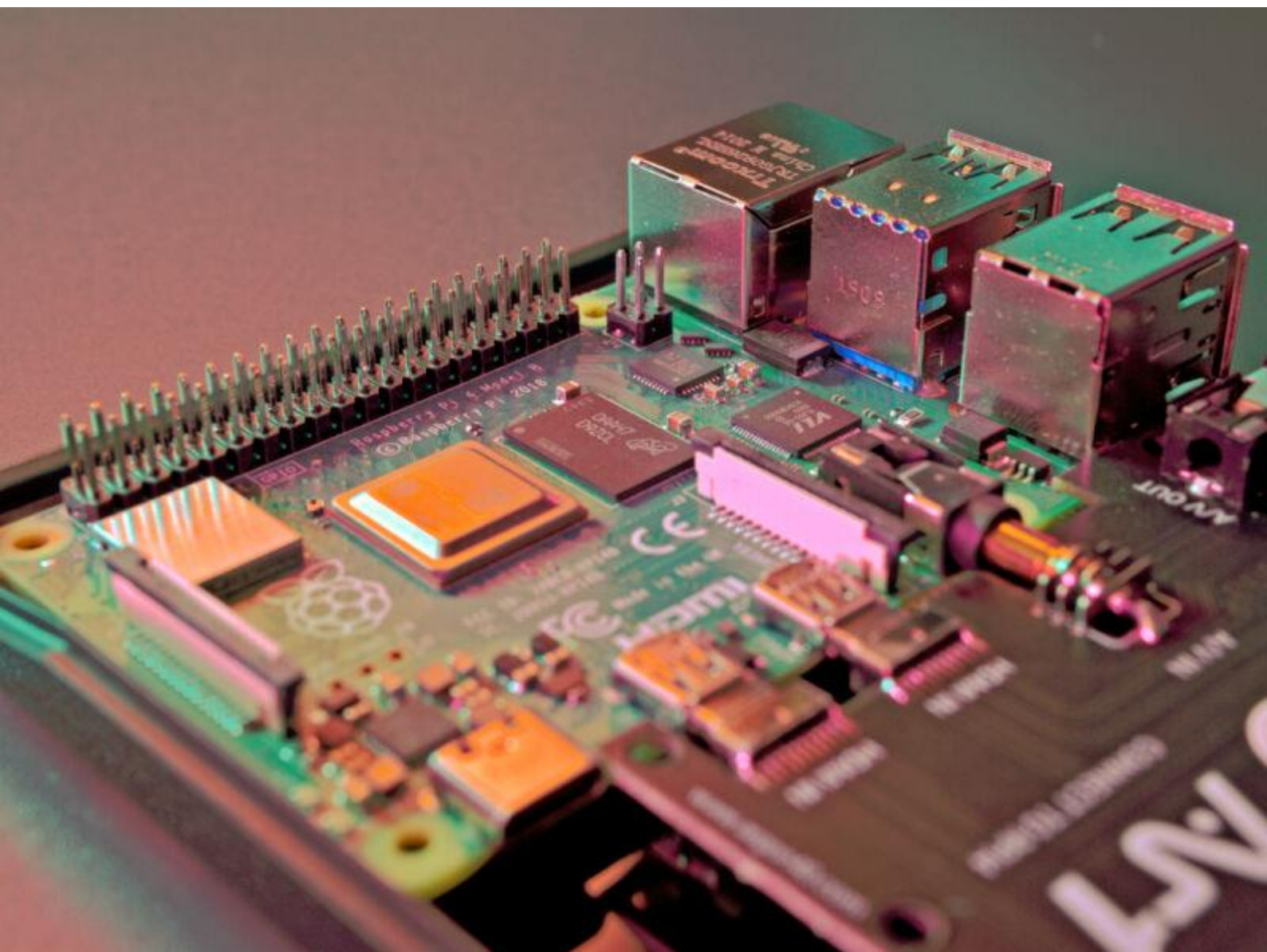


Priročnik Raspberry Pi

Zagon, namestitev, konfiguracija in programiranje



Boris Vertuš

PRIROČNIK RASPBERRY PI, Zagon, namestitev, konfiguracija in programiranje

Boris Vertuš

Oblikovanje in prelom: Boris Vertuš

Besedilo in slike: Boris Vertuš

Letnica: 2023

Kraj: Novo mesto

Objava: digitalna knjižnica (format PDF)

Založnik: samozaložba

Spletni naslov za dostop: <https://borisvertus.splet.arnes.si/files/2023/04/Prirocnik-Raspberry-PI-Zagon-namestitev-konfiguracija-in-programiranje.pdf>

CIP – Kataložni zapis o publikaciji

Narodna in univerzitetna knjižnica, Ljubljana

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani

COBISS.SI-ID 147461123

ISBN 978-961-07-1557-3 (PDF)

Kazalo vsebine

1. Raspberry Pi – Uvod	1
Modeli in generacije	1
2. Raspberry Pi — Prvi koraki	7
Uporaba	7
Nakup naprave	7
Zahteve	8
Združljive naprave	8
3. Raspberry Pi — operacijski sistem	10
Izbira OS	10
4. Povezava Raspberry Pi	13
Vrata in priključki	13
Modul kamere	15
Zagon naprave RPi	16
5. Raspberry Pi — konfiguracija	24
RPi konfiguracija	24
Konfiguracija Wi-Fi	25
6. Raspberry Pi — Delo z Linux OS	27
PIXEL namizno okolje	27
Navigacija po namizju	27
Meni z aplikacijami	28
Dostop do podmenijev	28
7. Raspberry Pi — PIXEL namizno okolje	29
Nameščanje novih aplikacij	29
8. Raspberry Pi — Linux Shell	31
Odpiranje okna Shell Window	31
Pregled podatkov in datotek	32

Pravice	34
Uporabni ukazi terminala	35
9. Raspberry Pi — GPIO	36
PiniGPIO	36
Alternativne funkcije	37
Povezava GPIO na Raspberry Pi	37
Naprave I2C	38
10. Raspberry Pi — Upori in upornost	40
Izračun upornosti	40
Vrste uporov	40
Karakteristične vrednosti upornosti	41
11. Raspberry Pi — načrtovanje vezave	44
12. Raspberry Pi — primeri projektov	45
Zaznava pritiska tipke	45
Priklop lučke LED	48
Izmenično utripanje svetilk LED	52
Temperaturni senzor	54
Izmerimo razdaljo	59
Izdelajmo predvajalnik zvokov (brenčalo)	62
13. Raspberry Pi — Pico simulator	65
Pico	65
Micropython	66
Utrpanje svetilke LED	67
Uporaba	68
14. Zaključek	69

Kazalo slik

Slika 1: Raspberry PI imager.....	10
Slika 2: Izbira OS za RPi	11
Slika 3: Vstavimo kartico z izbranim OS v napravo RPi	12
Slika 4: RPi model B.....	14
Slika 5: RPi model A.....	14
Slika 6: Kamera za napravo RPi	15
Slika 7: Začetna konfiguracija RPi	16
Slika 8: Izbira lokalizacije.....	17
Slika 9: Nameščanje komponent.....	17
Slika 10: Upravljanje z diskom	18
Slika 11: Nameščanje OS.....	18
Slika 12: Namestitev nalagalnika GRUB	19
Slika 13: Zaključevanje namestitve	19
Slika 14: Prvi zagon naprave RPi	20
Slika 15: Prvi zagon OS	20
Slika 16: Konfiguracija OS.....	24
Slika 17: Konfiguracija preko terminala	25
Slika 18: Okolje PIXEL.	27
Slika 19: Dostop do menija	28
Slika 20: Upravljanje s programi	30
Slika 21: Konzolna vrstica.....	31
Slika 22: Barvna shema in pripadajoče oznake.....	42
Slika 23: Prikaz izračuna na podlagi barvnih obročkov	43
Slika 24: Orodje Fritzing za načrtovanje sheme.....	44
Slika 25: Shema vezav RPi 3	46
Slika 26: povezovanje testne plošče	46
Slika 27: Shema vezave tipke	47
Slika 28: Pritisk na tipko in zaznava pritiska.....	48
Slika 29: Svetilka LED.....	49
Slika 30: Povezava dveh sijalk LED	53
Slika 31: DS18B20.....	55
Slika 32: Namestitev paketa za podporo senzorju.....	57
Slika 33: Nastavitev naprave RPi.....	57
Slika 34: 1-wire vmesnik vklop.....	58
Slika 35: Shema vezave ultrasoničnega senzorja.....	60
Slika 36: Shema vezave mini zvočnika	63
Slika 37: Simulator RPi Pico.....	65
Slika 38: GPIO shema vezav na napravi Pico.....	66
Slika 39: Utripanje svetilke LED Pico	67

1. Raspberry Pi – Uvod

Raspberry Pi je naprava, ki je bila razvita v sodelovanju s podjetjem Broadcom. Predstavlja mini računalnik, ki nam omogoča razvojno okolje z uporabo zunanjih naprav/komponent.

Naprava je v velikosti bančne kartice z dodanimi priključki. Z njo se lahko povežemo v svetovni splet, se pozabavamo z igrami, priključimo in upravljamo zunanje naprave, se učimo programiranja in mnogo drugega. Prednost je predvsem cenovna dostopnost, ki je v kombinaciji z relativno dobro zmogljivostjo in spodobnostjo učenja visoko cenjena.

Namenjena je predvsem učencem, dijakom, študentom, kar pa ne pomeni, da ni primerna za kakšne entuziasta, ki ima željo po novih znanjih in malce žilice za raziskovanje in učenje.

Raspberry Pi ali pogosto krajše kar RPi je razširjena naprava, ki omogoča izdelavo igračarskih pripomočkov, fitnes pripomočkov, zbiranja vremenskih podatkov in mnogo drugega. Omogoča prve korake v svet računalništva in svet programiranja. Uporablja se lahko kot nizkocenovni računalnik za manj zahtevne ali predele sveta, kjer je nakupna moč manjša.

Modeli in generacije

Leta 2012 je bila predstavljena prva generacija naprav RPi. Razvoj je pripeljal vse od številka ena pa do številke štiri. Nastale so tudi drugačne izvedbe, kot je npr. RPi Zero.

Prva generacija naprav RPi se je delila na štiri različne modele:

- Model A
- Model A +
- Model B
- Model B +

Kot smo omenili uvodu, gre za naprave velikosti kreditne kartice, kjer pa se je z modelom Zero spremenila velikost in oblika. Ključna razlika med modeloma A in B je bila v priključku LAN.

Raspberry Pi Zero models, ponujajo ali ne ponujajo priključkov GPIO (angl. general-purpose input output) s katerimi lahko priključimo in upravljamo zunanje naprave, kot so npr. senzorji temperature, vlage, lučke LED, motorji itn.

Podatki o velikosti

V tabeli so prikazane naprave po generacijah in modelih, da si lažje ustvarimo predstavo, kaj nam ponujajo.

Raspberry Pi različica	Datum lansiranja	Velikost	Dimenzije (v mm)
Raspberry Pi 4 Model B	2019-2020	Standard	85.6 x 56.5
Raspberry Pi 3 Model B+	2018	Standard	85.6 x 56.5
Raspberry Pi 3 Model B	2016	Standard	85.6 x 56.5
Raspberry Pi 3 Model A+	2018	Compact	65 x 56.5
Raspberry Pi Zero Wireless with Headers	2017	Mini	65 x 30 x 5
Raspberry Pi Zero Wireless	2016	Mini	65 x 30 x 5
Raspberry Pi Zero	2015	Mini	65 x 30 x 5
Raspberry Pi 2 Model B	2015	Standard	85.6 x 56.5
Raspberry Pi 1 Model B +	2014	Standard	85.6 x 56.5
Raspberry Pi 1 Model B	2012	Standard	85.6 x 56.5
Raspberry Pi 1 Model A+	2014	Compact	65 x 56.5
Raspberry Pi 1 Model A	2013	Standard	85.6 x 56.5

V naslednji tabeli so podane specifikacije strojne opreme glede na različne generacije.

Raspberry različica	Pi	Teža (v gramih)	GPIO	Hitrost Procesorja (CPE)	Število jeder	RAM
Raspberry Model B	Pi 4	46	40 Pin	1.5 GHz	Quad	1,2,4, or 8 GB
Raspberry Model B+	Pi 3	50	40 Pin	1.4 GHz	Quad	1 GB
Raspberry Model B	Pi 3	40	40 Pin	1.2 GHz	Quad	1 GB
Raspberry Model A+	Pi 3	28	40 Pin	1.4 GHz	Quad	512 MB
Raspberry Wireless Headers	Pi Zero with	10	40 Pin	1 GHz	Single	512 MB
Raspberry Wireless	Pi Zero	10	40 Pin Unpopulated	1 GHz	Single	512 MB
Raspberry Pi Zero		8	40 Pin Unpopulated	1 GHz	Single	512 MB
Raspberry Model B	Pi 2	42	40 Pin	1.2 GHz	Quad	1 GB
Raspberry Model B +	Pi 1	42	40 Pin	700 MHz	Single	512 MB
Raspberry Model B	Pi 1	38	21 Pin (26 Pin Header)	700 MHz	Single	512 MB
Raspberry Model A+	Pi 1	23	40 Pin	700 MHz	Single	512 MB
Raspberry Model A	Pi 1	30	21 Pin (26 Pin Header)	700 MHz	Single	256 MB

Povezljivost

Naslednja tabela prikazuje, kakšne so možnosti povezovanja naprav glede na modele in generacije naprav. Prikazujejo vhodno/izhodne priključke in možnosti uporabe priklopa zaslona.

Raspberry različica	Pi	Priključki USB	Ostali priključki	Napajanje	Priključki HDMI
--------------------------------	-----------	---------------------------	------------------------------	------------------	------------------------

Raspberry Pi Model B	4	2 USB3.0 2 USB2.0	1 USB-C	5.1V at 3A	2 micro-HDMI
Raspberry Pi Model B+	3	4 USB2.0	1 MicroUSB	5.1V at 2.5A	HDMI, Composite (TRRS)
Raspberry Pi Model B	3	4 USB2.0	1 MicroUSB	5.1V at 2.5A	HDMI, Composite (TRRS)
Raspberry Pi Model A+	3	1 USB2.0	1 MicroUSB	5.1V at 3A	HDMI, Composite (TRRS)
Raspberry Pi Zero Wireless with Headers		—	1 MicroUSB	5.1V at 1.2A	Mini-HDMI, GPIO Composite
Raspberry Pi Zero Wireless		—	1 MicroUSB	5.1V at 1.2A	Mini-HDMI, GPIO Composite
Raspberry Pi Zero		—	1 MicroUSB	5.1V at 1.2A	Mini-HDMI, GPIO Composite
Raspberry Pi Model B	2	4 USB2.0	1 MicroUSB	5.1V at 1.8A	HDMI, Composite (TRRS)
Raspberry Pi Model B +	1	4 USB2.0	1 MicroUSB	5.1V at 1.2A	HDMI, Composite (TRRS)
Raspberry Pi Model B	1	2 USB2.0	1 MicroUSB	5.1V at 3A	PAL and NTSC, HDMI or DSI, RCA
Raspberry Pi Model A+	1	1 USB2.0	1 MicroUSB or GPIO	5.1V at 700mA	HDMI, Composite (TRRS)
Raspberry Pi Model A	1	1 USB2.0	1 MicroUSB or GPIO	5.1V at 700mA	PAL and NTSC, HDMI or DSI, RCA

Spodnja tabela nam prikazuje, kakšne so zmožnosti komunikacije posamezne izmed naprav, ki so nam na voljo. Zanimajo nas področja videa, omrežja, tehnologije Bluetooth, Wi-Fi in druge.

Raspberry Pi Version	Video Out Quality	Video In	Ethernet	Bluetooth	Wi-Fi	External Storage
-----------------------------	--------------------------	-----------------	-----------------	------------------	--------------	-------------------------

Raspberry Pi 4 Model B	4kp60	CSI Camera Connector	Gigabit Ethernet	Bluetooth 5.0	Dual Band-2.4 GHz and 5GHz	MicroSD
Raspberry Pi 3 Model B+	1080p60	CSI Camera Connector	10/100 Mbit/s	Bluetooth 4.2/BLE	Dual Band-2.4 GHz and 5GHz	MicroSD
Raspberry Pi 3 Model B	1080p60	CSI Camera Connector	10/100 Mbit/s	Bluetooth 4.1	2.4 GHz	MicroSD
Raspberry Pi 3 Model A+	1080p60	CSI Camera Connector	—	Bluetooth 4.2/BLE	Dual Band-2.4 GHz and 5GHz	MicroSD
Raspberry Pi Zero Wireless with Headers	1080p60	CSI Camera Connector	—	Bluetooth 4.1	2.4 GHz	MicroSD
Raspberry Pi Zero Wireless	1080p60	CSI Camera Connector	—	Bluetooth 4.1	2.4 GHz	MicroSD
Raspberry Pi Zero	1080p60	CSI Camera Connector	—	—	—	MicroSD
Raspberry Pi 2 Model B	1080p60	CSI Camera Connector	10/100 Mbit/s	—	—	MicroSD
Raspberry Pi 1 Model B +	1080p60	CSI Camera Connector	10/100 Mbit/s	—	—	MicroSD

2. Raspberry Pi — Prvi koraki

V modernih časih, ko računalniki postajajo vedno manjši in lažji, se vseeno naprava RPi zdi, kot da bi prišla iz prihodnosti. Njena velikost je še vedno tako majhna, a hkrati tako uporabna naprava.

Uporaba

Z napravo RPi lahko bolj kot ne počnemo vse, kar bi počeli z namiznim računalnikom ali prenosnikom. Dostopamo lahko do interneta, odpiramo aplikacije, zaženemo igrico ali grafični vmesnik operacijskega sistema. Uporabimo jo lahko tudi kot terminalsko napravo, kjer preko ukazov upravljamo z napravo, podatki ali čim drugim.

Uporaba naprave ni omejena in omogoča.

- ☐ Igranje igric.
- ☐ Brskanje po spletu.
- ☐ Obdelavo in pisanje besedil.
- ☐ Izdelavo preglednic.
- ☐ Urejanje fotografij in video vsebin.
- ☐ Programiranje.
- ☐ Oddaljen dostop in drugo.

Najboljše pri vsem skupaj je to, da lahko spoznamo delovanje računalnika in ne zgolj njegovo uporabo. Naučimo se lahko preko različnih projektov, kako elektronske naprave delujejo in kako jih lahko programiramo.

Programiranja se lahko naučimo preko vizualnega vmesnika, kot je npr. Scratch ali pa se odločimo za koncept programiranja, kjer sami napišemo kodo npr. v okolju python. Skozi GPIO upravljamo z napravami, ki jih priključimo. Pošiljamo in beremo podatke preko vhodov/izhodov in tako pripravimo kakšno rešitev po lastnih željah.

Nakup naprave

Napravo RPi lahko kupimo preko različnih ponudnikov v Sloveniji. Na voljo so tudi različni t. i. kit kompleti, kjer imamo poleg naprave v kompletu tudi ostalo začetno opremo in nekaj senzorjev.

Zahteve

Kaj potrebujemo, da bomo napravo RPi uspešno priključili in zagnali?

Monitor

Za prikaz podatkov potrebujemo zaslon, ki ga bomo priključili preko priključka HDMI. Na voljo so nam tudi različne rešitve, kjer lahko neposredno preko vodila na napravi priključimo manjši zaslon, ki ima podporo tudi na dotik.

USB hub

Ker imajo različni modeli in različice različno število priklopov USB, nam bo prišel prav tudi razdelilec USB. Pazite le, da ima svoje napajanje, saj je moč napajanja priključka USB omejena in pogojena s samim napajanjem naprave RPi.

Tipkovnica in miška

Naprava RPi podpira in hkrati za upravljanje potrebuje tipkovnico in miško preko vtiča USB.

Kartica SD ali MicroSD

Za razliko od klasičnega računalnika, naprava RPi nima trdega diska. Namesto tega ima spominsko kartico, kamor namestimo operacijski sistem in kasneje shranjujemo vse spremembe, podatke, programe itn. Kartica mora biti dovolj velika, naj bo vsaj 16 GB in ustrezne hitrosti, da bo delovanje sistema sprejemljivo.

Adapter USB Wi-Fi

Zavisí od različice izbrane naprave. Če potrebujemo dostop do brezžičnega omrežja WiFi, nam bo USB Wifi prišel prav.

Zvočniki

Na voljo nam je tudi standardni priključek 3,5 mm za zvočnike, kamor lahko priključimo zvočnike ali slušalke za predvajanje zvoka.

Napajanje

Teoretično lahko uporabimo napajalnik telefona/tablice, če imamo ustrezní priključek. Težava je predvsem, če je napajalnik dovolj močan. Priporoča se napajalna moč 1,5 – 2 A po potrebi tudi več.

Združljive naprave

Na naslovu https://elinux.org/RPi_VerifiedPeripherals velja preveriti, če so izbrane naprave

združljive z napravami RPi. Načeloma večjih težav ni, se pa lahko pojavijo kakšni problemi. Da ne bo presenečenj, je zato priporočljivo, da se pred nakupom preveri ustreznost in združljivost.

3. Raspberry Pi — operacijski sistem

Preden pričnemo uporabljati napravi RPi, moramo na spominsko kartico naložiti izbrani operacijski sistem. Na voljo imamo več različnih distribucij, kjer se uporabnik za odloči, katera mu je najbližje. Običajno so to različne distribucije OS Linux.

Skozi čas se je postopek namestitve malo spreminjal. Danes to izvedemo najhitreje s pomočjo orodja Raspberry Pi Imager, ki nam je na voljo preko uradne spletne strani [Raspberry Pi OS – Raspberry Pi](#).



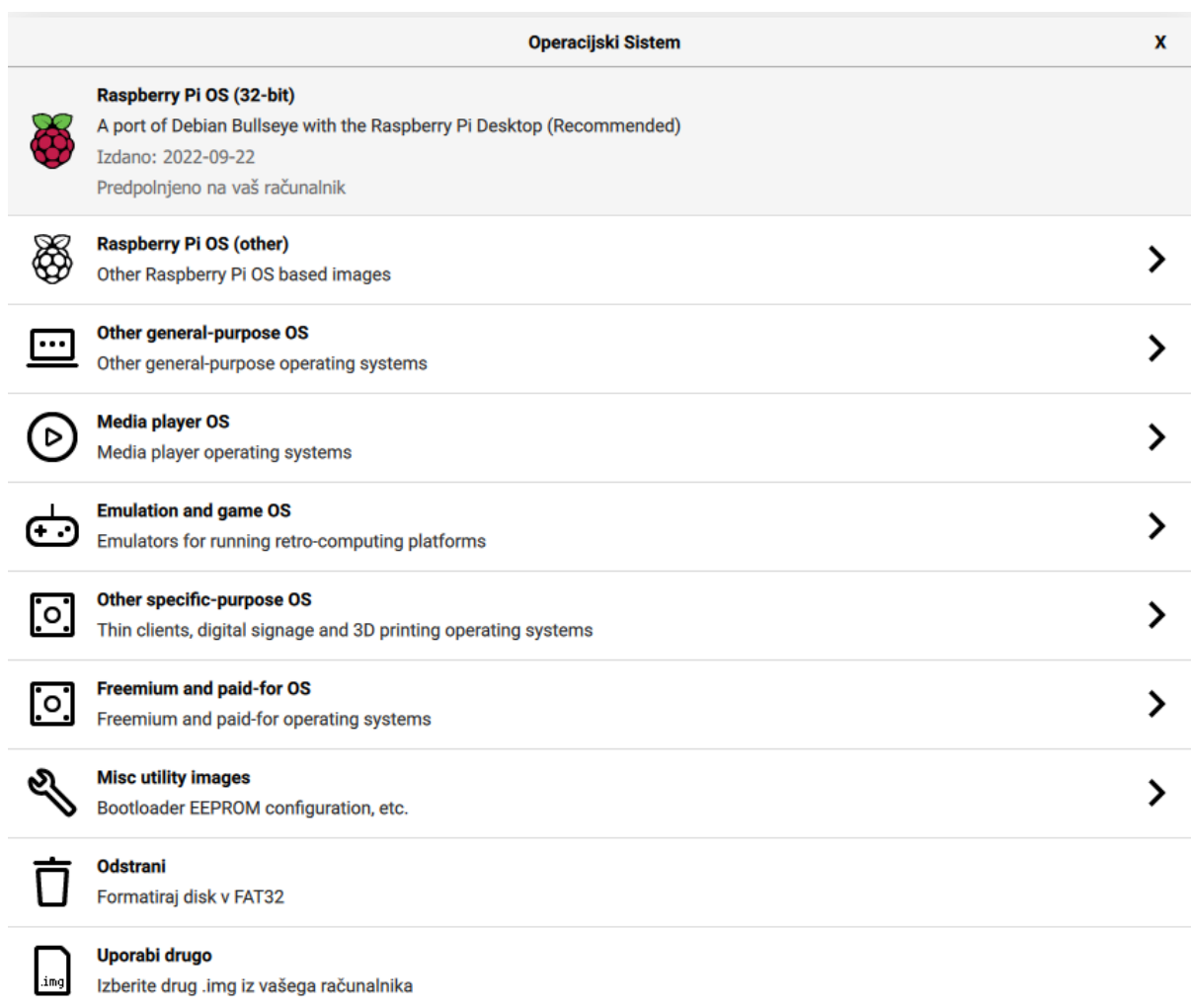
Slika 1: Raspberry Pi imager

Program prenesemo in pripravimo ustrezno spominsko kartico. Orodje nam je na voljo za vse večje OS, in sicer Windows, Linux in OS X.

Izbira OS

Sprva izberemo različico, ki jo želimo namestiti. Vidimo lahko, da je izbor zelo širok in prilagojen različnim potrebam,

ki jih bomo prilagajali namenu uporabe naprave RPi.



Slika 2: Izbira OS za RPi

Vidimo lahko, da je večina izbora namenjena sistemu Linux. Za začetnike bo najprimernejši Raspberry Pi OS, kjer so ključni programi že pred nameščeni. Izberemo le še spominsko kartico in pričnemo s postopkom »nameščanja«. Sistem se bo skopiral na kartico in ob zaključku smo pripravljeni na prvi zagon.

Uporaba okolja Linux

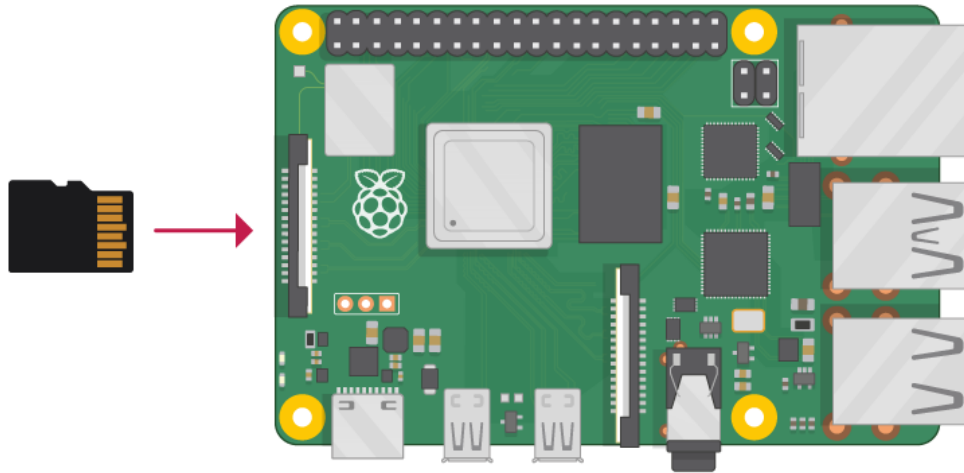
Namestitev programa za okolje Linux se razlikuje od okolja Windows in OS X. Tukaj je potrebno preko ukazne vrstice zagnati ukaz:

```
sudo apt install rpi-imager
```

Namestil so bo program, kjer bomo po enakem sistemu poskrbeli, da se bo izbrani OS namestil na spominsko kartico.

Prvi zagon RPi

Ko imamo spominsko kartico pripravljeno, jo vstavimo v ustrezno režo na napravi RPi.



Slika 3: Vstavimo kartico z izbranim OS v napravo RPi

Izvedemo zagon naprave in vmesnik nam prikaže prve korake, ki jih je potrebno izvesti ob prvem zagonu.

4. Povezava Raspberry Pi

Povezavo napravo izvedemo hitro in preprosto. Poglejmo si priključke, da bomo razumeli vse potrebno.

Vrata in priključki

Zadnja stran naprave je večinoma ravna zelena ploskev, kjer se nahaja reža za vstavitve spominske kartice. Vsi ostali priključki so na vrhni strani oz. v zgornjem delu, kjer do priključkov dostopamo. Tudi povezave GPIO se nahajajo na zgornji strani.

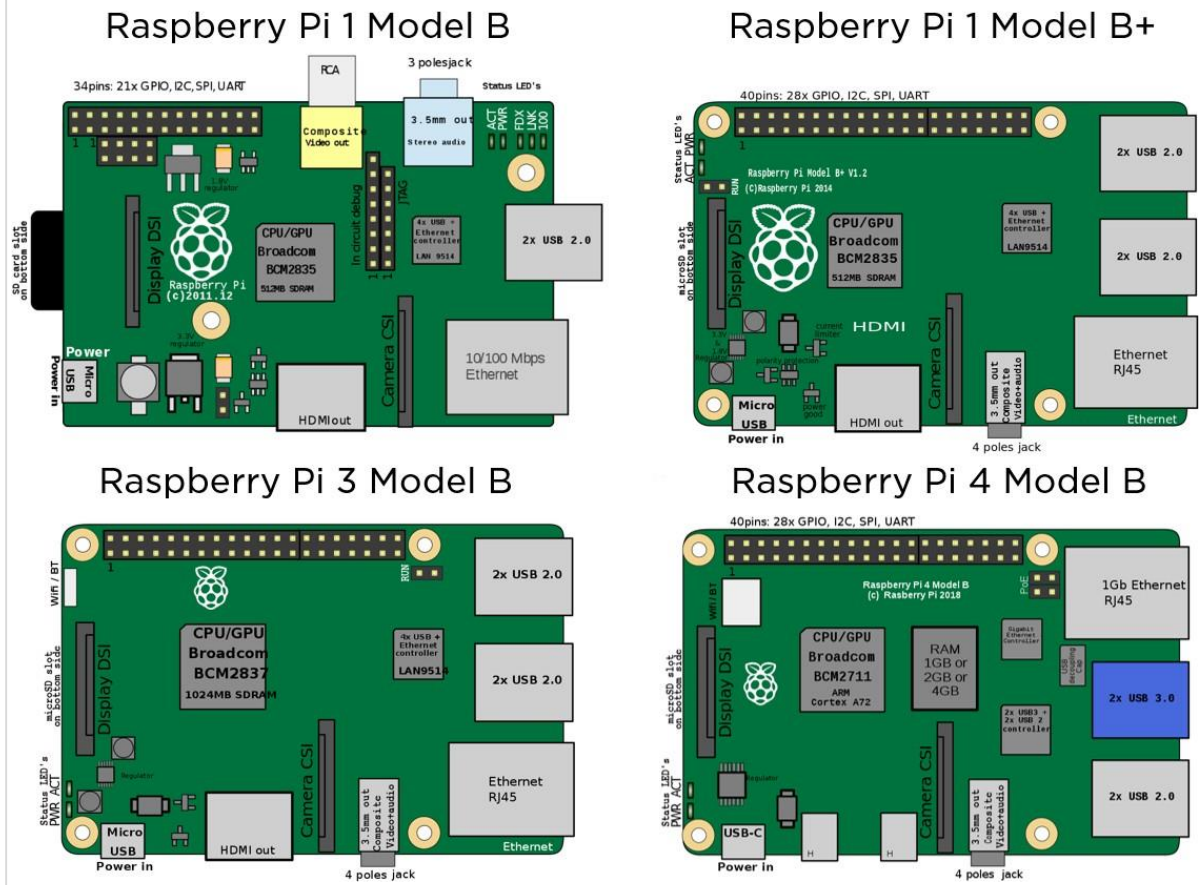
Poglejmo si diagrame, kako se priključki razlikujejo po modelih in generacijah.

Original sheme lahko najdemo tudi preko naslova: <https://core-electronics.com.au>

Diagram 1

Poglejmo si prikaz za model **Raspberry Pi Model B**:

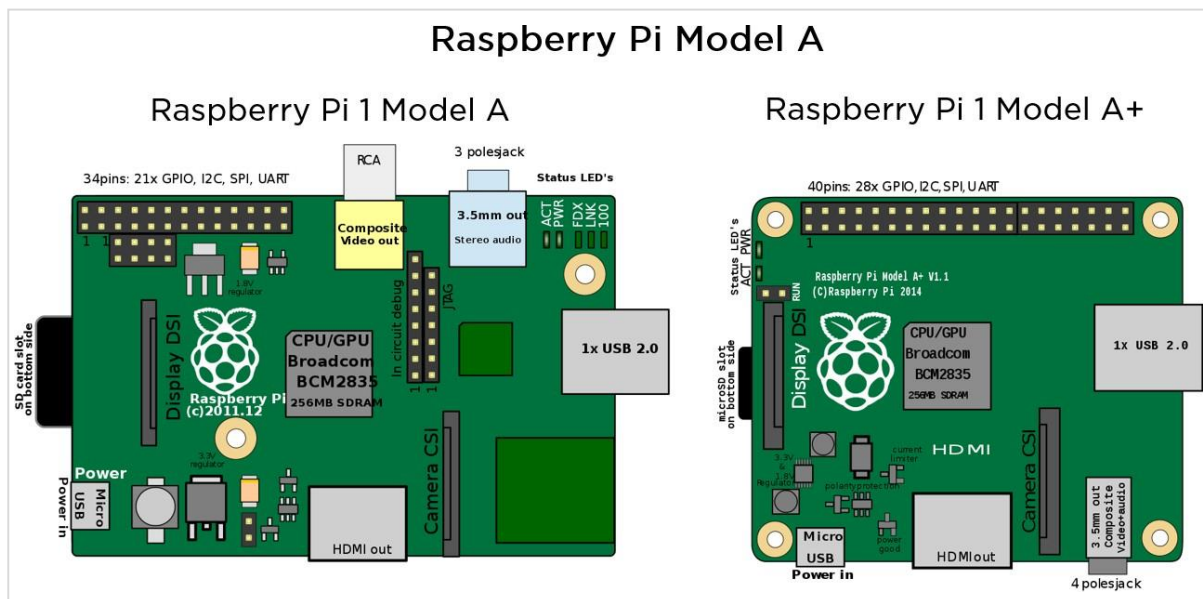
Raspberry Pi Model B



Slika 4: RPi model B

Diagram 2

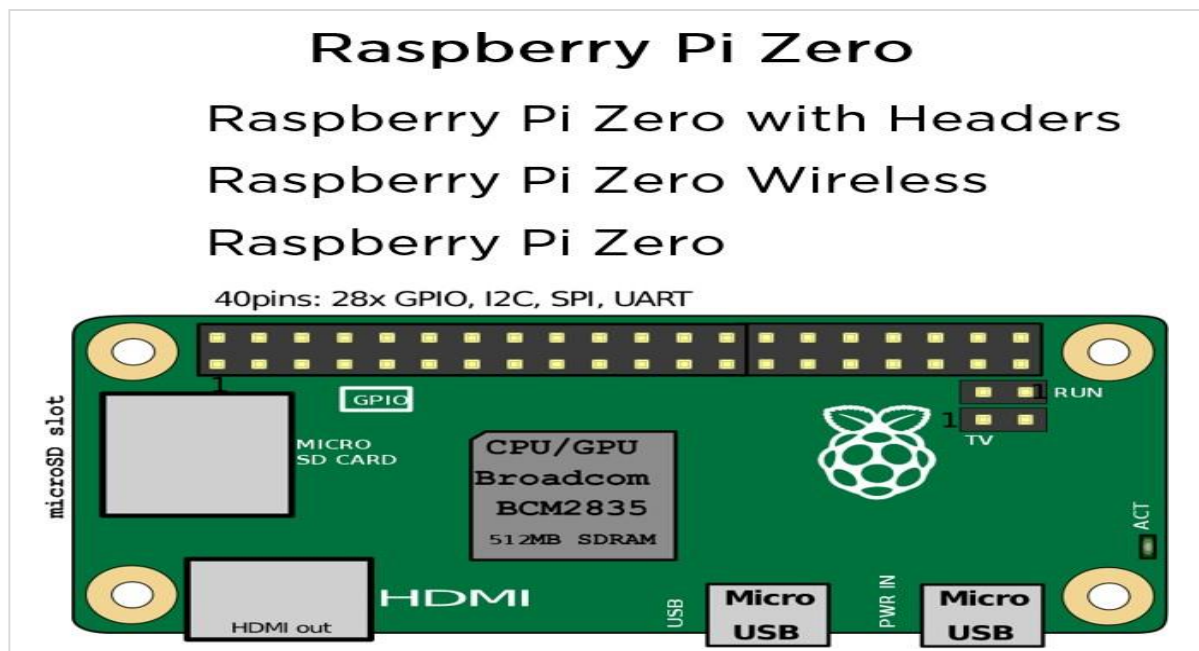
Poglejmo si diagram za model Raspberry Pi Model A:



Slika 5: RPi model A

Diagram 3

Prikaz priključkov za model **Raspberry Pi Zero**:



Modul kamere

Modul kamere je uradna naprava, ki jo lahko kupimo za našo napravo RPi. Priključimo jo lahko neposredno na vmesnik, ki se nahaja na RPi napravi s posebnim kablom, ki je priložen kameri.



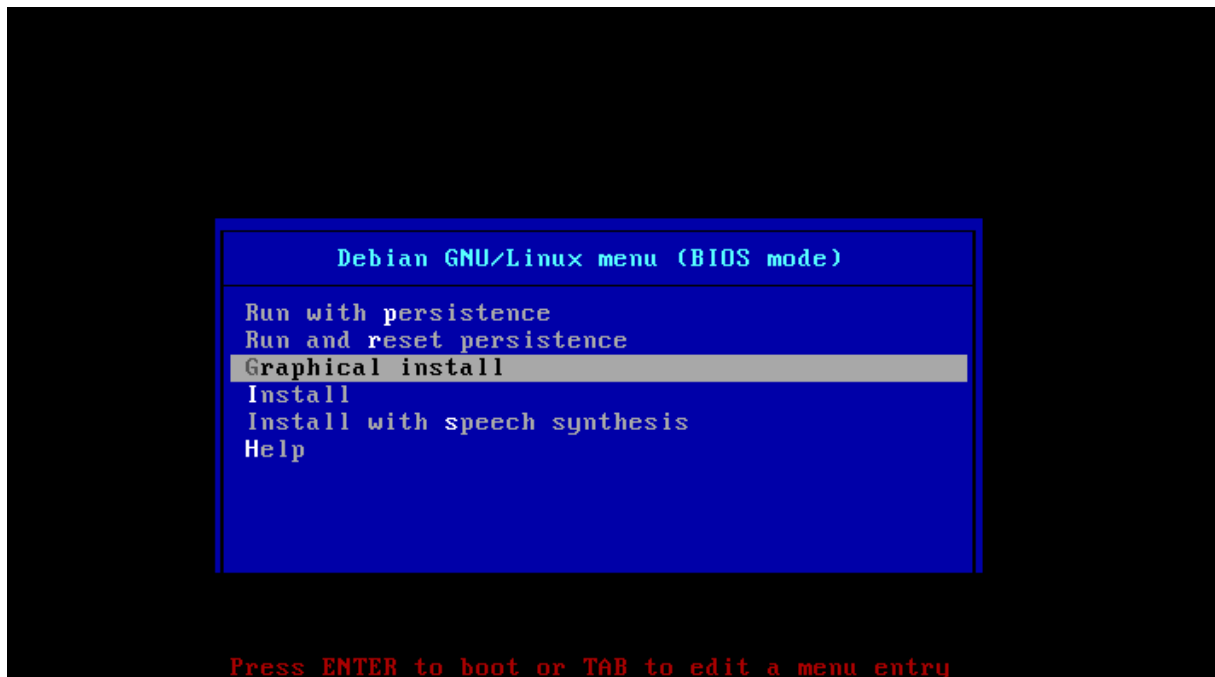
Slika 6: Kamera za napravo RPi

Pri priklopu pazimo, da bomo kabel obrnili, kot je označeno. Previdni moramo biti tudi pri rokovanju s priklopom, saj je plastika krhka in lahko hitro zatič tudi odlomimo.

Zagon naprave RPi

Priključimo vhodne naprave (miška, tipkovnica) in izhodne naprave (zaslon/ekran). Po potrebi poskrbimo, da napravo povežemo v svetovni splet preko priključka LAN. Potrebujemo ustrezen kabel s priključkom RJ-45 in dostop do interneta. Gumba za vklop nimamo, saj se naprava zažene sama ob vklopu napajanja.

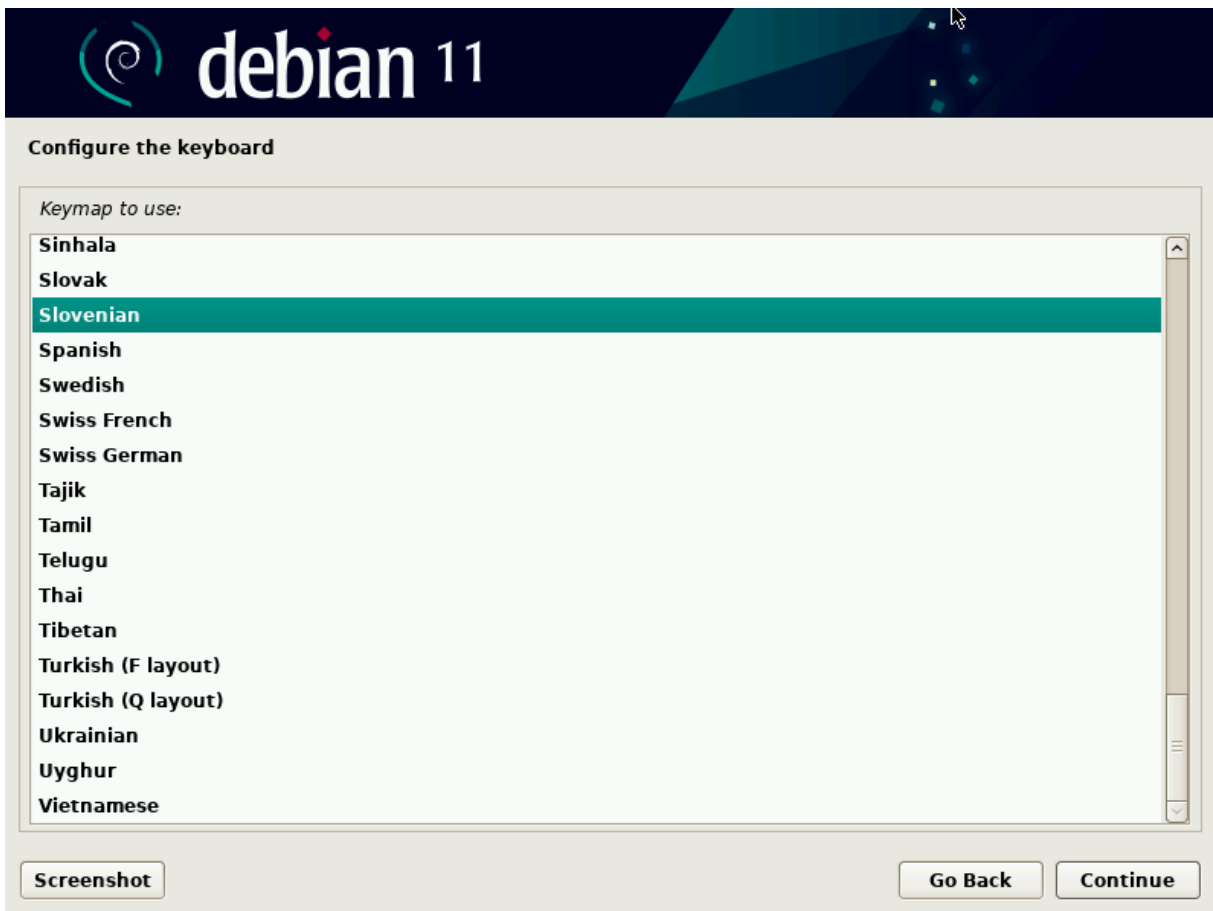
Sprva sledi vmesnik za namestitev, kjer seveda prikaz zavisi od izbrane različice.



Slika 7: Začetna konfiguracija RPi

Vkolikor želimo namestitev z grafičnim vmesnikom, izberemo opcijo Graphical install. Sicer lahko kasneje tudi v primeru, da grafike nismo namestili, poskrbimo, da jo bomo prikazali.

Tekom namestitve bo sledilo okno, kjer bomo izbrali ustrezen znakovni nabor in postavitev tipk na tipkovnici.



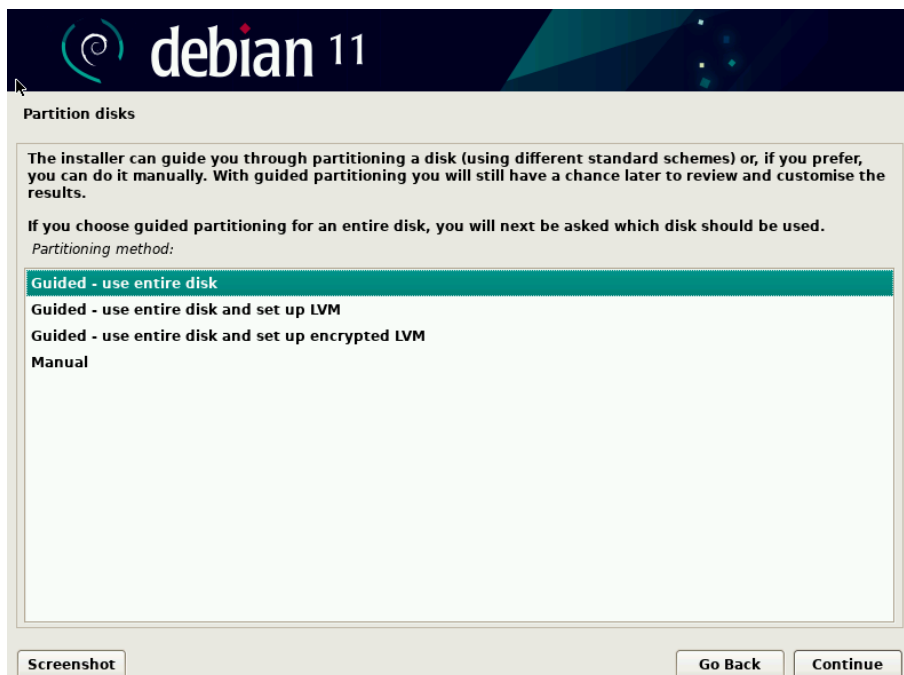
Slika 8: Izbira lokalizacije

Večje zadrege v tem delu ni, saj je jasno prikazano in voden, kaj bomo nastavili. Počakamo le, da se ustrezni moduli namestijo in to bo vezlo nekaj trenutkov, kot je razvidno tudi na naslednji sliki.



Slika 9: Nameščanje komponent

Tekom namestitve sledi tudi upravljanje z diskom oz. particijami. Nekdo, ki ni večč upravljanju znotraj okolja Linux, naj ohrani privzeto vrednost. S tem se zasede celotna spominska kartica oz. trdi disk.



Slika 10: Upravljanje z diskom

V primeru virtualizacije to predstavlja rezervirani del diska za sam virtualni sistem.



Slika 11: Nameščanje OS

Potrdimo nekaj sledečih korakov, da se zaključi postopek nameščanja in počakamo, da se sistem uspešno namesti. Pustimo lahko privzete vrednosti, ki nam jih ponudi, vkolikor nismo

vešči ročnega upravljanja s particijami, GUID itn.



Slika 12: Namestitev nalagalnika GRUB

V tem delu moramo biti pazljivi, da izberemo drugo možnost, sicer bomo morali ročno vnesti pot, kjer se bo izvedel zagon OS.



Slika 13: Zaključevanje namestitve

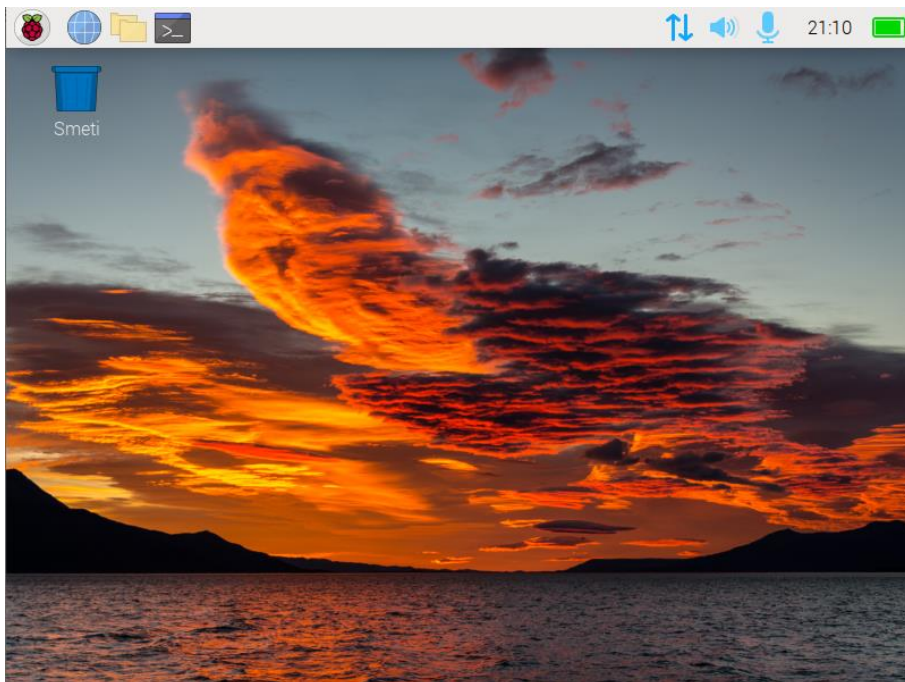
Počakamo še nekaj trenutkov, da se namestitev zaključi in se po ponovnem zagonu prikaže

vmesnik za konfiguracijo.



Slika 14: Prvi zagon naprave RPi

Prikaz prvega zagona se bo razlikoval med izbranimi distribucijami, ki so nam na voljo. Prikazan je predlagani vmesnik za začetnike, in sicer Raspberry PI OS, kjer je takoj na voljo tudi grafični vmesnik in osnovna orodja, ki so običajno potrebna za delo. Sledimo čarovniku, kjer nastavimo osnovne nastavitve.



Slika 15: Prvi zagon OS

Prikazana je vstopna stran OS, ki smo ga ravnokar uspešno namestili. Naj še enkrat opomnimo, da se vstopna stran razlikuje

5. Raspberry Pi — konfiguracija

V tem razdelku bomo spoznali konfiguracijo naprave RPi. Razumevanje in izvedba sta ključna, da bo delo z napravo potekalo nemoteno.

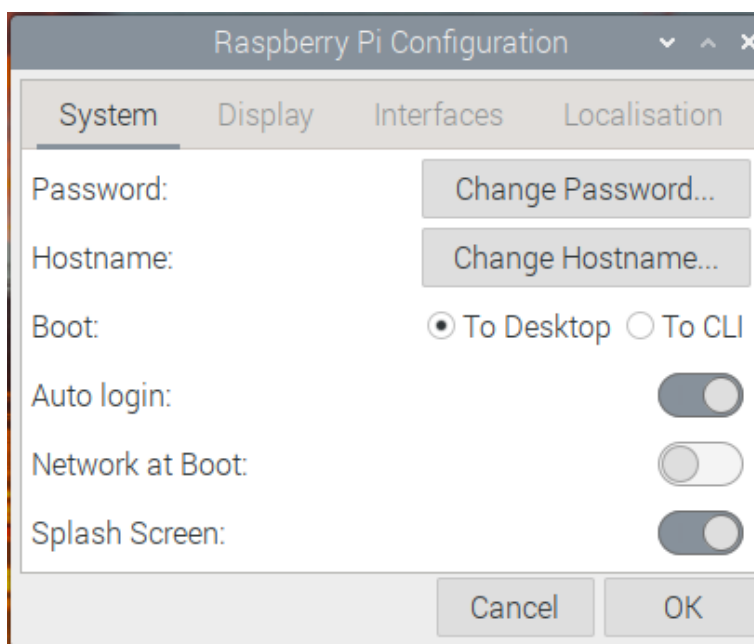
RPi konfiguracija

Raspberry Pi OS sloni na jedru Debian, kjer je uporabljeno namizje Pixel. Za začetnike je to najboljša rešitev, ki je hkrati tudi najbolj enostavna. Izvedli bomo nekaj osnovnih sprememb v konfiguraciji, kar si pogledamo sedaj. Vkolikor se želite pozabavati, lahko popolnoma enako okolje testirate tudi preko virtualiziranega sistema, ki si ga postavimo v okolju Virtual Box.

Dostop do nastavitve izvedemo preko klika na ikono maline in izberemo **Raspberry Pi configuration**.

Orodje za konfiguracijo

Sledimo nastavitvam, ki so razporejene preko štirih zavihkov.



Slika 16: Konfiguracija OS

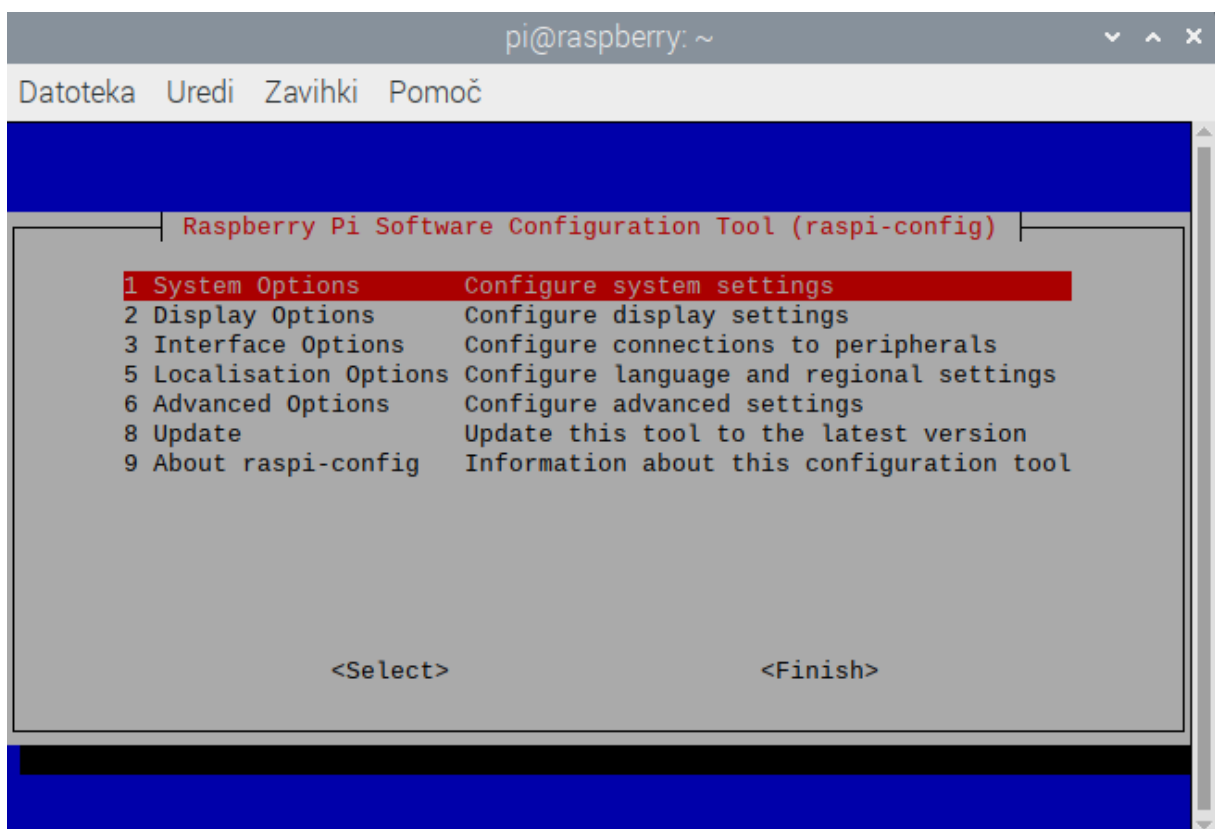
Odpre se nam polje z nastavitvami, kjer imamo običajno sledeče možnosti:

- **Change Password:** privzeto he geslo **raspberry**, katerega je seveda priporočljivo spremeniti.

- **Change the hostname:** Privzeto ime naprave je **raspberry**. Tudi tukaj se priporoča, da se ime naprave ustrezno spremeni.
- **Boot:** Izberemo lahko, kako naj se izvede zagon v grafični način ali konzolni način.
- **Auto Login:** Opcija je na voljo, če želimo samodejno izvesti prijavo v sistem.
- **Network at Boot:** Izbira ali je omrežje oz. zagon preko omrežja na voljo ali ne.
- **Splash screen:** Možnost, ki omogoča prikaz grafike RPi ob zagonu naprave.

Sledijo trije zavihki, kjer so na voljo Display, Interfaces in Localisation. S pogledom na njim bo hitro jasno, čemu služijo.

Do nastavitvev lahko pridemo tudi preko ukazne vrstice, kjer vnesemo ukaz `sudo raspi-config` in odpre se nam vmesnik, kjer je konfiguracija vsega zapisanega mogoča.



Slika 17: Konfiguracija preko terminala

Konfiguracija Wi-Fi

V desnem kotu zgoraj lahko vidimo dve puščici, ki kažeta status povezave brezžičnega omrežja. S klikom na ikono se nam prikažejo razpoložljiva omrežja, če seveda vmesnik WiFi imamo. Sicer nam izpiše, da vmesnik ni na voljo.

6. Raspberry Pi — Delo z Linux OS

V tem razdelku bomo spoznali okolje Linux, ki je nameščeno na našo napravo RPi.

PIXEL namizno okolje

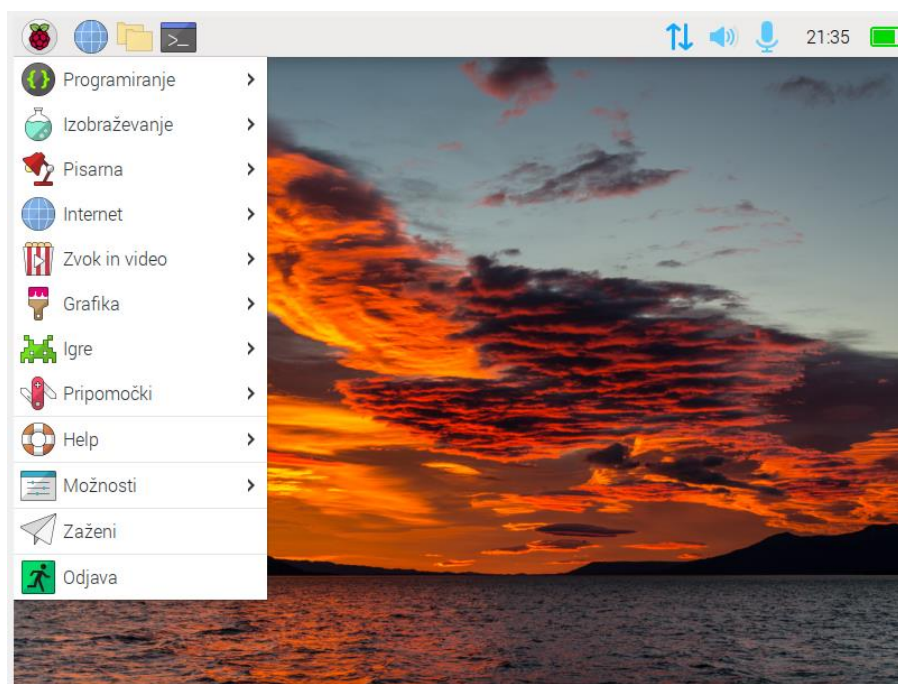
PIXEL (Pi Improved Xwindows Environment, Lightweight) predstavlja vizualno namizno okolje, ki je del okolja distribucije Linux-a. Predstavlja najhitrejši način, ki nam omogoča pričetek uporabe naprave RPi in vsebuje vse ključne elemente.

Nekaj značilnosti okolja PIXEL:

- Sloni na platformi LXDE (Lightweight X11 Desktop Environment, ki je odprtokodna.
- Okolje LXDE je bilo v sklopu projekta RPi preoblikovano in prilagojeno napravi RPi.
- Okolje PIXEL deluje in izgleda podobno, kot okolji Windows ali OS X.
- S podatki lahko upravljamo s pomočjo miške, tipkovnice in potegov.
- Uporaba okolja je enostavna in uporabniku prijazna.

Navigacija po namizju

Slika spodaj predstavlja okolje PIXEL. Vidimo lahko orodno vrstico, hitri dostop do menijev in pregled nad odprtimi aplikacijami.

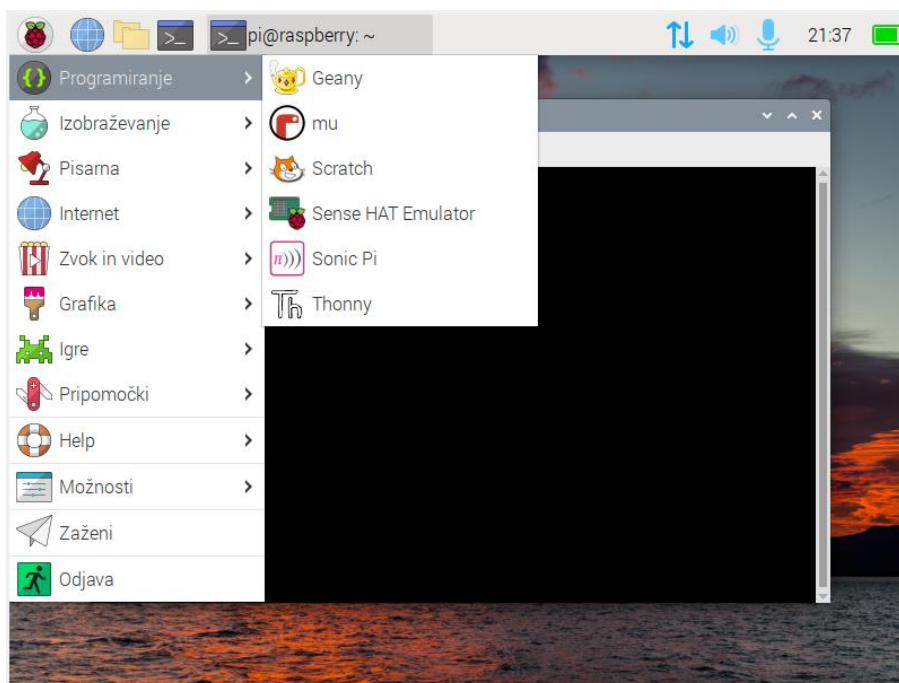


Slika 18: Okolje PIXEL.

Meni z aplikacijami

Za dostop do aplikacij bo služil meni, ki nam omogoča pregled nad nameščenimi aplikacijami in njihov zagon. Seveda ni nujno, da to storimo preko menija, saj imamo vse opcije vedno odprte preko ukazne vrstice. Je pa najhitreje in najbolj enostavno za začetnike.

Dostop izvedemo preko klika na malino, ki se nahaja v levem kotu zgoraj.



Slika 19: Dostop do menija

Dostop do podmenijev

S pomikom miške nad ustrezno glavno kategorijo, se nam odprejo podmeniji. Običajno so urejeni glede na področje, ki ga predstavljajo npr. programiranje.

Kateri so vsi programi, ki so pred nameščeni, si bo ogledal vsak bralec sam, saj se lahko glede na distribucijo in različico tudi razlikujejo.

7. Raspberry Pi — PIXEL namizno okolje

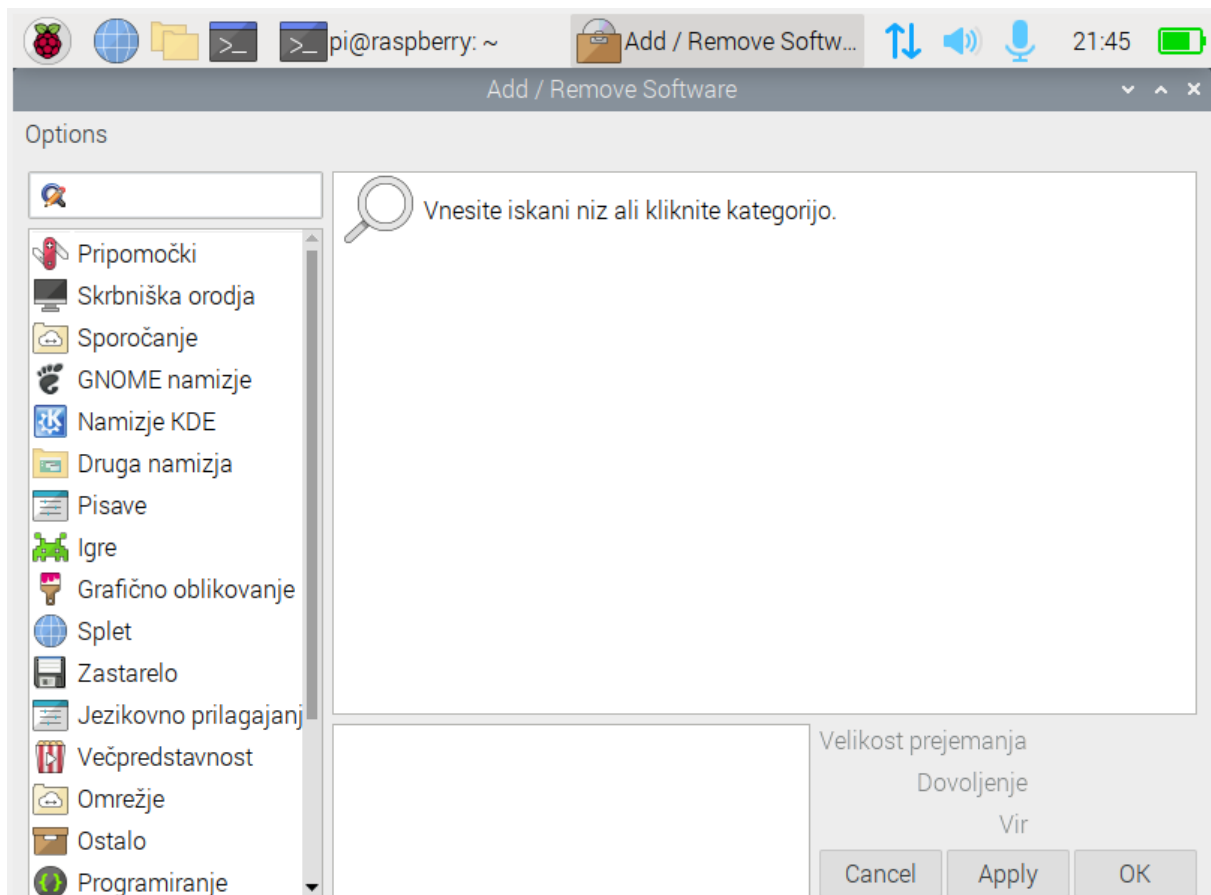
V tem razdelku bi lahko pričakovali, da si bomo ogledali delo z datotekami, kreiranjem, brisanjem itn. Lahko bi, vendar se smatra, da so osnovne operacije dela z operacijskim sistemom, ki je bolj kot ne podoben vsem dobro znanemu Windows, neko predznanje, ki ga ni potrebno obnavljati.

Zato bomo ta del skrčili in se posvetili temu, kjer bi se lahko pri delu zataknilo.

Nameščanje novih aplikacij

V okolju Linux je sicer bolj pogost način nameščanja aplikacij preko ukazne vrstice, vendar imamo tukaj, podobno kot v okolju Ubuntu vmesnik, ki nam bo to delo olajšal.

Z dostopom preko ikone maline → Možnosti → izberemo Add/remove software. Odpre se nam vmesnik, kjer lahko enostavno upravljamo z nameščenimi paketi ali izberemo nove.



Slika 20: Upravljanje s programi

8. Raspberry Pi — Linux Shell

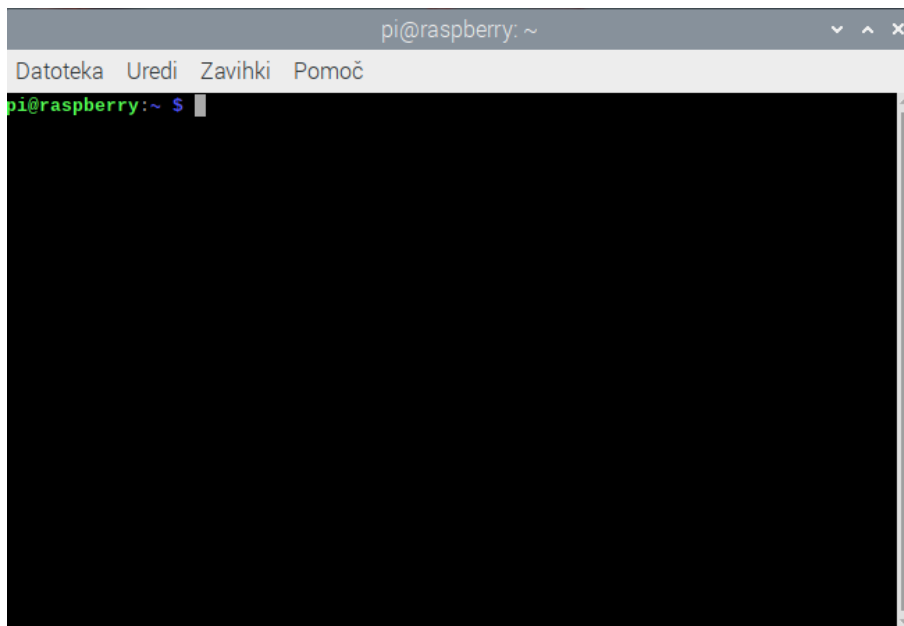
Shell oz. lupina v okolju RPi je osnova preko katere z napravo upravljamo, nameščamo pakete ali si pomagamo pri testiranju in komunikaciji z zunanjimi napravami.

Odpiranje okna Shell Window

Okno terminala lahko odpremo na enega izmed dveh načinov:

- Kliknemo na ikono Terminal, kjer imamo simbol >_ .
- Terminal lahko odpremo preko ustreznega menija.

V obeh primerih se nam bo odprla konzolna vrstica.



Slika 21: Konzolna vrstica

Razumevanje ukazne vrstice

Utripalka v ukazni vrstice je videti takole:

```
pi@raspberrypi ~ $
```

Vsebuje niz znakov, ki pomenijo sledeče.

pi

Predstavlja ime uporabnika, ki je trenutno prijavljen oz. aktiven.

raspberrypi

Predstavlja ime sistema, ki ga trenutno uporabljamo.

Simbol (~)

Simbol nam povem, kateri direktorij uporabnik gleda. Takšen znak predstavlja domači imenik.

Simbol dolarja (\$)

Predstavlja navadnega uporabnika brez posebnih pravic.

Pregled podatkov in datotek

Torej ob zagonu terminala se nahajamo v domačem imeniku.

Za prikaz vseh podatkov znotraj trenutnega imenika uporabimo ukaz ls, kot je prikazano.

```
pi@raspberrypi ~ $ ls
```

Izhod

Sledil bo prikaz podatkov, kot je prikazano na primeru.

```
Desktop Downloads Pictures python_games  
Videos Documents Music Public Templates
```

Pomembno je, da pazimo na velike in male črke, saj je okolje Linux na njih občutljivo.

Sprememba imenika

S prikazom datotek in imenikov se pojavi tudi potreba po spremembah. Pomikanje izvedemo z ukazom **cd**.

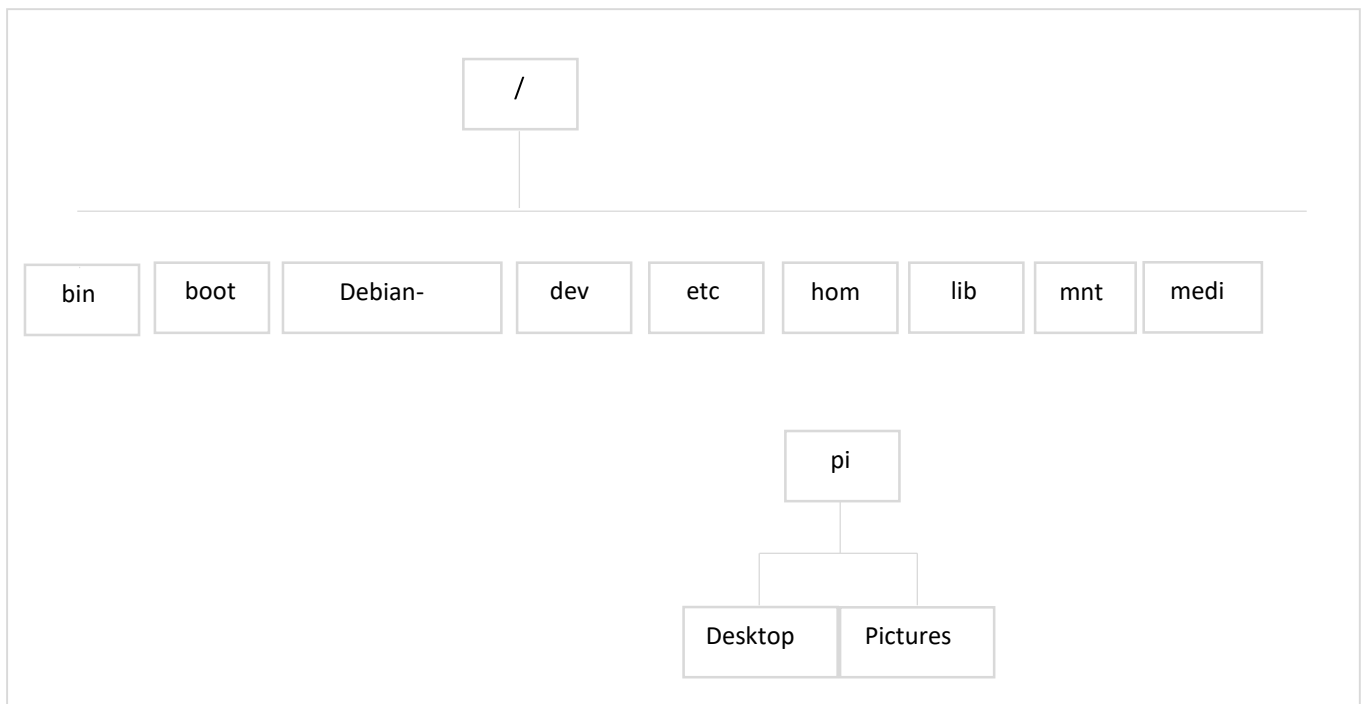
Želimo se pomakniti v imenik slik, torej vpišemo **cd Slike**.

```
pi@raspberrypi ~ $ cd Slike
```

Za pomik imenik nižje, bi vpisali **cd..** .

Drevesni prikaz

Spodnji diagram prikazuje, kako so organizirani podatki po imenikih znotraj sistema Linux.



Oznake pomenijo sledeče:

bin

Bin ali okrajšava za binaries vsebuje manjše programe, ki se vedejo kot ukazi v ukazni vrstici. Primer ukaz ls, ki smo ga že spoznali.

boot

Ta imenik je srce okolja Linux, kjer se nahaja t. i. Kernel.

dev

V tem imeniku so vse naprave, ki so priključene na sistem. Primer je omrežni vmesnik ali ključek USB.

etc

Imenik vsebuje različne konfiguracijske dokumente, ki so vezani na delovanje sistema.

home

Predstavlja domači imenik uporabnika.

lib

Vsebuje različne knjižnice, ki jih potrebuje OS.

lost+found

Ta imenik je uporabljen, če pride do napake v delovanju sistema.

media

Omogoča prepoznavo priključenih naprav, kot je npr. ključek USB.

mnt

Predstavlja izmenljive naprave, ki so priključene v sistem.

root

Rezerviran za root uporabnike, kjer pravic nimamo in ga ne moremo spreminjati.

Pravice

Kako so urejene pravice v okolju Linux? Delimo jih v tri skupine, in sicer Owner, Group in World.

Owner

Oseba, ki je kreirala dokument. Vsebuje navodila, kaj uporabnik s tem dokumentom lahko počne.

Group

Skupina ljudi, katerim se pravica dodeljuje.

World

Vsi ostali, ki dostopajo do dokumenta in pravice, ki so nanj vezane.

Vrste pravic

Vsaka izmed treh skupin ima lahko tri različne pravice:

- **Read permission:** Uporabnik ima možnost branja podatkov.
- **Write permission:** Uporabnik ima možnost spreminjanja/pisanja podatkov.
- **Execute permission:** Uporabnik ima pravico izvajanja podatka.

Ponovni zagon naprave RPi

Če želimo izvesti ponovni zagon OS na RPi, vnesemo v ukazno vrstico sledeči ukaz.

```
sudo reboot
```

Zaustavitev naprave RPi

Za zaustavitev, ki je izvedena pravilno, vnesemo v terminal sledeči ukaz.

```
sudo halt
```

Uporabni ukazi terminala

sudo raspi-config : konfiguracija naprave

startx: zagon grafičnega vmesnika

sudo halt : nastavitev zaustavitve naprave in priprava na izklop

ifconfig : prikaže nam aktivne mrežne povezave in njihove podrobnosti

ifconfig -a : prikaz naslova IP, ki ga lahko uporabimo za oddaljeno povezavo

sudo reboot : ponovni zagon naprave po konfiguraciji ali zaradi drugih razlogov

cat /etc/network/interfaces : prikaz vseh omrežnih vmesnikov in prikaz uporabe dinamičnega DHCP naslavljanja

iface eth0 inet dhcp/static : sprememba načina delovanja mrežne kartice

ping naslov_IP : preverjanje odzivnosti naslova IP ali domenskega imena

sudo apt-get update : izvedba posodobitve sistema

sudo apt-get remove <ime_paketa> : odstrani izbrani paket

sudo apt-get install <ime_paketa> : prenos paketa in njegova namestitve

ls -la : pregled strukture in podatkov o dokumentih

cp <ime_datoteke> <nova_lokacija> : kopiramo izbrano datoteko na novo lokacijo

mv <ime_datoteke> <nova_lokacija> : premik izbrane datoteke na novo lokacijo

rm <ime_datoteke> : izbrišemo izbrano datoteko

mkdir <ime_imenika> : ustvarimo novi imenik z izbranim imenom

cat <ime_datoteke> : izpis vsebine datoteke

raspi-config: dostop do konfiguracijskega vmesnika

clear: počisti trenutni vmesnik Terminala

date: izpiše trenutni datum

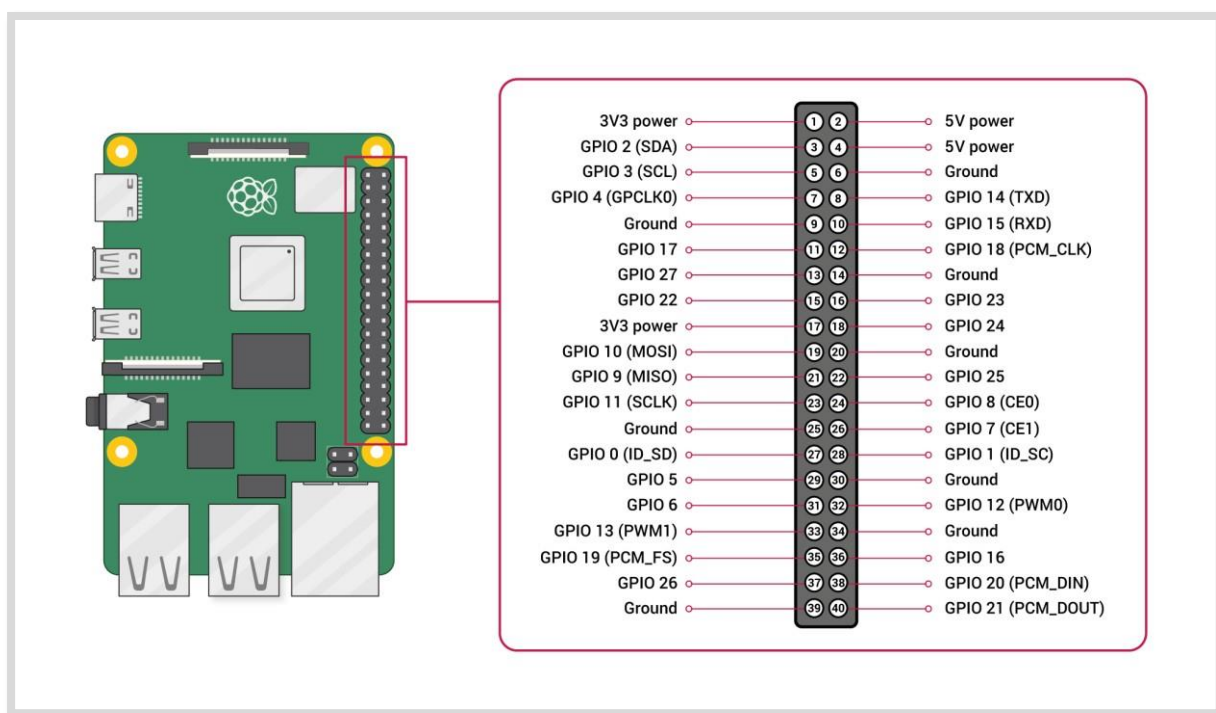
9. Raspberry Pi — GPIO

V tem razdelku bomo spoznali priključek GPIO in njegovo uporabo.

Pini GPIO

Ena izmed ključnih prednosti naprave RPi je uporaba priključkov GPIO (angl. General purpose input output). Njihov namen je priklop zunanjih naprav in branje/pošiljanje podatkov s priključenimi napravami.

Spodaj na sliki je podan diagram povezav prikljopa GPIO, kar je ključno, ko sledimo navodilom za priklop posamezne naprave. Prikazana shema 40-pinov je aktualna v vseh novejših generacijah naprav RPi.



Izbirni diagram lahko najdemo na uradni spletni strani www.raspberrypi.org

Napetosti

V zgornjem diagramu lahko vidimo, da imamo različni napetosti 5V in 3,3V. Na voljo je tudi nekaj pinov za ozemljitev (GND). Našteti priključki niso programirljivi, torej jim ne moremo spreminjati vloge.

Izhodi

Priključek GPIO lahko deluje kot izhodni priključek. Napetost v takšnem primeru je lahko visoka, torej (angl. High) 3,3V, ali nizka (angl. Low) 0V.

Vhodi

Priključek GPIO lahko nastavimo tudi kot vhod. Beremo lahko vrednosti 3,3V (angl. High) ali 0V (angl. Low). Uporabimo lahko t. i. vlečni upor (angl. Pull-up) ali spustni (angl. Pull-down) upor. Priključki oz. pini so v nekaterih primerih na GPIO2 in GPIO3 fiksni z nastavitvijo vlečnega oz. spustnega upora. Ostalim priključkom lahko to programsko določamo.

Alternativne funkcije

Priključke GPIO lahko uporabimo z različnimi alternativni funkcijami. Nekatere so dostopne na vseh priključkih, druge so omejene na določene priključke.

PWM: Pulse-width modulation

PWR funkcija je na voljo na vseh priključkih, kjer je to glede na shemo na voljo npr. GPIO12, GPIO13, GPIO18 in GPIO19.

SPI: Serial Peripheral Interface

SPI je na voljo na naslednjih priključkih:

SPIO: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)

SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

I2C: Inter-integrated Circuit

I2C priklop je na voljo na sledečih priključkih:

Data: (GPIO2); Clock (GPIO3)

EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)

Serial

Serijska komunikacija je na voljo preko spodnjih dveh priključkov:

TX(GPIO14)

RX(GPIO15)

Povezava GPIO na Raspberry Pi

Podanih je nekaj pravil, ki jih je potrebno upoštevati, ko izvajamo povezovanje GPIO na

napravi RPi. V nasprotnem primeru lahko pride do poškodbe naprave RPi.

- Na priključek GPIO naj ne bo večje napetosti, kot je 3.3.
- Tokovna obremenitev na priključek, naj ne bo višja od 3mA. Obremenitev je sicer lahko višja, vendar to skrajša življenjsko dobo vezja RPi. Zato se držimo načela, da je to zgornja meja.
- Povezovanje naprave izvajamo takrat, ko naprava ni zagnana. Tudi stikov ne povzročajmo med tem, ko naprava deluje.
- Napajanje 5V je več kot dovolj za delovanje naprave RPi. Ne poskušajte dovesti večje napetosti, sicer lahko trajno poškodujete vezje.
- Skupna tokovna obremenitev na izhodnih priključkih GPIO naj ne bo višja od 50mA.

Izhod priključkov GPIO

Za upravljanje z vhodi in izhodi GPIO preko okolja python, potrebujemo knjižnico RPi.GPIO, ki nam bo omogočila ustrezno komunikacijo.

Namestitev python knjižnice RPi.GPIO

Za namestitev/uporabo knjižnice python za komunikacijo z GPIO vnesemo sledeči ukaz v terminal na napravi RPi.

```
sudo apt-get install python-  
dev sudo apt-get install
```

Večina zadnjih distribucij OS ima knjižnico že nameščeno. V primeru, da vam python javi, da le-ta ni na voljo, uporabite zgornja ukaza.

Naprave I2C

Poglejmo, kako lahko uporabljamo naprave s priklopom I2C.

Primer 1: Uporabimo Adafruit Occidentalis 0.2 ali novejši

Vkolikor uporabljamo Adafruit Occidentalis, nam ni potrebno storiti ničesar. Distribucija je že pripravljena in ima vse potrebno, da komunikacijo lahko izvedemo.

Primer 2: Uporaba Raspbian različice

Vkolikor uporabljamo različico Raspbian moramo sprva izvesti sledečo spremembo.

Uredimo dokument `/etc/modules` s pomočjo sledečega ukaza.

```
sudo nano /etc/modules
```

Sedaj dodamo spodaj podani vrsti v dokument.


```
i2c-  
bcm2708
```

V naslednjem koraku moramo urediti dokument `/etc/modprobe.d/raspi-blacklist.conf` in odstraniti komentar pri spodnjih dveh vrsticah. Komentar je podan z oznako lojtre #.

```
blacklist      i2c-  
bcm2708 #blacklist
```

Ko zapisano izvedemo, namestimo knjižnico I2C s pomočjo spodnjega ukaza.

```
sudo apt-get install python-smbus
```

Sledi le še ponovni zagon naprave RPi in I2C je pripravljen za uporabo.

Iskanje naslovov I2C

Vkolikor imamo na priključek I2C priključeno zunanjo napravo in želimo ugotoviti njen naslov za dostopanje, potem si pomagamo z orodjem, ki ga je potrebno namestiti.

Orodje I2C namestimo s sledečim ukazom v ukazni vrstici:

```
sudo apt-get install i2c-tools
```

Ko je postopek zaključen, sledi priklop naprave na I2C na napravi RPi in izvedba spodnjega ukaza:

```
sudo i2cdetect -y 1
```

Paziti moramo na dve stvari, in sicer:

- Sprva moramo paziti, če imamo novejšo različico operacijskega sistema oz. zavisi od različice izbranega OS, je orodje za I2C lahko že nameščeno.
- Vkolikor imamo starejši tip naprave RPi, spremenimo vrednost drugega ukaza za prikaz priključenih naprav iz vrednosti 1 v vrednost 0.

10. Raspberry Pi — Upori in upornost

Izračun upornosti

V klasičnem elektronskem vezju s številnimi elektronskimi elementi so najštevilčnejši valjasti elementi, ki jim pravimo upori. Njihova naloga je, da povezujejo aktivne elektronske komponente v neko celoto, ki ji pravimo elektronska naprava. Upori predstavljajo pasivne elektronske elemente, ki se upirajo pretoku električnega toka. Pri tem velja pravilo, da če je upornost večja, je električni tok manjši in obratno.

Povezavo med napetostjo (U), tokom (I) in upornostjo (R) je prvi odkril nemški fizik Ohm. Po njem se tudi imenuje Ohmov zakon. Izračun povezave izračunamo z matematično formulo:

$$R = \frac{U}{I}$$

Z imenom Ohm je poimenovana tudi osnovna merska enota za praktično upornost, in sicer je to 1 W (ohm).

Vrste uporov

Standardne upore lahko razdelimo v dve osnovni skupini:

- ☐ stalni upori: nazivna vrednost upornosti se ne spreminja, glede na tehnologijo izdelave pa jih lahko razdelimo na ogljene, žične in metal-film upore.
- ☐ spremenljivi upori: predstavljajo upore, katerim lahko spreminjamo vrednost upornosti, sestavljeni pa so iz štirih osnovnih delov: uporovne plasti, drsnika, osi in priključnih kontaktov. Spremenljivi upori se delijo v osnovni na dve osnovni izvedbi, in sicer:
 - potenciometri
 - trimer potenciometri

Poznamo tudi posebne vrste uporov:

- termistorji: uporna vrednost se spreminja v povezavi s spremembo temperature. Poznamo dva tipa:
 - NTC (negativni temperaturni koeficient) – s porastom temperature se upornost manjša.
 - PTC (pozitivni temperaturni koeficient) – s porastom temperature se upornost večja.
- VDR (varistor): upornost se spreminja glede na priključeno napetost

- LDR (foto upor): upornost se spreminja glede na osvetlitev

Karakteristične vrednosti upornosti

Nazivna vrednost upornosti se izraža v osnovni merski enoti 1 W ali v večjih merskih enotah. Primer večjih enot:

1 k Ω = 1000 Ω (KILO OHM)
1 M Ω = 1000 k Ω = 1000000 Ω (MEGA OHM)

Vsi upori se izdelujejo po v naprej določenih standardih. V trgovini ne bomo dobili ravno vsake vrednosti, ki si jo zaželimo, vendar se bomo morali zadovoljiti z najbližjo standardno vrednostjo. V kolikor bi nam izračun pokazal, da bi nam najbolj ustrezal upor z upornostjo 14,2 kW, bomo morali uporabiti z najbližjo standardno vrednostjo, ki je v tem primeru 15 kW.

Označevanje uporov

Na vsakem uporu najdemo podatke, kot so: vrednost upornosti, toleranca, dovoljena disipacija moči, tip upora, proizvajalec, tovarniška oznaka itn. Običajno najdemo same prve tri podatke, ki so za nas tudi najpomembnejši, torej vrednost upora, toleranca in dovoljena disipacija moči (nazivna moč upora).

Vrednost upora se izraža v osnovnih enotah (W, kW ali MW). Kažejo nam, v kolikšni meri se upor upira pretoku električnega toka. Dovoljena disipacija moči označuje maksimalno moč, ki jo lahko razvije upor brez poškodb. Moč upora je določena z dimenzijami upora (večji upori zdržijo večjo disipacijo moči) ter z materialom iz katerega je upor narejen (večja moč uporov je običajno na keramičnem nosilcu). Dovoljena disipacija se izraža v Watih (W). Toleranca predstavlja dovoljeno odstopanje dejanske vrednosti od tovarniško označene vrednosti upornosti.

Za označevanje upornosti uporabljamo uporovno lestvico, ki je lahko:

- alfanumerična (črke in številke)
- barvna lestvica

Alfanumerično označevanje najpogosteje uporabljamo za označevanje uporov večjih moči. Vsi potrebni podatki so zapisani oz. natisnjeni na sami površini upora. Namesto standardne oznake se za mersko enoto uporabljajo nadomestne črke:

Ω \rightarrow E (z ali brez črke E)
k Ω \rightarrow k
M Ω \rightarrow M

Nadomestne črke so lahko tudi na mestu decimalne vejice (npr. 8k2 Ω 8,2 kW). Toleranco lahko izrazimo tudi z nadomestnimi črkami:

F = 1 %
G = 2 %
J = 5 %
K = 10 %
M = 20 %

Označevanje s pomočjo barvnih prstanov bomo srečali pogosteje, kot alfanumerično označevanje. Pri tem načinu je na upor naneseo nekaj barvnih prstanov, tipično 3, 4, 5 ali celo 6 barvnih prstanov. Vsaka izmed barv predstavlja eno številko, vrednost upora določimo na podlagi tabele barvnih oznak, ki je podana spodaj na sliki.

Slika 22: Barvna shema in pripadajoče oznake

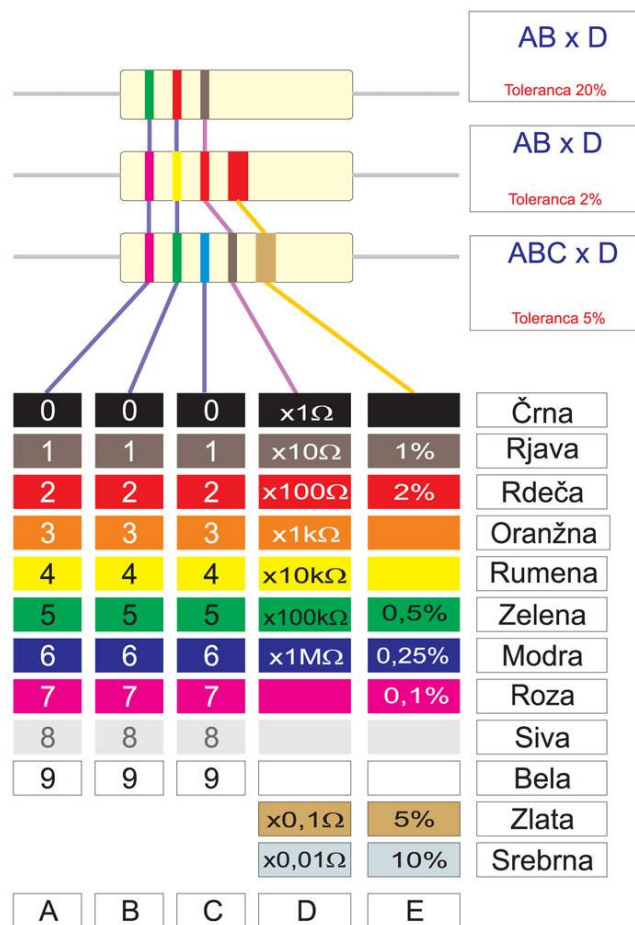


Tabela ima 5 stolpcev in 12 vrstic. Stolpci A, B in C določajo številke vrednosti upornosti. Stolpec D predstavlja multiplikator (torej množitelj vrednosti prikazane s prvimi tremi stolpci), stolpec E pa kaže toleranco (dovoljeno odstopanje).

Kako določimo prvi prstan na upor? Prvi prstan je vedno tisti, ki je najbližji robu upora, zadnji prstan pa je pogosto a ne vedno širši od ostalih. V veliko pomoč so nam najpogosteje uporabljene tolerance uporov vrednosti 5 % in 10 %, kateri sta predstavljeni z zlato ali srebrno barvo, ki se ne uporabljata za označevanje vrednosti v prvih treh prstanih.

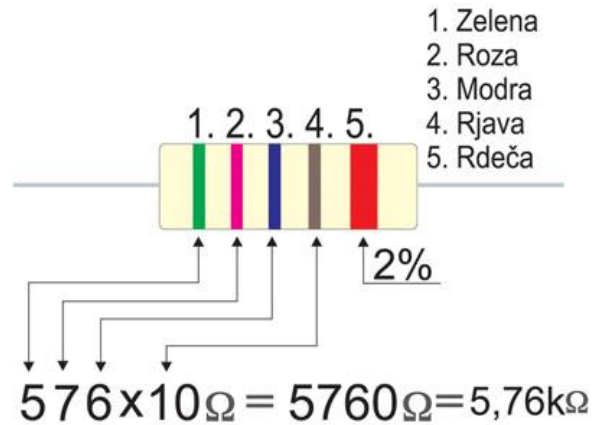
Označevanje s 4 barvami

Označevanje s tremi barvami je ekvivalentno označevanju s štirimi barvami, le da je izpuščena četrta barva. Toleranca v tem primeru znaša 20 %.

Primeri označevanja:

1. barva: prva številka,
2. barva: druga številka,
3. barva: multiplikator (množitelj),
4. barva: toleranca.

Slika 23: Prikaz izračuna na podlagi barvnih obročkov



Označevanje s 5. barvami

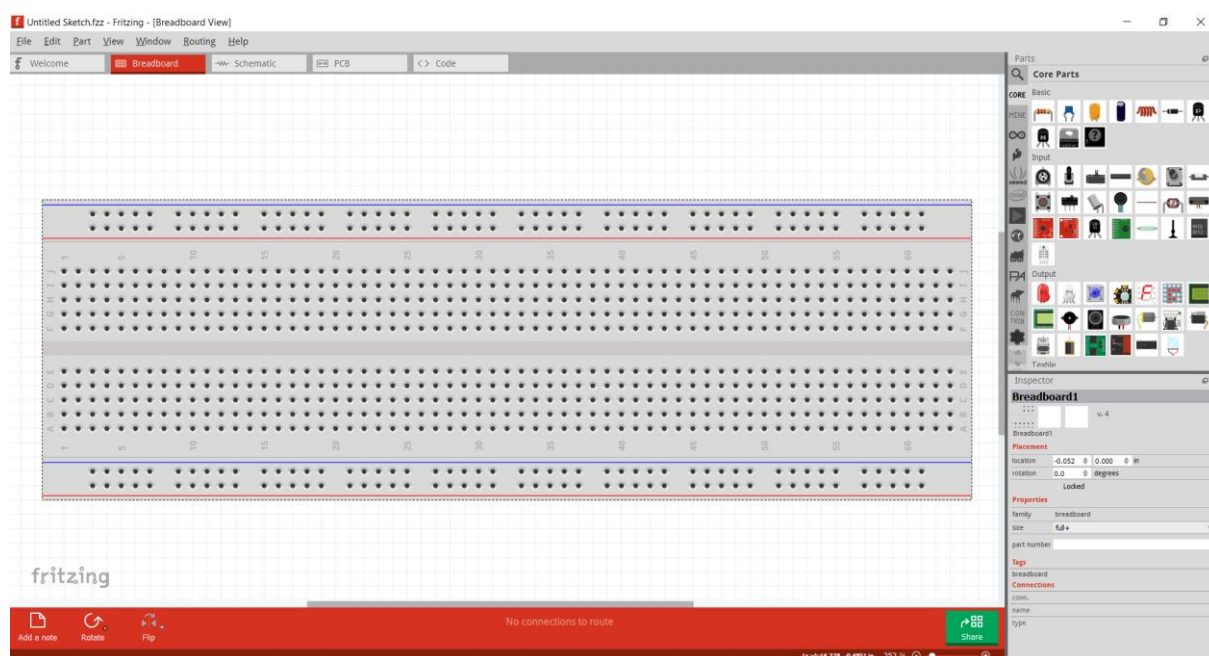
Označevanje uporov s pomočjo petih barv je najpogostejše pri preciznih uporih s toleranco 2 %, 1 % ali nižjo.

Za vse tiste, ki bi radi izračun upornosti le ustrezno preverili, lahko uporabite katero izmed spletnih strani, kjer vrednost upora le izračunate s pomočjo izbranih barv npr. <https://www.allaboutcircuits.com/tools/resistor-color-code-calculator/>.

11. Raspberry Pi — načrtovanje vezave

Preden se lotimo zagona naprave in priklopa prvih komponent si pogledjmo, kako lahko načrtujemo samo izvedbo priklopa v ustreznem programskem okolju.

Na voljo nam je trenutno še brezplačno orodje Fritzing, kjer lahko načrtujemo shemo vezave za različne naprave, med drugim tudi za naprave RPi. Prednost je vsekakor shema, ki nam bo pomagala pri priklopu in samem razumevanju načrtovanja.



Slika 24: Orodje Fritzing za načrtovanje sheme

Fritzing je odprtokodno orodje, ki nam pomaga pri načrtovanju izdelave sheme vezave za različne naprave in različne komponente. Namen je načrtovanje, risanje in deljenje projektov, ki bodo na voljo tudi drugim. Prenos lahko izvedemo preko Github ali preko spletne strani fritzing.org/download. V primeru prenosa preko spletne strani je potreben prispevek 8 € za prenos programske opreme.

12. Raspberry Pi — primeri projektov

Prikazali bomo nekaj primerov, kjer bomo priključili zunanje komponente in napisali ustrezno python kodo za komunikacijo. Izdelali bomo shemo in testirali delovanje projekta.

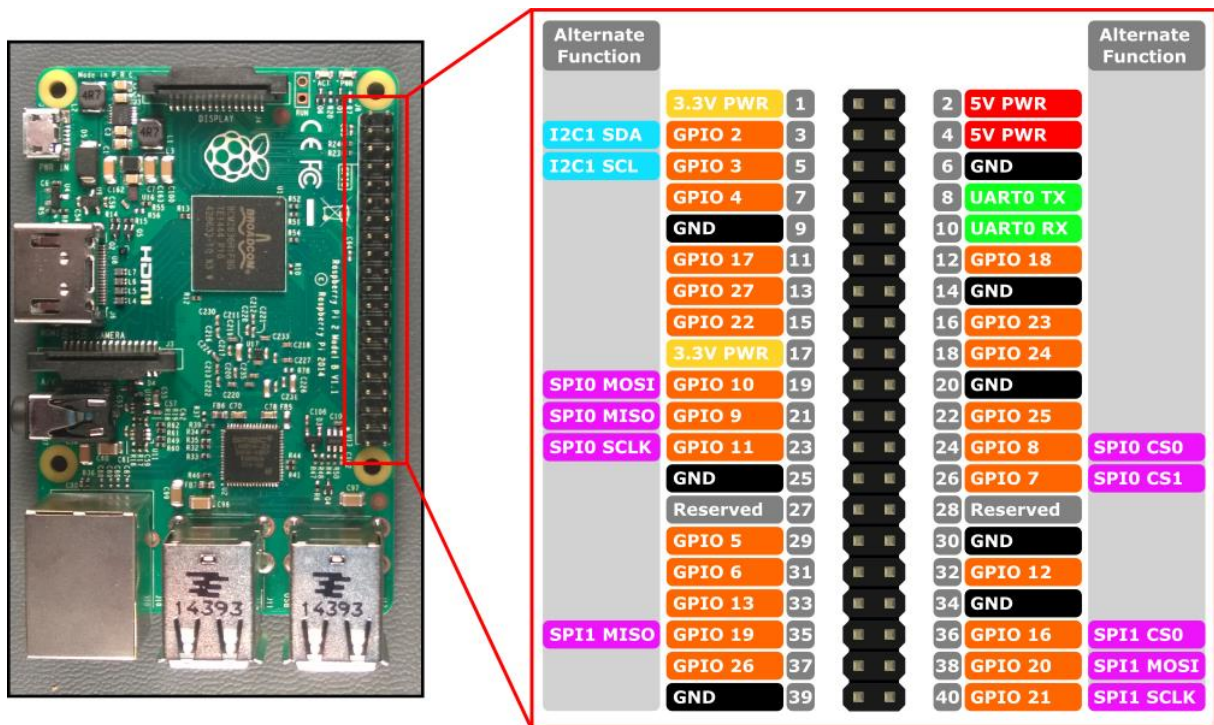
Zaznava pritiska tipke

Za začetek si pogledjmo preprost primer, kjer bomo priključili tipko in preko ustrezne python kode zaznali, kdaj je bila tipka pritisnjena.

Za izdelavo projekta potrebujemo:

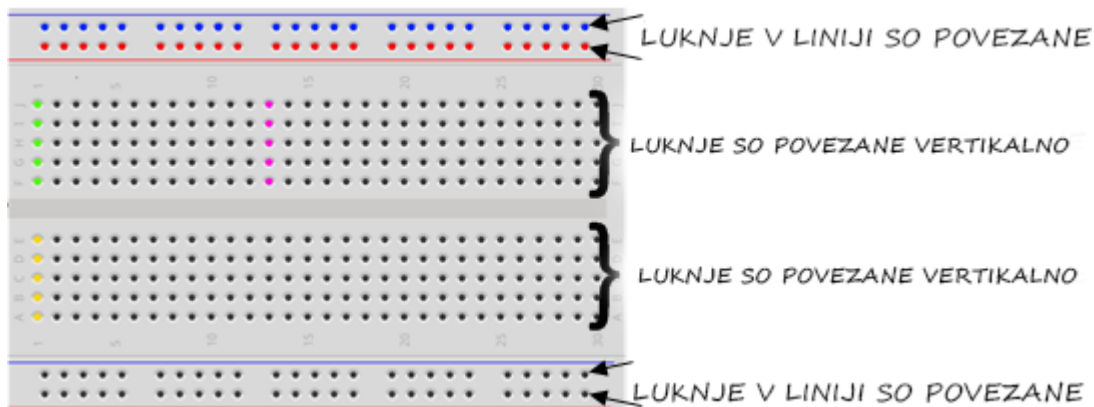
- RPi napravo
- tipko
- testna plošča (angl. Breadboard)
- ustrezne žičke
- ustrezen upor. V našem primeru bo uporabljen 230 Ω .

V okolju Fritzing izberemo ustrezno napravo, v našem primeru je to kar RPi 3 model B. Na priključku GPIO izberemo za komunikacijo enega izmed GPIO priklpov npr. 7, 8, 9, 10, 11, ...



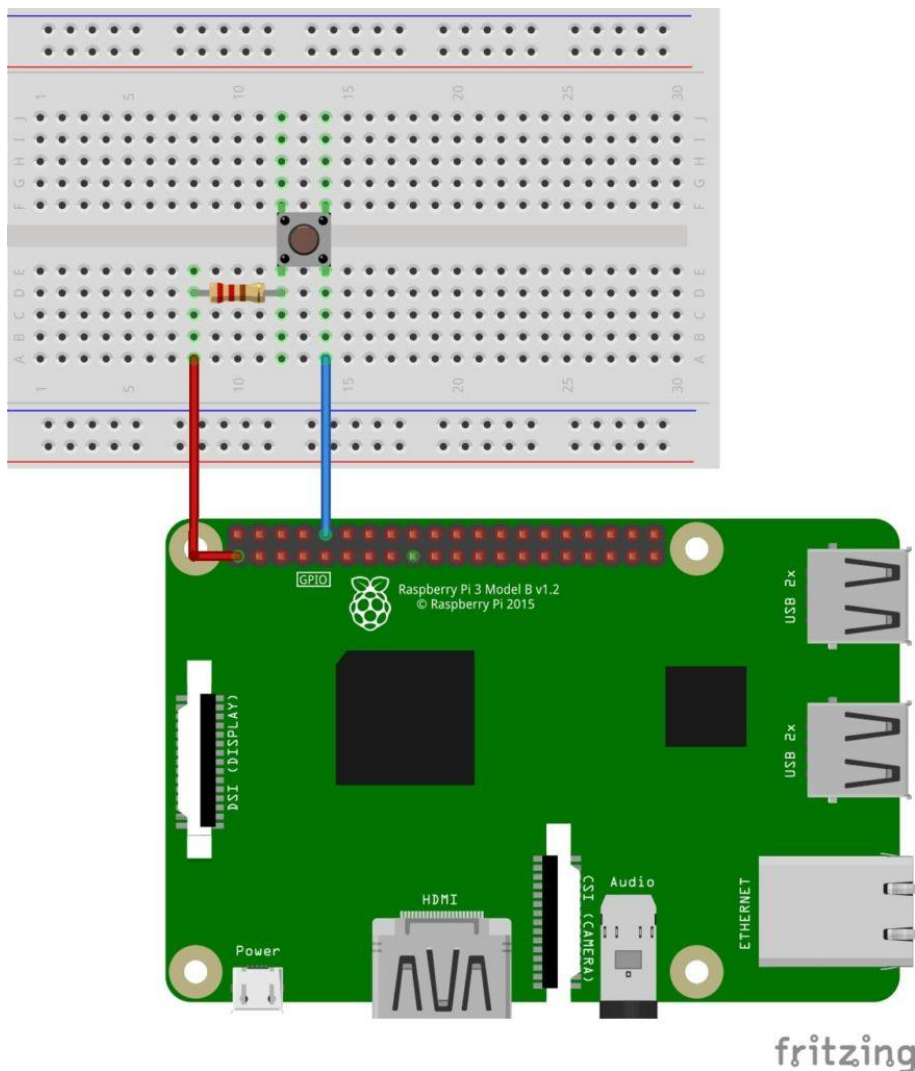
Slika 25: Shema vezav RPi 3

S pomočjo programa Fritzing izdelamo ustrezno shemo vezave. GPIO 1 predstavlja napetost 3,3 V, GPIO 8 pa nam bo služil kot GND ob zaznavi tokokroga, ki ga bomo ustvarili s pritiskom na tipko. Na sliki lahko vidimo testno ploščo (angl. Breadboard) v beli barvi, kjer povezujemo v linijah ustrezne povezave. Med napetost 3,3V in tipko vstavimo ustrezen upor. Pogledjmo še način vezave na testni plošči, da ne bo kakšne dileme.



Slika 26: povezovanje testne plošče

Glede na prikazano sliko izvedemo ustrezno vezavo.



Slika 27: Shema vezave tipke

Ko imamo vezavo izdelano, sledi le še vezava na pravi napravi RPi in ustrezen zapis programa python. Za samo testiranje kodo bomo potrebovali še paket RPi.GPIO, ki bo omogočal komunikacijo preko GPIO priklpov.

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

Po vnosu ukaza bo sledilo izpisovanje sekvenc nameščanja paketov. Poglejmo še kodo, ki bo izvajala komunikacijo in ob pritisku na tipko izpisala besedilo "Button press".

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(8, GPIO.IN,pull_up_down=GPIO.PUD_UP)
```

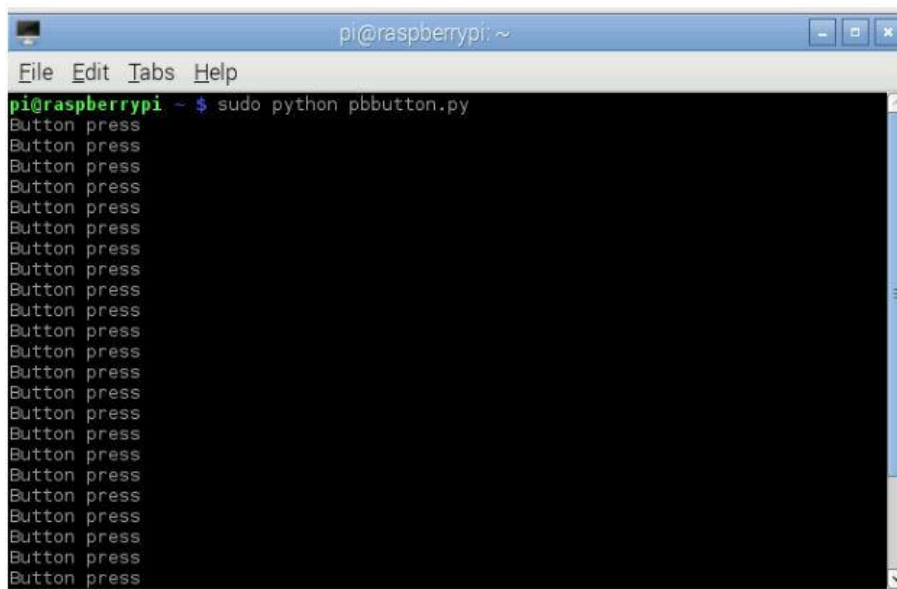
```
while True:
    inputValue = GPIO.input(8)
    if (inputValue == False):
        print("Button press ")
        time.sleep(0.5)
```

Ob zagonu programa na naši napravi RPi, se bo izvedla komunikacija in izpisala besedilo ob vsakem pritisku na tipko.

Še enkrat opozorimo na zagon programa.

```
pi@raspberrypi ~ $ sudo python tipka.py
```

In še pripadajoči izpis, ki ga dobimo ob pritisku na tipko znotraj ukazne vrstice.



```
pi@raspberrypi ~ $ sudo python pbutton.py
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
Button press
```

Slika 28: Pritisk na tipko in zaznava pritiska

Tako imamo preprost primer zaznave pritiska tipke zaključen.

Priklop lučke LED

Naša naloga bo sila preprosta, in sicer želimo priključiti lučko LED, ki bo utripala.

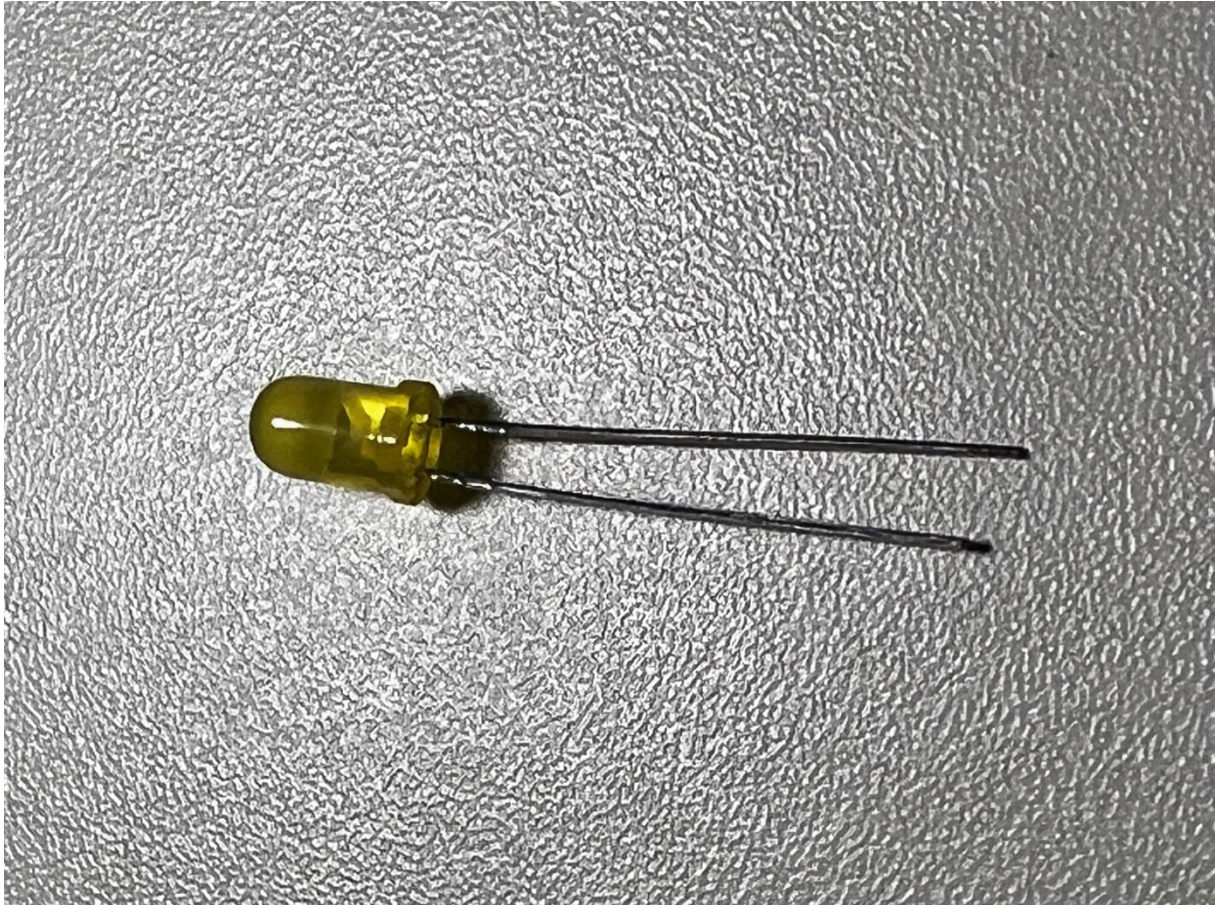
Za project potrebujemo:

- lučko LED poljubne barve
- napravo RPi

- ustrezen upor glede na lučko LED (običajno je v kompletu priložen 230 Ω ali 1 k Ω).

Svetilka LED

Na tem mestu se ustavimo pri svetilki LED, kjer bova ob prvem pogledu nanjo opazili, da ima eno nožico daljšo kot drugo.

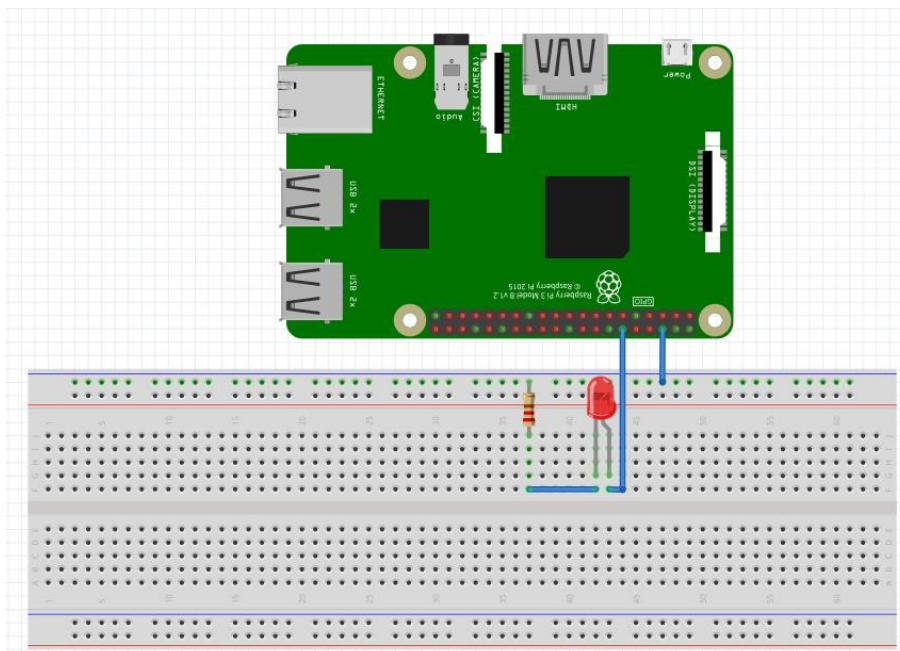


Slika 29: Svetilka LED

Daljša nožica, kateri pravimo tudi anoda, vedno povežemo na pozitivni del, torej +. Zavis od vrste svetilke LED je to lahko 3,3 V ali 5V. V kompletih so običajno priložene različice s 3,3 V. Krajša nožica predstavlja negativno vezavo, torej – oz. GND, kjer govorimo o katodi. Sijalka LED bo svetila le, če skozi njo pravilno spustimo ustrezen tok.

Shema vezave

V programu Fritzing izrišemo shemo vezave, kot bomo to storili kasneje na dejanski napravi.



Kot smo povedali, smo pozorni na ustrezno postavitev svetila LED (predvsem + in – glede na nožice). Poskrbimo tudi za vmesno vezavo ustreznega upora.

Program za utripanje svetila LED

Sledi zapis kode v okolju python, ki bo poskrbela za utripanje. Pri vezavi ni pomembno, kateri GPIO izberemo, pazimo le, da izberemo tistega, ki ga lahko programiramo npr. GPIO 18, kot je prikazano na shemi. Vkolikor boste izbrali drugo PIN, pazite na ustrezno spremembo kode python.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
print ("LED on")
GPIO.output(18,GPIO.HIGH)
time.sleep(1)
print ("LED off")
GPIO.output(18,GPIO.LOW)
```

Poglejmo si še kratko razlago zapisane kode.

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

Kot smo že omenili, potrebujemo paket RPi.GPIO za komunikacijo s priključki GPIO.

```
import time
```

Sledi uvoz časa, kjer bomo nastavili, koliko časa sijalka LED gori.

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

Z ukazom setmode bomo napravi RPi povedali, katero postavitev oz. način delovanja naj izbere pri izvajanju programa, da bo komunikacija tekla ustrezno.

```
GPIO.setwarnings(False)
```

Z nastavitvijo setwarning(False) izključimo morebitna opozorila, ki bi se pojavila ob zagonu našega programa.

```
print ("LED on")
```

Izvede izpis besedila, da se bo sijalka LED vključila oz. je vključena.

```
GPIO.output(18,GPIO.HIGH)
```

S tem ukazom izvedemo vklop priključka GPIO številka 18. Z vrednostjo HIGH skozenj spustimo tok 3,3V, kar omogoči, da sijalka zasveti.

```
time.sleep(1)
```

Časovnik uporabimo za zamik, ki se bo ustvaril na tem mestu. Konkretno bomo imeli eno sekundo zamika, preden se izvede naslednji korak v kodi.

```
print ("LED off")
```

Sledi izpis, da smo sijalko ugasnili in takoj prožimo naslednji ukaz, ki bo izključil tokorog.

```
GPIO.output(18,GPIO.LOW)
```

Poskrbimo, da se sijalka ugasne. Vrednost LOW izklopi napajanje 3,3 V.

Pri vsem skupaj vidimo, da smo sicer poskrbeli za vklop in izklop sijalke, vendar smo zapisali, da bomo imeli sijalko LED, ki bo utripala. Zato je potreben majhen popravek v našem python programu, kjer bomo vklope in izklope izvajali v zanki.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
while (True):
    print "LED on"
    GPIO.output(18,GPIO.HIGH)
    time.sleep(1)
    print "LED off"
    GPIO.output(18,GPIO.LOW)
```

Kot lahko vidimo, smo dodali le zanko while. Dokler je pogoj resničen (angl. True), toliko časa se izvaja. Koliko časa konkretno se bo izvajal? Dokler ne prekinemo izvajanja programa.

Tako smo uspešno podali primer utripanja sijalke LED.

Izmenično utripanje svetilk LED

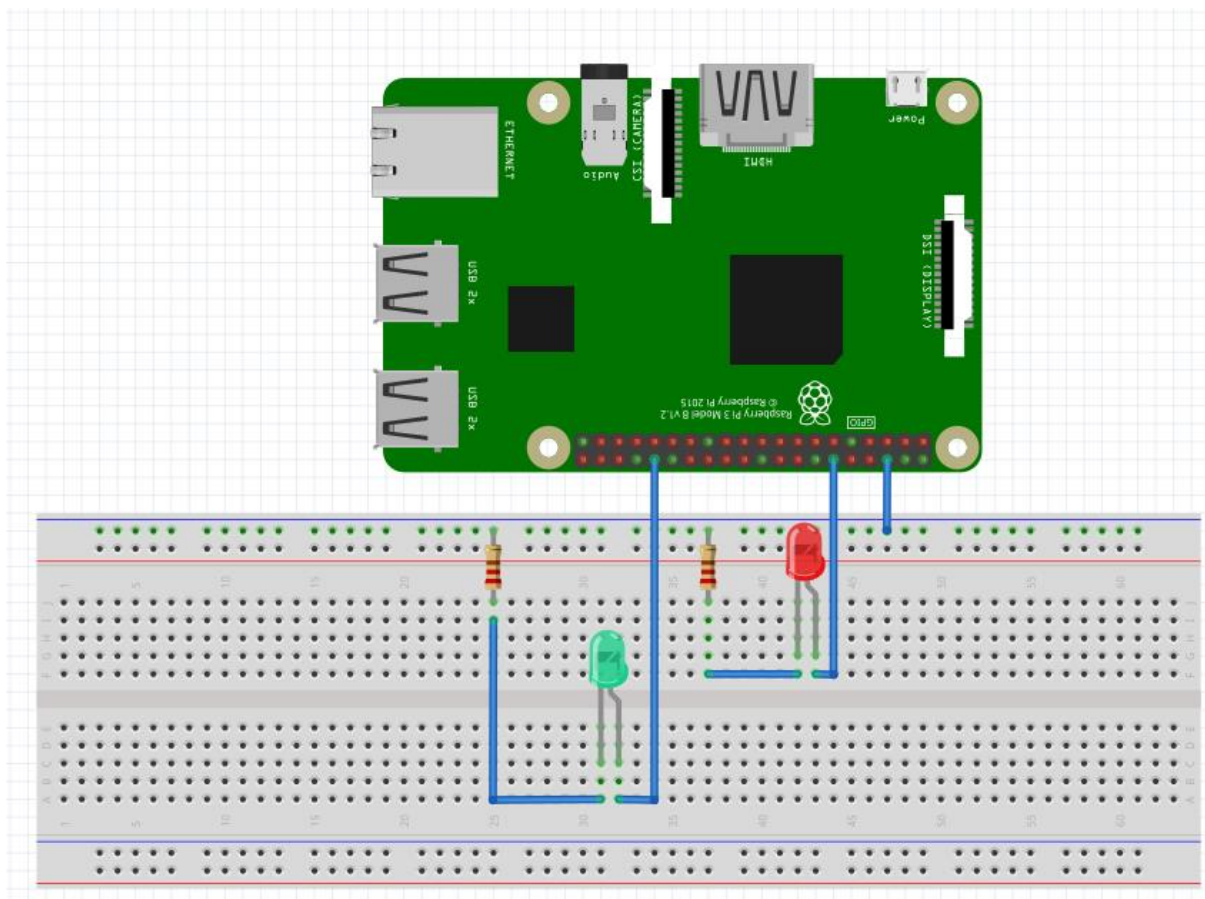
Prejšnjo nalogo si malce popestrimo in dodamo še eno svetilko LED. Tokrat želimo, da se bosta prižigali izmenično.

Za projekt potrebujemo:

- lučki LED poljubne barve
- napravo RPi
- ustrezna upora glede na lučko LED (običajno je v kompletu priložen 230 Ω ali 1 k Ω).

Shema vezave

Tudi tokrat si bomo narisali to, kar želimo izdelati kot končno rešitev.



Slika 30: Povezava dveh sijalk LED

Na obstoječo shemo iz prejšnjega primera, dodamo še eno sijalko LED, kjer smo spremenili bravo s klikom na lastnosti, da bo zelene barve. Priklop je izveden na GPIO številka 12.

Program za izmenično utripanje sijalk LED

V program dodamo ustrezní priključek GPIO številka 12, katerega smo uporabili v shemi in končni vezavi. Ker imamo dve sijalki LED, pred vstopom v zanko preventivno vrednosti obeh postavimo na LOW, torej poskrbimo, da sta preverjeno obe sijalki izključeni. Sledi izmenično utripanje rdeče in zelene sijalke z zamikom ene sekunde.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(12,GPIO.OUT)
GPIO.output(12, GPIO.LOW)
GPIO.output(18, GPIO.LOW)
while (True):
    GPIO.output(18,GPIO.HIGH)
```

```
time.sleep(1)
GPIO.output(18,GPIO.LOW)
GPIO.output(12,GPIO.HIGH)
time.sleep(1)
GPIO.output(12,GPIO.LOW)
```

Tudi tokrat se zanka izvaja toliko časa, dokler ne prekinemo izvajanja programa python.

Temperaturni senzor

V kit kompletih je pogosto priložen senzor temperature, ki ga lahko uporabimo za branje temperature okolja. Običajno je to senzor DS18B20, v novejših kompletih pa ga nadomešča senzor temperature in vlage.

Način reševanja projekta zavisi od sensorja, ki ga bomo uporabili. V našem primeru bo to kar klasični DS18B20.

Za projekt potrebujemo:

- Napravo RPi.
- Temperaturni senzor npr. DS18B20.
- Upor vrednosti 4,7 k Ω (barvna lestvica rumena, vijoličasto rdeča, zlata).
- Testna plošča (angl. Breadboard).
- Kable za povezavo.

Temperaturni senzor DS18B20

Za uporabljeni temperaturni senzor pred izdelavo sheme pogledamo, kakšna je shema vezave.



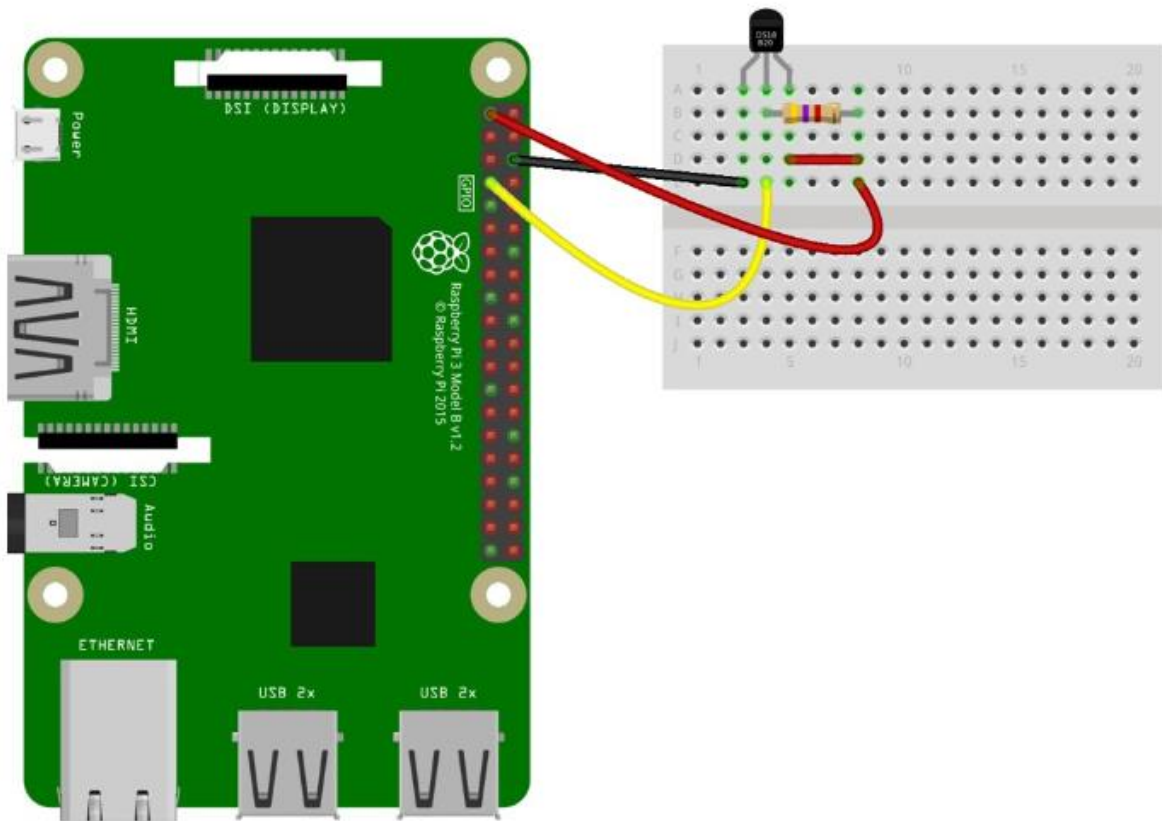
Slika 31: DS18B20

Vidimo lahko, da moramo gledati vezavo od leve proti desni, če upoštevamo, da je ploski del senzorja obrnjen proti nam.

Nožica 1 predstavlja GND, nožica 2 komunikacijo in nožica 3 napajanje 3,3 do 5 V.

Shema vezave

Priporočamo, da si za izbrani senzor preverite, kakšna je zahtevana vezava v sami dokumentaciji (navodilih), da ne bo prišlo do napačne vezave in uničenja senzorja.



fritzing

Prikazan imamo primer vezave, kjer smo se navezali na prej prikazano dokumentacijo senzorja, da ne bo prišlo do težav pri delovanju senzorja.

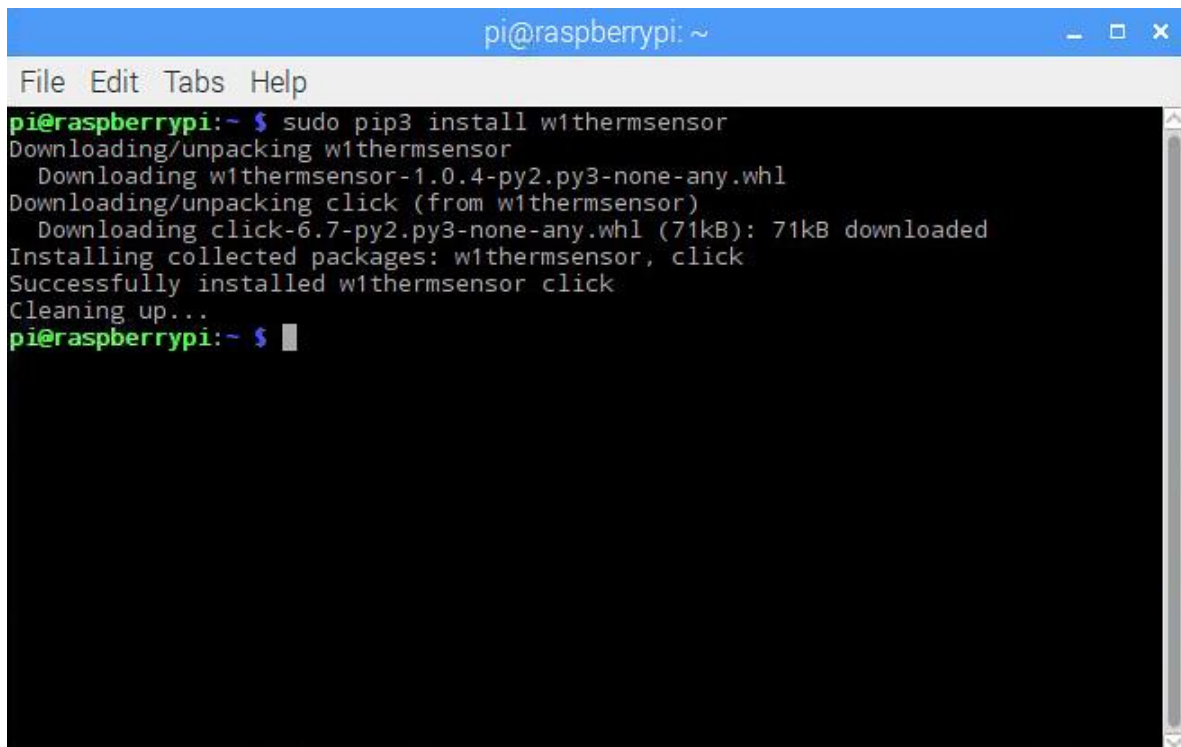
Program za branje temperature

Preden napišemo kodo za branje podatkov, moramo namestiti ustrezen paket, ki nam bo komunikacijo s senzorjem omogočal. Namestiti je potrebno paket `w1thermsensor` preko ukazne vrstice.

To izvedemo tako, da v ukazno vrstico (angl. Terminal) vnesemo spodnji ukaz.

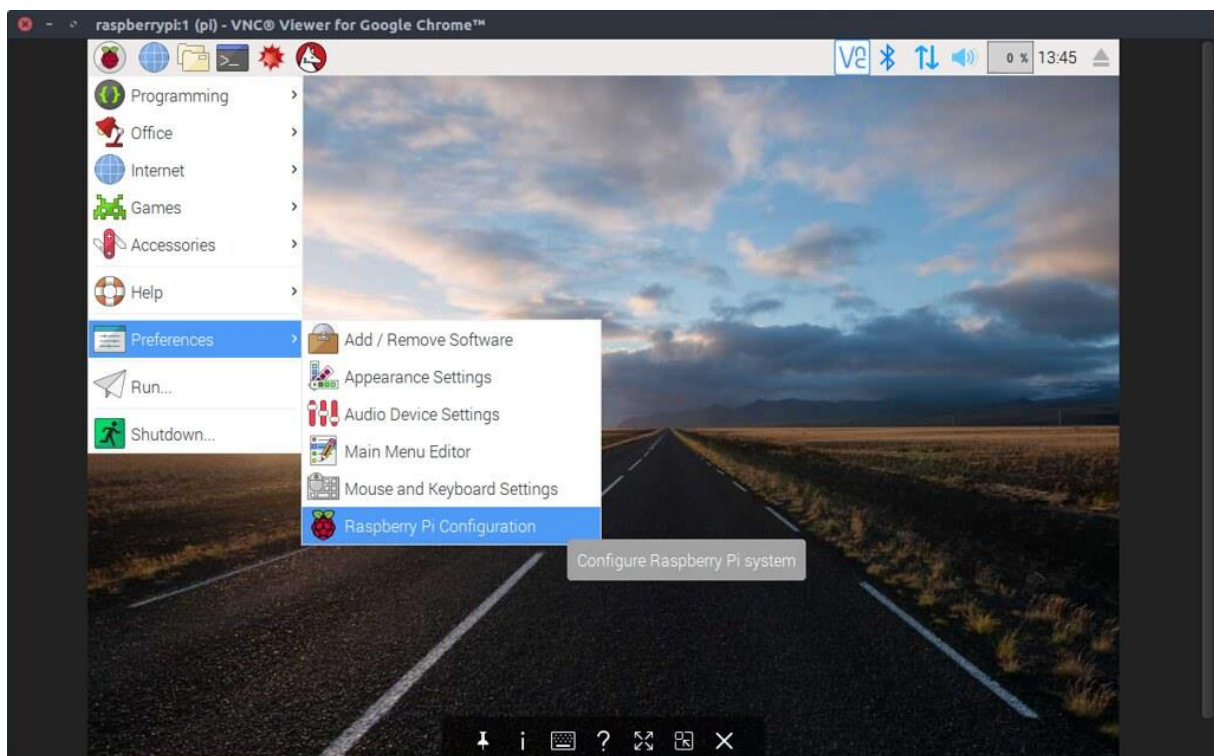
```
sudo pip3 install w1thermsensor
```

Sledilo bo nameščanje paketov, ki nam bo vzelo nekaj trenutkov.



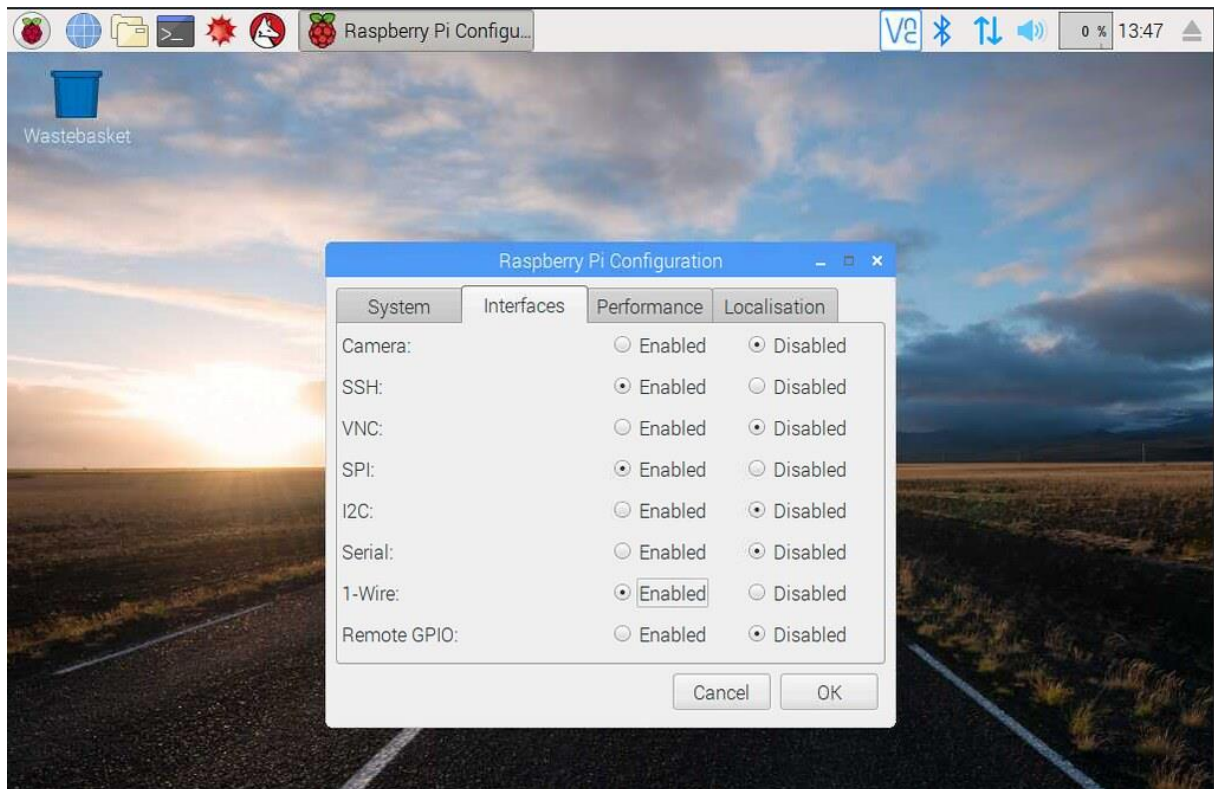
Slika 32: Namestitev paketa za podporo senzorju

Izbrani senzor omogoča komunikacijo preko t. i. 1 wire vmesnika, zato moramo to opcijo vključiti tudi preko nastavitve naprave RPi.



Slika 33: Nastavitev naprave RPi

Ko sem vmesnik za konfiguracijo odpre, spremenimo vrednost 1-wire na omogočeno.



Slika 34: 1-wire vmesnik vklop

S potrditvijo izberemo novo nastavitvev, ki pa bo veljavna šele, ko se bo naprava ponovno zagnala.

Sledi pisanje kode, kjer uporabimo nameščeno knjižnico in komunikacijo preko 1-wire komunikacije.

```
import time
from w1thermsensor import W1ThermSensor
senzor = W1ThermSensor()

while True:
    temperatura = senzor.get_temperature()
    print("Temperatura je %s v stopinjah" % temperatura)
    time.sleep(1)
```

Iz podanega primera bi lahko hitro izpeljali zanimivo kombinacijo, in sicer, če je temperature npr. Višja od 25 °C, potem naj se prižge rdeča sijalka LED, ki bi jo dodatno povezali, sicer naj gori zelena sijalka LED.

Izmerimo razdaljo

Ideja projekta je, da s pomočjo ustreznega ultrasoničnega senzorja izmerimo razdaljo, kar nam lahko pride prav pri projektih, kjer naredimo robota, ki se bo samodejno gibal v prostoru in zaznaval ovire.

V našem primeru želimo zaznati oviro oz. predmet, kar nam omogoča zaznavanje razdalje.

Za projekt potrebujemo:

- Napravo RPi.
- Ultrasonični senzor npr. HCSR04P. Pozor obstaja tudi različica brez oznake P.
- Ustrezne žičke za vezavo.

Še enkrat opozorimo, da imamo dve različici senzorja, ki sta si med seboj sila podobni. V našem projektu bomo uporabili HCSR04P različico, ki se od običajne HCSR04 razlikuje v samem vezju (izdelavi).



Na levi strani se nahaja senzor HCSR04P, na desni strani pa se nahaja senzor HCSR04, torej brez oznake P. Z ostrim očesom boste hitro opazili razliko, in sicer srebrni element v sredini med senzorjema. V primeru uporabe slednjega moramo paziti, saj potrebujemo napetostni delilnik, ki ga lahko ustvarimo s pomočjo dveh ustreznih uporov in ustrezne vezave. Če se temu lahko izognemo, ob izbiri prvega, zakaj ne bi naredili tako?

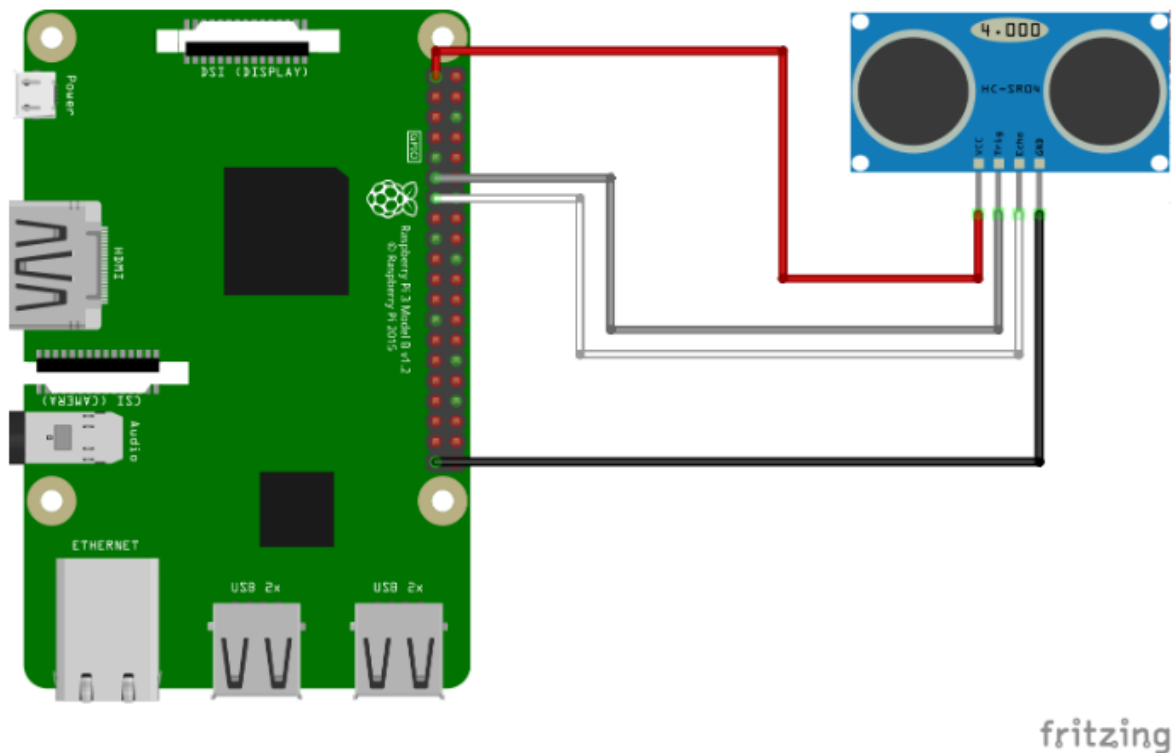
Vezavo izbranega senzorja lahko izvedemo neposredno, in sicer preko sledeče sheme.

SENZOR	RPI
VCC	3,3 V
Trig	GPIO17
Echo	GPIO27
GND	Uporabimo enega izmed GND

Ponovimo, da izbira GPIO zapisi od razpoložljivih priključkov GPIO.

Shema vezave

Ob upoštevanju vsega zapisanega, izvedemo načrtovanje sheme vezave v okolju Fritzing.



Slika 35: Shema vezave ultrasoničnega senzorja

Upoštevač prikazano tabelo, izvedemo ustrezno vezavo in se lotimo povezave na naši napravi RPi.

Program za branje razdalje

Preden zapišemo kodo moramo razumeti še izračun razdalje. Za izračun razdalje upoštevamo sledečo formulo:

$$\text{Razdalja} = \text{hitrost} * \text{čas}$$

Če torej delamo z enoto metri, hitrost predstavlja hitrost zvoka v metrih na sekundo (m/s). Čas je tako izračun z deljenjem začetnega časa pulza, do končnega časa pulza.

```
from gpiozero import DistanceSensor
from time import sleep

senzor = DistanceSensor(trigger=17,echo=27)

while True:
    print('Razdalja do najbližjega objekta je: ', senzor.distance, 'm')
    sleep(1)
```

Na srečo imamo tudi tokrat paket, ki ga lahko uvozimo in uporabimo. DistanceSensor je namenjen komunikaciji in delu s senzorjem, kar nam olajša pisanje kode.

```
from gpiozero import DistanceSensor
```

Izvedemo uvoz gpiozero paketa, kjer uporabimo DistanceSensor za izračun razdalje.

```
from time import sleep
```

Uvoz paketa time nam omogoči, da uporabimo funkcijo sleep, ki je namenjena časovnemu zamiku.

```
senzor = DistanceSensor(trigger=17,echo=27)
```

V spremenljivko senzor dodamo vrednosti, ki ju preberemo preko priključkov GPIO, in sicer 17 in 27, kot smo načrtovali v samem začetku.

V zanki izvajamo izpisovanje podatkov, dokler ne prekinemo delovanja kode programa. Sicer bi se to izvajalo v neskončnost.

```
print('Razdalja do najbližjega objekta je: ', senzor.distance, 'm')
```

Izpisujemo razdaljo med senzorjem in približanim objektom.

```
sleep(1)
```

Po vsakem izpisu počakamo 1 sekundo in ponovimo zanko.

Program nam bo v ukazni vrstici izpisoval želeno vrednost, ki pa bo v dolgi obliki s cca 20 števili za vejico. Da bi takšen izpis olepšali, lahko uporabimo funkcijo round(), ki nam bo omejila in zaokrožila vrednost na želeno število decimalnih mest.

```
from gpiozero import DistanceSensor
from time import sleep
```

```
senzor = DistanceSensor(trigger=17,echo=27)

while True:
    print('Razdalja do najbližjega objekta je: ', round(senzor.distance, 2), 'm')
    sleep(1)
```

V prikazanem primeru nam funkcija round() zaokroži izpis na dve decimalni mesti.

Izdelajmo predvajalnik zvokov (brenčalo)

Namen projekta je izdelava preproste vezave, kjer bomo uporabili brenčalo (angl. buzzer) za predvajanje preprostih zvokov.

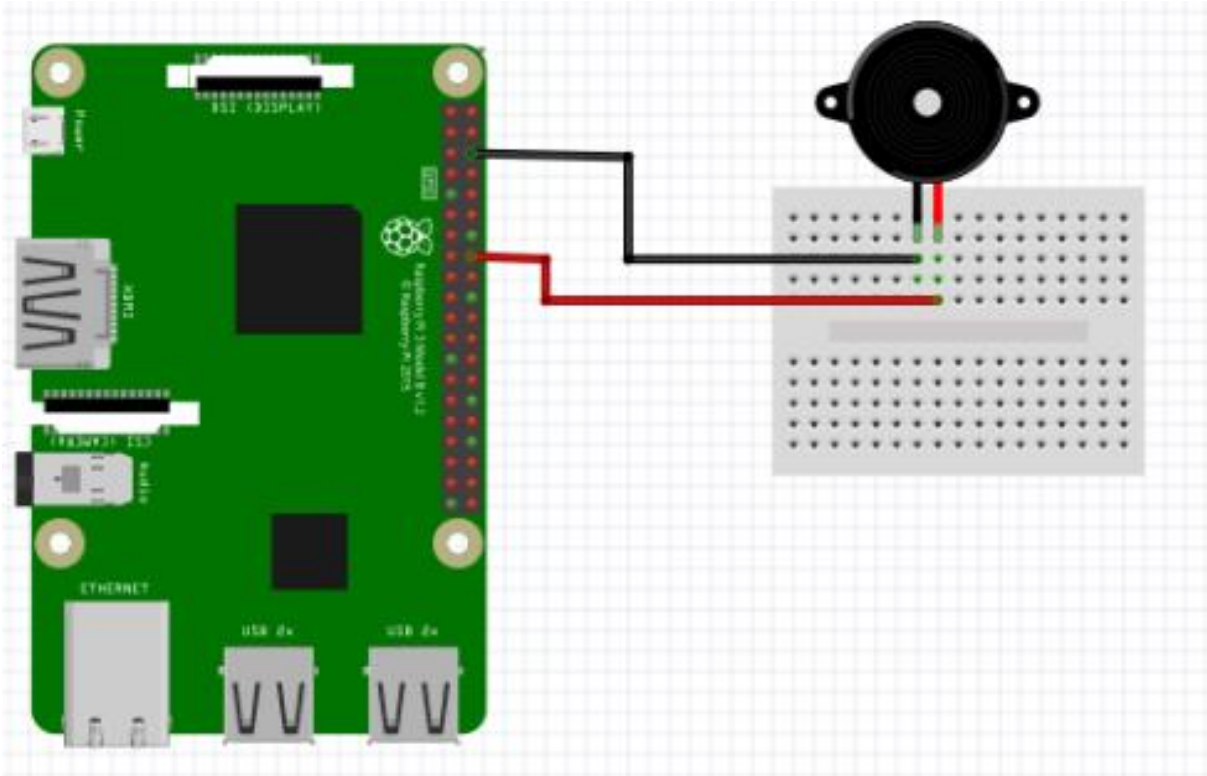
Kaj potrebujemo?

- Napravo RPi z ustreznim sistemom OS.
- Testno ploščo.
- Zvočnik (angl. buzzer).
- Ustrezne žičke za povezavo.

Izdelajmo ustrezno shemo vezave.

Shema vezave

Tudi tokrat si pripravimo ustrezno shemo, ki nam bo služila za osnovo izdelave projekta. Ker imamo lahko različne zvočnike, od tega zavisi tudi način vezave. Da pojasnimo, od tega je odvisno, če bo potreben kakšni vmesni upor ali ne. Pri klasičnem primeru, kjer govorimo o 12 mm okroglem mini zvočniku, ki predvaja zvoke v zvočnem območju 2 kHz upor niti ne bo potreben.



Slika 36: Shema vezave mini zvočnika

Vidimo lahko, da je vezava preprosta in ne potrebujemo dodatnih vmesnih členov.

Program za predvajanje zvoka

Za ustrezno komunikacijo moramo uvoziti ustrezno knjižnico, in sicer RPi.GPIO, kot smo to počeli že do sedaj. Tudi tokrat uvozimo knjižnico za upravljanje časa, ki bo služila za časovnik.

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
buzzer=23
GPIO.setup(buzzer,GPIO.OUT)
while (True):
    GPIO.output(buzzer,GPIO.HIGH)
    print ("Pisk")
    sleep(0.5)
    GPIO.output(buzzer,GPIO.LOW)
    print ("Brez piska!")
    sleep(0.5)
```

Če pogledamo kodo, potem lahko vidimo, da je precej podobna do sedaj videni.

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

Uvozimo modul za upravljanje s priključki GPIO.

```
from time import sleep
```

Časovnik nam bo služil, da bomo lahko ustvarili kratke prekinitve med predvajanjem piska.

```
GPIO.setwarnings(False)
```

Če želimo, lahko onemogočimo obvestila, ki so vezana na opozorila oz. dogodke.

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

Nastavimo način delovanja priključkov GPIO kot BCM.

```
buzzer=23
```

Spremenljivki buzzer shranimo vrednost 23, kar je glede na shemo vrednost priključka GPIO, preko katerega bomo upravljali vklop/izklop zvočnika.

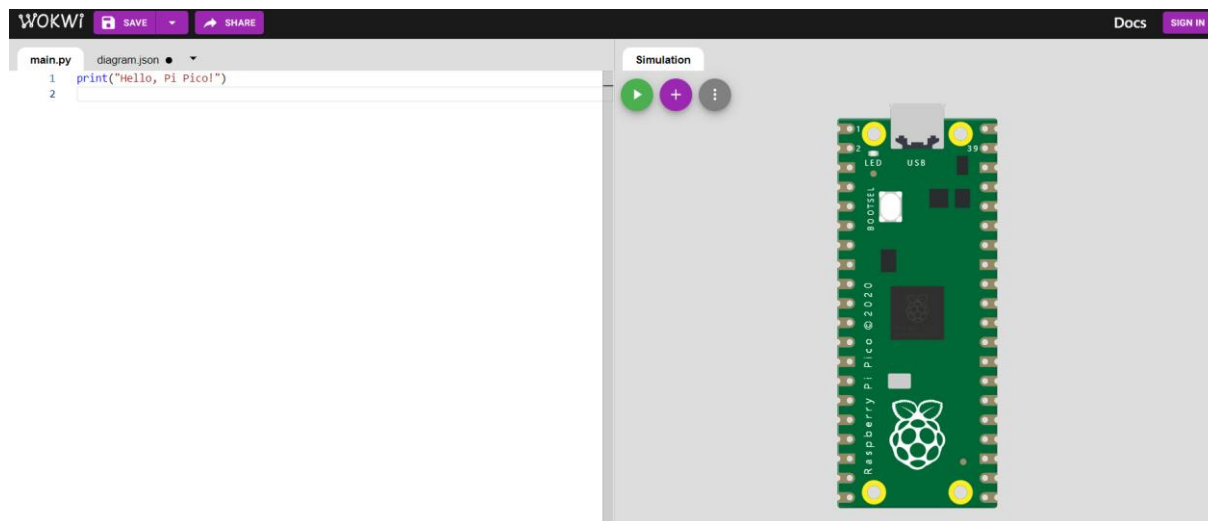
```
while (True):  
    GPIO.output(buzzer,GPIO.HIGH)  
    print ("Pisk")  
    sleep(0.5)  
    GPIO.output(buzzer,GPIO.LOW)  
    print ("Brez piska!")  
    sleep(0.5)
```

V zanki, ki se bo izvajala dokler je ne bomo prekinili, se bo izvajal vklop in izklop zvočnika preko GPIO priključka, kjer bomo imeli zamike po 0,5 sekunde.

13. Raspberry Pi — Pico simulator

Za vse tiste, ki bi se radi učili in testirali svoje znanje a hkrati ne želite kupiti naprave zgolj za testiranje, potem obstaja tudi rešitev, ki bo primerna za vas, in sicer simulacija delovanja naprave, kjer priključimo komponente, napišemo kodo i dejansko testiramo delovanje v spletnem okolju.

Ena izmed takšnih spletnih rešitev je obisk naslova <https://wokwi.com/projects/new/micropython-pi-pico>, kjer imamo možnost, da preverimo kodo in vezavo komponent.



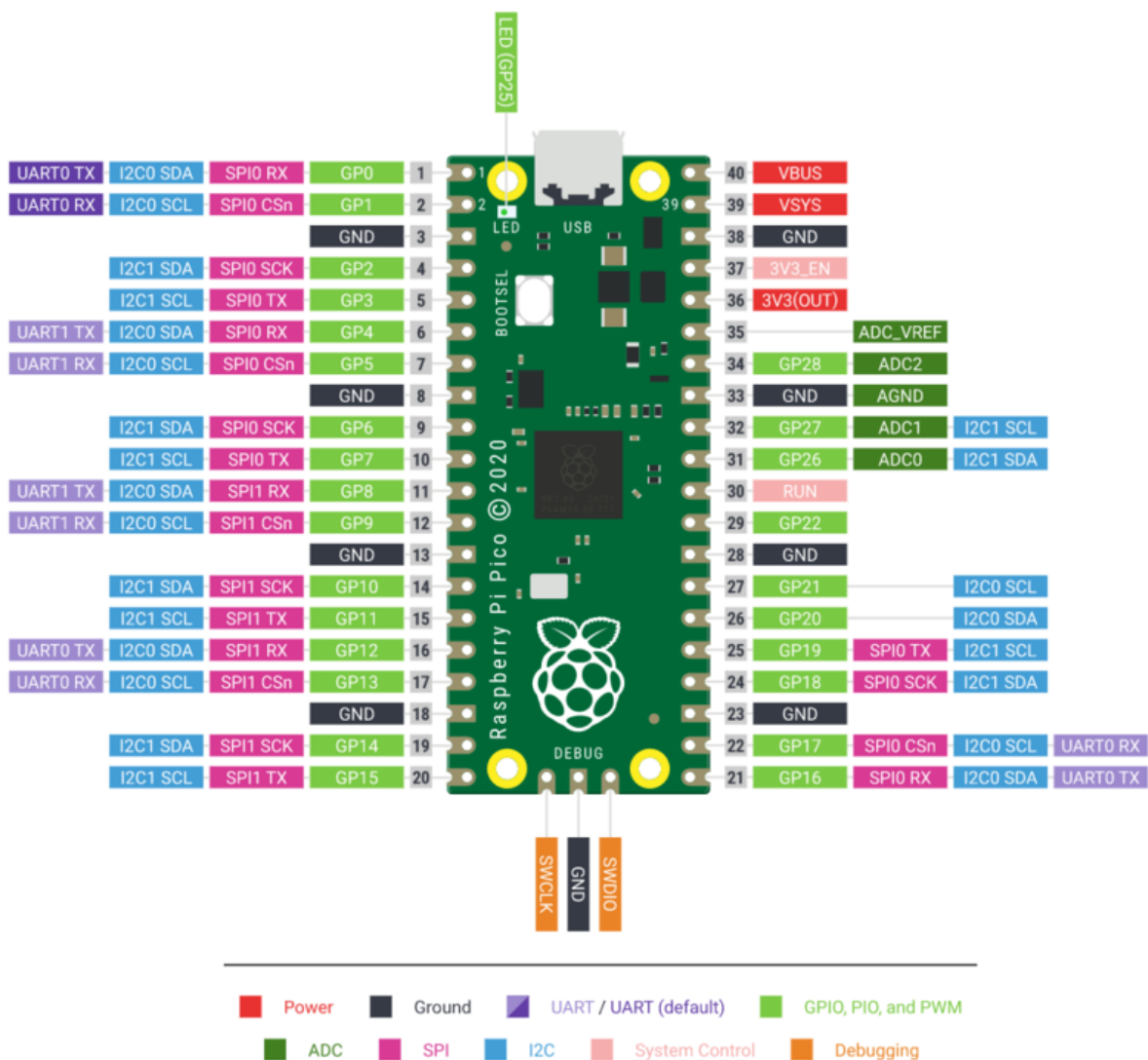
Slika 37: Simulator RPi Pico

Res je, da smo vseskozi pisali, kako naredimo projekt na napravi RPi 3, vendar ne pozabimo, da smo v začetku knjige zapisali, da obstaja več modelov naprav. Zato bomo tokrat vzeli za osnovo model Pico, kjer bomo pokazali, da je ključno razumevanje načina dela in ne katero napravo konkretno uporabljamo. Model naprave bolj določa zmogljivost in tehnične lastnosti, kot so priključki, število priklopov, naprave za komunikacijo itn.

Pico

Naprava Pico spada v skupino naprav RPi. Zanimiva je predvsem zaradi velikosti, cene in uporabnosti, kjer smo še bolj omejeni z velikostjo.

Shema vezav je sledeča.



Slika 38: GPIO shema vezav na napravi Pico

Micropython

Micropython predstavlja enostaven in učinkovit način, da implementiramo jezik python 3 z ustreznimi knjižnicami v projekte, ki jih načrtujemo. Okolje je prilagojeno različno mikro krmilnikom, med drugim tudi platformi RPi. Osnova za delovanje je plošča pyboard, ki simulira delovanje RPi Pico, kot je tudi prikazano na sliki.

Micropython je v celoti kompatibilen z običajnim okoljem python, kar omogoča prenos in izkoristek napisane kode brez popravkov. Deluje tudi na drugih mikro krmilnikih in podpora različne programske jezike ne zgolj okolja python.

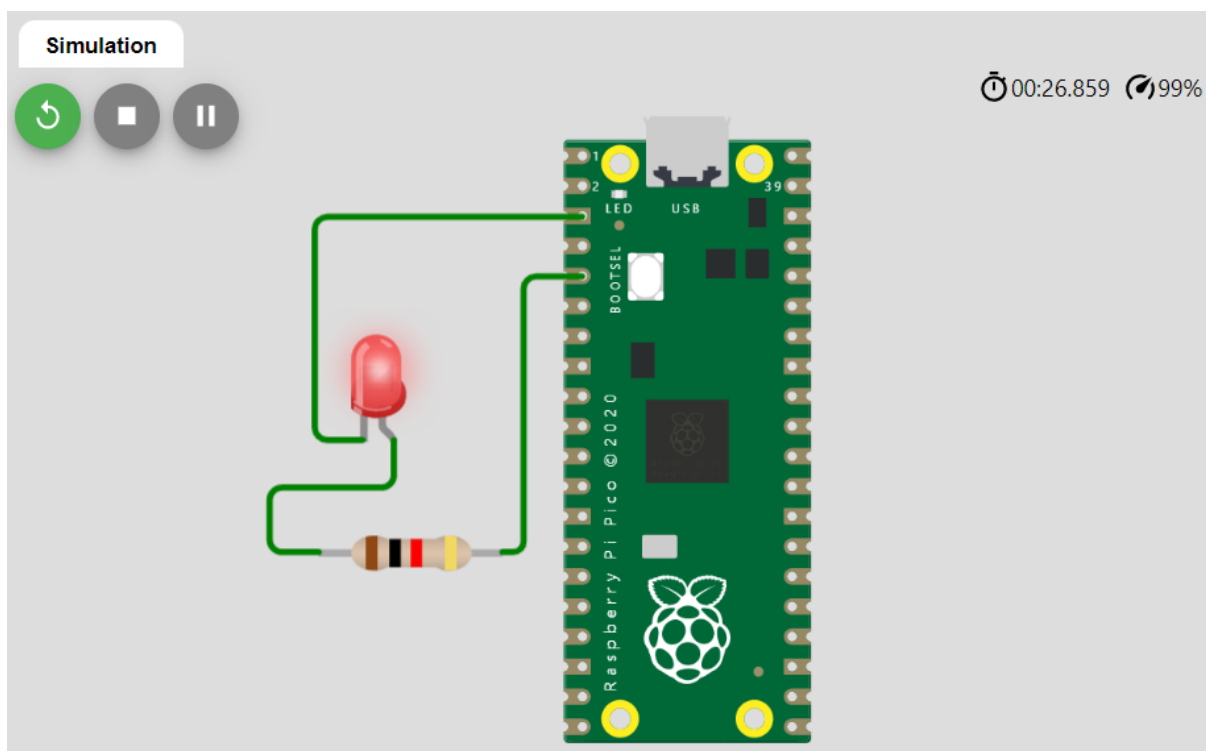
Utripanje svetilke LED

Poglejmo si osnovni primer, in sicer utripanje svetilke LED. Tokrat seveda to izvedemo na projektu Pico.

Kaj potrebujemo? Okolje, ki smo ga navedli v začetku tega poglavja. To je vse, res je. Preprosto kajne?

Še vedno lahko narišemo shemo, če tako želimo. Ker pa tukaj izvajamo virtualne povezave, ki jih povezujemo preko spleta, lahko izvedemo kar neposredno povezavo. Kje nastane težava? Da iz prikazanega ne bomo na prvi pogled razbrali, kam smo vezavo izvedli. No razbrati se da, le malce bolj podrobno je potrebno pogledati.

Vidimo lahko, da smo izbrali PIN številka 3 oz. GPIO 1 za GND in PIN številka 5 oz. GPIO 3 za +3,3V. Paziti moramo na številke in oznake, ki se med seboj razlikujejo.



Slika 39: Utripanje svetilke LED Pico

Na sliki bomo težko prikazali utripanje svetilke, lahko pa zajemamo sliko tako, da se vidi prižgana svetilka v nekem trenutku.

Program za utripanje

Koda na napravi Pico bo malce drugačna od kode, ki smo jo uporabljali do sedaj. Poglejmo si preprost primer, kako lahko upravljamo s priključki GPIO.

```
from machine import Pin, Timer
led = Pin(3, Pin.OUT)
timer = Timer()

def blink(timer):
    led.toggle()

timer.init(freq=2.5, mode=Timer.PERIODIC, callback=blink)
```

Vidimo lahko, da namesto uvoza GPIO uvozimo priključke t. i. Pin-e in časovnik.

```
from machine import Pin, Timer
```

Sledi kreiranje spremenljivke, kjer bomo imeli vrednost priključka, kjer bomo vklop in izklop izvajali.

```
led = Pin(3, Pin.OUT)
```

Kreiramo spremenljivko, kjer bomo upravljali s časom.

```
timer = Timer()
```

Ustvarimo funkcijo, ki bo poskrbela za utripanje.

```
def blink(timer):
    led.toggle()
```

Sledi še klic časovnika, kjer podamo frekvenco utripanja in klic funkcije za utripanje (angl. blink).

```
timer.init(freq=2.5, mode=Timer.PERIODIC, callback=blink)
```

Uporaba

Prikazali smo le možnost, ki nam je na voljo. Ni edina, je pa uporabna različica. Zakaj bi se je poslužili? Da testiramo način delovanja, morda pred nakupom naprave ugotovimo, če nam ustreza njeno delovanje ali zgolj uporabimo napredno tehnologijo, da privarčujemo in ne kupimo fizične naprave. Vsak se bo odločal sam in našel poti, kako rešitev čim bolje izkoristiti.

14. Zaključek

Od začetka priročnika do konca smo podali nekaj usmeritev, kako se v prvih korakih spoznati z napravo Raspberry PI. Pogledali smo njene ključne lastnosti in zmogljivosti, po katerih se modeli med seboj razlikujejo. Sledil je prikaz nameščanja in upravljanja z operacijskim sistemom in njegova prilagoditev.

Ko smo imeli sistem pripravljen, smo se lotili povezave komponent in pisanja kode python. Pomagali smo si z okolje Fritzing, kjer smo izrisali ustrezno shemo. Ključno pri tem je, da si z narisano shemo pomagamo do cilja. Vsekakor pa je ključni korak prilagajanje kode. Ko to počnemo sami je dobro, da se malce pozabavamo in priključimo na svoje vrednosti priključke. Na srečo večje škode običajno ne povzročimo, se pa zato toliko več naučimo.

Pri samem programiranju je sicer dobro, da imamo kakšno predznanje v okolju python. Uporaba priključkov GPIO je le dodatni korak, ki ga v povezavi z izbrano napravo uporabimo, da izvedemo ustrezno vezavo in komunikacijo (izmenjavo podatkov). Skozi primere hitro spoznamo, da potrebujemo tudi znanje okolja Linux in uporabe ukazne vrstice (terminala).

Skratka naprava Raspberry PI predstavlja projekt, kjer povezujemo različna znanja v zaključeno celoto. Projekti so zabavni in širijo naše razumevanje. Seveda pa je to priročnik, ki bo prišel prav predvsem tistim, ki boste osnovna znanja imeli in boste ob tem prejeli tudi ustrezno razlago z demonstracijo.

Pregled znanj seveda ne predstavlja poglobitve v vsa zapisana področja, saj bi vsako izmed njih lahko imelo svojo obsežno knjigo. Namen je, da imamo povzeta ključna znanja, ki nam bodo podala smernice za delo in prve korake. Od tukaj naprej bo na posamezniku, da se v zadane izzive poglobi in najde ustrezne rešitve.