



## Uvod

U ovoj laboratorijskoj vježbi ćemo se upoznati sa računarom Raspberry Pi 2B, baziranim na procesoru sa četiri ARM Cortex-A7 jezgra, koja rade na 1GHz, te sa 1GB RAM memorije.

Informacije o Raspberry Pi se mogu pronaći na <https://www.raspberrypi.org/>.

## Raspberry Pi OS

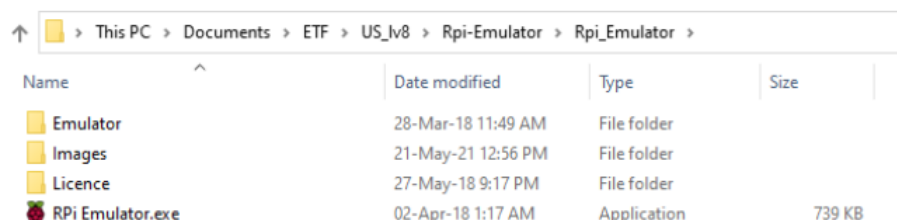
Tipično, na Raspberry Pi je instaliran operativni sistem, putem kojega aplikacija onda koristi hardverske resurse. Najrašireniji operativni sistem za Raspberry Pi je **Raspberry Pi OS**. Radi se o verziji Debian distribucije operativnog sistema Linux kompajlirane za ARM, sa potrebnom podrškom za hardverske resurse koji se nalaze na Raspberry Pi.

Raspberry Pi OS se može preuzeti sa <https://www.raspberrypi.com/software/operating-systems/>. Od ponuđenih verzija, mi ćemo koristiti najnoviju verziju bez desktopa (Raspberry Pi OS Lite - bullseye).

Da bismo se upoznali sa Raspberry Pi OS, koristićemo emulator **QEMU**. Radi se o open source emulatoru različitih arhitektura, kao što su: x86, PowerPC, ARM, MIPS, SPARC, OpenRISC i druge. QEMU se može preuzeti sa sajta <https://www.qemu.org/>, ali se studentima koji koriste PC sa Windows operativnim sistemom preporučuje korištenje pripremljenog QEMU emulatora u okviru aplikacije RPi Emulator. Studenti koji koriste Linux mogu QEMU instalirati sa repozitorija distribucije.

## RPi Emulator

RPi Emulator se može preuzeti sa [ovog linka](#). Nakon preuzimanja je potrebno raspakirati zip fajl, a njegov sadržaj će izgledati kao na slici 1.



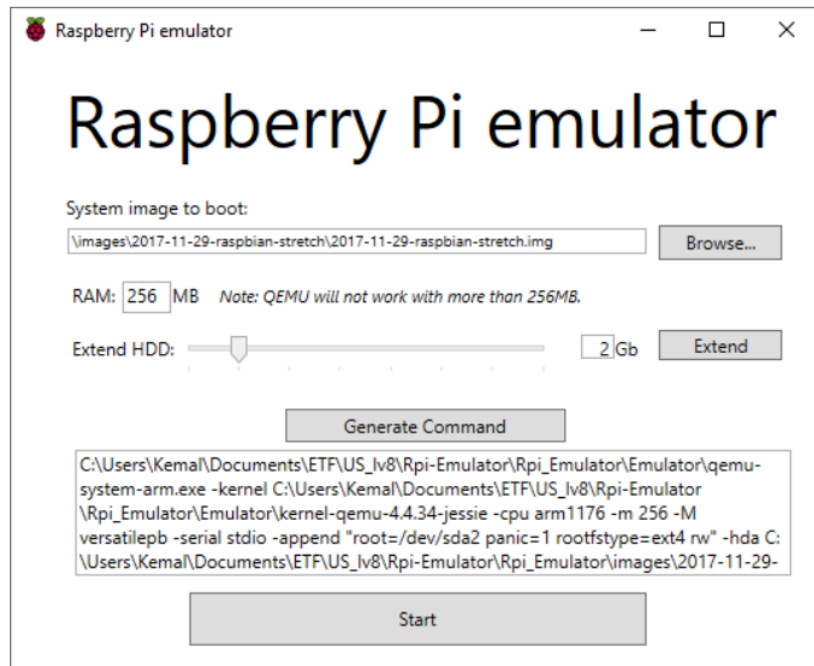
Name	Date modified	Type	Size
Emulator	28-Mar-18 11:49 AM	File folder	
Images	21-May-21 12:56 PM	File folder	
Licence	27-May-18 9:17 PM	File folder	
RPi Emulator.exe	02-Apr-18 1:17 AM	Application	739 KB

Slika 1: Sadržaj zip fajla sa RPi Emulatorom.

### Napomena

Voditi računa da staza do foldera u koji je raspakiran zip fajl ne smije sadržavati razmake (space), inače RPi Emulator neće raditi.

Image Raspberry Pi OS-a treba raspakirati u folder u kome se nalazi RPi Emulator. Nakon pokretanja aplikacije *RPi Emulator.exe*, otvoriće se prozor kao na slici 2.



Slika 2: Inicijalni prozor RPi Emulatora.

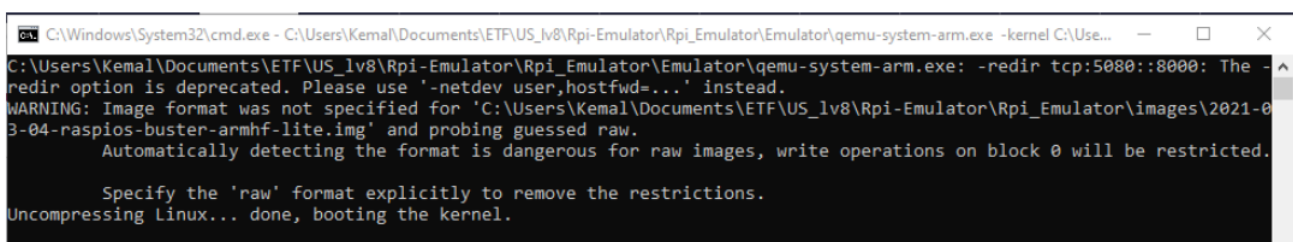
Prije pokretanja emulatora je potrebno podesiti nekoliko opcija. Kao prvo, potrebno je odabrati Image fajl koji će se emulirati, što se postiže pritiskom na <Browse> sa prethodne slike, te lociranjem željenog fajla. Osim toga, potrebno je odabrati količinu RAM-a (ostaviti 256MB) i prostora za pohranu (minimalno 2GB) koji se dodjeljuju RPi Emulatoru.

Aplikacija RPi Emulator u stvari samo pokreće QEMU koji je raspakovan u folderu aplikacije. Da bi se QEMU ispravno pokrenuo, potrebno je navesti ispravne parametre. Linija naredbe koja pokreće QEMU je prikazana u donjem dijelu prozora RPi Emulatora. Za potrebe realizacije laboratorijske vježbe će biti potrebno izmijeniti naredbu, tako što ćemo izmijeniti `-redir tcp:2222::22` u `-redir tcp:5080::8000`. Opcija `-redir` preusmjerava port host računara na kome se izvršava QEMU na port emuliranog računara.

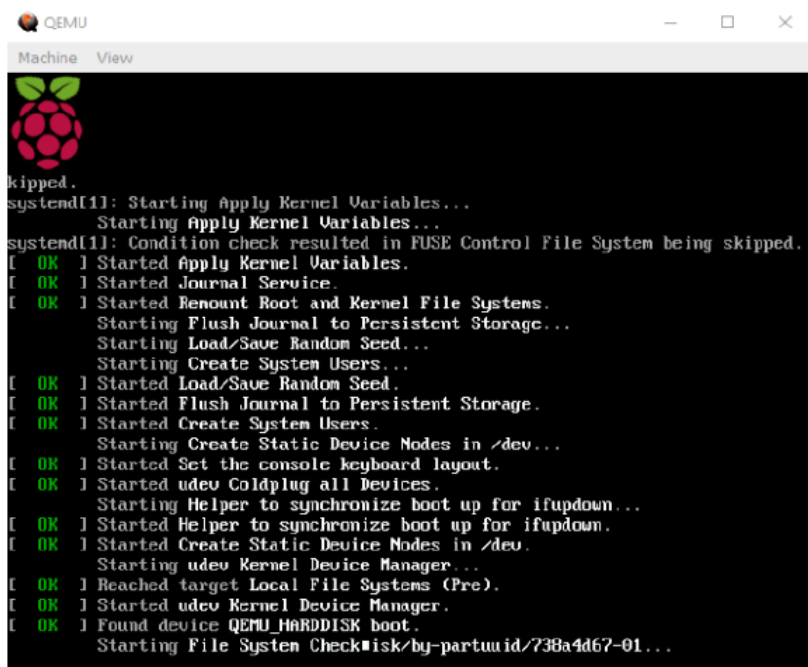
### Napomena

Nemojte pritisnuti dugme <Generate Command>, jer će to dovesti do poništavanja izmjena unesenih u komandu.

Sada možemo pokrenuti QEMU pritiskom na <Start>, što će rezultirati otvaranjem dva prozora. Jedan je Command Prompt kao na slici 3, i u njemu je u stvari pokrenut QEMU. U drugom prozoru (slika 4) će se odvijati interakcija sa emuliranim Raspberry Pi. Svaki ispis na *stdout* emuliranog Raspberry Pi će biti prikazan u ovom prozoru.

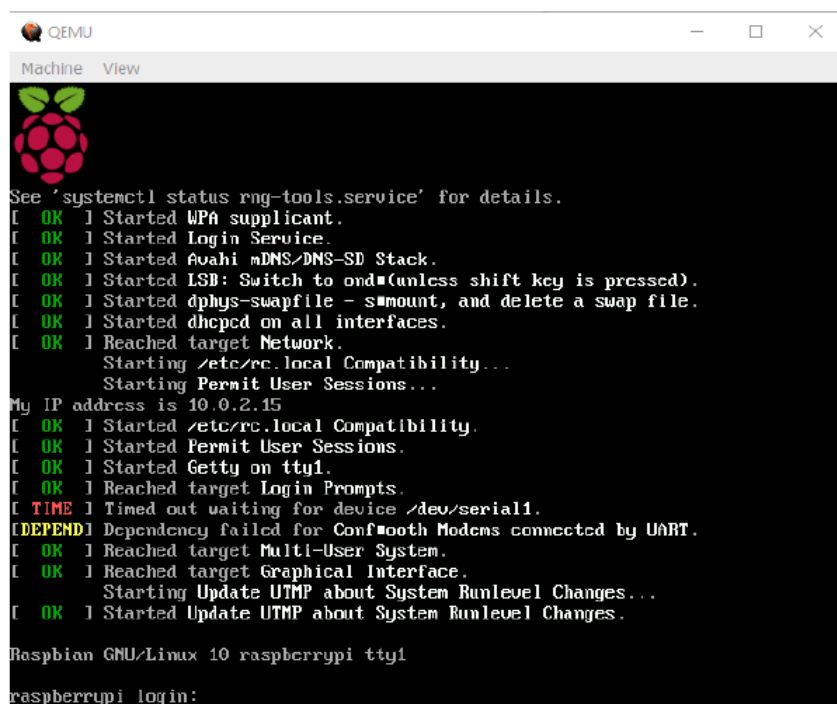


Slika 3: Izgled Command Prompt-a u kome je pokrenut QEMU.



Slika 4: Izgled prozora u kome se odvija interakcija sa emuliranim Raspberry Pi.

Biće potrebno sačekati da se pojavi *login* prompt emuliranog Raspberry Pi (slika 5, što će potrajati nekoliko minuta.



Slika 5: Login prompt emuliranog Raspberry Pi.

## Rad sa Raspberry Pi OS

Prije početka rada sa emuliranim Raspberry Pi, potrebno se prijaviti na sistem sa korisničkim imenom i lozinkom:

- login: **pi**
- password: **raspberrypi**

## Napomena

Voditi računa da inicijalno raspored znakova na tastaturi može biti engleski (EN), u kojem slučaju će tasteri znakova Y i Z biti zamijenjeni.

Nakon uspješne prijave, pojaviće se komandni prompt terminala kao na slici 6. Na komandni prompt se mogu unositi i izvršavati standardne *shell* naredbe Linux-a<sup>1</sup>.

Najosnovnije *shell* komande koje ćemo koristiti su:

- `clear` - briše sadržaj terminala;
- `pwd` (*print working directory*) - ispisuje u terminalu trenutno aktivni direktorij (kompletnu putanju) - ako se komanda izvrši nakon prijave na sistem, to je `/home/pi`, što znači da se nalazimo u direktoriju `pi`, koji je poddirektorij u okviru `home`.
- `ls` - ispisuje sve fajlove i poddirektorije koji se nalaze u trenutno aktivnom direktoriju.
- `mkdir` (*make directory*) - kreira novi poddirektorij u trenutno aktivnom direktoriju (ili u direktoriju čija staza je navedena).
- `cd` (*change directory*) - mijenja aktivni direktorij.
- `cp` (*copy*) - kopira fajl ili fajlove.
- `sudo` (*super-user do*) - daje korisniku root/super-user privilegije.

Još neke *shell* komande koje ćemo koristiti su:

- `who`,
- `whoami`,
- `uname -a`,
- `df`,
- `free`,
- `shutdown now`,
- `reboot`.

Više informacija o osnovnim komandama `bash shell`-a se može pročitati u [1].

## Hint

Uz *shell* komande se tipično mogu koristiti različite opcije, koje specificiraju detaljnije način izvršavanja komande. Tako npr. `ls -l` ispisuje detaljne informacije o sadržaju aktivnog direktorija. Više informacija za svaku komandu se može dobiti ispisivanjem *help* teksta komande (`ls --help` ili `ls -h`) ili putem *man*-stranice (npr. `man ls`, pa odabirom opcije 1). Iz prikaza *man*-stranice se izlazi pritiskom na taster Q.

## Hint

Prilikom unosa staze, imena direktorija ili fajla, pritiskom na tipku <TAB> se može automatski dopuniti unos. Ukoliko postoji više fajlova/direktorija koji zadovoljavaju uslov, novim pritiskom na <TAB> će biti ispisane sve opcije.

<sup>1</sup>Više informacija se može pročitati na [https://en.wikipedia.org/wiki/Unix\\_shell](https://en.wikipedia.org/wiki/Unix_shell) i [https://en.wikipedia.org/wiki/Bash\\_\(Unix\\_shell\)](https://en.wikipedia.org/wiki/Bash_(Unix_shell))

## Hint

U terminalu strelicama gore/dole na tastaturi možete browse-ati historiju komandi. Tako je puno lakše za ponoviti neku komandu, ili editirati prethodnu, sličnu komandu.

## Instalacija Raspberry Pi OS na SD karticu

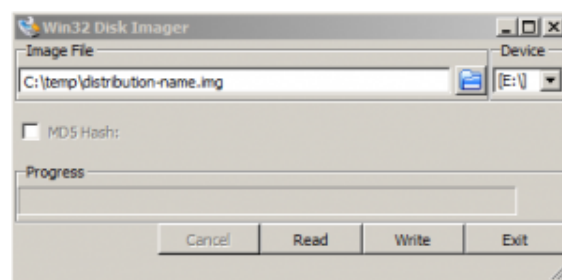
Da bi se na stvarnom Raspberry Pi mogao pokrenuti operativni sistem, potrebno ga je pohraniti na SD karticu. U nastavku će biti pojašnjen način na koji se to radi.

Za prebacivanje image-a Raspberry Pi OS (koji je prethodno preuzet, kako je gore pojašnjeno) na SD karticu se može koristiti npr.:

- Win32DiskImager (Windows),
- Raspberry Pi Imager (Windows, Linux, macOS),
- dd (Linux).

Način korištenja **Raspberry Pi Imager**-a je pojašnjen u kratkom videu na linku <https://www.youtube.com/watch?v=ntaXWS8Lk34>.

Aplikacija **Win32DiskImager** se može preuzeti sa linka <https://sourceforge.net/projects/win32diskimager/files/latest/download>. Nakon pokretanja aplikacije, pojaviće se prozor kao na slici 6. Treba voditi da se aplikacija mora pokrenuti sa administratorskim privilegijama.



Slika 6: Izgled prozora Win32Disk Imager-a.

Nakon odabira željenog image fajla, potrebno je ubaciti SD karticu i odabrati je, te pritisnuti na dugme **Write**. Nakon toga će početi pisanje na SD karticu, a progress bar će prikazivati tok operacije.

Aplikacija **dd** se poziva sa komandnog prompta Linux terminala unošenjem i izvršavanjem komande `dd bs=4M if=./ime_image_fajla.img of=/dev/sdX`, gdje **sdX** predstavlja određeni uređaj (može se odrediti izvršavanjem komande `df`).

## Napomena

Voditi računa da se odabere ispravan određeni uređaj (SD kartica), jer će podaci sa određeniog uređaja biti izgubljeni.

## Pokretanje Raspberry Pi OS sa SD kartice i konfigurisanje sistema

Nakon što je image Raspbery Pi OS distribucije prebačen na SD karticu, karticu možemo staviti u odgovarajući slot na Raspberry Pi računaru. Da bismo mogli završiti postupak konfigurisanja Raspberry Pi, moguće je:

- koristiti tastaturu, miša i odgovarajući monitor (televizor) sa HDMI ili sa kompozitnim video ulazom. Ako Raspberry Pi računaru pristupamo putem tastature, miša i sa

prikazom na monitoru/televizoru, rad se ne razlikuje od rada sa bilo kojim drugim PC računarom. Nakon što se Raspberry Pi priključi na napajanje, pokrenuće se operativni sistem sa SD kartice, te će se na kraju prikazati desktop sa uobičajenom organizacijom (ako je instalirana desktop verzija) ili login prompt konzole (ako je instalirana lite verzija).

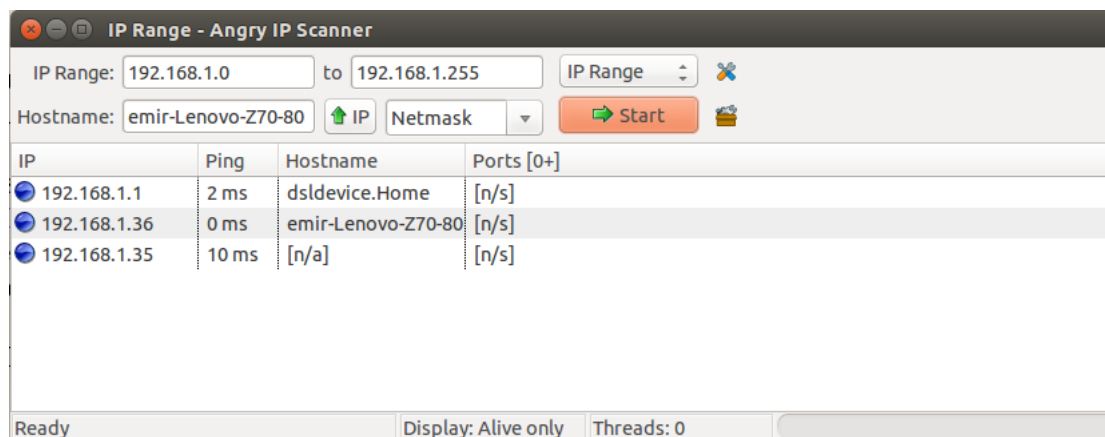
- korištenjem aplikacije koja podržava SSH protokol i komandnog prompta PC računara, ukoliko je Raspberry Pi spojen putem LAN-a. U tom slučaju se ne zahtijeva spajanje tastature/miša/monitora na Raspberry Pi. **Ovaj pristup će biti korišten na laboratorijskim vježbama iz US.** Na Windows operativnim sistemima se može koristiti neka aplikacija koja omogućava SSH (npr. *Putty*), dok na Linux operativnim sistemima postoji komanda `ssh`.

### Napomena

Da bi na Raspberry Pi OS-u bio omogućen pristup putem SSH, potrebno je aktivirati SSH server. To se može postići tako što se na *boot* particiji SD kartice kreira fajl sa imenom `ssh` (fajl treba biti prazan). Pri tome treba voditi računa da se na Windows sistemu ne doda ekstenzija `.txt` (treba biti samo ime fajla, bez ekstenzije).

Ukoliko Raspberry Pi računar pristupamo putem mreže, biće potrebno poznavati IP adresu Raspberry Pi računara. Obzirom da se ova adresa u inicijalnoj konfiguraciji Raspberry Pi OS distribucije dobiva od DHCP servera, adresu je moguće pročitati kada pristupimo routeru na kome se izvršava DHCP server. Ukoliko nemamo pristup ovom routeru, adresu je moguće odrediti i skeniranjem lokalne mreže na koju je povezan Raspberry Pi računar.

Na laboratorijskim vježbama iz Ugradbenih sistema preporučljivo je koristiti *Angry IP scanner* (<http://angryip.org/download/>). Izgled prozora ove aplikacije je prikazan na slici 7, a ona omogućava jednostavno lociranje uređaja spojenih u LAN. Tako se jednostavno može odrediti IP adresa Raspberry Pi računara.



Slika 7: Izgled prozora aplikacije *Angry IP scanner*.

### Napomena

Obzirom da će se pri radu u laboratoriji u istu LAN spajati više Raspberry Pi računara, da bi se znalo tačno koji računar pripada kojem radnom mjestu (timu), potrebno je da se spajanje LAN kablova sinhronizira među timovima (odnosno da predmetni asistent koordinira uključivanje računara na LAN). U protivnom se može desiti da timovi pristupaju Raspberry Pi računarima njihovih kolega.

Ako za pristup Raspberry Pi računarima koristimo *Putty*, nakon pokretanja aplikacije na PC računar je potrebno odabrati **SSH** i postaviti **IP adresu** Raspberry Pi računara. Nakon što

se pojavi zahtjev za unos korisničkog imena i lozinke, potrebno je unijeti inicijalne vrijednosti (korisnik `pi`, lozinka `raspberrypi`). Ako za pristup Raspberry Pi računar koristimo komandu `ssh`, onda ćemo komandnom promptu na Raspberry Pi računar pristupiti izvršavanjem komande `ssh -l pi xxx.xxx.xxx.xxx`. Ovdje je umjesto `xxx.xxx.xxx.xxx` potrebno upisati adresu Raspberry Pi računara.

Da bi se izvršila inicijalna konfiguracija Raspberry Pi računara (ili naknadna konfiguracija), potrebno je izvršiti komandu `sudo raspi-config`. Ova komanda pokreće konfiguracijsku skriptu (slika 8), koja omogućava npr.:

- Proširivanje fajl sistema na cijeli kapacitet kartice (akcija **Advanced Options->Expand Filesystem**) - Raspberry Pi OS image je kreiran tako da bude manji od SD kartice kapaciteta 4GB, te je ova akcija neophodna kako bi se mogao iskoristiti cijeli kapacitet SD kartice (16GB ili više).
- Promjena korisničke lozinke (akcija **System Options->Password**) - voditi računa da se nova lozinka zapamti, inače će biti potrebno ponoviti cijeli postupak prebacivanja image-a i konfigurisanja.
- Podešavanje lokalnih postavki, kao što su tastatura, jezik i sl. (akcija **Localisation Options**) - ovdje se može odabrati željeni jezik, vremenska zona, kao i raspored znakova na tastaturi.

Nakon izlaska iz skripte za konfigurisanje, pritiskom na `<Enter>` na opciji **Finish**, nova konfiguracija će biti zapisana. Ovim je Raspberry Pi u potpunosti konfigurisan i funkcionalan.

### Napomena

Raspberry Pi računar ne treba isključivati tako što će se prekinuti napajanje. Na taj način rizikujemo da se ošteti sadržaj SD kartice, nakon čega bi često jedino rješenje bilo ponavljanje procedure instaliranja OS, uz neminovni gubitak podataka na SD kartici. Da bismo Raspberry Pi isključili na ispravan način, potrebno je ili u terminalu izvršiti komandu `halt` ili `shutdown now` (za ovo treba imati privilegije root korisnika), ili izvršiti *Shut down* korištenjem miša, u slučaju da sa Raspberry Pi računarom komuniciramo putem tastature, miša i monitora. Isključivanje napajanja je dozvoljeno tek nakon što se na Raspberry Pi ugase sve LED, izuzev crvene LED koja indicira prisustvo napajanja.

## ZADACI

### Zadatak 1

Raspberry Pi 2 Model B ima određen broj GPIO (Digitalnih ulaza/izlaza opšte namjene eng. **General Purpose Input/Output**) koji rade na logičkim nivoima (0/3.3V) i nisu tolerantni na 5V. Ovi digitalni ulazi/izlazi se mogu koristiti na više načina<sup>2</sup>.

U ovom zadatku ćemo se upoznati sa korištenjem virtualnog fajl sistema `sysfs` za ispis vrijednosti na digitalne izlaze, odnosno čitanje vrijednosti sa digitalnog ulaza.

GPIO SoC-a su izvedeni na *header* (slika 8).

#### Digitalni izlazi

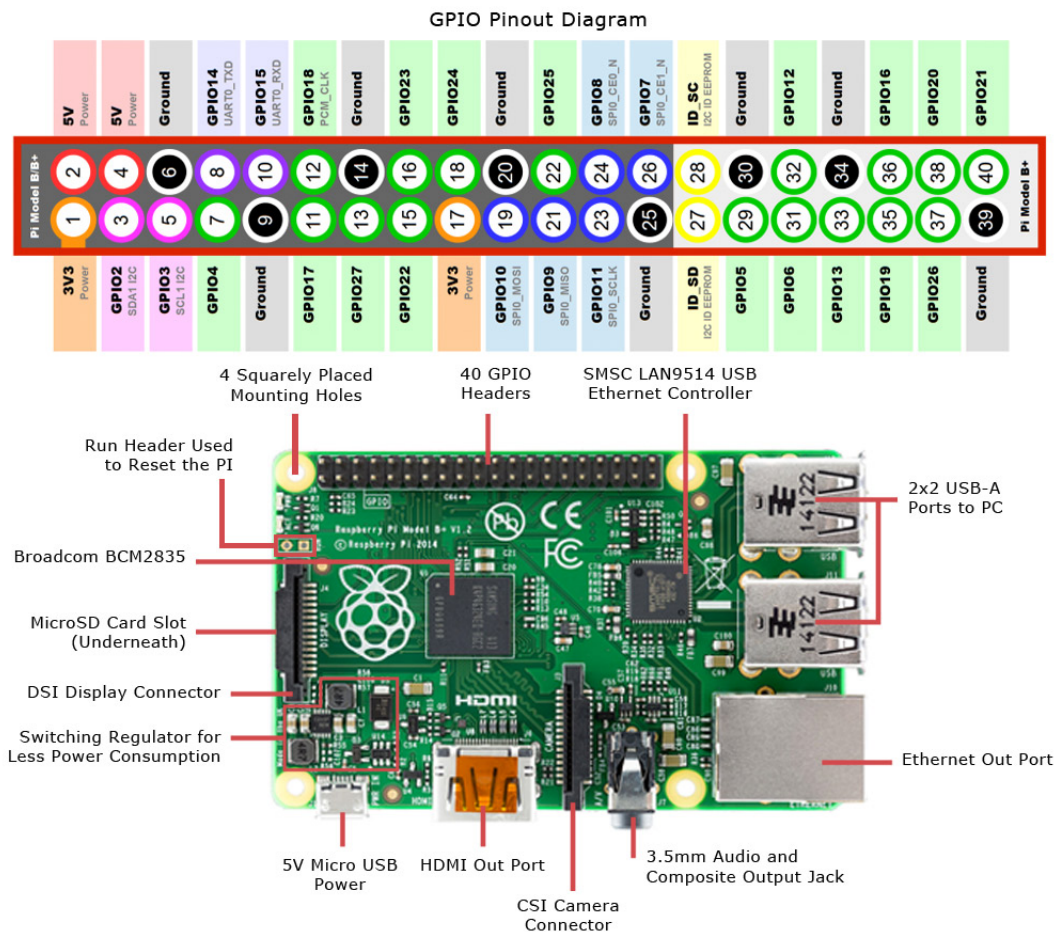
Na GPIO Raspberry Pi računara koje koristimo na laboratorijskim vježbama su povezane LED:

---

<sup>2</sup>Vidjeti <http://codeandlife.com/2012/07/03/benchmarking-raspberry-pi-gpio-speed/> za prednosti i nedostatke različitih načina korištenja GPIO



- plava LED - GPIO2,
- zelena LED - GPIO3,
- crvena LED - GPIO4.



Slika 8: Raspored GPIO na *header*-u na Raspberry Pi 2B.

Da bi se npr. aktivirala plava LED (koja je povezana na GPIO2), potrebno je izvršiti sljedeće komande u terminalu (kao administrator / sudo):

- Kreiranje GPIO pristupnog fajla (2 se odnosi na GPIO2):  
`echo 2 > /sys/class/gpio/export`
- Konfiguracija smjera digitalnog pina (out - izlaz, in - ulaz):  
`echo out > /sys/class/gpio/gpio2/direction`
- Postavljanje digitalne vrijednosti:  
`echo 1 > /sys/class/gpio/gpio2/value`  
`echo 0 > /sys/class/gpio/gpio2/value`
- Brisanje (dealokacija) GPIO se obavlja korištenjem:  
`echo 2 > /sys/class/gpio/unexport`

### Digitalni ulazi

Digitalni ulazi se koriste veoma slično kao i digitalni izlazi. Kako bi se analizirao način korištenja digitalnih ulaza, na pin GPIO17 je spojen taster. Sljedeće komande omogućavaju čitanje tog digitalnog ulaza:



- Kreiranje GPIO pristupnog fajla:  
`echo 17e > /sys/class/gpio/export`
- Konfiguracija smjera digitalnog pina:  
`echo in > /sys/class/gpio/gpio17/direction`
- čitanje digitalne vrijednosti:  
`cat /sys/class/gpio/gpio17/value`  
(uočite razliku kada je taster pritisnut a kada ne!).
- Brisanje (dealokacija) GPIO se obavlja korištenjem:  
`echo 17 > /sys/class/gpio/unexport`

## Zadatak 2

Komande u terminalu se mogu pisati i u okviru tzv. *bash* skripti. One omogućavaju da se više komandi izvršava pokretanjem samo jedne skripte. Prva linija takve skripte je obavezno `#!/bin/bash` (čime se definira koja aplikacija izvršava skriptu), fajl se tipično snima sa ekstenzijom `.sh`. Da bi se *bash* skripta mogla izvršiti, potrebno je prethodno izvršiti komandu `chmod +x skripta.sh`. Nakon toga, skripta se pokreće korištenjem `./skripta.sh` (gdje je *skripta* naziv fajla). Više detalja oko sintakse se može naći na <http://ss64.com/bash/>. Skripte se mogu editirati korištenjem *nano* editora (pokretanje fajla u nano editoru vrši se naredbom `nano`, ili `nano ime_fajla` za editovanje postojećeg fajla)

Napraviti i izvršiti *bash* skriptu koja omogućava da LED dioda automatski mijenja stanje svake sekunde. Primjer *bash* skripte se nalazi na linku [https://c2.etf.unsa.ba/pluginfile.php/128632/mod\\_folder/content/0/US-LV7-zad2.sh?forcedownload=1](https://c2.etf.unsa.ba/pluginfile.php/128632/mod_folder/content/0/US-LV7-zad2.sh?forcedownload=1).

## Zadatak 3

GPIO računara Raspberry Pi se mogu koristiti i pomoću neke od raspoloživih biblioteka. Demonstriraćemo korištenje C-biblioteke *WiringPi*. Da bi se ova biblioteka mogla koristiti, potrebno je prvo preuzeti izvorni kod biblioteke, te ga kompajlirati. Postupak preuzimanja i kompajliranja je pojašnjen je u nastavku.

Prvo je potrebno instalirati **git** (izvršiti sljedeće instrukcije):

```
sudo su
apt-get install git-core
apt-get update
apt-get upgrade
```

Zatim je potrebno instalirati i kompajlirati *WiringPi*:

```
git clone https://github.com/WiringPi/WiringPi.git
cd WiringPi
./build
```

Nakon pripreme biblioteke *WiringPi* je potrebno napraviti program u C-u, koji omogućava da LED dioda automatski mijenja stanje svakih pola sekunde. Primjer koda se nalazi na linku [https://c2.etf.unsa.ba/pluginfile.php/128632/mod\\_folder/content/0/US-LV7-zad3.c](https://c2.etf.unsa.ba/pluginfile.php/128632/mod_folder/content/0/US-LV7-zad3.c). Aplikacija se pokreće naredbom `./US-LV7-zad3`.

## Zadatak 4

U ovom zadatku ćemo demonstrirati kako se mjerenja tri ose akcelerometra na mobilnom telefonu mogu iskoristiti za generiranje PWM signala na LED diodama Raspberry Pi računara.

Za tu svrhu ćemo koristiti namjensku Android aplikaciju za ovu laboratorijsku vježbu, koja omogućava očitavanje mjerenja akcelerometra i slanje izmjerenih vrijednosti pomoću http protokola http serveru implementiranom u Python-u, koji se izvršava na Raspberry Pi. Ovaj server će onda mjerenja koristiti da bi generirao vrijednosti duty cycle-a za plavu, zelenu i crvenu LED, u rasponu od 0-100%. Nako toga će biti moguće pomjeranjem telefona dobiti različite boje spektra, koje predstavljaju kombinaciju intenziteta svjetla koje daju plava, zelena i crvena LED. U nastavku su pojašnjeni koraci koje je potrebno provesti.

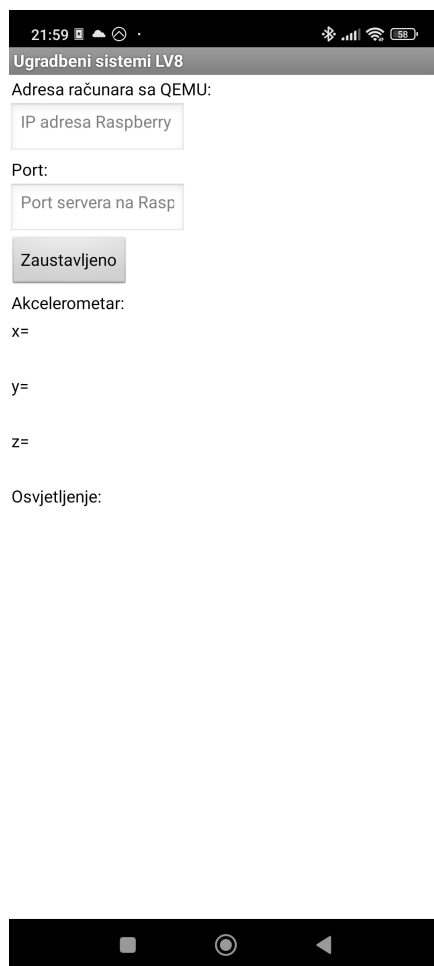
Na Raspberry Pi je potrebno kreirati fajl `http_svr.py`, čiji sadržaj se nalazi na linku [https://c2.etf.unsa.ba/pluginfile.php/128632/mod\\_folder/content/0/http\\_svr.py](https://c2.etf.unsa.ba/pluginfile.php/128632/mod_folder/content/0/http_svr.py). U okviru fajla je potrebno napraviti izmjenu linije `hostName = "192.168.0.154"`, u koju je potrebno upisati ispravnu IP adresu. Nakon toga je potrebno omogućiti izvršavanje fajla na način kao u zadatku 2, te pokrenuti http server izvršavanjem komande `./http_svr.py`.

Sa Android telefona je potrebno pristupiti linku [https://c2.etf.unsa.ba/pluginfile.php/128632/mod\\_folder/content/0/US-LV7.apk?forcedownload=1](https://c2.etf.unsa.ba/pluginfile.php/128632/mod_folder/content/0/US-LV7.apk?forcedownload=1), te preuzeti i instalirati mobilnu aplikaciju.

### Napomena

Da bi se APK fajl aplikacije mogao instalirati, biće potrebno omogućiti instaliranje aplikacija iz nepoznatih izvora.

Nakon pokretanja mobilne aplikacije, na telefonu će biti prikazan prozor kao na slici 9.



Slika 9: Izgled prozora mobilne aplikacije.

Sada je potrebno unijeti IP adresu računara Raspberry Pi, te port na kome se nalazi http server (8000). Klikom na dugme <Zaustavljeno> će započeti očitavanje akceleratorometra i izmjerene vrijednosti će se periodično slati http serveru. Na osnovu izmjerenih vrijednosti će http server generirati PWM signale za plavu, crvenu i zelenu LED.

#### Hint

Podešavanje PWM LED dioda na Raspberry Pi se može postići i bez mobilne aplikacije, unošenjem u browser URL-a `http://IP_ADRESA_RASPBERRY_PI:8000/?ax=10&ay=10&az=10&lux=130`, gdje se za očitavanje ax, ay i az mogu unijeti vrijednosti između -10 i 10.

## Zadatak 5

Upoznati se sa Python bibliotekom `RPi.GPIO` (<https://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/>), koja omogućava korištenje digitalnih ulaza i izlaza, te PWM. Napisati kôd koji omogućava da se realizira PWM na plavoj LED, pri čemu se svakim pritiskom na taster *duty cycle* uvećava za 10%. Nakon što *duty cycle* dosegne 100%, ponovo se vraća na 0.

## Literatura

- [1] V. Gedris (2003) *An Introduction to the Linux Command Shell For Beginners*