Računalo Raspberry Pi

Redžić, Leon

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:196136

Rights / Prava: Attribution 3.0 Unported/Imenovanje 3.0

Download date / Datum preuzimanja: **2024-03-28**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Leon Redžić

Raspberry PI

ZAVRŠNI/DIPLOMSKI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Leon Redžić
Matični broj: 43284/14–R
Studij: Informacijski sustavi

Raspberry PI

ZAVRŠNI/DIPLOMSKI RAD

Mentor:
Doc.dr.sc. Igor Balaban

Varaždin, lipanj 2018.

Sadržaj

1. Uvod 1
2. Računalno sklopovlje 2
2.1. Cortex A53 procesor 2
2.2. Memorija 3
2.3. Ethernet i 802.11 6
2.4. Napajanje 6
2.5. Grafički procesor i displej 7
3. Usporedba sa prijenosnim računalom (Lenovo G505) 7
3.1. Usporedba arhitekture 8
3.2. Usporedba mogućnosti periferije i portova 8
3.3. Usporedba brzina 9
3.4. Usporedba brzine izvođenja sortnih algoritama 11
3.5. Omjer cijene i brzine stolnog i Raspberry Pi računala 12
4. Praktični rad 14
4.1. Operacijski sustavi 14
4.1.1. Raspbian 14
4.1.2. Windows 10 IOT Core 15
4.1.3. Open Source Media center 15
4.1.4. LIBREELEC 15
4.1.5. RISC OS 16
4.2. Podizanje operacijskog sustava (Raspbian) na SD karticu i USB 16
4.3. LAMP Wordpress server 17
4.4. GPIO program 19
4.5. Internet zvučnik 24
5. Zaključak 29
6. Literatura 30

1. Uvod

Raspberry Pi računalo je namijenjeno da bude jeftino, dostupno i lagano za upotrebu te konfiguraciju svojim korisnicima. To je i uspjelo sa svojom cijenom od 300 kuna samog uređaja te performansama jačeg današnjeg mobitela sredinom 2018 godine. Računalo je veličine kreditne kartice te ima dosta portova za spajanje periferije. Kao i HDMI izlaz za povezivanje sa većinom današnjih televizora tako i monitora. Napaja se preko Micro USB porta što je također vrlo rašireno u doba pisanja ovog rada.

U svrhu završnog rada ću koristiti Raspberry Pi 3 računalo model 3B. Ovo računalo se razlikuje od ostalih modela tako da ima proširenu radnu memoriju, ugrađen čip za mrežno povezivanje koji podržava WiFi te Bluetooth povezivanje bez dodatnih modula koji zauzimaju portove. Prvo će se opisati arhitektura računala te njegovih komponenti poput procesora, RAM pločice, mrežnog povezivanja pomoću Ethernet-a te WiFi-a, nakon toga proučit će se brzina izvođenja radnji na računalu.

Nakon što se opiše cijeli sustav računala onda će se opisati ukratko prijenosno računalo Lenovo G505 koje će se koristiti za usporedbu sa Raspberry Pi računalom. Usporedit će se teoretske brzine te će se usporediti brzine koristeći isti benchmark sustav na obje platforme. Nakon toga provest će se testovi brzine izvođenja izabranih sortnih algoritama.

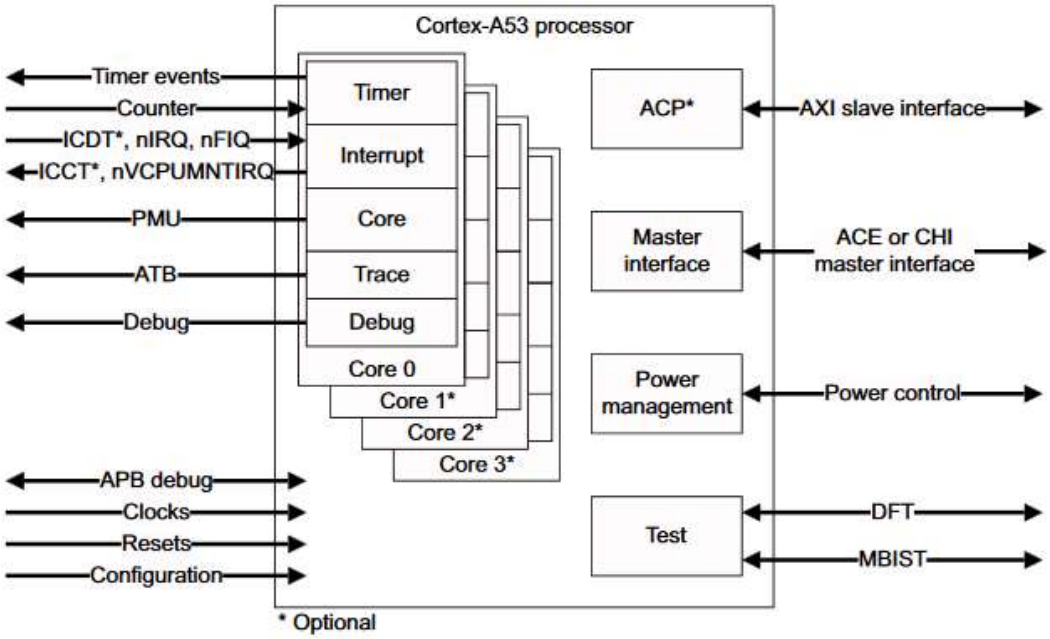
Za kraj će se izraditi mali sustav na računalu koji će pokretati web server koji će posluživati sučelje za upravljanje GPIO pinovima te audio izlazima samog Raspberry Pi računala. Nakon kojeg ću u zaključku dati osvrt na lakoću korištenja Raspberry Pi sustava tijekom razvoja.

2. Računalno sklopovlje

Promatrajući komponente Raspberry Pi uređaja, vidljivo je da su komponente birane na način da imaju nisku potrošnju energije te da budu maksimalno kompaktne u smislu svoje fizičke veličine. Kroz prve verzije Raspberry PI uređaja su komponente bile većinom iste obitelji komponenti te su samo napredovale i razvijale se kroz godine. Uređaj sadrži većinu izvornih komponenti samo kasnije generacije istih. Kroz modele se dodaju i druge komponente. Tako je npr. u verziji koju ću opisivati kroz rad (Raspberry 3 Model B) ugrađen Bluetooth modul te WiFi modul. Koji uvelike olakšava rad na uređaju. Stvar koju valja primijetiti jest da sklopovlje napreduje te se dodaju nove stvari no ostaje unazad kompatibilan odnosno ne uklanjanju stare funkcionalnosti te je stari kod i primjene računala kontinuirano ostvarive kroz generacije.

2.1. Cortex A53 procesor

Na Raspberry Pi 3 model B računalu nalazi se Cortex A53 mobilni procesor. Odnosno on je dio BCM2837 Broadcom čipa. Pošto je ovo računalo niske potrošnje energije i zamišljeno je da bude kompaktno ovo je mobilni procesor koji podržava i 64 bitni set instrukcija (AArch64) kao i 32 bitni set instrukcija (AArch32) prema Cortex A53 dokumentaciji.



Slika 1. Arhitektura procesora (Izvor: Cortex A53 Dokumentacija, ARM)

Cortex A53 je procesor koji implementira ARMv8-A arhitekturu. Procesor ima primjene u mobilnim uređajima, avijaciji, mrežnim uređajima, mreži medija za pohranjivanje (

SSD, HDD) te u automobilskim multimedijskim info sustavima. Ključne prednosti korištenja ovog procesora su: mala potrošnja energije, povećana brzina izvođenja instrukcija omogućavanjem dualnog instrukcijskog dekodiranja i izvršavanja (32-bit i 64-bit), velika štednja energije.

Procesor je super skalarni. Odnosno ima mogućnost da jedna procesorska jezgra izvršava više instrukcija u jednome ciklusu. Prema Flynn-ovoj arhitekturi procesor je MIMD (Multiple Instruction stream, multiple data stream) zbog svoje četiri jezgre. Ova arhitektura je povoljna za ovu primjenu jer dopušta veću količinu podataka da se obradi kao i pri većoj brzini na maloj pločici što je i sama ideja Raspberry Pi uređaja.

ARMv8-A arhitektura je donijela veliku promjenu naspram ARMv7 jer donosi opcionalni A64 64-bitni instrukcijski set. Funkcionalnosti Aarch64 arhitekture su:

- 31 64-bitni opći registar
- Posebni pokazivač stoga registar
- PC više nije dostupan kao registar
- Instrukcije su i dalje 32 bitne no dohvaćaju se u parovima
- Adrese su 64-bitne

L1 Cache sistem se sastoji od zasebnih instrukcijskih i podatkovnih cache sustava. Veličina L1 Cache sistema je 8KB. L1 cache je zaseban za svaku jezgru što znači da svaka jezgra ima po 8KB cache memorije. L1 cache memorija se puni principom kritična riječ prva(CWF)[1]. Odnosno bitniji instrukcijski blokovi se dohvaćaju prvi te se uštedi djelić vremena tokom izvođenja instrukcija. Ukupna veličina L1 cache memorije je 32 KB pošto imamo 4 memorije po 8 KB.

L2 Cache sistem se sastoji od SCU[2] (Snoop Control Unit) koji sadrži duplikate L1 Cache memorije svake jezgre kako bi se omogućilo komuniciranje među jezgrama te opcionalnu dodatnu L2 cache memoriju koja služi kao normalna L2 cache memorija. Raspberry Pi 3 Model B sadrži takvu vrstu L2 Cache memorije te je ona veličine 512KB

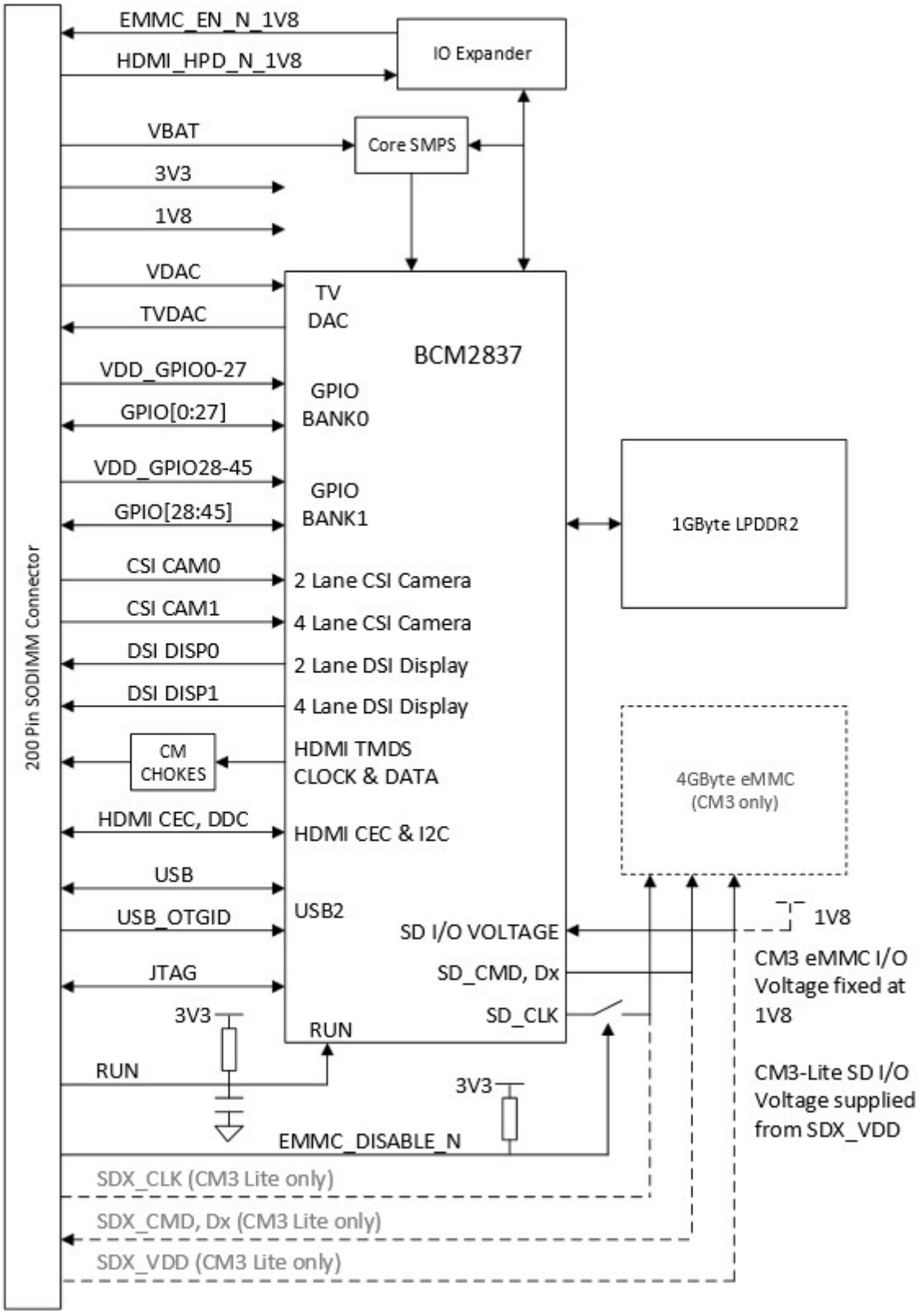
2.2. Memorija

[1] Cortex A53 dokumentacija, Poglavlje 2.1.1

[2] Cortex A53 dokumentacija,Poglavlje 7.1

Raspberry Pi 3 model B koristi 1GByte LPDDR2 (Low Power Double Data Rate 2 memorija) koja je specifična po tome da je niske potrošnje energije. Ova vrsta memorije ima brzinu unutarnje pohrane od 200 MHz odnosno 800 MT/s te zahtjeva 1.2 V napajanja.

Raspberry Pi 3 model B koristi Elpida B8132B4PB-8D-F čip koji se nalazi na dnu tiskane pločice Raspberry Pi računala.



Slika 2. Block dijagram RAM (Izvor: Raspberry Pi računalski modul dokumentacija)

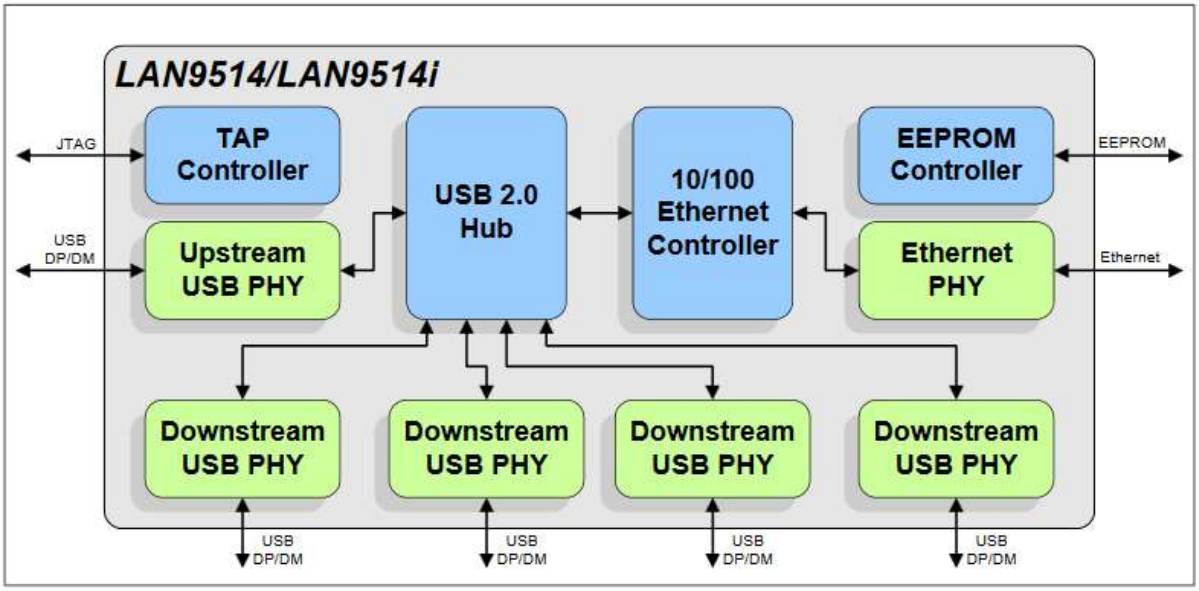
.

2.3. Ethernet i 802.11

Pošto se radi o Raspberry Pi 3 modelu B ima mogućnost Etherneta, te WiFi modul je ugrađen na ploču. Modul koji se koristi na Raspberry Pi računalu jest Elpida B8132B4PB-8D-F, modul je glavni zaslužan za ulazno izlazne operacije Raspberry Pi računala pošto podržava:

- 4 klijentska ulaza
- 1 poslužiteljski ulaz
- Ethernet MAC 10/100 sa punim dupleksom

Odnosno postoje 4 USB 2.0 porta koji služe kako bi se povezali vanjski periferni uređaji na Raspberry Pi te 1 poslužiteljski ulaz koji prenosi podatke sa 4 klijentska ulaza na procesor. Te Ethernet ulaz koji služi za povezivanje na računalne mreže



Slika 3. Blok dijagram mrežnog modula (Izvor: Dokumentacija LAN95414 čip)

2.4. Napajanje

Pločica Raspberry Pi 3 model B uređaja prima fiksnih 5 Volti za napajanje svih komponenti uređaja. Ova voltaža dopustila je implementaciju napajanja koristeći Micro USB koji je u današnje vrijeme vrlo široko rasprostranjen jer je trenutno standard za napajanje mobilnih uređaja uz skori prijelaz na USB tip C konektore. Koristeći Micro USB ima jednu implicitnu svrhu još a to je da ne dopušta korisniku da greškom spoji uređaj na pogrešnu voltažu te tako oštetiti sam uređaj. Pločica ima na sebi diodu koja sprječava da se zamjene polovi istosmjerne struje te otpornik koji služi za preuzimanje dijela tereta ukoliko napon varira.

Jakost struje koju uređaj treba primiti jest od 700 do 1500 mA uz normalnu upotrebu te osigurati moguću nadoknadu ukoliko se rade veće operacije. Potrošnja energije modela B jest 3.5W bez povezanih uređaja (HDMI, Ethernet, USB itd).

Još jedna mogućnost napajanja uređaje jest preko GPIO pinova. No to nije preporučeno pošto ne sadrže nikakvu zaštitu protiv male ili velike voltaže/struje te se treba osigurati svoja. Napajanje koje ću koristiti tokom izrade praktičnog rada jest napajanje preko predviđenog Micro USB napajanja koristeći napajanje karakteristika 5V/2000 mA pošto ću malo više koristiti resurse Raspberry Pi uređaja.

2.5. Grafički procesor i displej

VideoCore je obitelj grafičkih procesora koja se koristi unutar Raspberry Pi uređaja. Ova obitelj grafičkih procesora ima naglasak na niskoj potrošnji energije. Razvio ju je Broadcom. Procesor se koristi kao dodatak na Cortex A53 procesor kako bi se omogućila bolja reprodukcija multimedijskih uređaja. Zbog arhitekture procesora Cortex A53 on ne bi mogao u visokoj kvaliteti ni brzini dekodirati i kodirati video pa se koristi VideoCore kao dodatak tome.

Bitna stvar kod grafičkog procesora jest da podržava široki spektar algoritama kodiranja videa. Ovaj grafički procesor podržava neke od glavnih u današnje vrijeme koji su najrašireniji i smatram da je to dovoljno za njegovu primjenu.

Kompresijski algoritmi koje podržava su:

- H.263
- H.264/MPEG-4
- AVC
- MPEG-2
- H.265
- VP9

Novi VideoCore IV koji se postavlja u Raspberry Pi 3 model B je donio poboljšanje tako da podržava 1080p video reprodukciju odnosno prikazivanje zaslona. Te je podigao operacijsku frekvenciju sa 250 na 300 MHz

3. Usporedba sa prijenosnim računalom (Lenovo G505)

Prijenosno računalo Lenovo G505 je računalo pušteno u prodaju krajem 2013 godine. Cijena računala pri kupnju je bila 2000 HRK krajem 2014 godine. Računalo je namijenjeno

standardnim korisnicima računala koji žele dobar omjer cijene i brzine. Ideja pri izradi računala je sudeći prema specifikacijama i cijeni u rangu prijenosnih računala bila ista kao i za računalo Raspberry Pi. Uzmimo u obzir da je sveukupna konfiguracija Raspberry Pi računala, sa periferijom i potrebnim adapterima za rad jednak prijenosnom računalu ukupnom cijenom iznosila 480 HRK sredinom 2016 godine. Što je četvrtina cijene prijenosnog računala.

3.1. Usporedba arhitekture

Iz razloga što je ovo prijenosno računalo, njegove komponente su birane ciljano da imaju naglasak na prenosivosti i na potrošnji energije. Lenovo G505 također sadrži SOC (sustav na čipu) kao svoj primarni računalni module.

Procesor koji se nalazi u računalu jest AMD A5-5000. Procesor se također sastoji od 4 jezgre procesor. Radni takt procesora je 1.5Ghz što je relativno sporo za današnje standarde koji se puštaju u prodaju. Ovaj radni takt, koristeći računalo dugi niz godina, je dovoljan za normalnu upotrebu, pregled multimedije, pretraživanje interneta i uređivanje dokumenata. One zadatke za koje je i Raspberry Pi namijenjen. Cortex A53 ima radni takt sporiji za 200 MHz. Sporiji takt možemo zanemariti s obzirom da je A5-5000 CISC procesor koji ima prošireni instrukcijski set, naspram ARM instrukcijskog seta koji je nastao iz RISC-a pa ima smanjeni broj instrukcija. To bi značilo da može obaviti isti posao sa manjim brojem instrukcija, odnosno u nekim slučajevima mu treba manje procesorskog vremena, tako da razliku od 200 MHz-a možemo zanemariti.

Cache memorija također utječe na brzinu rada na računalu. Računalo G505 ima L1 cache veličine 256 KB što je znatno više od 32KB L1 cache memorije Cortex A53 chipset-a. Sada dolazi L2 cache memorija nakon L1 te je ona jednaka za oba procesora i iznosi 512 KB ukupno. Uzeći to u obzir možemo dati prednost Cortex A53 procesoru zbog svoje RISC arhitekture.

3.2. Usporedba mogućnosti periferije i portova

Prednost	Raspberry Pi	Lenovo g505	Prednost
3xUSB 2.0	1xUSB 2.0	1xUSB 2.0	2xUSB 3.0
	Ethernet	Ethernet	

	3.5mm ulaz	3.5mm ulaz	
	HDMI port	HDMI port	VGA
	Napajanje	Napajanje	

Tablica 1. Usporedba mogućnosti periferije

Prema priloženoj tablici vidimo da prijenosno računalo sadrži dva USB 3.0 porta, no Raspberry Pi sadrži dodatna 3 USB porta. Trebamo uzeti u obzir da na Raspberry Pi računalu jedan USB port na periferiju tipkovnice i miš-a. Većina ljudi danas koristi bežičnu kombinaciju. Što zauzima jedan port. Dok za odvojenu žičnu periferiju bi se koristila 2 USB porta. Dok prijenosno računalo Lenovo G505 ne mora popunjavati portove u tu svrhu.

Ethernet port je jednak na obje konfiguracije. Te obje konfiguracije sadrže WIFI modul za internet povezanost. Audio povezanost se na Raspberry Pi računalo omogućava preko 3.5mm ulaza te HDMI izlaza koji može prenositi zvuk ukoliko periferni uređaj to podržava. Prijenosno računalo ima ugrađene zvučnike za reprodukciju audio sadržaja dok Raspberry Pi to ne podržava. Također prijenosno računalo sadrži i web kameru. Što je dodatna prednost.

HDMI port prenosi video i audio sadržaj do maksimalne rezolucije tzv. Full HD što se odnosi na rezoluciju 1920x1080 što obje konfiguracije podržavaju. Valja napomenuti da Raspberry Pi računalo podržava to no ima problem sa zagrijavanjem ukoliko se prenosi tom rezolucijom. No to se može regulirati pomoću dodatnog, te podosta jeftinog hladnjaka za Raspberry Pi računalo što rješava problem zagrijavanja. Prijenosno računalo sadrži dodatni VGA port što mu daje prednost povezivanja s periferijom no treba napomenuti da na konfiguraciju tog prijenosnog računala uvijek će se koristiti samo jedan port, VGA ili HDMI.

Za napajanje konfiguracije, prijenosno računalo koristi svoju izvedbu utora za svoj strujni adapter. S time je ograničen na to da se može koristi samo pripadajući Lenovo strujni adapter za napajanje konfiguracije. Raspberry Pi računalo za svoje napajanje koristi Micro USB utor koji je danas dosta raširen i standardiziran da se može puno lakše doći do alternativnog izvora struje nego za Lenovo računalo za koje se obavezno mora koristiti njihov strujni adapter.

3.3. Usporedba brzina

Za mjerenje benchmarking sustavom korist će se alat naziva sysbench iz razloga što se može koristiti na Windows i na Linux platformi. Ovo je vrlo jednostavan alat te nudi dobro objašnjenje svojih komandi i niz testova. Koristit će se procesorski test, test pristupa memoriji te test zapisivanja i čitanja datoteka na tvrdom disku.

Popis komanda korištene za benchmark

```
sysbench --test=cpu --num-threads=4 --cpu-max-prime=10000 run
sysbench --test=cpu --num-threads=1 --cpu-max-prime=10000 run

sysbench --test=memory --memory-block-size=1K --memory-total-size=1G
--memory-oper=write run
sysbench –test=memory –memory-block-size=1K –memory-total-size=1G –
memory-oper=read run

sysbench --test=fileio --file-total-size=512M prepare
sysbench --test=fileio --file-total-size=512M --file-test-mode=rndrw
--max-time=300 --max-requests=0 run
sysbench --test=fileio --file-total-size=512 cleanup
```

Test	**Raspberry pi**	**Prijenosno računalo**
Procesorski test više dretva	34.8855s	10.0015s
Procesorski test 1 dretva	139.6411s	10.0010s
Zapisivanje u memoriju	544.90 MB/sec	231.85 MB/sec
Čitanje iz memorije	719.10 MB/sec	240.44 MB/sec
UI protok	9.1197Mb/sec	0.44MB/sec

Tablica 2. Brzine računala po testnim stavkama

Testovi pokazuju da što se tiče procesorske brzine da prijenosno računalo je brže 3.4 puta za više-dretvene procese dok je čak 13 puta brži za jedno-dretvene sustave. Što je i očekivano s obzirom na arhitekturu i izvedbu. Raspberry Pi odnosi pobjedu na brzinama pristupa memoriji. 2.3 puta brža brzina pisanja u memoriju i 3 puta brža čitanja memorije. Razliku možemo obrazložiti jer Raspberry PI koristi LPDDR memoriju koja i treba biti 2 puta brža od radne memorije, LPDDR radna memorija se koristi u mobilnim uređajima te je namijenjen da ima veće brzine pristupa. Protok podataka kroz ulazno izlazne jedinice, odnosno, pisanje i čitanje datoteka na sustavu je velika razlika. Oko 20 puta je brži Raspberry Pi. Rezultat je malo čudnovat, te je provjeren više puta. Ovakav je jer Raspberry Pi koristi SD karticu za pohranu svojih podataka a ona koristi flash memoriju dok prijenosno računalo koristi magnetski

tvrdi disk tako znamo da će sigurno Raspberry Pi biti brži u pogledu permanentnog zapisa podataka.

Za teoretske brzine uzet ću benchmark performanse više sustava koje sam pridobio preko sustava zvan geekbench.com[3]. Na tom sustavu se uzimaju rezultati korisnika koji objavljuju svoje rezultate na svojim platformama. Tako možemo dobiti kvalitetne i prosječne rezultate za Raspberry Pi računalo te za prijenosno računalo Lenovo G505. U obzir ćemo uzeti FLOPS (floating operations per second) za više jezgri, iz razloga što su oba sustava napravljena da se koriste kao više-jezgreni sustavi te ću ih tako i razmatrati i ocjenjivati. Raspberry Pi ima prosječnu brzinu 1.2 GFLOPS-a[4] dok Lenovo ima prosječnu brzinu od 7.57 [5]GFLOPS-a. Iz rezultata možemo vidjeti da je prijenosno računalo 6,3 puta brže u operacijama za decimalne brojeve. To su teoretske razlike, no prijašnjim benchmark testovima možemo vidjeti da zbog svoje arhitekture i operacijskih sustava koji svako od računala pokreće ta razlika se drastično smanjuje na samo 2 puta bolje performanse što uzeći u obzir 4 puta manju cijenu je odlično. No više o tome u daljnjem poglavlju.

Jedna mjera koju možemo uzeti u obzir, iako ne igra baš veliku ulogu u performansama jest brzina paljenja sustava. Mjereno je 10 paljenja svakog sustava te uzet prosjek. Rađeno je više paljenja sustava iz razloga što na prijenosnom računalu Windows Update može raditi nepredvidivih problema i dati rezultat dugog paljenja. Dok je Raspberry Pi dosta konzistentan. Mjerenjem smo došli do vremena paljenja za Raspberry Pi od 37.07s što je solidno za tu platformu koju možemo smatrati mobilnom, dok je prijenosno računalo završilo s rezultatom od 138.31s odnosno 2:18 što je velik broj za prijenosna računala od kojih očekujemo da budu spremna za rad što prije.

3.4. Usporedba brzine izvođenja sortnih algoritama

Napisan je program (s jednom dretvom) za c++11 platformu te je kompiliran na računalima pomoću g++ prevoditelja. Kod je dostupan na javnom repozitoriju referenciranom kasnije u praktičnom radu[6]. Nakon toga provedeni su testovi na računalima te su utvrđene sljedeće brzine.

[3] Servis Geekbench http://geekbench.com
[4] Raspberry pi na servisu Geekbench http://browser.geekbench.com/geekbench2/2603281
[5] Lenovo G505 na servisu Geekbench http://browser.geekbench.com/v4/cpu/7838924
[6] Vanjska poveznica do autorovog repozitorija rada https://github.com/Eptick/zavrsni.git

Algoritam	Uzorak	Raspberry Pi	Prijenosno računalo
Quicksort	1 000 000	26,8401s	11,1377s
	10 000	0,0159s	0,0041s
	1000	0,0012s	0,0003s
Mergesort	1 000 000	1,2228s	0,5782s
	10 000	0,0174s	0,0043s
	1000	0,0014s	0,0003s
Heapsort	1 000 000	3,0439s	1,3662s
	10 000	0,0331s	0,0077s
	1000	0,0025s	0,0007s

Tablica 3. Brzine sortnih algoritama (Izvor: Sastavljeno od strane autora)

Vidljivo je da je prijenosno računalo znatno brže u izvođenju sortnih algoritama. U ovim brojkama ćemo uzeti nekakav prosjek da je prijenosno računalo do 3 puta brže u izvođenju sortnih algoritama. Što se povećava na manjem broju elemenata polja za sortiranje, a smanjuje na većim poljima za sortiranje.

3.5. Omjer cijene i brzine stolnog i Raspberry Pi računala

Cijena prijenosnog računala Lenovo G505 je 2000 HRK, dok je sveukupna cijena Raspberry Pi računala 480 HRK. 300 kuna je sam uređaj dok je 180 kuna odlazi na periferiju i potrebne elemente za rad. Za omjer cijene i performansi ću uzeti performanse benchmarking programa te performanse sortnih algoritama na 1 000 000 elemenata. Cilj je dobiti koliko je kuna potrebno za jednu sekundu mjernog elementa. Pošto su benchmark i performanse sortnih algoritma izražene u sekundama. Uzmimo za primjer benchmark test procesora sa više dretvi. Raspberry Pi računalu je trebalo 34.88 sekundi dok je prijenosnom računalu trebalo 10 sekundi. Za izračun uzmemo cijenu računala i podijelimo sa brojem sekundi koje je trebalo za izvedbu. Dakle 480 HRK / 34.88 s i dobijemo da je omjer kune i performanse jednak 13.76 HRK za jednu sekundu. Dok kada uzmemo cijenu za prijenosno računalo i podijelimo sa brojem sekundi dobijemo 2000 HRK / 10 sekundi to je 200 kuna za jednu sekundu izvedbe. Sada je to malo teže shvatiti, Vidimo da je jedinica dosta skuplja. Uzmimo za fiktivni primjer da je i Raspberry

Pi računalo završilo benchmark u 10 sekundi. 480/10 je 48 kuna za jednu sekundu izvedbe. Tako je vidljivo da bi platili četiri puta manje za iste performanse.

Kroz rad možemo uzeti u obzir da je nekakav prosjek razlike između Raspberry Pi računala i prijenosnog računala 3 puta brže računalo je prijenosno. Ako uzmemo tu mjeru obzir te pokušamo dobiti sa klasterom Raspberry Pi računala slične performanse te ih spojimo 3 u klaster došli bi do ukupne cijene od 1440 HRK što je za četvrtinu manje od cijene prijenosnog računala odnosno 28% manju cijenu dobijemo iste performanse ukoliko bi se klaster koristio kao npr. Web poslužitelj.

4. Praktični rad

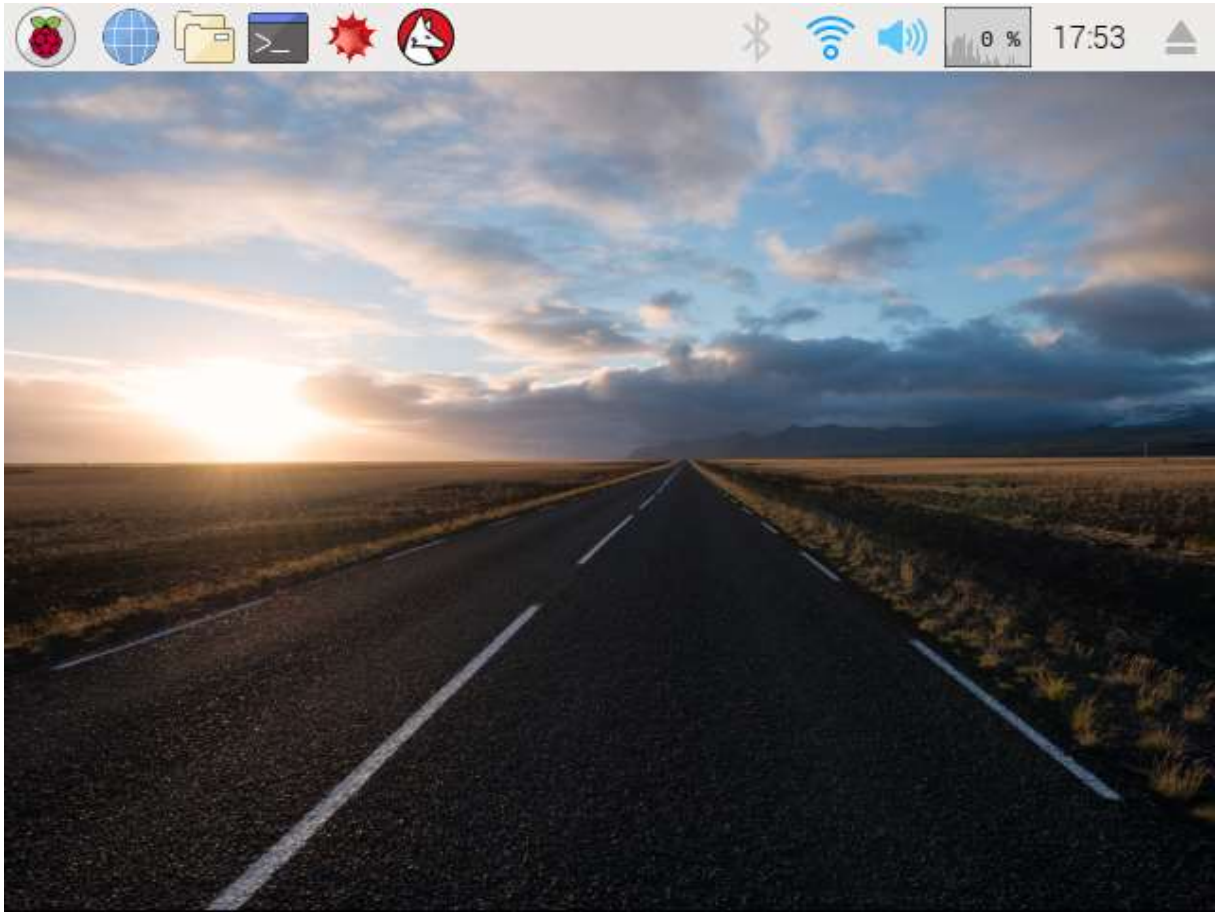
Pošto je ovo praktični završni rad u ovom dijelu će se obraditi najviše sadržaja. Odnosno napravit će se najviše sadržaja jer će se iskoristiti Raspberry Pi da se napravi funkcionalni sustav koji će pogoniti sva poglavlja u ovom glavnom četvrtom poglavlju. Operacijski sustav će pokretati Raspberry Pi te će on pokretati LAMP (Linux, Apache, MySql, PHP) web sustav koji će posluživati Wordpress stranicu na kojoj će se nalaziti web sučelja za program pokretanja GPIO pinova Raspberry Pi računala te web sučelje za Internet zvučnik. Sav kod korišten u završnom radu se nalazi na sljedećoj adresi https://github.com/Eptick/zavrsni

4.1. Operacijski sustavi

Iako se poglavlje zove Praktični rad u ovom dijelu će se opisati moguće instalacije operacijskih sustava, njihove prednosti i nedostatci, kada koji koristiti. Ti operacijski sustavi su, Raspbian, Windows IOT Core, Open Source Media Center, LIBREELEC koji služi za multimedijski centar te na kraju RISC OS što je operacijski sustav isključivo za Raspberry Pi no nije na Linux kernelu.

4.1.1. Raspbian

Raspbian je sustav koji je nastao od Debian sustava te je specifičan u tome da je održavan od strane zajednicice koja ima svrhu da što više Debian paketa prenese na Raspbian sustav kao takav. Raspbian ima identičan sustav instaliranja dodatnih paketa kao i Debian no ima par specifičnih konfiguracijskih datoteka poput datoteke raspi-config koji je vizualni konfigurator svih opcija Raspberry Pi sustava. Postoji datoteka config.txt koja služi u svrhu kvazi BIOS-a te nju čita grafički procesor prije nego su procesor i operacijski sustav inicijalizirani. Ovo je preporučeni operacijski sustav za Raspberry Pi pošto je pisan specifično za njega.



Slika 4. Inicijalni ekran Raspbian operacijskog sustava

4.1.2. Windows 10 IOT Core

Ovo je platforma koja je dostupna na Raspberry Pi uređaja kako bi Windows svoju platformu proširio i na ovu domenu. Odnosno ona služi kako bi se DOTNET aplikacije specifično UWP (Univeral Windows Platform) aplikacije mogle povezati sa Raspberry Pi uređajima. Ona se instalira kao i ostatak stvari te je ona opcija samo za razvijanje a ne za normalnu upotrebu pošto ne pruža nikakvo sučelje za upravljanje ulazno izlaznim jedinicama na klasičan način nego samo sučelje za pristupanje preko drugih aplikacija.

4.1.3. Open Source Media center

OSMC je operacijski sustav u svrhu reprodukcije medija koji je otvorenog koda odnosno održavan od strane zajednice te je dosta dobar ako će se Raspberry Pi koristiti sam u svrhu toga da se reproducira medijski sadržaj. OS je testiran i funkcionira solidno, nema pregrijavanja uređaja ukoliko budu veće rezolucije te je intuitivan za korištenje.

4.1.4. LIBREELEC

LIBREELEC je nastao kao kopija KODI sustava za preuzimanje i reprodukciju medijskog sadržaja. Kao ideja je dobar no kao izvedba nakon testiranja je razočaravajuća pošto je ograničen dodatcima koji su neophodni za rad. Ukoliko se koristi samo kao medijski centar, inferioran je opciji OSMC-a.

4.1.5. RISC OS

RISC OS je osnovni operacijski sustav koji sadrži samo osnovne stvari za rad sa računalom. Ovo je sustav koji nije na *NIX kernelu te kao takav sadrži vrlo minimalan skup aplikacija. Sustav je brz i omogućava da se lakše dolazi do svakog sloja upravljanja uređajem no sa svojim ograničenim aplikacijama ne preporuča se za upotrebu.

4.2. Podizanje operacijskog sustava (Raspbian) na SD karticu i USB

Podizanje Raspbian sustava je vrlo jednostavno, omogućeno je na dva načina. Preko NOOBS SD kartice koja se često dobije uz kupovinu Raspberry Pi računala. NOOOBS omogućava da se podigne puno različitih operacijskih sustava, u trenutku pisanja rada. Raspbian, Windows IOT Core, LIBRELEC i OSMC. Ovi operacijski sustavi su jednostavni za instalaciju. NOOBS sustav traži od korisnika da spoji računalo na Internet vezu preko WiFi-a ili Ethernet modula te nakon toga se odabere OS za instalaciju te se preuzme sa interneta te instalira na SD karticu.

Drugi način je možda čak i jednostavniji, SD kartica se stavi u drugi uređaj sa OS po želji te se na karticu prebaci sadržaj koji se nalazi na https://www.raspberrypi.org/downloads/ nakon toga se SD kartica ubaci u Raspberry Pi računalo te je ono spremno za daljnju upotrebu.

Kako bi se operacijski sustav pokrenuo sa USB uređaja. Potrebno je staviti *NIX sliku na Raspberry Pi uređaj te izvršiti sljedeće komande prema stranici Raspberry Pi uređaja

```
sudo apt-get update && sudo apt-get upgrade
```

```
echo program_usb_boot_mode=1 | sudo tee -a /boot/config.txt
```

Ove komande pristupaju OTP (One Time Programable) memoriji te postave zastavicu koja ukazuje da će se operacijski sustav pokušati pokrenuti sa USB pohrane. Nakon ovoga potrebno je istim procesom kao i za SD karticu staviti željenu sliku na USB memoriju. Dakle formatirati USB na FAT32 sustav te i prebaciti sliku preuzetu sa službenih kanala na USB memoriju, izbaciti SD karticu ako se nalazi u uređaju te ubaciti konfigurirani USB disk. Nakon čega je Raspberry Pi spreman za upotrebu.

4.3. LAMP Wordpress server

Raspberry Pi nije moćno računalo niti je namijenjeno da bude korišteno kao web server ali za manje potrebe i jednostavno konfiguraciju Raspberry Pi može poslužiti kao jeftina opcija koja će funkcionirati za skromne primjene. Primjena može biti za manje obrte da osposobe web stranicu za manju novčanu cifru. Može biti u osobne svrhe da se koristi kao uređaj za razvijanje web tehnologija.

Wordpress je sustav za upravljanje sadržajem koji omogućava jednostavnu početnu konfiguraciju i veliku korisničku podršku. Njegovo veliko tržište osigurava da ima dosta resursa dostupnih za osobnu personalizaciju web stranice i lagano objavljivanje novog sadržaja. Dok developerima omogućava da se na jednostavan i brz način razviju velike količine sadržaja i konfigurira svaki dio sustava.

Sustav koji će se konfigurirati na RaspberryPi sustavu će pokretati Apache web server koji koristi php 7.0. Baza podataka koju zahtjeva Wordpress je MySql. Pošto imamo unix sustav tipa Raspbian na Raspberry Pi instalacija će biti jednostavna. Potrebno je prvo instalirati Apache web server, nakon toga serveru instalirati dodatke za rad sa PHP 7.0. Nakon toga instalirati i konfigurirati MySQL za rad sa apache2 sustavom i PHP 7.0.

Nakon toga dolazi konfiguracija Wordpress sustava. Potrebno ga je preuzeti sa službenog repozitorija. Te ga raspakirati u root direktorij apache web direktorija. Za povezivanje s bazom podataka je potrebno kreirati korisnika posebnog za wordpress sustav. Nakon što kreiramo korisnika te mu damo privilegije za upravljanje sustavom, potrebno je kreirati bazu podataka u koju će Wordpress spremati svoje podatke.

Nakon što se konfigurirala baza podataka, možemo pokrenuti apache web server te otvoriti putanju http://localhost ukoliko se radi preko grafičkog sučelja ili upisati lokalnu IP adresu Raspberry Pi računala kako bi došao korak konfiguracije Wordpress-a. Nakon što se konfigurira sustav, stranica je namještena na lokalnu mrežu.

https://projects.raspberrypi.org/en/projects/lamp-web-server-with-wordpress

```
sudo apt-get update

sudo apt-get install apache2 php7.0 libapache2-mod-php7.0 mysql-server php7.0-mysql

sudo service apache2 restart

sudo wget http://wordpress.org/latest.tar.gz /  /var/www/

Sudo mysql -uroot -p
```

```
$ create user 'wordpress'@'localhost' identified by 'wordpress
$ GRANT ALL PRIVILEGES ON *.* TO 'wordpress'@'localhost'';
mysql -uwordpress -pwordpres
$ create database wordpress
```

Nakon izvršenih komandi potrebno je sve datoteke koje su preuzete sa Wordpress stranice postaviti u `/var/www/html` direktorij kako bi ih apache2 web server servirao pri pristupanju.

1. Otvori se web mjesto
   Ukoliko se pristupa lokalno sa Raspberry Pi: http://localhost
   Ukoliko se pristupa sa drugog računala potrebno je upisati lokalnu IP adresu Raspberry Pi računala koja se može pronaći pomoću komande `ifconfig`
2. Database Name 'wordpress'
   Username: wordpress
   password: worpdress
   host: localhost
   table_prefix: wp_
3. Site title: Završni rad
   Korisničko ime: wordpress
   Lozinka: wordpress
   email: redzicleon@gmail.com
4. Pokrenuti instalaciju nakon čega je Wordpress konfiguriran

Daljnje komande nisu potrebne za normalan rad Wordpress sustava no potrebne su ovaj rad pošto je daljnji praktični rad tako koncipiran da su 2 poglavalja integrirana kao Wordpress stranice i ovo im je potrebno da im se omogući normalan rad. Ovim komandama je omogućen Proxy koji govori da sve zahtjeve koji stižu na /ajax/ rutu servera, server prosljeđuje dalje na lokalni port 8080 na kojem će se nalaziti node.js server#2 koji služi za Internet zvučnik te rada sa GPIO sustavom. Provo će se omogućiti proxy i websocket prosljeđivanje. Nakon toga će se postaviti ruta za početni Virtualhost.

```
sudo a2enmod proxy rewrite proxy_http proxy_wstunnel
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

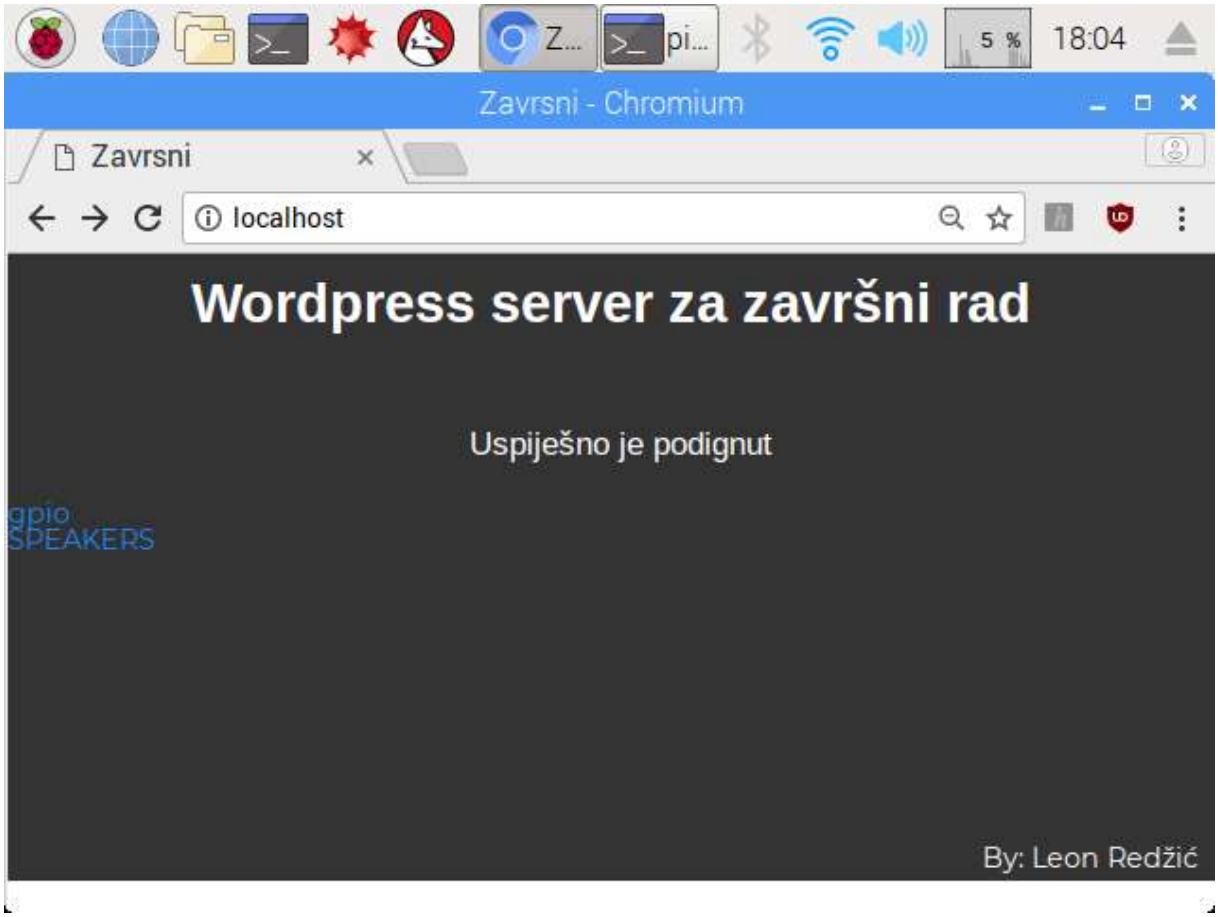
Dodati sljedeći sadržaj na kraj <VirtualHost> elementa

```
ProxyPreserveHost On
ProxyPass        /ajax/ http://127.0.0.1:8080/
ProxyPassReverse /ajax/ http://127.0.0.1:8080/
```

Nakon toga ponovo pokrenuti Apache servis

```
Sudo service apache2 restart
```

Nakon što se server ponovo pokrene može se pristupiti wordpress server po lokalnoj adresi Raspberry Po računala koju možemo dobaviti pomoću ifconfig komande. Te vidimo da je Wordpress server uspješno podignut.



Slika 5. Wordpress server sa posebno napisanom temom u svrhu završnog rada

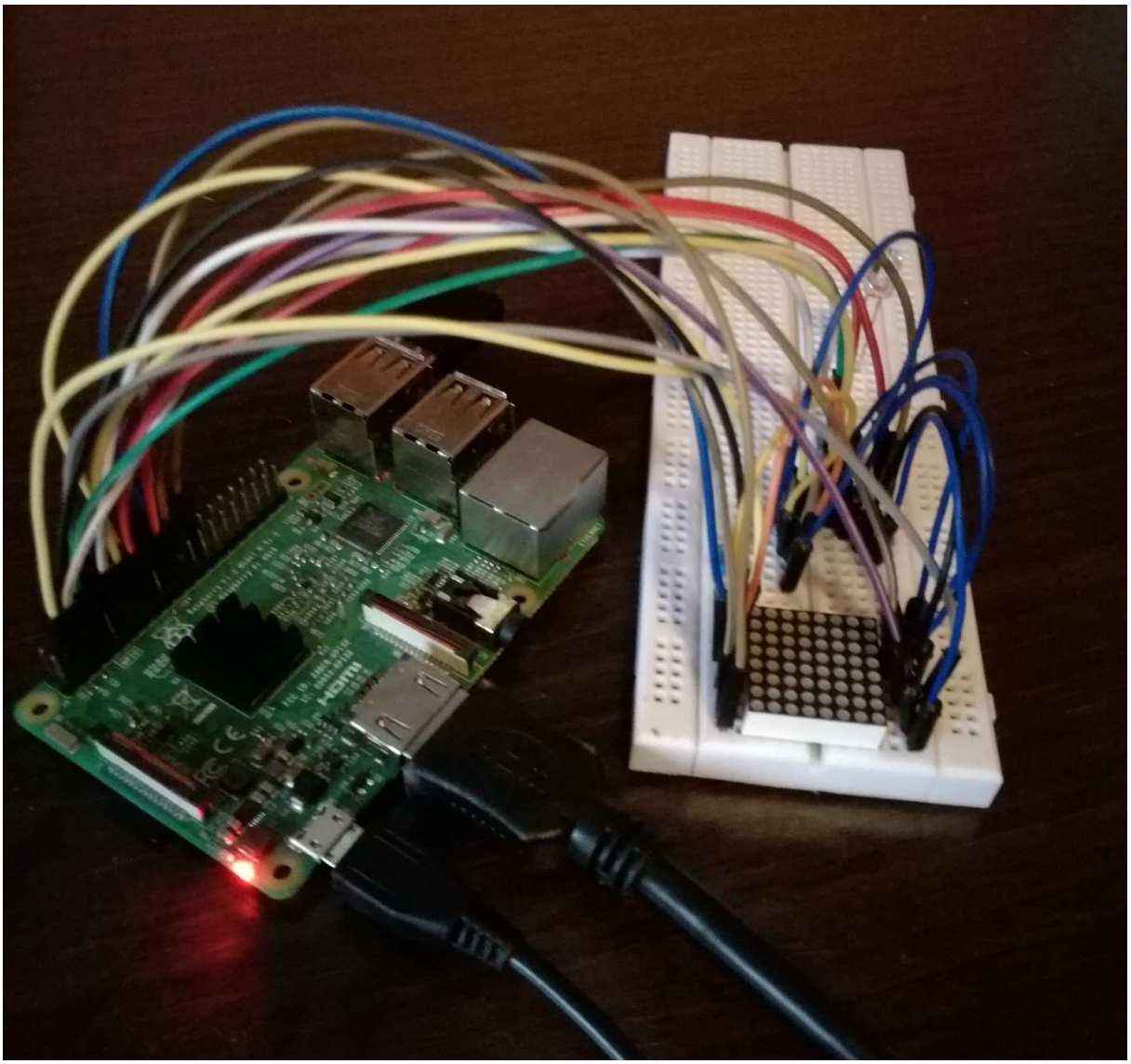
4.4. GPIO program

Raspored pinova koji su spojeni je sljedeći. Sljedeći prikaz pokazuje kako su pinovi na dot matrix zaslonu raspoređeni da bi se koristili za aktivaciju diode. Prema ovom rasporedu će se spojiti na GPIO pinove Raspberry Pi uređaja po predodređenom rasporedu. Prikazan niže.

```
  13 3 4 10 6 11 15 16
9
14
8
12
1
7
2
5
```

Columns (PiPin : DisplayPin)	Rows (PiPin : DisplayPin)
14 : 13	2 : 9
15 : 3	3 : 14
18 : 4	4 : 8
23 : 10	17 : 12
24 : 6	27 : 1
25 : 11	22 : 7
8 : 15	10 : 2
7 : 16	9 : 5

Za 8x8 dot matrix zaslon potrebno mi je bilo 8pinova za redove te 8 pinova za stupce. Proizvoljno sam odredio te pinove te ih naveo gore koji pin se spaja na koji pin dot matrix zaslona što sam našao na datasheet dokumentaciji zaslona.



Slika 6. Spojen Dot matrix displej na GPIO pinove Raspberry Pi računalo

Prema datasheet dokumentaciji, algoritam za upravljanje koja LED je upaljena je sljedeći. Pinovi koji su dodijeljeni redovima, da bi LED svijetlila trebaju imati napon na sebi, a stupci moraju biti uzemljeni, onda ako je pin za redak pod naponom, a pun za stupac uzemljen, struja će proći kroz LED te će ona zasvijetliti. Algoritam ide redoslijedom red po red, dakle stavi napon na jedan red, sve ostale redove isključi. Onda postavi pinove na stanje visokog otpora za LED koje trebaju biti upaljene. Tako svijetli samo 1 red LED koje trebaju biti osvjetljene, to stanje zadrži 1ms, nakon toga prelazi na idući red, te drži njegovo stanje 1 red. Ukupno vrijeme onda za prikaz cijelog zaslona je 8ms, no kada se to prikazuje konstantno i veoma brzo, zbog tromosti oka, nama izgleda kao da su upaljene sve odjednom. Izmjene sličica poslanih na Raspberry Pi računalo su izvede na sljedeći način, kada računalo primi podatke, ono primi ih u 2 dijela, prvi dio su podaci koji sadrže informacije o tome koje LED trebaju svijetliti a drugi dio je podatak koji sadrži vremenski razmak između sličica. Računalo kada primi te podatke,

pokreće dvije dretve, jedna dretva izmjenjuje sličice ovisno o razmaku između sličica, a druga dretva konstanto i neprekidno ispisuje trenutnu sličicu na dot matrix zaslona.

Pređimo sada na izvedbu koda servera za gpio. Pri pokretanju samog programa se vrši inicijalizacija pinova prema prijašnje određenom rasporedu koristeći on/off biblioteku. Svi se pinovi inicijaliziraju kao izlazni odnosno da će se na njih zapisivati određena razina napona.

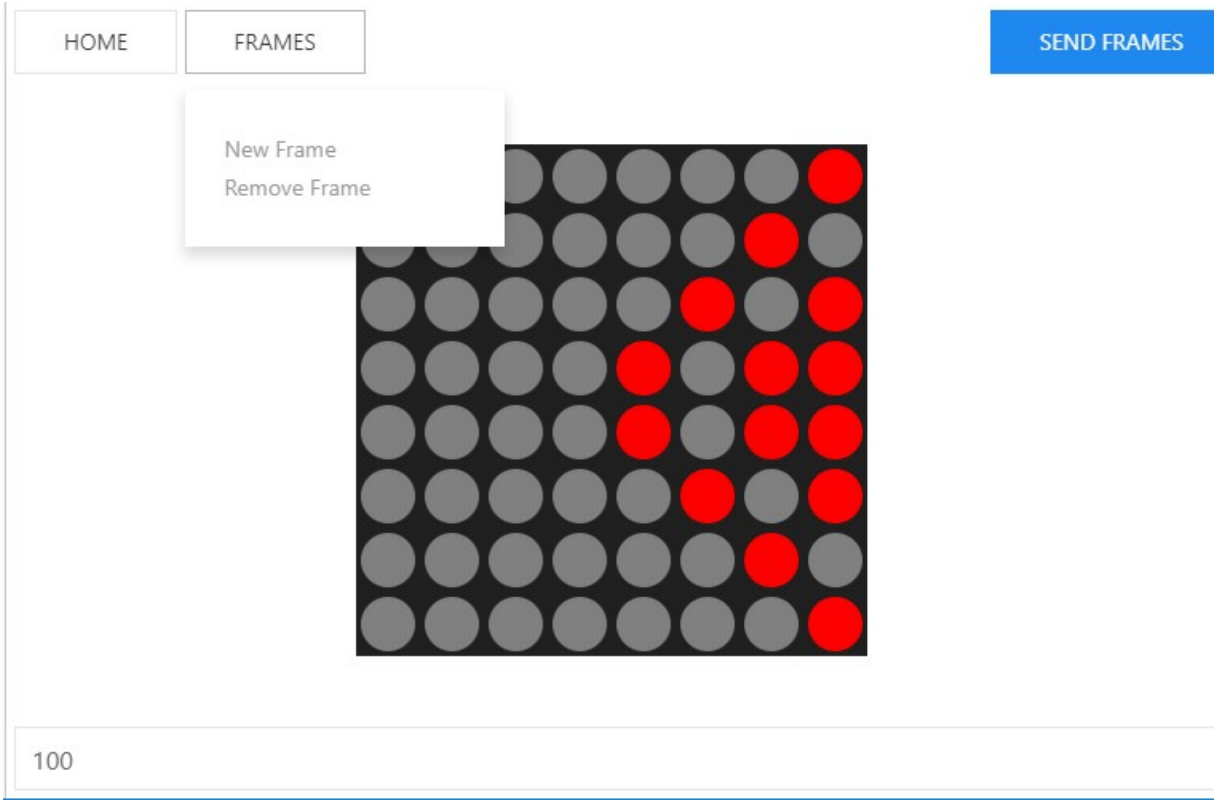
```
var Gpio = require('onoff').Gpio;
var stupacPinMap = [14, 15, 18, 23, 24, 25,  8, 7];
var redakPinMap  = [2,   3,  4, 17, 27, 22, 10, 9];
var piStupacPins = [];
var piRedakPins = [];
function initPins() {
    for(let pin of stupacPinMap) {
        piStupacPins.push(new Gpio(pin, 'out'));
    }
    for(let pin of redakPinMap) {
        piRedakPins.push(new Gpio(pin, 'out'));
    }
}
var frames = [];
var keys = [];
var key;
var playing = false;
var delay = 250;
function printFrame(i=0, old = 0){
    piRedakPins[old].writeSync(0);
    piRedakPins[i].writeSync(1);
    for(var j = 0; j < 8; j++) {
        // Ovdje ide 1 - state jer mora biti inverz (ground je 0 ) 1 je visoki
otpor odnosno brokiran
        piStupacPins[j].writeSync(1-frames[key][i][j]);
    }
    if(playing)
    setTimeout(printFrame, 4,(i+1)%8, i);
}
function selectKey(i = 0) {
    key = keys[i];
    // console.log("Key change: ", key, "Delay", delay);
    if(playing)
        setTimeout(selectKey, delay,(i+1)%(keys.length));
}
```

Ispod toga se nalazi kod koji je zaslužan za dretvu koji se sastoji od dvije metode printFrame koja služi za ispisivanje jednog okvira slike poslane na Raspberry Pi preko web sučelja. Dok druga metoda služi za biranje okvira koji se trenutno ispisuje te je ona napravljena da bi se omogućile animacije. printFrame ide algoritmom red po red te postavlja vrijednosti na pinove, zadrži ih 4ms te pređe na idući red. I tako u beskonačnost.

```
app.post('/gpio', (req, res) => {
    if(req.body.action === undefined) {
        res.status(500).send({ error: 'action not sent' });
        return;
    }
    switch(req.body.action) {
        case 'display':
            keys = [];
            delay = parseInt( req.body.delay );
            frames = JSON.parse( req.body.frames );
            for(let k in frames) {
                keys.push(k);
            }
            key = keys[0];
            if(!playing) {
        playing = true;
                setTimeout(selectKey, 1);
                setTimeout(printFrame, 2);
            } else {
        playing = false;
        setTimeout(function() {playing = true;setTimeout(selectKey,
1);setTimeout(printFrame, 2); }, 500);
        }
        break;
    }
    res.send();
})
```

Ovaj odsječak koda se vrti na serveru te služi da server obrađuje sve zahtjeve poslane sa web sučelja. Napravljen je tako da se sve šalju na istu rutu servera te se kao podatak šalje akcija koju sučelje zahtjeva. Trenutno se sastoji od jedne akcije a to je zaslon koja prima okvire odnosno sličice, razmak između svake sličice u milisekundama. Ona pokreće dretve ispisa ukoliko nisu pokrenute.

Web sučelje je napisano koristeći vue.js[7] kako bi bilo vizualno privlačno i intuitivno za korištenje. Sučelje se sastoji od dijela za dodavanje i uklanjanje sličica, polja za unos razmaka između sličica. Centralnog sučelja za uređivanje izgleda pojedinačne sličice te gumba za slanje sličica na server odnosno Raspberry Pi.



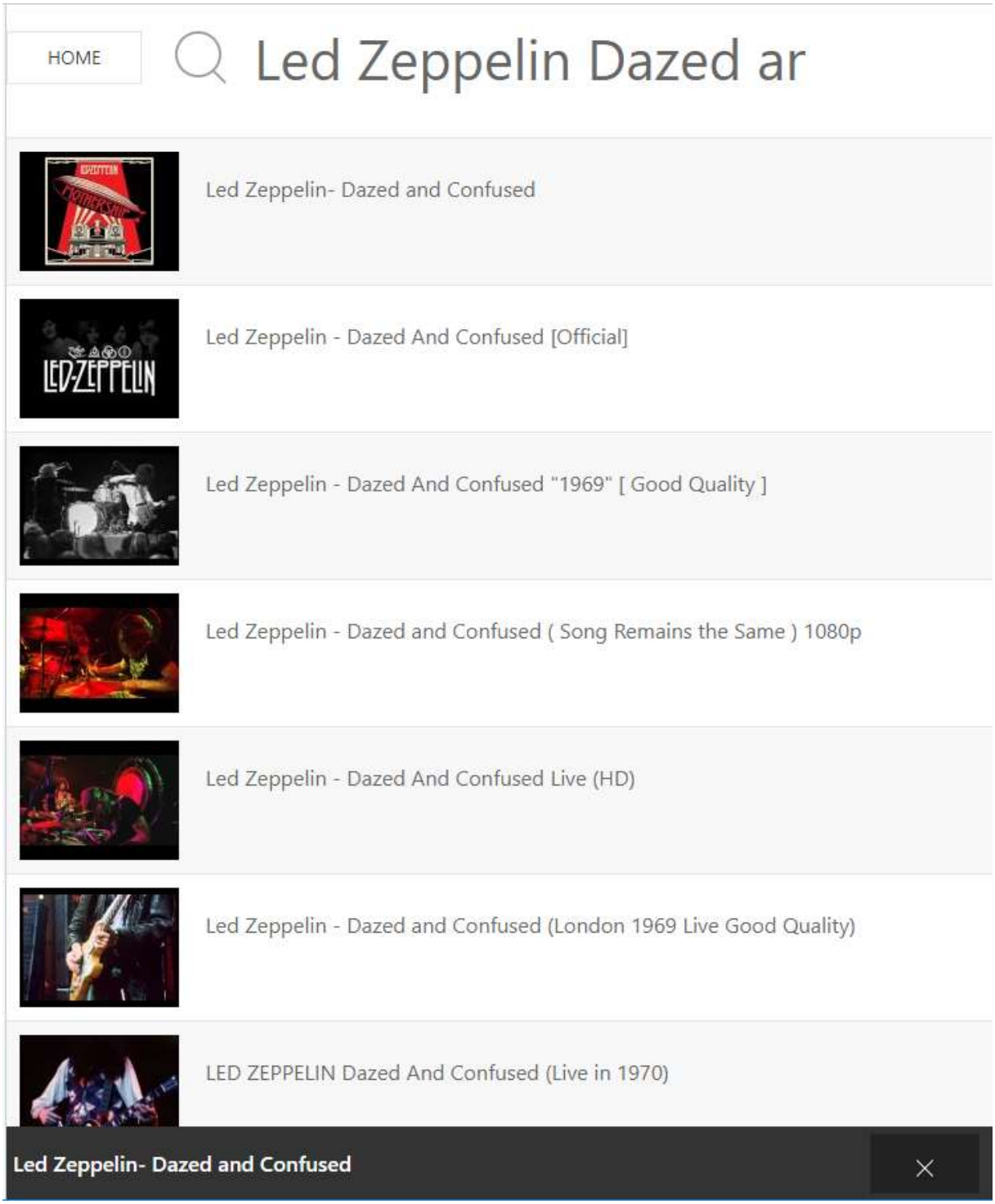
Slika 7. GPIO program web sučelje

4.5. Internet zvučnik

Internet zvučnik je mini projekt na koji je zamišljen na sljedeći način. Raspberry Pi nudi web sučelje preko kojeg se korisnici spajaju te na izlaze (HDMI, 3.5mm ili čak USB) reproduciraju medijski sadržaj sa Youtube servisa.

Server je podignut pomoću node.js Express biblioteke te ona radi nove procese koji reproduciraju medijski sadržaj na izlaze Raspberry Pi računala koristeći Speaker i yt-download-media biblioteke. Web sučelje je realizirano kroz Wordpress sustav prijašnje podignut te sučelje šalje komande na node.js server koji prima te HTTP POST zahtjeve i obrađuje ih te prosljeđuje naredbe daljnjim procesima. Server je ujedinjen na server prijašnjeg GPIO poglavlja zbog uštede resursa i jednostavnosti konfiguracije.

[7] Vue.js početna stranica https://vuejs.org



Slika 8. Speaker web sučelje

Sučelje će sadržavati polje za unos pojma za pretraživanje, listu pronađenog sadržaja te informacijsku traku na kojoj će svi korisnici moći dobiti pregled što se trenutno reproducira. Sada ćemo proći kroz proces sustava.

Kada se na sučelju unese pojam te se pošalje sadržaj odradi se sljedeći dio koda

```
$("#search").submit((event) => {
        event.preventDefault();
        $("#search-results").hide();
        $(".spinner").show();
        let term = $("#search-input").val();
        $.post( ajaxurldata.link + "/speaker/search", { term : term },
function( data ) {
            $("#search-results").html(" ");
            data.forEach(song => {
                let img = "<a href='#' class='play-song' data-
id='"+song.id+"'><div class='uk-inline-clip uk-transition-toggle'><img
class='song-image' src='"+song.thumb+"' alt='"+song.title+"'/><div class='uk-
transition-fade uk-position-cover uk-position-small uk-overlay uk-overlay-
default uk-flex uk-flex-center uk-flex-middle'><p class='uk-h4 uk-margin-
remove'>Play</p></div></div></a>";
                $("#search-results").append("<li>"+img+"<p class='song-
title'>"+song.title+"</li>")
            });
            initPlayButtons();
            $("#search-results").show();
            $(".spinner").hide();
        });
    });
```

Dohvati se traženi pojam te se pošalje zahtjev do servera koji ima registriranu tu putanju sa podacima pretrage. Nakon što prime se rezultati unutar funkcije koju odrađuje uspješni poziv rezultata funkcije, prvo obrišu prijašnji rezultati onda se za svaki pretraženi pojam kreira jedan element liste koji sadrži sliku elementa te njezin naziv i doda se na kraj liste pronađenih elemenata. Nakon toga se odradi funkcija koja inicijalizira sve gumbe za reprodukciju sadržaja te se prikaže sadržaj. Akcija na serveru koja prima sadržaj pojma za pretraživanje je sljedeća.

```
app.post('/speaker/search', (req, res) => {
    if(req.body.term === undefined) {
        res.status(500).send({ error: 'term not sent' });
        return;
    }
    search(req.body.term, YoutubeOpts, function(err, results) {
        if(err) return console.log(err);
        let response = [];
        results.forEach(elem => {
            response.push({
                id : elem.id,
                title: elem.title,
                thumb: elem.thumbnails.default.url
            })
        });
        res.send(response);
    });
})
```

Prvo se verificira da je poslan traženi pojam. Ako nije, odgovori s greškom. Ako postoji traženi pojam šalje se zahtjev do Youtube servisa funkcijom koju nam omogućuje biblioteka youtube-search. Te se u polje odgovora dodaje svaki element medijskog sadržaja uzimajući informacije id, naslov elementa te putanja do slike elementa. Nakon svega se pošalje odgovor web sučelju. Kada se klikne na određeni element šalje se play zahtjev serveru koji sadrži id elementa za reprodukciju. Na serveru se izvršava sljedeći kod

```
app.post('/speaker/play', (req, res) => {
    if(req.body.id === undefined) {
        res.status(500).send({ error: 'id not sent' });
        return;
    }
    try {
        player.play(req.body.id);
        getSongTitle(req.body.id);
        res.send();
    }catch(e){console.error(e)};
    console.log("ID: requested for reproduction: " + req.body.id)
})
```

Provjeri se dali je u zahtjevu poslan id, ako nije odgovori se pogreška. Ako postoji, istancira se novi proces koji je zadužen za reprodukciju medijskog sadržaja kako bi se rasteretio

proces web servera. Nakon toga se dobavi naslove elementa kako bi se obavijestili svi trenutni korisnici o promjeni. Kod za Player je sljedeći

```
const stream = require('youtube-audio-stream')

const decoder = require('lame').Decoder
const Speaker = require('speaker')
var id = process.argv[2]
if(id !== undefined)
    stream('http://youtube.com/watch?v='+id).pipe(decoder()).pipe(new
Speaker())
```

Koristeći biblioteke youtube-audio-stream za dobavljanje medijskog sadržaja te biblioteku speaker za reprodukciju na izlaze Raspberry Pi uređaja. Dodatne stvari koje su na playeru je sustav za live obavještavanje korisnika i promjenu elemente reprodukcije.

5. Zaključak

Tokom pisanja ovog rada prvi puta sam koristio Raspberry Pi računalo te sam ga nabavio novog, što bi značilo da sam ga morao od nule konfigurirati sam. Uzeći to u obzir mogu reći da mi nije stvaralo problema tokom cijelog procesa. Podizanje samog sustava bilo je iznimno bez muke te nakon toga sve dalje je išlo bez problema. Što se tiče pristupa informacija također je bilo jednostavno, možemo reći da je razvijanje na Raspberry Pi računalo pokretano zajednicom te se to osjeti tokom razvoja jer su informacije lagano dostupne.

Što se tiče brzine samog uređaja, to, to me iznenadilo koliko je zapravo responzivno bilo. Dok je pokretalo 2 web servera bez problema se grafičko sučelje prikazivalo i odgovaralo na moje zahtjeve. Samo grafičko sučelje je također, mada minimalno, odlično za korištenje u smislu da je sve prema današnjim konvencijama, pozicije elemenata i slično. Za cijenu uređaja koju sam naveo u radu ugodno sam iznenađen računalom. Podaci testiranja koje sam navodio tokom rada, da je prosjek 3x brže prijenosno računalo me iznenadilo jer se tokom korištenja samog Raspberry Pi računala ne primijeti razlika velika.

Sve u svemu, računalo je nadmašilo moja očekivanja te sam uživao razvijati praktični dio na njemu zbog lakog pristupa informacijama te laganom pristupu bibliotekama.

6. Literatura

[1] Dokumentacija Cortex A53 Procesora, ARM, Pristupano Listopad 2017
https://developer.arm.com/products/processors/cortex-a/cortex-a53

[2] Dokumentacija VideoCore IV, Broadcom Corporation, Pristupano Listopad 2017
https://docs.broadcom.com/docs/12358545

[3] Elinux stranica vođena zajednicom, Pristupano, Listopad 2017
http://elinux.org/RPi_Documentation

[4] Dokumentacija BCM2836 čipa, Gert van Loo, 18 August 2014, Pristupano Listopad 2017
https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2836/QA7_rev3.4.pdf

[5] LAN9514 dokumentacija, Microchip, Pristupano Listopad 2017
http://www.microchip.com/wwwproducts/en/LAN9514

[6] Raspberry Pi, stranica preuzimanja softvera, Pristupano Travanj 2018
https://www.raspberrypi.org/downloads/

[7] Raspberry PI compute module dokumentacija, Pristupano Listopad 2017
https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM_1p0.pdf

[8] Risc Open početna stranica, Pristupano, Travanj 2018
https://www.riscosopen.org/content/downloads/raspberry-pi

[9] Dokumentacija Dot Matrix displeja, Sure Electronics, Pristupano Veljača 2018
https://cdn.instructables.com/ORIG/F06/78D6/IOO54LGQ/F0678D6IOO54LGQ.pdf