

# Informacijski sustav umjeravanja etalona duljine

---

**Braun, Teo**

**Master's thesis / Diplomski rad**

**2016**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:235:149293>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-05-06**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering  
and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Teo Braun**

Zagreb, 2016.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

Mentor:

Prof. dr. sc. Dubravko Majetić, dipl. ing.

Student:

Teo Braun

Zagreb, 2016.

*Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.*

*Zahvaljujem se svima koji su na ikakav način pomogli pri izradi ovog diplomskog rada.*

*Najviše bi se zahvalio svom mentoru prof. dr. sc. Dubravku Majetiću na prilici da razvijem vještine i znanja u području koji volim, te na iznimnoj susretljivosti i strpljivosti.*

*Zahvalio bi se svima sa Katedre za mjerenje i kontrolu bez kojih ovaj rad ne bi ni postojao, posebice prof. dr. sc. Biserki Runje te Josipu Smoljiću.*

*Posebna zahvala odlazi kolegama i prijateljima Davidu Rupčiću i Luki Ceraju, koji su kroz godine bili vjerna podrška puni dosjetki i zabavnih trenutaka, gurajući studiranje prema svojem kraju.*

*Zahvaljujem se svojoj djevojci Sandri Mak što je brižno i empatično kimala glavom slušajući moje probleme upitne razumljivosti.*

*Na kraju zahvaljujem se svojoj obitelji bez koje ne bi postojao ni ja, ni ovaj rad, niti prilika da vidim koliko strpljivi i brižni mogu biti u količini potpore koju su pružali kroz sve ove godine.*

*Teo Braun*



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo  
materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

## DIPLOMSKI ZADATAK

Student: **Teo Braun**

Mat. br.: 0035177714

Naslov rada na hrvatskom jeziku: **Informacijski sustav umjeravanja etalona duljine**

Naslov rada na engleskom jeziku: **Gauge block calibration information system**

Opis zadatka:

Mahr Federal Millitron 1240 je elektronski komparator koji usporednom metodom mjeri razliku između referentnog i umjeravanog etalona iste nazivne duljine. Dobivenu razliku prikazuje samo na svojem ekranu. Stoga se u Laboratoriju za precizna mjerenja dužina FSB-a dobiveni rezultati s mjernog uređaja ručno upisuju u Radnu podlogu. Ona obuhvaća izračune statističkih veličina koje opisuju centralnu tendenciju i rasipanje rezultata mjerenja, izračune varijacije duljine te potrebne korekcije zbog utjecaja temperature i mjerne sile. Dobiveni rezultati se tada unose u računalu te se za umjereni etalon izrađuje mjerni certifikat. U navedenom postupku postoji puno mjesta gdje u sustav unosimo potencijalnu ljudsku pogrešku prilikom čitanja, pisanja, računanja ili prepisivanja podataka. Javlja se potreba za sveobuhvatnim softverskim rješenjem gdje se mjerni podaci očitavaju direktno sa mjernog uređaja i obrađuju na računalu ne bi li ljudsku pogrešku sveli na minimum.

U radu treba načiniti sljedeće:

1. Sastaviti sklopovlje potrebno za komunikaciju s mjernim uređajem koje će ujedno i obnašati funkciju servera koristeći računalu Raspberry Pi.
2. Ostvariti komunikaciju i prijenos mjernih podataka između mjernog uređaja i računala.
3. Razviti sučelje za unos, pohranu, analizu i pregled mjernih podataka.
4. Generirati certifikat o umjeravanju etalona duljine.
5. Tijekom izrade programske podrške koristiti web tehnologije.
6. Izvesti zaključke rada.

Zadatak zadan:

29. rujna 2016.

Rok predaje rada:

1. prosinca 2016.

Predviđeni datum obrane:

7., 8. i 9. prosinca 2016.

Zadatak zadao:

Prof. dr. sc. Dubravko Majetić

Predsjednik Povjerenstva:

Prof. dr. sc. Franjo Cajner

## SAŽETAK

Informacijski sustav umjeravanja etalona duljine gradi na završnom radu: Izrada web aplikacije za analizu rezultata usporedbenih mjerenja etalona duljine, Teo Braun - FSB, Zagreb, 2015.

Unutar završnog rada izrađuje se web aplikacija za automatizaciju papirologije u svrhu certificiranja etalona duljine, koristeći HTML, CSS, PHP, SQL, Javascript, Bootstrap i AJAX.

Ovaj diplomski rad nastavlja suradnju sa Nacionalnim Laboratorijom za duljinu, te gradi na već učinjenom i koncentrira se na izradu sklopovlja potrebnog za komunikaciju sa mjernim uređajem Mahr Millitron 1240. te sigurnosnim protokolima potrebnim za industrijsku primjenu.

Za potrebe rada koristit će se računalo veličine kreditne kartice, Raspberry Pi 3, te modul za serijsku komunikaciju.

Raspberry Pi pretvoriti će se u server korištenjem Apache2, PHP 5, MariaDB i phpMyAdmin softverskih paketa, dok mogućnost rada kao bežične pristupne točke, postići će se softverskim paketima DNSmasq i HostAPD.

Komunikaciju sa mjernim uređajem biti će postignuta korištenjem Python-a i PHP-a, te njihovim izmjenjivanjem varijabli korištenjem virtualne memorije.

U konačnici će se cijeli sustav osigurati implementacijom autentifikacije korisnika te sigurnosnim protokolima za zaštitu od SQL injektiranja, otimanje sesija, fizičkih i brute force napada.

Kao rezultat rada imati ćemo gotov proizvod, spreman za industrijsku primjenu, koji potpuno uklanja ljudsku pogrešku iz procesa certificiranja etalona duljine, te čini sam proces drastično bržim i efikasnijim.

### Ključne riječi:

Etalon duljine, Raspberry Pi, RS232, Mahr Millitron 1240, web programiranje, web server, bežična pristupna točka, Python, PHP, AJAX, sigurnost web stranica

## SUMMARY

The Gauge block calibration information system builds upon an undergraduate thesis titled: Web Application for Analysis of Length Standards Measurement Results Obtained by Differential Method, Teo Braun – FSB, 2015.

The undergraduate thesis covered the topic of automating paperwork for the purpose of making gauge block certificates, utilizing web technologies such as HTML, CSS, Javascript, PHP, SQL and AJAX.

Continuing the cooperation with the National laboratory for length, this work will build upon the previous topic and expand it with developing the hardware for establishing a communication link between the Mahr Milltron 1240 measurement device and our website. This work will also focus on implementing web security to provide an industry ready product.

The centerpiece of our hardware, we'll be a credit card-sized computer, the Raspberry Pi 3, along with a serial port module.

We will transform our Raspberry Pi computer into a web server using the following software: Apache 2, PHP 5, MariaDB and phpMyAdmin. We will also be adding access point capabilities with the use of DNSmasq and HostAPD software.

The communication link between the measurement device and our website will be established by utilizing Python and PHP with the help of virtual memory for interchanging variables.

We will round up this thesis by implementing user authentication with special regard for security measures in order to protect against SQL injecting, session hijacking, physical and brute force attacks.

The end result of this thesis will be an industry ready product, that will save time, money and increase reliability by completely eliminating the human error in the process of gauge block measurement and certification.

### Key words:

Gauge blocks, Raspberry Pi, RS232, Mahr Millitron 1240, web development, web server, access point, Python, PHP, AJAX, web security

## SADRŽAJ

Sažetak .....	I
Summary .....	II
Sadržaj .....	III
Popis slika .....	VI
1. Uvod .....	1
2. Sklopovlje .....	3
2.1. Mahr Federal Millitron 1240 [3] .....	5
2.2. Raspberry Pi 3 [4] .....	6
2.3. Serijsko sučelje - Max3232 RS232 to TTL [5] .....	7
2.4. Tipka za sigurno gašenje .....	9
2.5. Android tablet .....	9
3. Softver .....	10
3.1. Operativni sustav .....	10
3.1.1. Preuzimanje OS-a .....	10
3.1.2. Priprema SD kartice .....	10
3.1.3. Instalacija OS-a .....	12
3.2. SSH protokol .....	14
3.3. Python 3 .....	16
3.4. Serijsko sučelje .....	16
3.4.1. Zamjena portova .....	17
3.4.2. Isključivanje serijskog terminala .....	18
3.4.3. Serijska komunikacija sa Millitron-om .....	18
3.4.3.1. Konfiguriranje serijskog sučelja Mahr Millitron-a 1240 .....	18
3.4.3.2. Testiranje serijskog sučelja .....	19
3.5. Tipka za sigurno gašenje .....	21
3.6. Web server .....	22
3.6.1. Apache HTTP Server .....	22



---

3.6.2.	PHP .....	23
3.6.3.	SQL .....	25
3.6.4.	phpMyAdmin .....	26
3.7.	Bežična pristupna točka .....	27
3.7.1.	Statička IP adresa .....	27
3.7.2.	Konfiguracija hostapd-a .....	28
3.7.3.	Konfiguracija DNSMASQ-a .....	30
3.7.4.	Konfiguracija IP4 prosljeđivanja .....	30
3.7.5.	Dijeljenje Interneta .....	31
3.7.6.	Pristupno ime serveru .....	31
3.8.	Upotreba .....	32
4.	Sigurnost web stranica .....	34
4.1.	Korisničko ime i lozinka .....	34
4.2.	Zaštita .....	38
4.2.1.	SQL injektiranje .....	38
4.2.2.	Otimanje sesija .....	40
4.2.2.1.	SSL .....	41
4.2.3.	Fizički napad .....	42
4.2.4.	Hashiranje lozinke .....	43
4.2.5.	Brute Force napad .....	44
4.2.6.	DDOS napad .....	45
4.2.6.1.	Zaštita DNS-om .....	45
4.3.	Primjena .....	46
5.	Komunikacija .....	50
5.1.	Sloj 4 - Python .....	50
5.2.	Sloj 5 - /dev/shm/ .....	52
5.3.	Sloj 6 – AJAX, PHP .....	53

5.4. Sloj 7 – AJAX, Javascript .....	53
5.5. Sučelje .....	54
6. Zaključak .....	55
Literatura .....	56
Resursi .....	57

## POPIS SLIKA

Slika 2.1 Dijelovi potrebni za izvedbu sklopovlja .....	4
Slika 2.2 Induktivno ticalo (lijevo) i Millitron (desno) u laboratoriju za precizna mjerenja dužina .....	5
Slika 2.3 Pinout RS232 9-pin D-sub muškog priključka na poledini Millitron-a 1240.....	5
Slika 2.4 Raspberry Pi 3 .....	6
Slika 2.5 Usporedba RS232 (gore) i TTL komunikacije (dolje).....	7
Slika 2.6 MAX3232 modul (lijevo) i MAX3232 shema korištenja.....	8
Slika 2.7 Spajanje Raspberry Pi-a sa MAX3232 modulom .....	8
Slika 3.1 SD Card Formatter 4.0 .....	11
Slika 3.2 Win32 Disk Imager .....	12
Slika 3.3 sudo raspi-config.....	13
Slika 3.4 Putty .....	15
Slika 3.5 info.php odaziv.....	25
Slika 3.6 phpMyAdmin grafičko sučelje.....	26
Slika 3.7 FileZilla.....	33
Slika 4.1 SSL protokol .....	41
Slika 4.2 Početno sučelje.....	46
Slika 4.3 Sučelje za unos lozinke .....	47
Slika 4.4 Sučelje za registraciju novih korisnika .....	48
Slika 4.5 Unutar web stranice – pregled zapisa .....	49
Slika 5.1 Sučelje za očitavanje mjerenja.....	54

## 1. UVOD

Etaloni duljine su referentna tvar ili mjerni sustav namijenjen za određivanje, ostvarivanje, čuvanje ili obnavljanje mjernih jedinica. Najčešće se koriste za umjeravanje mjernih uređaja unutar poduzeća, tvornica i proizvodnih pogona kako bi se osigurala univerzalna kompatibilnost među poduzećima i svijetom.

Kako bi etaloni duljine bili referentni moraju se umjeravati od strane ovlaštenog laboratorija za mjerenja. Jedan od takvih laboratorija je i Nacionalni laboratorij za duljinu koji se nalazi na FSB-u. Ovaj rad se radi za njih uz njihovu suradnju.

Samo umjeravanje se provodi usporednom metodom referentnog i umjeravanog etalona.

Problematika umjeravanja etalona čini sam način na koji se ona provodi. Ručno, sa papirom i olovkom.

Mjeritelj je ipak ljudsko biće i sklono je povremenim pogreškama. Te pogreške uključuju krivo očitavanje rezultata sa mjernog ekrana, krivo prepisivanje mjera, krivo računanje izračuna, krivo referenciranje popratnih tablica, te u konačnici krivo prepisivanje u računalu za potrebe izrade certifikata. Kako se radi o preciznim mjerenjima gdje je svaki mikron važan, evidentno je da postoji stvarna potreba za uklanjanje tih pogrešaka.

Informacijski sustav umjeravanja etalona duljine, što je tema ovog rada, obuhvaća potpunu informatizaciju procesa prikupljanja i obrade podataka. Taj proces obuhvaća izradu potrebnog sklopovlja za komunikaciju sa mjernim uređajem, softver za prikupljanje i obradu podataka, spremanje podataka u mrežno dostupnu bazu te u konačnici izrade certifikata za izmjerenu garnituru etalona duljine.

Proces informatizacije je započet 2015. u Završnom radu „Izrada web aplikacije za analizu rezultata usporedbenih mjerenja etalona duljine, Teo Braun“ [1], čija je glavna tema bila izrada softvera za automatizaciju potrebne papirologije, kroz koji su se detaljno razradile teme etalona duljine, interneta, web tehnologija i izrade baze podataka. Taj rad je smanjio ljudsku pogrešku na minimum, ali je nije potpuno uklonio zbog potrebe ručnog unosa mjernih podataka u aplikaciju.

Ovaj diplomski rad nastoji još više smanjiti ljudsku pogrešku te gradi na spomenutom softveru, koncentrirajući se na izradu sklopovlja i popratnih softverskih nadogradnji za komunikaciju s mjernim uređajem, te obuhvaća izgradnju sigurnosnih protokola i autentifikaciju korisnika, ne bi li dobili potpuni proizvod spreman za industrijsku primjenu.

Kao rezultat ovog rada ostat će samo jedina ljudska pogreška, rukovanje samim etalonima duljine.

## 2. SKLOPOVLJE

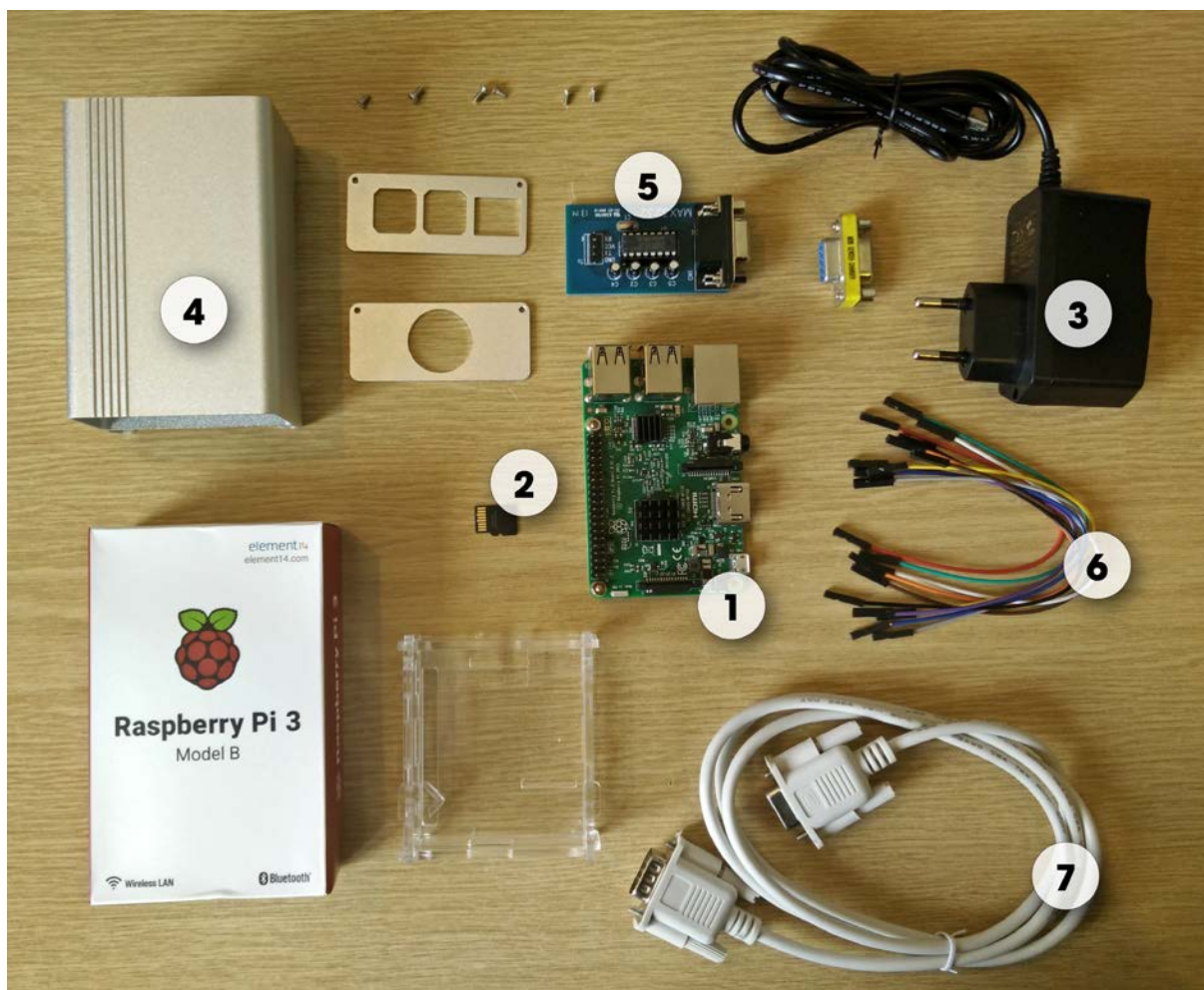
Sklopovlje je definirano zahtjevima i uvjetima okoline, tj. komunikacijskim mogućnostima mjernog uređaja, vrstom softvera koji ćemo koristiti, kontroliranim uvjetima prostorije u kojoj se vrše mjerenja i cijenom izvedbe.

Imajući to u vidu dolazimo do sljedećih zahtjeva:

- Mogućnost korištenja serijskog RS232 sučelja
- Mogućnost mrežne komunikacije
- Mogućnost bežične pristupne točke
- Mogućnost obnašanja funkcija web servera
- Mali toplinski trag
- Pristupačna cijena

Kako bi zadovoljili te uvijete izabran je Raspberry Pi 3 kao centralni sklop naše izvedbe. To je malo računalo veličine kreditne kartice koje će uz odgovarajući OS i softverske pakete zadovoljiti sve naše uvijete osim RS232 porta. RS232 port, tj. serijsko sučelje ćemo postići dodavanjem zasebnoga modula na TTL pinove Raspberry Pi-a.

Raspberry Pi će u konačnici raditi bezglavo (bez monitora, tipkovnice i miša) spojen samo sa napajanjem i mjernim uređajem. Softverom za akviziciju mjernih podataka pristupati će se Android tabletom koji će biti spojen sa Raspberry Pi-om preko WiFi-a.



**Slika 2.1** Dijelovi potrebni za izvedbu sklopovlja

Popis dijelova:

- 1) Raspberry Pi 3 – malo računalo
- 2) Micro SD kartica klase 10 8GB – služi kako disk Raspberry Pi-u, na njega ćemo staviti OS i sav popratni softver
- 3) Napajanje 5V 2.5A sa mikro USB utorom – za Raspberry Pi
- 4) Kućište za Raspberry Pi
- 5) RS232 Modul – za ostvarivanje serijskog sučelja sa mjernim uređajem
- 6) Ženski-ženski Pin konektor kablovi – za spajanje RS232 modula sa Raspberry Pi-om
- 7) RS232 D-sub 9-pin produžni kabel – za spajanje RS232 modula sa mjernim uređajem

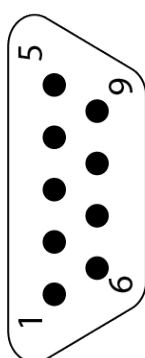
## 2.1. Mahr Federal Millitron 1240 [3]

Mahr Federal Millitron 1240 je elektronski komparator koji služi za prikazivanje razlike između referentnog i mjerenog etalona. Samo mjerenje se vrši na postolju s induktivnim ticalom dok Millitron uspoređuje rezultate i prikazuje ih na svojem ekranu s obzirom na vrstu mjerenja i mjernog područja.



Slika 2.2 Induktivno ticalo (lijevo) i Millitron (desno) u laboratoriju za precizna mjerenja dužina

Millitron 1240 je na poledini uređaja opremljen sa serijskom vezom koju ćemo koristiti za spajanje na Raspberry Pi.



Pin	Oznaka	Funkcija
1	NC	nekorišteno
2	RxD	Ulaz podataka
3	TxD	Izlaz podataka
4	DTR	Rukovanje izlaz
5	Gnd	Uzemljenje, 0 V
6	DSR	Rukovanje ulaz
7	NC	nekorišteno
8	NC	nekorišteno
9	NC	nekorišteno

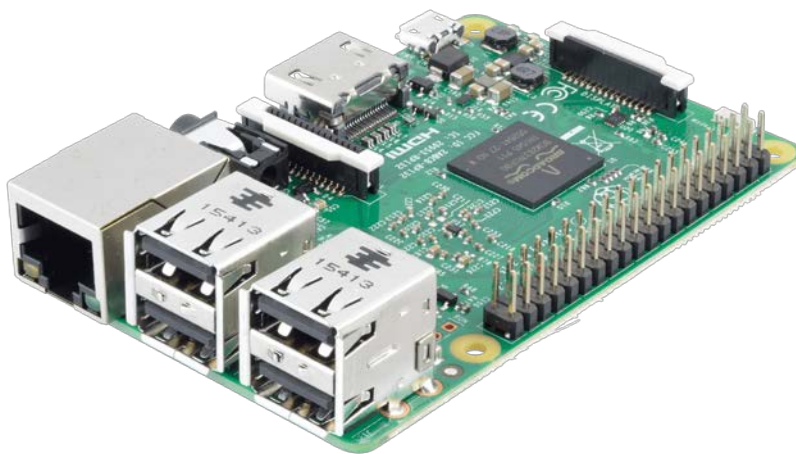
Slika 2.3 Pinout RS232 9-pin D-sub muškog priključka na poledini Millitron-a 1240



## 2.2. Raspberry Pi 3 [4]

Raspberry Pi 3 je malo računalo veličine kreditne kartice. Od samog začeca 2012. zamišljeno je kao jeftini i dostupni uređaj. Imajući to u vidu nije uopće čudno što kao disk koristi micro SD karticu, kao i većina digitalnih kamera, te za napajanje koristi USB punjač od mobitela jačine 2.5A.

Kako bi se Raspberry Pi pretvorio u potpuno računalo potrebno je samo HDMI ekran, tipkovnica i miš, koji će nama u okviru ovog rada trebati samo za inicijalno podešavanje softvera.



Slika 2.4 Raspberry Pi 3

Specifikacije računala:

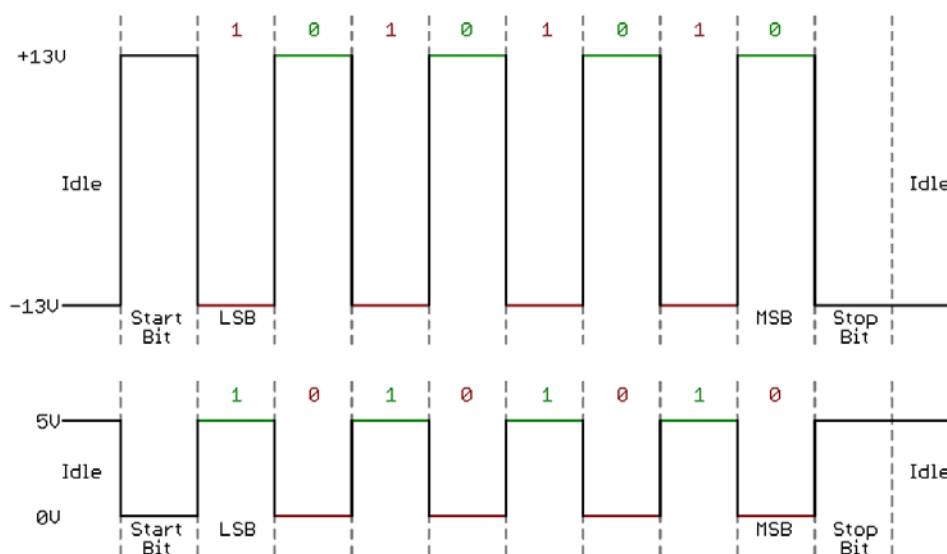
- CPU: 4 jezgri ARM Cortex-A53, 1.2GHz
- RAM: 1GB (900 Mhz)
- GPU: Broadcom VideoCore IV
- Mreža: 10/100 Ethernet, 2.4GHz 802.11n WiFi
- Bluetooth: Bluetooth 4.1
- Memorija: micro SD
- GPIO: 40-pin (digitalni ulazi i izlazi)
- Portovi: HDMI, 3.5 mm audio-video priključak, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

### 2.3. Serijsko sučelje - Max3232 RS232 to TTL [5]

Kako Raspberry Pi nema fizički ugrađen RS232 D-sub 9-pin port. Potreban nam je modul za serijsku komunikaciju.

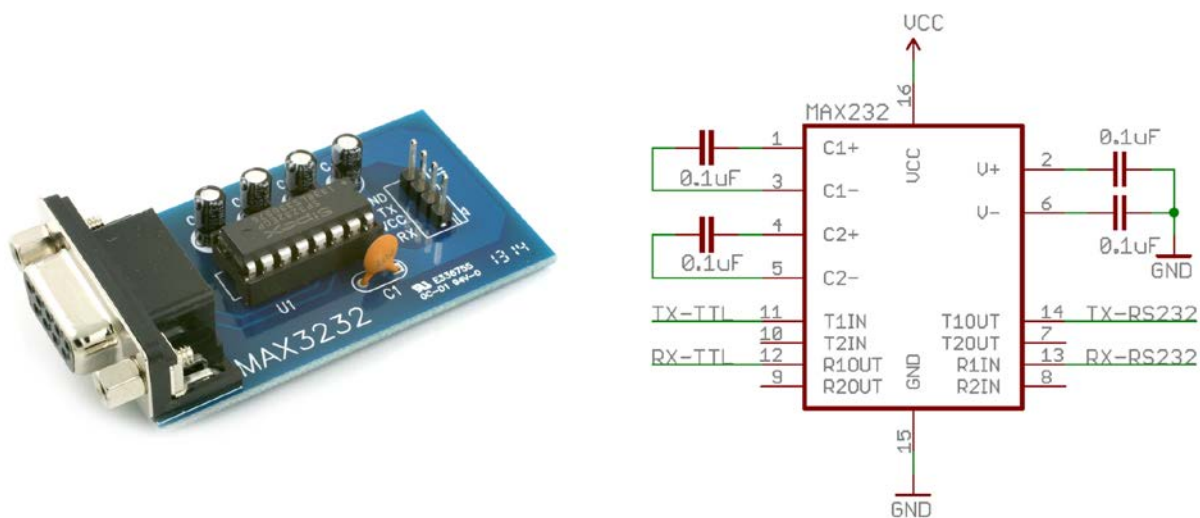
Raspberry Pi na svojim GPIO (General-purpose input/output) pinovima nudi i UART protokol (universally asynchronous receiver/transmitter) preko svojih RxD i TxD pinova (Receive and Transmit Data). UART ima isti komunikacijski protokol kao i RS232, ali koriste drugačije napone. UART na Raspberry Pi-u koristi TTL (transistor-transistor logic), što znači da koristi napon između uzemljenja 0V i napajanja od 5V. U tom protokolu 0V predstavlja logičku nulu dok 5V logičku jedinicu.

RS232 (Recommended Standard 232) za logičku jedinicu koristi -13V (aktivirano između -25V i -3V), dok je logička nula +13V (između 3V i 25V).



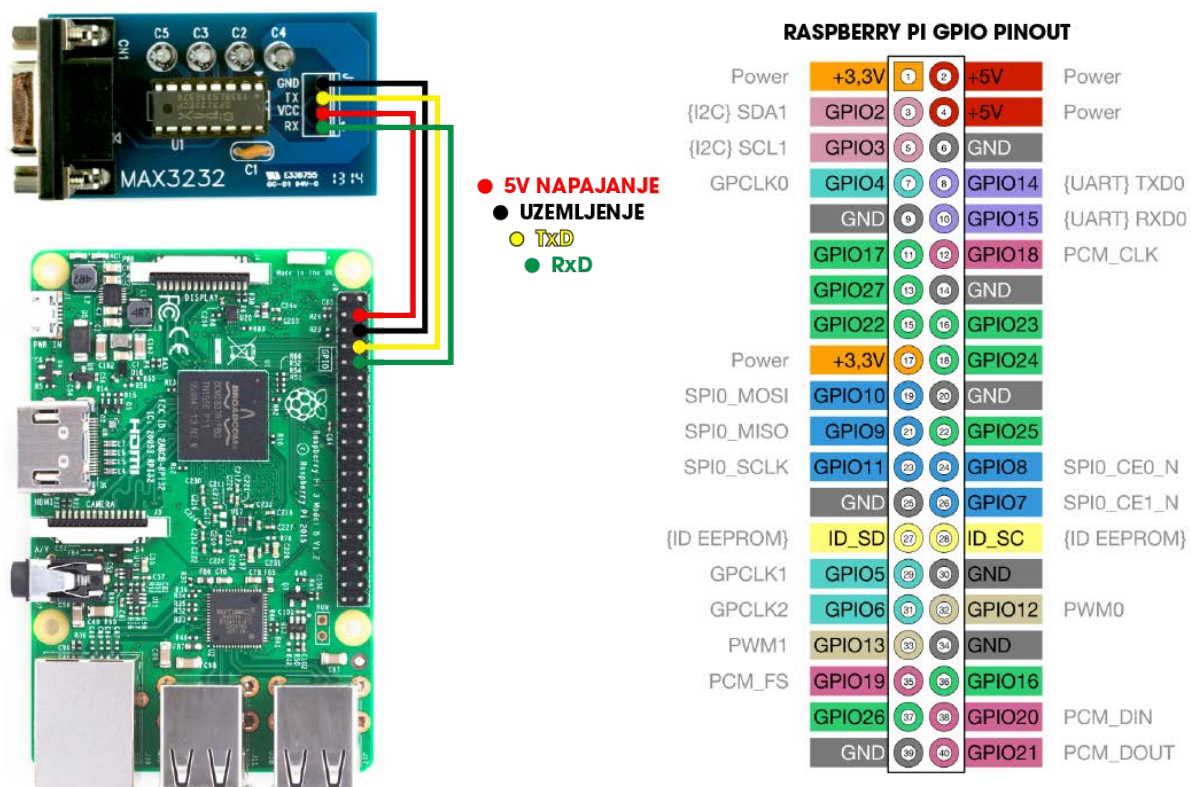
**Slika 2.5 Usporedba RS232 (gore) i TTL komunikacije (dolje)**

Taj problem rješavamo sa MAX232 mikrokontrolerom koji je osmišljen za upravo taj problem.



Slika 2.6 MAX3232 modul (lijevo) i MAX3232 shema korištenja

Imajući to sve u vidu, MAX232 modul spajamo na način prikazan na slici 2.7.



Slika 2.7 Spajanje Raspberry Pi-a sa MAX3232 modulom

Kako bi upotpunili komunikaciju sa Millitron-om potreban nam je RS232 D-sub 9-pin muško ženski produžni kabel.

Postoje dvije vrste kabla za serijska sučelja. Jedni su produžni, što samo produžuju izlaz do drugog mjesta (pin-to-pin), te postoje Null-Modem kablovi koji invertiraju podatkovne Txd i Rxd pinove, te pinove za rukovanje DTR i DSR (Data Terminal Ready, Data Set Ready).

Naš MAX3232 modul već ima invertirane pinove stoga je potrebno koristiti RS3232 produžni kabel.

## **2.4. Tipka za sigurno gašenje**

Raspberry Pi, da zadrži nisku cijenu, nažalost nema tipku za isključivanje. Računalo se uključi kada spojimo napajanje te se isključi kada prekinemo dovod napajanja.

Posljedica toga je mogućnost oštećenja podataka na micro SD kartici prilikom gašenja, pa čak i oštećenja cijelog operativnog sustava.

S obzirom da to nikako nije poželjan efekt potrebno je ugraditi sigurnosno gašenje kroz sustav koji će se pokrenuti na pritisak tipke.

Kako bi to implementirali, potrebno je samo jedno tipkalo spojiti na jedan od slobodnih GPIO pinova, npr. GPIO21 i GND (uzemljenje).

Kasnije ćemo implementirati skriptu koja će hvatati ostvareni kontakt ukoliko je tipka pritisnuta i pokrenuti postupak sigurnog gašenja.

## **2.5. Android tablet**

S obzirom da se mjerenje provodi u kontroliranim uvjetima od 20°C, tlak zraka 101325 Pa i tlaka vodene pare u zraku 1333 Pa, poželjno je koristiti uređaj koji previše ne poremećuje te uvijete. Iz tog razloga preporučuje se Android tablet zbog svoje jednostavnosti, malog grijanja i pristupačne cijene. Unatoč našem izboru, zapravo se može koristiti bilo koji uređaj sa Internet preglednikom i WiFi mogućnostima za spajanje na Raspberry Pi.

Kako smo za izgled web stranice, kojoj će se pristupati, koristili prilagodljiv kostur Bootstrap [1], možemo biti sigurni da veličina ekrana uređaja neće utjecati na preglednost i funkcionalnost stranice.

### 3. SOFTVER

#### 3.1. Operativni sustav

Raspberry Pi bez operativnog sustava je obična hladna kutija koja troši energiju. Kako bi smo mogli pokrenuti serverske softvere, potrebno je prvo stvoriti okruženje u kojima ti softveri mogu izvršavati svoje dužnosti, tj. operativni sustav.

Operativni sustav je skup osnovnih sustavnih programa koji upravljaju sklopovljem računala radi ostvarivanja osnovnih funkcija računala: ulaz, memoriranje, obrada i izlaz podataka.

Za izvedbu ovog rada koristit ćemo Raspbian Jessie Lite [I]. To je operativan sustav otvorenog koda baziran na Linux-u posebno podešen za rad sa Raspberry Pi-om. Sadrži samo najosnovnije programske pakete kako bi koristio što manje resursa našeg računala ostavljajući više za naš softver. Kako Raspbian Jessie Lite nema ni korisničko sučelje, njegov ukupni trag na RAM-ima iznosi 33Mb, ostavljajući gotovo jedan gigabyte za naše programe.

Iako ćemo u konačnici Raspberry Pi koristiti bezglavo, bez monitora i računalne periferije, ono će nam inicijalno biti potrebno za postavljanje samo operativnog sustava.

##### 3.1.1. Preuzimanje OS-a

Za pripremu operativni sustava Raspbian Jessie Lite, biti će nam potrebno zasebno računalo sa Internet konekcijom kako bi pristupili službenoj stranici udruge Raspberry Pi [II] i preuzeli OS. Datoteka koju ćemo dobiti imati će nakon ekstrakcije iz arhive, ekstenziju **\*.img**.

**img** datoteke su savršene bit-po-bit kopije SD kartice ili USB memorije. Pomoću odgovarajućeg softvera mogu se koristiti kao rezervne kopije SD kartice koje se spremaju na računalo ili za lako i jednostavno vraćanje već prethodno spremljenog stanja.

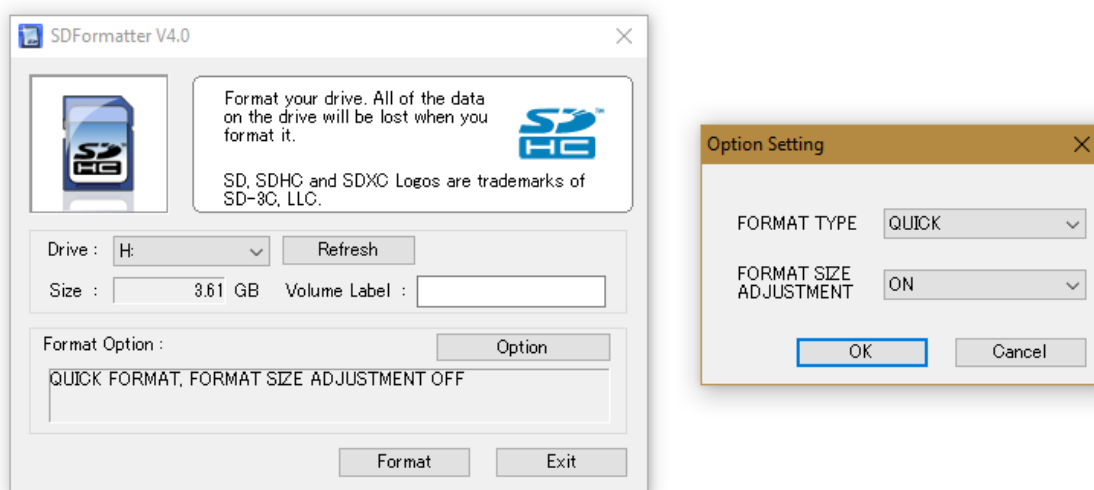
##### 3.1.2. Priprema SD kartice

U svrhu rada sa Raspberry Pi-om izabrana je micro SD kartica veličine 8 GB, brzine klase 10 UHS1 (40 Mb/s). S obzirom da će na njoj biti operativni sustav, što veća brzina SD kartice je poželjna.

Prije nego prenesemo OS na SD karticu, moramo je prvo formatirati, tj. definirati datotečni sustav kartice.

Datotečne sustave koriste mediji za pohranu podataka na način da jedan dio memorije čini indeks svih podataka, dok je drugi, puno veći dio, spremište podataka. Na taj način kada kreirate neku datoteku, stvori se njen indeks koji pokazuje na memorijsku lokaciju gdje se zapravo nalaze bitovi koji čine vašu datoteku. Taj efekt se najbolje može vidjeti prilikom premještanja datoteke unutar istog diska, jer djeluje da se premještanje događa gotovo instantno bez razlike na veličinu datoteke. Jedina stvar koja se tada promijenila je indeks datoteke. Kod brisanja datoteke, samo se zapravo izbriše njen indeks, dok u spremištu ostaju i dalje bitovi datoteke ali su sada proglašeni slobodni za korištenje.

Za formatiranje SD kartice koristi ćemo SD Card Formatter 4.0 [II] od organizacije koja je osmislila i održava SD standard.



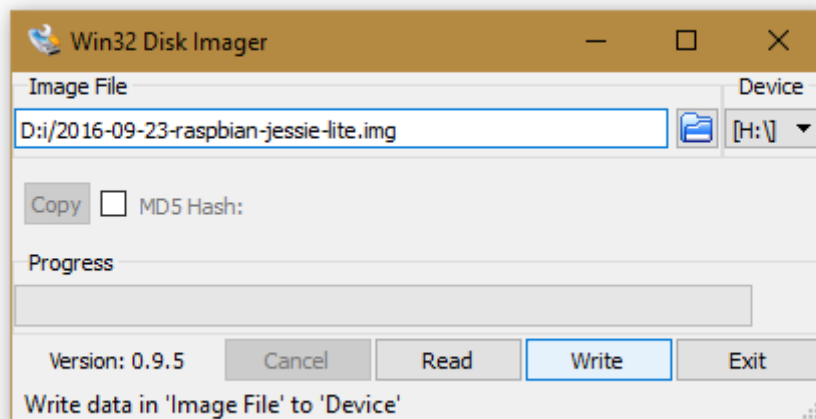
Slika 3.1 SD Card Formatter 4.0

U samom softveru trebamo pripaziti da smo izabrali pravilno slovo koje je dodijeljeno našoj kartici, te u opcijama uključiti „Format Size Adjustment“ jer će Raspbian prilikom prvog paljenja se prilagođavati veličini naše SD kartice.

Nakon uspješnog formatiranja potrebno je otpakirati **img** datoteku Raspbiana na SD karticu.

To ćemo učiniti softverom otvorenog koda Win32 Disk Imager [III]. Tim softverom možemo raditi rezervne kopije SD kartica ili USB memorija koje se spremaju u **img** datoteke ili restaurirati snimljeno stanje iz **img** datoteke na SD kartice.

Koristeći Win32 Disk Imager napraviti ćemo posljednje.



Slika 3.2 Win32 Disk Imager

Prije samog formatiranja važno je izabrati dobro slovo dodijeljeno našoj SD kartici te izabrati odgovarajuću **img** datoteku.

### 3.1.3. Instalacija OS-a

Nakon uspješnog pisanja Raspbiana na SD karticu, potrebno je:

- Umetnuti SD karticu u Raspberry Pi
- Spojiti monitor ili TV preko HDMI sučelja
- Spojiti tipkovnicu preko USB sučelja
- Spojiti Raspberry Pi na lokalnu mrežu s Internetom koristeći Ethernet sučelje
- Spojiti punjač od mobitela sa micro USB portom (5V 2.5A)

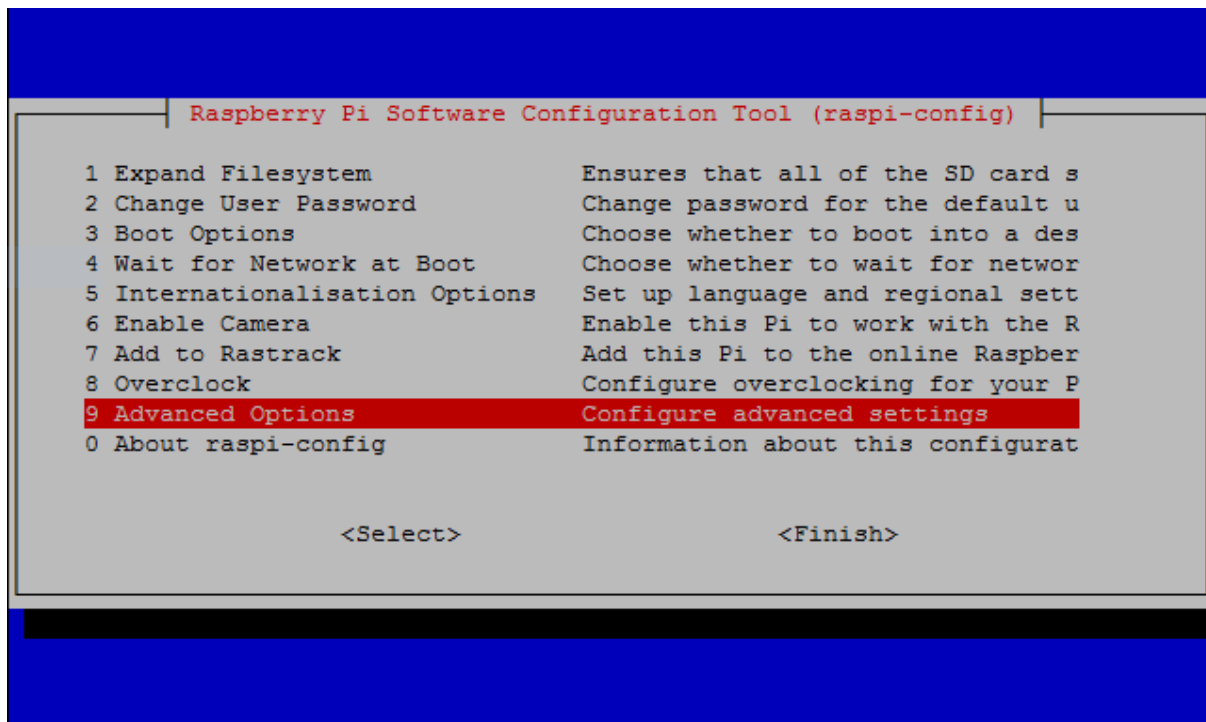
Prilikom paljenja Raspberry Pi-a, na ekranu će nas dočekati terminal koji će od nas tražiti korisničko ime i lozinku.

Predodređeno korisničko ime je **pi**, a lozinka **raspberrypi**.

Nakon što dobijemo pristup sustavu, osnovne postavke namjestit ćemo upisivajući naredbu:

```
sudo raspi-config
```

Riječ **sudo** će se pojavljivati često u naredbama terminala jer on označava administratorska prava prilikom izvođenja naredbe koje nam trebaju za uspješno podešavanja i instalaciju softverskih paketa.



Slika 3.3 sudo raspi-config

**sudo raspi-config** otvara sučelje za osnovne postavke sustava. U ovom sučelju promijeniti ćemo sljedeće:

- Internationalisation Options
  - Change Keyboard Layout – izabrati Hrvatski raspored tipkovnice
  - Change Timezone – izabrati vremensku zonu
- Change user password – promijeniti predefiniranu lozinku
- Advanced Options
  - Hostname – promijeniti ime Raspberry Pi-a koje će biti vidljivo na lokalnoj mreži. U ovom radu to je **gblockpi** (gauge block pi)
  - SSH – potrebno je uključiti SSH protokol za daljinsko upravljanje



Kako bi nove postavke počele djelovati, Raspberry Pi je potrebno ponovno pokrenuti naredbom:

```
sudo reboot
```

Nakon pokretanja, potrebno je ažurirati operativni sustav na najnoviju verziju kako ne bi došlo do nekompatibilnosti u verzijama softvera koje ćemo koristiti.

Kako bi ažurirali sustav potrebno je pokrenuti sljedeće naredbe:

```
sudo apt-get update  
sudo apt-get dist-upgrade  
sudo apt-get upgrade
```

**apt-get** je dio Advanced Packaging Tool-a (APT), koji je zadužen za instalaciju, nadogradnju i uklanjanje softverskih paketa. Te naredbe ćemo često koristiti prilikom instalacije novih softverskih paketa, a koriste se na sljedeći način:

```
apt-get [-opcija] [naredba]
```

Naredba **apt-get update** spaja se na repozitorij softvera i preuzima popis s najnovijim inačicama dostupnog softvera preko alta **apt-get**.

Naredba **apt-get dist-upgrade**, temeljem usporedbe trenutno instaliranih inačica softvera s bazom najnoviji, prikuplja instalacije te vrši ažuriranje temeljem posebnih pravila.

**dist-upgrade** pazi na međusobne ovisnosti softvera za rad, pametno odlučuje koji softverski paketi se trebaju ukloniti, a koji ažurirat dajući prednost važnijim softverima na štetu manje bitnih.

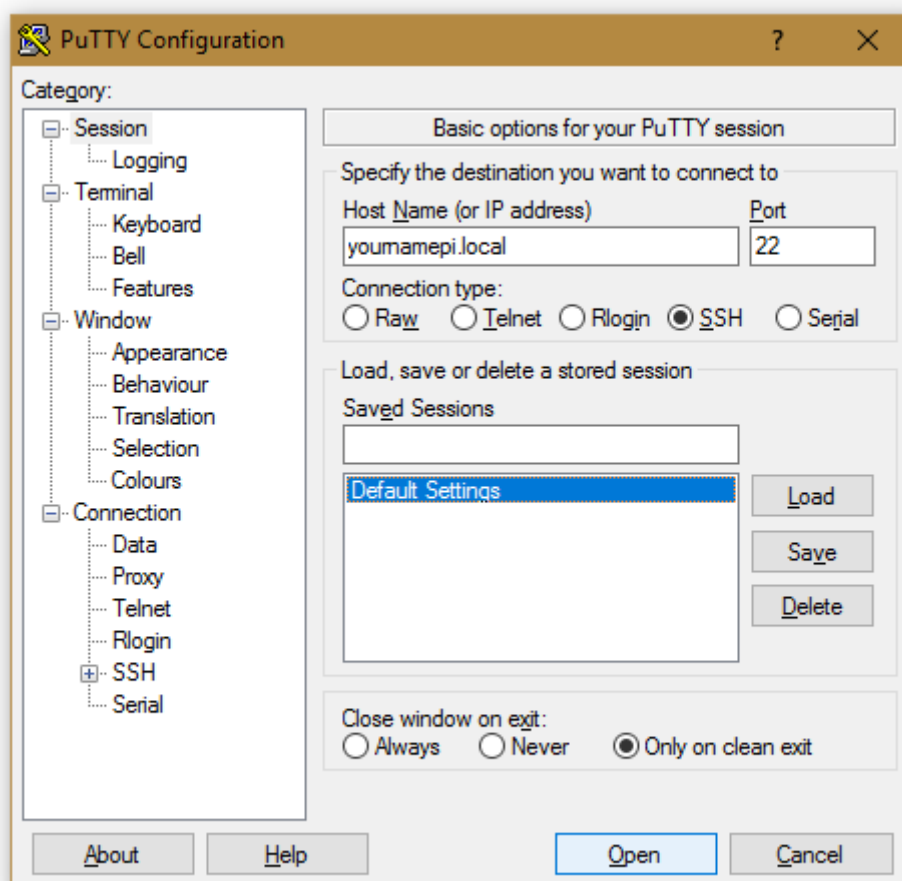
Naredba **apt-get upgrade**, temeljem usporedbe trenutno instaliranih inačica softvera s bazom najnoviji, prikuplja instalacije te vrši ažuriranje.

### 3.2. SSH protokol

SSH (Secure Shell) je mrežni protokol koji korisnicima omogućuje uspostavu sigurnog komunikacijskog kanala između dva računala putem nesigurne računalne mreže. Na taj način moguće je pristupiti terminalu računala kojim želimo upravljati i upisivati naredbe kao da se fizički nalazimo na tom računalu.

S obzirom da smo u prethodnom koraku uključili SSH protokol na Raspberry Pi-u, možemo već početi koristiti Raspberry Pi u bezglavom stanju.

Kako bi smo pristupili Raspberry Pi-u koristit ćemo softver otvorenog koda Putty [IV].



Slika 3.4 Putty

Za ispravno korištenje Putty-a potrebno je selektirati SSH protokol, koristiti port 22, te znati IP adresu računala na kojeg se spajamo ili njegov hostname ako se nalazi na lokalnoj mreži.

Za pristupiti Raspberry Pi-u, potrebno je dodati ekstenziju **\*.local** na hostname koji smo koristili u prethodnom poglavlju. U našem radu to je **gblockpi.local**. Ekstenzija **local** nije uobičajena praksa već je specifično za rad Raspbian.

Nakon uspješnog spajanja, otvara se terminal u koji se, kao i pri paljenju Raspberry Pi-a, treba ulogirati nakon čega možemo slobodno pisati naredbe kao i prilikom korištenja tipkovnice i monitora.

U ostatku poglavlja 3. SOFTVER, podrazumijeva se upisivanje svih naredbi kroz Putty terminal koristeći SSH protokol ili se direktno sve upisuje na Raspbeebry Pi-u.

### 3.3. Python 3

Python [6] je programski jezik opće namjene, interpretiran i visoke razine. Python dopušta programerima korištenje nekoliko stilova programiranja. Objektno orijentirano, strukturno i aspektno orijentirano programiranje. Uz tu fleksibilnost Python automatski alocira memoriju za varijable te ih nije potrebno inicijalizirati kao u drugim jezicima.

Sam izvršni kod Python nešto je sporiji od drugih programskih jezika kao što su C++, no unatoč tome vrlo je popularan diljem svijeta zbog navedenih prednosti i čistoće koda kojim se programira.

U ovom radu Python koristimo za komunikaciju sa Millitron-om preko serijskog sučelja te za detekciju pritisnutosti tipke za sigurno gašenje sustava.

Za instalaciju koristi ćemo **apt-get** alat kojim ćemo instalirati Python 3, biblioteku za serijsko sučelje **Pyserial** te **rpi.gpio** biblioteku za korištenje GPIO pinova.

```
sudo apt-get install python3
sudo apt-get install python3-serial
sudo apt-get install python3-rpi.gpio
```

Koristeći naredbu **apt-get install imepaketa**, alat **apt-get** dohvaća najnoviju instalaciju **imepaketa** preko interneta sa repozitorija te vrši instalaciju.

### 3.4. Serijsko sučelje

Kako bi smo koristili serijsko sučelje potrebno je ispravno konfigurirati port koji će se koristiti. Na Windows sustavima port koji se uobičajeno koristi za serijsko sučelje nosi ime **COM1** do **COM9**. Na Raspberry Pi-u to ime je **/dev/ttyAMA0** i **/dev/ttyS0**.

### 3.4.1. Zamjena portova

Raspberry Pi je do svoje druge inačice za serijsku komunikaciju koristio **/dev/ttyAMA0**, no kako je Raspberry Pi 3 dobio Bluetooth mogućnosti taj port je dodijeljen Bluetooth-u, pa je port **/dev/ttyS0** pripao GPIO serijskom sučelju.

Port **/dev/ttyAMA0** u potpunosti radi poduprijet sklopovljem što osigurava pouzdan rad, dok port **/dev/ttyS0** je poduprijet djelomično softverom što zna uzrokovati trenutke nepouzdanosti.

Kako bi bili sigurni da naše sklopovlje pouzdano očitava mjerenja, potrebno je zamijeniti funkcije navedenih portova.

To ćemo učiniti upisivanjem sljedeće naredbe:

```
sudo nano /boot/config.txt
```

Naredbom **nano** pokrećemo tekst editor kojim ćemo editirati datoteku **config.txt** u kojoj se nalaze inicijalne postavke Raspberry Pi-a.

U zadnji red datoteke potrebno je dodati sljedeći tekst:

```
enable_uart=1
```

Promjene se spremaju pritiskom **Ctrl+X** i potvrdom spremanja utipkavanjem znaka **Y**.

Kako bi se postavke postale aktivne potrebno je ponovno pokrenuti Raspberry Pi naredbom:

```
sudo reboot
```

Nakon ponovnog pokretanja možemo provjeriti status portova upisivanje sljedeće naredbe koja će nam izlistati sve portove na sustavu.

```
ls -l /dev
```

Ukoliko smo uspjeli zamijeniti portove, trebali bi u listi vidjeti sljedeće:

```
serial0 -> ttyAMA0  
serial1 -> ttyS0
```

**serial0** i **serial1** su aliasi koje možemo koristiti prilikom programiranja, gdje **serial0** uvijek ukazuje na port za serijsko sučelje.

### 3.4.2. Isključivanje serijskog terminala

Raspberry Pi je prepodešen za korištenje serijskog sučelja za upravljanje terminalom. U slučaju da ne koristimo lokalnu mrežu za upravljanje Raspberry Pi-om koristeći SSH protokol, bili bi u mogućnosti spojiti računalo na Raspberry Pi preko serijske veze i na taj način postići isto.

Kako bi mogli koristiti serijsku vezu za mjerenja, potrebno je isključiti mogućnost serijskog terminala upisivanjem sljedeće naredbe:

```
sudo raspi-config
```

Naredba otvara sučelje za postavke prikazano na slici 3.3.

Potrebno je navigirati do sljedeće opcije kako bi isključili korištenje serijskog terminala.

```
>Advanced options >Serial > No
```

Postavke će postati aktivne nakon ponovnog pokretanja Raspberry Pi-a.

```
sudo reboot
```

### 3.4.3. Serijska komunikacija sa Millitron-om

#### 3.4.3.1. Konfiguriranje serijskog sučelja Mahr Millitron-a 1240

Za testiranje komunikacije sa Millitron-om, potrebno je najprije namjestiti parametre komunikacije.

DIN66348-2 standard za serijska mjerenja propisuje korištenje **7e1** protokola.

**7e1** označava korištenje:

- **7**, označava 7 bita za prijenos podataka (dovoljan broj za ASCII znakovlje)
- **e** (even), označava parni paritet. Uz svaki poslani znak slati će se još jedan kontrolni bit koji ovisno o poslanom znaku čini sumu poslanih bitova parnim. Ukoliko dostavljene vrijednosti bitova nisu parne, zna se da je došlo do greške prilikom prijenosa.
- **1**, označava stop bit, tj. kraj prijenosa znaka

Na Mahr Millitronu 1240 režim komunikacije **7e1** podešava se pritiskom „Konfig“ tipke na poleđini uređaja, te navigiranjem do sljedećih postavki:

- Konfig
  - Input/Output Parameters
    - Baud – 9600 (Brzina znakova u sekundi. Odabrana je najbrža ponuđena opcija)
    - Rec – X ON (Koristi softversku inačicu kontroliranja prometa podataka, tj. davanje dopuštenja kada tko smije pričati. Ovime garantiramo veću kompatibilnost s drugim uređajima jer smanjujemo broj pravilno korištenih pinova na D-sub 9-pin konektoru na samo dva. Txd i Rxd)
    - RS – 232 C (Izabran D-sub 9-pin port na poledini uređaja)
    - Char – 7e1 (režim komunikacije po DIN66348-2)

#### 3.4.3.2. Testiranje serijskog sučelja

Korištenjem alata **nano**, stvoriti ćemo python skriptu kojom ćemo testirati komunikaciju s Mahr Millitron-om 1240.

```
cd
sudo nano rs232.py
```

Naredba **cd** služi za navigaciju kroz mape sustava. Obično se koristi u obliku **cd primjer/mapa/podmapa** ili **cd ..** što nas vraća jedan nivo mape iznad. Kada se koristi bez lokacije, na Raspberry Pi-u vraća nas u **/home/pi**, gdje ćemo pomoću **nano** alata stvoriti python skriptu **rs232.py**.

```
import time
import serial
port = "/dev/ttyAMA0"
baudrate = 9600
ser = serial.Serial(port,
baudrate,bytesize=serial.SEVENBITS,parity=serial.PARITY_EVEN,
stopbits=serial.STOPBITS_ONE, timeout=1, xonxoff=True)
if ser.isOpen():
    print(ser.name + ' je uspostavljen...')
while True:
    cmd = input("Naredba ili 'izlaz':")
    if cmd == "izlaz":
```

```
ser.close()
print (ser.name + ' je isključen')
break
else:
    ser.write((cmd + '\r\n').encode())
    out = ''
    time.sleep(0.08)
    while ser.inWaiting() > 0:
        out += (ser.read(1)).decode()
    if out != '':
        print ("=> " + out)
```

Najkritičniji dio skripte je pogoditi parametre komunikacije koje moraju odgovarati parametrima konfiguriranim na Millitron-u. Koristeći biblioteku **serial** [7] dolazimo vrlo lako do inicijalizacije veze koristeći naredbu **serial.Serial()**.

Posebnost svih serijskih veza je što naredbe završavaju sa znakom <CR> (Carriage return), što postizemo u pythonu dodavanjem znakova „\r\n“ na kraj našeg upita Millitron-u.

Naredbu, tj. upit prije slanja potrebno je kodirati naredbom **.encode()** kao što i odgovor moramo dekodirati naredbom **.decode()**.

Nakon što skriptu spremimo, možemo je pokrenuti sa naredbom:

```
sudo python3 rs232.py
```

Program je napravljen da čeka ulaz korisnika, koji nakon slanja upita mjernom uređaju čeka 80 milisekundi odgovor.

Ukoliko se utipka riječ „izlaz“, program prestaje sa radom.

Mahr Millimar 1240 ima već predodređene upite na koje odgovara:

- **I** – upit za inicijalizaciju. Prije nego li Millitron dobi upit „I“, gluh je na svaki drugi znak. Nakon uspješnog slanja upita vraća odgovor: „**I,MARH GMBH, M1240, 1.73**“ te poslije prihvata i ostale upite.
- **M** – upit za rezultat mjerenja. Nakon uspješnog slanja upita vraća rezultat mjerenja: „**E7, M1, -332.49**“ ovisno o mjernom području u kojem se nalazi prilikom korištenja.
- **X** – upit kojim se gasi komunikacija tj. čini Millitron gluhim na sve upite osim znaka „I“. Nakon uspješnog slanja upita vraća odgovor: „**X**“.

### 3.5. Tipka za sigurno gašenje

Kako je objašnjena problematika nedostatka tipke za sigurno gašenje Raspberry Pi u poglavlju 2.4, rješenje tog problema dolazi u obliku tipkala spojenih na GPIO21 i GND-a, te python skripte **offtipka.py** koja će se pokrenuti prilikom paljenja Raspberry Pi-a.

Da bi smo kreirali skriptu koristiti ćemo sljedeću naredbu:

```
cd
sudo nano offtipka.py
```

**nano** alat otvara tekst editor u kojem stvaramo sljedeću skriptu:

```
import RPi.GPIO as GPIO
import os
gpio_broj = 21
GPIO.setmode(GPIO.BCM)
GPIO.setup(gpio_broj, GPIO.IN, pull_up_down=GPIO.PUD_UP)
try:
    GPIO.wait_for_edge(gpio_broj, GPIO.FALLING)
    os.system("sudo shutdown -h now")
except:
    pass
GPIO.cleanup()
```

Skripta koristi **RPi.GPIO** biblioteku namijenjenu za rad sa GPIO pinovima Raspberry Pi-a s kojom vršimo detekciju padajućeg napona na GPIO21 pinu prilikom pritiskanja tipkala, dok je **os** biblioteka potrebna za aktiviranje naredbe za terminal **sudo shutdown -h now**.

Kako bismo skriptu aktivirali pri svakom paljenju Raspberry Pi-a, potrebno je dodati naredbu za pokretanje u datoteku **/etc/rc.local** sljedećom naredbom:

```
sudo nano /etc/rc.local
```

Sljedeću naredbu dodajemo u predzadnji red, odmah iznad naredbe **exit 0**.

```
python3 /home/pi/offtipka.py &
```

S obzirom da skripta **offtipka.py** traje dok se ne pritisne tipkalo, znak **&** stavlja se na kraj naredbe kako bi se proces izvršavao u pozadini i omogućio pokretanje daljnjih naredbi ukoliko ih odlučimo dodati.



Kako bi isprobali rad tipkala, potrebno je spremati promjene **rc.local** datoteke i ponovno pokrenuti Raspberry Pi naredbom **sudo reboot**.

Kada pritisnemo tipkalo izvršit će se naredba **sudo shutdown -h now**, što će pokrenuti proces sigurnog gašenja. Zelena lampica tj. indikator aktivnosti blizu porta za napajanje će se pravilno upaliti i ugasiti 10 puta prije li nego ostane trajno ugašen. To je znak da možemo sigurno odspojiti napajanje bez rizika oštećivanja podataka na SD kartici.

### 3.6. Web server

Računalo može postati web serverom ako na njega instaliramo i konfiguriramo serverski softver kao što je Apache HTTP Server sa dodacima kao što je PHP, te zasebnu SQL bazu podataka.

#### 3.6.1. Apache HTTP Server

Apache HTTP Server [8] je softver koji stoji u pozadini 60% svih servera na svijetu. Uvelike njegovoj popularnosti pomaže što je to softver otvorenog koda, tj. besplatan je.

Njegova glavna zadaća je posluživati klijentima tj. posjetiteljima web stranica HTML datoteke preko HTTP protokola na portu 80 [1] te korištenjem raznih modula kao što je PHP, Perl ili Python za generiranje HTML datoteke.

Osim Apache-a zadnjih godina se pojavio još Nginx server (engine x) [9] koji lagano ali sigurno raste u popularnosti, zbog strahovito većih brzine izvođenja i stabilnost. Nažalost prati ga manja kompatibilnost s drugim modulima i trenutno manja raširenost a time i dokumentiranost. U posljednjih 8 godina tržišni udio narastao mu je sa 0 na 16% svih web poslužitelja, a ta brojka će zasigurno i dalje rasti.

Za ovaj rad smo izabrali Apache zbog njene bolje dokumentiranosti te ćemo ga instalirati sljedećom naredbom:

```
sudo apt-get install apache2
```

Nakon instalacije bazna mapa za naše web stranice postaje:

```
/var/www/html/
```

Te ćemo tamo smještati sve naše web stranice.

Za pristup web stranici možemo s bilo kojeg uređaja, koji se nalazi na istoj lokalnoj mreži kao i Raspberry Pi, otvoriti Internet preglednik i upisati sljedeću adresu tj. hostname Raspberry Pi-a:

```
http://gblockpi.local/
```

Dobra praksa je aktivirati mogućnost korištenja **.htaccess** datoteka. Iako u ovom radu to nije potrebno. U slučaju da se odluči na server dodati WordPress blog, to će biti jedan od zahtjeva. **.htaccess** datoteke služe kako bi se konfigurirale određene postavke servera. Najčešće se koriste za izmjenu izgleda web adresa u ljudima prihvatljiviji oblik.

Za aktivaciju potrebno je pokrenuti sljedeću naredbu:

```
sudo a2enmod rewrite
```

te editirati sljedeću datoteku:

```
sudo nano /etc/apache2/apache2.conf
```

gdje je potrebno pronaći i promijeniti segment u kojem piše **AllowOverride None** u sljedeći oblik:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

### 3.6.2. PHP

PHP (Hypertext preprocessor) [10] je programski jezik koji se izvršava na serveru prije učitavanja stranice, nakon čega uz pomoć Apache-a softvera generira HTML kod koji se poslužuje posjetitelju.

Od prosinca 2015. izdana je stabilna verzija PHP 7 te bi službeno trebala zamijeniti verziju 5, dok je verzija 6 u razvoju bila preskočena. PHP 7 se može pohvaliti sa dvostruko bržim brzinama izvođenja od verzije 5, ali to postiže na račun izbacivanja podrške za određene zastarjele naredbe koji se još uvijek koriste u nekim bibliotekama.

Za ovaj rad je izabran PHP 5 kako bi osigurali kompatibilnost sa svim bibliotekama koje koristimo kao što su FPDF za generiranje PDF dokumenata.

Za instalaciju PHP-a potrebno je instalirati sljedeće:

```
sudo apt-get install php5 libapache2-mod-php5
```

**php5** je softver, koji je potreban za izvršavanje PHP skripti, dok je **libapache2-mod-php5** paket potreban za sukladan rad Apache-a i PHP-a.

Dobra je praska dodati još sljedeće najčešće korištene module za PHP, kako bi osigurali kompatibilnost sa budućim projektima koje ćemo potencijalno htjeti pokrenuti na Raspberry Pi-u. U ovome radu ti moduli nam neće biti potrebni.

```
sudo apt-get install php-pear php5-curl php5-gd php5-xsl
```

**php-pear** je programski okvir za ponovno korištenje PHP komponenti. **php5-curl** je biblioteka koja služi za dohvaćanje podataka izvan web stranice. **php5-gd** služi za kreiranje ili manipuliranje slikama. **php5-xsl** služi za rad sa **xsl** datotekama i strukturama.

Posljednje ćemo instalirati **php5-apcu**. Kako je PHP zapravo kod koji se mora interpretirati i kompajlirati kako bi se izvršio, pametno je određene dijelove kompajliranog koda čuvati u memoriji tj. RAM-u, ne bi li se sljedeći puta brže izvršavao. APCu (Alternative PHP Cache) je modul koji radi upravo to i njime postićemo deseterostruko veće brzine na pojedinim dijelovima stranice.

Instalaciju vršimo sljedećom naredbom:

```
sudo apt-get install php5-apcu
```

Kako bi primijenili učinjeno, Apache servis potrebno je ponovno pokrenuti:

```
sudo service apache2 restart
```


Za provjeru ispravnog rada PHP-a, stvorit ćemo malu PHP stranicu koja provjerava verziju PHP-a i vraća specifikacije instalirane verzije.

```
nano /var/www/html/info.php
```

**info.php** editirat ćemo na sljedeći način:

```
<?php
phpinfo();
?>
```

Odlaskom na web stranicu <http://gblockpi.local/info.php> vratiti će se sljedeći odaziv:

<div>PHP Version 5.6.24-0+deb8u1</div> 	
System	Linux gblockpi 4.4.21-v7+ #911 SMP Thu Sep 15 14:22:38 BST 2016 armv7l
Build Date	Jul 26 2016 23:12:41
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-mysqlnd.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-apcu.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mcrypt.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini, /etc/php5/apache2/conf.d/20-xsl.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, mcrypt.*, mdecrypt.*

Slika 3.5 info.php odaziv

### 3.6.3. SQL

SQL (Structured Query Language) je programski jezik posebne namjene, za svrhu rada s relacijskim bazama podataka [1] što podrazumijeva stvaranje baza i tablica te izradu, traženje, ažuriranje i brisanje podataka.

Najpoznatiji SQL softver otvorenog koda je zasigurno MySQL, koji pokriva najveći dio tržišta. Ali 2009. originalni kreatori MySQL-a napustili su MySQL zbog brojnih akvizicija (trenutno u vlasništvu Oracle-a) te osnovali udrugu MariaDB.

MariaDB [11] je softver za relacijske baze podataka. Može se pohvaliti savršenom kompatibilnošću s MySQL-om te puno većim brzinama rada, efektivnošću te bržim izdavanjem sigurnosnih zakrpa.

Za ovaj rad izabrali smo MariaDB te vršimo instalaciju sljedećim naredbama:

```
sudo apt-get install mariadb-server mariadb-client
sudo apt-get install php5-mysqldb
```

**mariadb-server** je softver za relacijske baze podataka. **mariadb-client** je alat za komunikaciju s bazom podataka iz terminala. **php-mysqldb** je paket koji omogućava sukladan rad PHP-a i baze podataka.

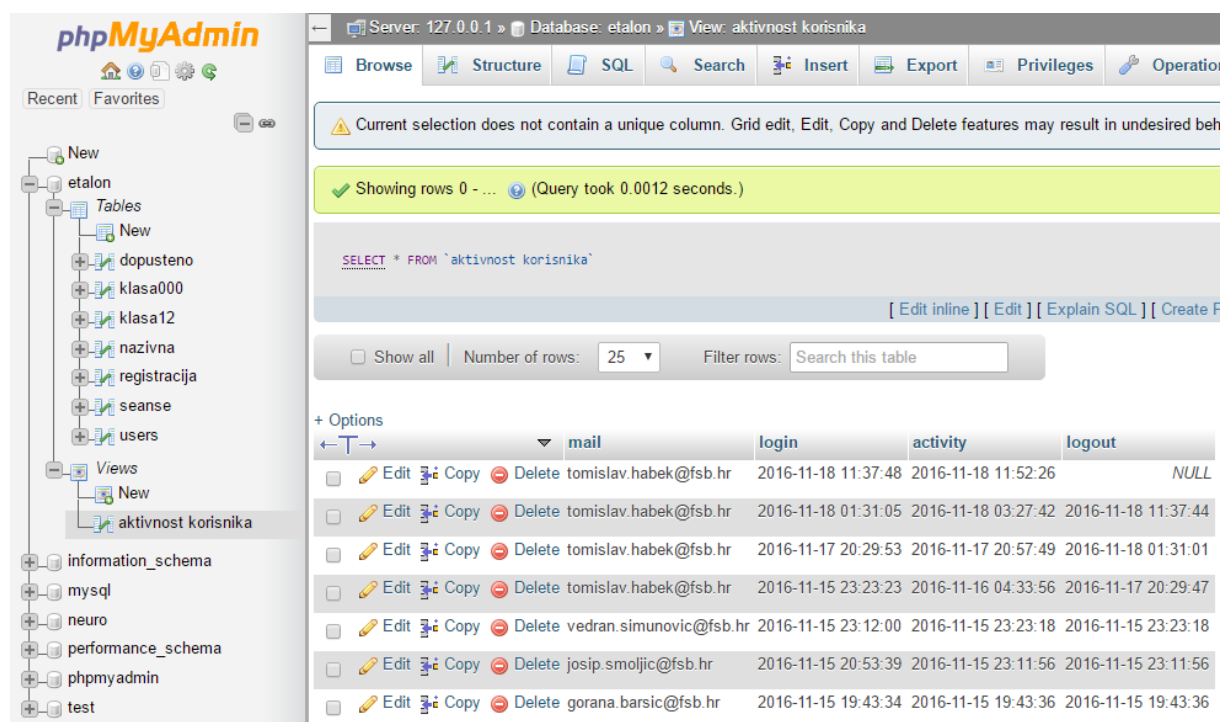
Sama instalacija traži stvaranje korisnika baze te definiranje njegove lozinke.

Kako bi primijenili učinjeno, Apache servis potrebno je ponovno pokrenuti:

```
sudo service apache2 restart
```

### 3.6.4. phpMyAdmin

phpMyAdmin [12] je softver temeljen na PHP-u, u obliku web stranice, koji služi za vizualno upravljanje bazom podataka.



Slika 3.6 phpMyAdmin grafičko sučelje

Instalaciju vršimo sljedećom naredbom:

```
sudo apt-get install phpmyadmin
```

Prilikom instalacije potrebno je unijeti podatke o serverskom softveru koji koristimo (Apache 2), te unijeti lozinku baze podataka koju smo definirali za MariaDB.

Naposljetku za dobru povezanost potrebno je editirati sljedeću datoteku:

```
sudo nano /etc/apache2/apache2.conf
```

te u zadnjem redu dodati sljedeći kod:

```
Include /etc/phpmyadmin/apache.conf
```

Kako bi primijenili učinjeno, Apache servis potrebno je ponovno pokrenuti:

```
sudo service apache2 restart
```

### 3.7. Bežična pristupna točka

Bežična pristupna točka (eng. Wireless Access Point) je uređaj koji omogućava bežičnim uređajima priključivanje na računalnu mrežu koristeći WiFi. Obično se uključuje u mrežni usmjerivač ako se želi omogućiti pristup žičanoj mreži, te se koristi kao relej za druge bežične uređaje unutar radijusa dometa.

Svaki puta kada se spajamo na WiFi, zapravo se spajamo na bežičnu pristupnu točku.

Raspberry Pi 3 u sebi ima WiFi mogućnosti za spajanje na druge mreže, ali uz poseban softver moguće ga je prenamijeniti da djeluje kao pristupna točka, što ćemo za potrebe ovog rada i napraviti.

Instalirati ćemo sljedeće pakete:

```
sudo apt-get install dnsmasq hostapd
```

**hostapd** (Host access point daemon) služi za pretvaranje ugrađenog WiFi modula u bežičnu pristupnu točku, dok **dnsmasq** služi kao DNS (Domain Name System) i DHCP (Dynamic Host Configuration Protocol) server. DNS usluge služe za korištenje imena umjesto IP adresa, dok DHCP služi za dodjeljivanje IP adresa uređajima koji će se spajati na Raspberry Pi.

#### 3.7.1. Statička IP adresa

Za osiguravanje uspješnog rada Raspberry Pi kao pristupna točka mora imati statičnu IP adresu na bežičnoj mreži.

Kako bi to postigli prvo moramo ukloniti početnu postavku Raspberry Pi-a editiranjem sljedeće datoteke:

```
sudo nano /etc/dhcpd.conf
```

te dodajući u zadnji red:

```
denyinterfaces wlan0
```

Nakon spremanja promjene možemo editirati datoteku za statičke postavke sljedećom naredbom:

```
sudo nano /etc/network/interfaces
```

Te editirati sekciju **wlan0** da izgleda kao sljedeći blok koda:

```
allow-hotplug wlan0
iface wlan0 inet static
    address 172.24.1.1
    netmask 255.255.255.0
    network 172.24.1.0
    broadcast 172.24.1.255
#    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Ovime postizemo statičnu IP adresu pristupne točke **172.24.1.1**.

Kako bi primijenili učinjeno, **dhcpd** servis potrebno je ponovno pokrenuti te nanovo učitati **wlan0** konfiguraciju:

```
sudo service dhcpd restart
sudo ifdown wlan0; sudo ifup wlan0
```

### 3.7.2. Konfiguracija hostapd-a

Za konfiguriranje pristupne točke potrebno je kreirati novu datoteku **hostapd.conf** sljedećom naredbom:

```
sudo nano /etc/hostapd/hostapd.conf
```

Gdje ćemo upisati i spremati sljedeće:

```
# ime Wifi sučelja
interface=wlan0
# Koristi nl80211 driver za WiFi, specifičan za Raspberry Pi 3
driver=nl80211
# Ime Wifi mreže
ssid=gBlockPi
# Koristi 2.4GHz frekvenciju
hw_mode=g
# Koristi kanal broj 6
channel=6
# Omogući korištenje 802.11n
ieee80211n=1
# Omogući WMM (WiFi Multimedia)
wmm_enabled=1
# Omogući 40MHz kanal sa 20ns intervalom zaštite
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
# Prihvaćaj sve uređaja, tj. MAC adrese (media access control address)
macaddr_acl=0
# Koristi WPA autentifikaciju
auth_algs=1
# Da li je potrebno da kornici prehodno zanju ime Wifi mreže?
ignore_broadcast_ssid=0
# Koristi WPA2
wpa=2
# Koristi pre-shared ključ
wpa_key_mgmt=WPA-PSK
# Lozinka mreže
wpa_passphrase=lozinkakojunitkonikadanecepogoditi
# Koristi AES, umjesto TKIP
rsn_pairwise=CCMP
```

Kako bi javili hostapd-u što je potrebno učitati prilikom paljenja Raspberry Pi-a, editirati ćemo sljedeću datoteku:

```
sudo nano /etc/default/hostapd
```



gdje ćemo pronaći dio koda **#DAEMON\_CONF**="" i zamijeniti sa:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

### 3.7.3. Konfiguracija DNSMASQ-a

Konfiguracijska datoteka koja je došla sa softverom **dnsmasq**, u sebi sadrži pregršt informacija o tome kako se koristi, ali većina podataka nije potrebna za naše svrhe. Originalnu konfiguracijsku datoteku **dnsmasq.conf** preimenovati ćemo u **dnsmasq.conf.orig** za referenciranje u slučaju problema, te kreirati novu datoteku za editiranje:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig  
sudo nano /etc/dnsmasq.conf
```

Alat **mv** (move), služi za premještanje ili preimenovanje datoteka ili mapa.

Novu **dnsmasq.conf** datoteku popuniti ćemo sljedećim kodom:

```
interface=wlan0      # Koristi wlan0  sučelje  
listen-address=172.24.1.1 # Slušaj adresu pristupne točke  
bind-interfaces      # Veže sučelja kako se podaci ne bi slali drugdje  
server=8.8.8.8       # Proslijedi DNS upite na Google DNS  
domain-needed       # Ne prosljeđuj kratka imena  
bogus-priv          # Ne prosljeđuj krive IP adrese  
dhcp-range=172.24.1.50,172.24.1.150,12h # Adrese se dodjeljuju u  
definiranom raspon, sa maksimalnim trajanjem od 12h
```

### 3.7.4. Konfiguracija IP4 prosljeđivanja

Kako bi uređaji spojeni na Raspberry Pi WiFi mrežu mogli pristupiti uređajima na žičanoj Ethernet mreži, potrebno je uključiti IP4 prosljeđivanje.

U okviru ovo rada, to nije potrebno ali je dobra praksa to učiniti za buduće primjene.

To činimo editiranjem konfiguracijske datoteke **sysctl.conf**:

```
sudo nano /etc/sysctl.conf
```

gdje je potrebno odkomentirati tj. ukloniti znak „#“ ispred naredbe:

```
net.ipv4.ip_forward=1
```

### 3.7.5. Dijeljenje Interneta

Kako bi Raspberry Pi Internet konekciju, koju dobiva od lokalne Ethernet mreže, dodijelio uređajima koji su spojeni na njegovu pristupnu točku, potrebno je konfigurirati IP tablice, tj. provesti određeno spajanje **eth0** (Ethernet) i **wlan0** sučelja.

To provodimo upisivajući sljedeće naredbe:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

Upravo napravljenu konfiguraciju IP tablica možemo spremiti u datoteku **iptables.ipv4.nat** sljedećom naredbom:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

IP tablice potrebno je pokrenuti prilikom svakog paljenja Raspberry Pi-a što ćemo učiniti dodajući **iptables.ipv4.nat** naredbu u **rc.local** datoteku, kao što smo već činili sa Python skriptom za sigurno gašenje sustava opisano u poglavlju 3.6.

Potrebno je editirati datoteku:

```
sudo nano /etc/rc.local
```

te dodati sljedeći red iznad naše python skripte za gašenje:

```
iptables-restore < /etc/iptables.ipv4.nat
```

### 3.7.6. Pristupno ime serveru

Dok smo na Ethernet mreži mogli pristupiti serveru odlaskom na **http://gblockpi.local/**, to nije moguće kada smo spojeni na pristupnu točku Raspberry Pi-a.

Kako bi omogućili spajanje na server tj. otvaranje stranice koristeći posebno ime, potrebno je editirati sljedeću datoteku:

```
sudo nano /etc/hosts
```

te na dno dodati red:

```
172.24.1.1      gblock.pi
```

gdje IP adresa označava statičku IP adresu **wlan0** sučelja, a **gblock.pi** ime s kojom ćemo pristupiti toj IP adresi.

Time smo konfigurirali sve postavke potrebne za obnašanje funkcije bežične pristupne točke, te ih možemo primjetni jednostavnim paljenjem servisa **hostapd** i **dnsmasq**, te vršenjem ponovnog pokretanja Raspberry Pi-a.

```
sudo service hostapd start  
sudo service dnsmasq start  
sudo reboot
```

### 3.8. Upotreba

Lokacija na Raspberry Pi-u na koju smještamo sve naše web stranice je sljedeća:

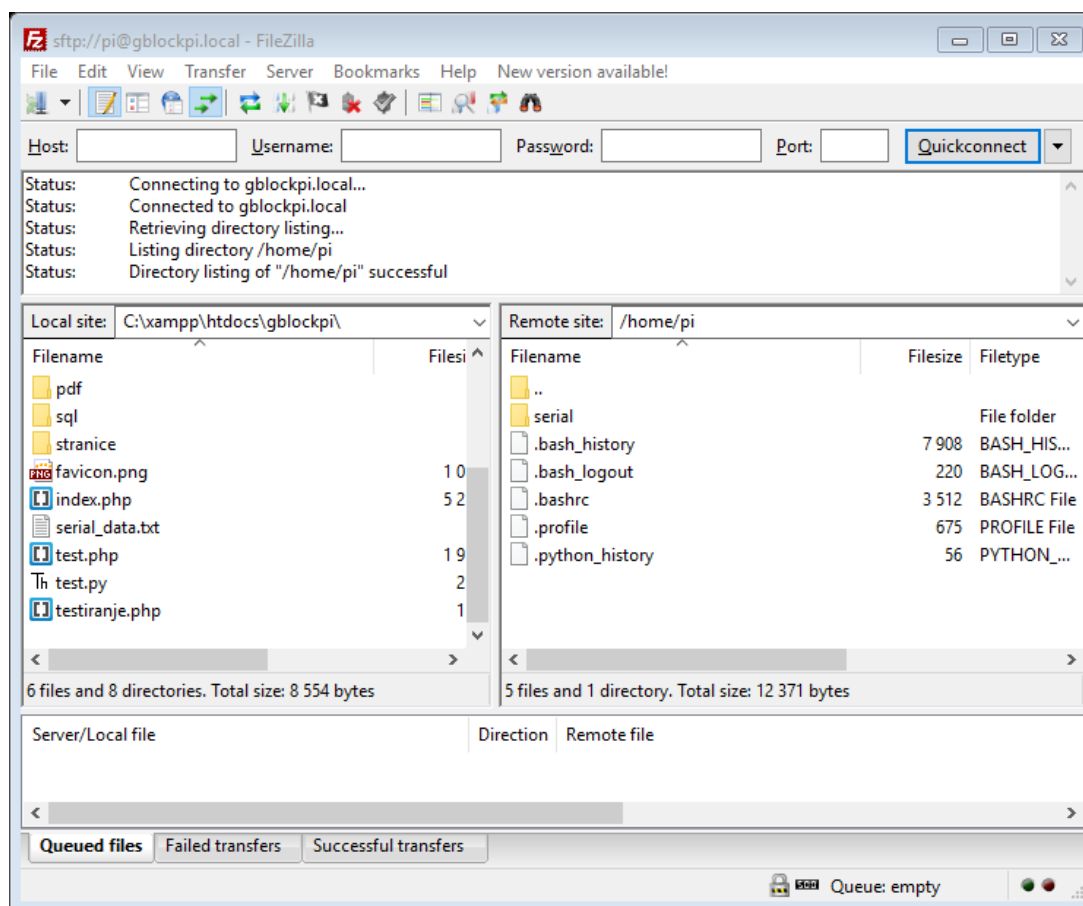
```
/var/www/html/
```

Samo prebacivanje vršimo pomoću softverskog paketa otvorenog koda Filezilla [V]. Računalo na kojem pokrećemo Filezilla-u mora biti na istoj lokalnoj mreži kao i Raspberry Pi.

SSH protokol koji smo koristili kroz poglavlje 3, za daljinsko konfiguriranje i instalaciju raznih softverskih paketa, također može služiti i za prebacivanje datoteka.

Filezilla čini upravo to.

Za uspješnu konekciju treba unijeti **hostname** (gblockpi.local), korisničko ime (pi), lozinku te definirati korišteni port 22 specifičan za SSH protokol.



Slika 3.7 FileZilla

Nakon prebacivanja web datoteka, možemo pristupiti našoj web stranici, spajanjem na WiFi Raspberry PI-a, koristeći Android tablet ili neki drugi uređaj.

WiFi mreža nosit će ime **gBlockPi**, na koju se spajamo lozinkom namještenom u poglavlju 3.8.2, te otvaramo web stranicu koristeći sljedeći adresu u Internet pregledniku:

```
http://gblock.pi/
```

## 4. SIGURNOST WEB STRANICA

Internet je globalna mreža računala. Sve što se nalazi na Internetu dostupno je bilo gdje u svijetu, u bilo kojem trenutku, bilo kojoj osobi. To predstavlja ogromno dostignuće čovječanstva, povezujući cijeli svijet u jednu veliku zajednicu te omogućujući kolaboracije i dijaloge, koji se bez Interneta nikada ne bi mogli ni dogoditi. Tom novom povezanošću omogućena je razmjena ideja na razini dosada nikada viđenoj, time drastično ubrzavajući razvoj svih znanstvenih grana te u konačnici i živote svih ljudi.

Uz sve prednosti koje Internet donosi, on donosi i noćne more svim ljudima zaduženim za njegovu sigurnost. Sve što se nalazi na Internetu nikada neće moći biti u potpunosti sigurno, koliko god zidova postavili, sve dok postoje ljudi koji će tu sigurnost gledati kao izazove te ljudi kojima je to u interesu.

Nemogućnost potpune zaštite vidimo i u svjetskim IT poduzećima i njihovoj svakodnevnoj borbi protiv neovlaštenog pristupa u njihove sustave. Sigurnost web stranica najjača je kada o njoj svakodnevno brine tim ljudi i vrši zakrpe jednakom brzinom kojom se rupe u sigurnosti i stvaraju, tj. otkrivaju.

Informacijski sustav mjerenja etalona duljine, koji se nalazi na Raspberry Pi-u, neće biti direktno dostupan na Internetu, ali će biti spojen na lokalnu mrežu FSB-a, koji ima brojne pristupne točke, uključujući i onu na samom Raspberry Pi-u. Pristupne točke možda jesu zaštićene lozinkama, ali nisu neprobojne, time izlažući naš sustav napadima.

Iako nećemo moći kroz ovaj rad stvoriti potpuno nepropusnu zaštitu, možemo poduzeti korake kojima ćemo neovlašteni pristup našoj stranici značajno otežati.

### 4.1. Korisničko ime i lozinka

Sigurnost web stranica se uglavnom svodi na povlaštenost korisnika, tj. kojim korisnicima je dopušten pristup i kojima nije.

Najlakši način provođenja identifikacije korisnika je korištenje korisničkih imena i lozinki.

Ta vrste autentifikacije se implementira jednostavnom formom za unos korisničkog imena i lozinke te se provjerava postojanje tog korisnika u bazi podataka.

Ukoliko se korisnik s odgovarajućim pristupnim podacima nađe u bazi podataka, stvara se sesija s jednim ili više podataka.

Sesiju čini skup podataka spremljenih na serveru. Tim podacima se može pristupiti samo ako posjetitelj stranice ima odgovarajući kolačić s identifikacijskim kodom sesije koja se automatski generira prilikom pokretanja sesije. Stoga su podaci sesije vezani uz uređaj kojim se pristupalo stranici. Kolačići u kojima je spremljen identifikacijski kod sesije se nalaze na uređaju kojim pristupamo web stranici. Sesija nam na taj način daje jednostavni repozitorij podataka koji se može dohvatiti na svim stranicama koje čine naš sustav. Sesija traje sve dok se ne izbriše odjavom korisnika.

Svaku stranicu koju želimo zaštititi, zaključavamo provjerom postojanja određenog podatka u sesiji te dodatnom provjerom da li podaci iz sesije postoje i u bazi podataka.

Taj proces pokazati ćemo na sljedećem primjeru:

#### **prijava.php**

```
<!DOCTYPE html>
<html>
<body>
  <form action = "provjera.php" method = "POST">
    <input type="text" name="korisnickoime">
    <input type="password" name="lozinka">
    <button type="submit">Prijava</button>
  </form>
</body>
</html>
```

Forma se u gornjem primjeru **prijava.php** sastoji od dva polja za unos podataka i jedne tipke za poslati podatke.

Atributom **type**, definirali smo kakve unose očekujemo u poljima te što radi tipka na pritisak.

**type="text"** označava očekivanje riječi i rečenica, tj. znakova.

**type="password"** označava lozinku te će se unešeni tekst automatski prikazivati prikriveno crnim kružićima kako smo navikli vidjeti diljem Interneta.

**type="submit"** označava tipku **Prijava**, čijim će se pritiskom svi podaci unutar forme proslijediti stranici **provjera.php** metodom POST [1].

## provjera.php

```
<?php
include("db_connect.php");
if($_SERVER["REQUEST_METHOD"] == "POST") {
    $ime=$_POST['korisnickoime'];
    $lozinka=$_POST['lozinka'];
    $stmt=$con->prepare("SELECT id FROM korisnici WHERE ime={$ime} and
lozinka={$lozinka}");
    $baza=$stmt->fetch();
    $id=$baza["id"];
    if (isset($id)) {
        session_start();
        $_SESSION['id'] = $id;
        header("location: tajna.php");
    }
    else
    {
        header("location: prijava.php");
    }
}
?>
```

Najprije se spajamo na bazu podataka dodavanjem skripte **db\_connect.php** objašnjenom u završnom radu [1].

Dohvaćamo proslijeđene podatke iz **prijave.php** koristeći naredbu **\$\_POST[]**, te ih spremamo kao varijable.

Postavljamo pitanje bazi podataka:

Dohvati **id** iz tablice **korisnici** gdje je **ime** jednako upisanom korisničkom imenu i **lozinka** jednaka upisanoj lozinci.

Kako bi ovo bilo uspješno, moramo najprije u bazi imati tablicu zvanu **korisnici** s popunjenim poljima **id**, **ime** i **lozinka**.

Ako baza uspije pronaći korisnika s odgovarajućim korisničkim imenom i lozinkom, dohvaća njegov **id** i sprema ga u sesiju.

Ako je prijava bila uspješna, preusmjeruje korisnika na stranicu **tajna.php**. U slučaju neuspjeha, sesija se ne postavlja i preusmjeruje korisnika natrag na stranicu **prijava.php**.

Kako bi zaštitili **tajna.php** stranicu, napraviti ćemo zasebnu skriptu **lock.php** koja provjerava je li korisnik prijavljen na stranicu.

### **lock.php**

```
<?php
include("db_connect.php");
session_start();
if (isset($_SESSION['id'])) {
    $id=$_SESSION['id'];
    $stmt=$con->query("SELECT count(*) as num FROM korisnici WHERE
id={$id}");
    $baza=$stmt->fetch();
    $num=$baza["num"];
}
if(!isset($num) && !($num=1))
{ header("Location: prijava.php"); }
?>
```

**lock.php** dohvaća **id** korisnika iz sesije, te provjerava u bazi podataka da li taj korisnik postoji. Ukoliko postoji, ništa se ne dogodi. Ako ne postoji, korisnik je preusmjeren na web stranicu **prijava.php**.

**lock.php** koristi se dodavanjem na vrh svih stranica koje želimo zaštititi naredbom **include**, kako je prikazano u web stranici **tajna.php**.

### **tajna.php**

```
<!DOCTYPE html>
<html>
<body>
    <?php include('lock.php'); ?>
    <p>Ultra super skrivene riječi, koje samo korisnici web stranice mogu
saznati prijavom.</p>
    <a href="odjava.php">Odjavite se putem ovog linka</a>
</body>
</html>
```



Za odjavu iz sustava potrebno je stisnuli link na stranici **tajna.php**, koji nas vodi na **odjava.php**. **odjava.php** radi na način da uništava sve podatke spremljene u sesiji koristeći sljedeće naredbe te nas vraća na stranicu **prijava.php**:

### **odjava.php**

```
<?php
    session_start();
    session_unset();
    session_destroy();
    header("Location: prijava.php");
?>
```

## **4.2. Zaštita**

Dodavanjem sloja prijave korisnika, eliminirali smo većinu znatiželjnih očiju koji bi u protivnom završili na našoj web stranici. Kako bi obeshrabрили i one najznatiželjnije, prethodni primjer iz poglavlja 4.1 moramo učiniti znatno kompleksnijim rješavanjem određenih sigurnosnih rupa u prethodnom pristupu.

### **4.2.1. SQL injektiranje**

Baza podataka funkcionira odgovaranjem na upite koje postavljamo. U primjeru iz prethodnog poglavlja 4.1, koristili smo upite koji u sebi sadrže varijable uvezenih iz forme na stranici **prijava.php**.

U slučaju da korisnik u forme unosi pokušaje korisničkog imena i lozinke, sve će raditi kako treba, te će stranica ostati zaštićena.

Ukoliko korisnik odluči umjesto uobičajenih riječi u forme upisati SQL naredbe, tada nastaje problem, jer dopuštamo neautoriziranoj osobi kontrolu nad našom bazom podataka.

Jednostavan način zaobilaženja naše provjere bi bilo sljedeći unos:

```
Korisničko ime: Mirko OR 1=1
Lozinka: Slavko OR 1=1
```

čime bi naš upit postao:

```
SELECT id FROM korisnici WHERE ime=Mirko OR 1=1 and lozinka=Slavko OR 1=1
```

S obzirom da je jednadžba **1=1** uvijek istinita te koristeći logički operator **OR** (ili), potpuno je nebitno što je uneseno korisničko ime i lozinka nevažeće. Baza će takvim upitom vratiti **id**-ove svih korisnika, te neovlaštenoj osobi omogućiti pristup.

Rješenje se krije u samom načinu provođenja SQL upita koristeći PHP PDO prepared statements (pripremljene upite).

Razlika korištenja pripremljenih upita od običnih upita je u samoj komunikaciji s bazom podataka. Upit i varijable upita se šalju bazi odvojeno, dok se u običnom upitu šalju zajedno.

To omogućuje pravilno obrađivanje pripremljenog upita, jer se SQL naredbe i logički operatori čitaj samo iz upita, dok se varijable upita gledaju isključivo samo kao varijable. Pripremljeni upiti trebaju malo više vremena da se obrade od običnih, ali vrijedno je problema koji rješavaju.

Na sljedećem ćemo primjeru pokazati pravilan način korištenja pripremljenih upita za **provjera.php** iz primjera u poglavlju 4.1:

```
$ime=$_POST['korisnickoime'];  
$lozinka=$_POST['lozinka'];  
$query="SELECT id FROM korisnici WHERE ime=:ime and lozinka=:lozinka";  
$stmt=$con->prepare($query);  
$stmt->bindParam(':ime', $ime);  
$stmt->bindParam(':lozinka', $mypassword);  
$stmt->execute();  
$baza=$stmt->fetch();  
$id=$baza["id"];
```

Glavna značajke korištenje pripremljenih upita naspram običnih je korištenje naredba **prepare**, **bindParam** i **execute** umjesto standardne **query** naredbe.

**prepare** šalje upit bazi.

**bindParam** šalje varijablu upita bazi.

**execute** izvršava kompletan upit pripremljen naredbama **prepare** i **bindParam**.

#### 4.2.2. Otimanje sesija

Otimanje sesije je čin krađe korisničkog kolačića, koji definira sesiju, od strane neovlaštena osobe.

Kako bi haker (neovlaštena osoba) došao do našeg sesijskog kolačića mora imati pristup našem računalu ili presresti dostavu kolačića prilikom njegovog stvaranja. Presretanje se događa postavljanjem hakera kao posrednika podatkovnih paketa između nas i servera.

Kada se haker domogne sesijskog kolačića može koristiti našu web stranicu kao da smo to mi. No, to traje sve dok se ne odjavimo sa stranice i učinimo kolačić nevažećim.

U tom vremenskom periodu, haker može pristupiti sesijskim podacima spremljenim na serveru. S obzirom da je naš kontrolni podatak **id** korisnika, koji je uvijek isti, s tom informacijom haker će uvijek moći stvarati svoje kolačiće bez razlike na našu prijavljenost u sustav ili promjenu korisničkog imena ili lozinke.

Zbog tog razloga, nikada nije pametno korisničke podatke koristiti kao kontrolni podatak.

Kako bi riješili ovaj problem kontrolni podatak mora biti varijabilan, tj. jedinstven za svaku sesiju. Taj problem rješavamo nasumično generiranim nizom znakova, tj. tokenom.

Za implementaciju takvog rješenja, najprije moramo unutar baze podataka stvoriti tablicu **sesije** sa poljima za **id** korisnika i **token-a**.

Prilikom autentifikacije korisnika, generirati ćemo token i spremiti ga u sesiju sljedećim naredbama:

```
$token=bin2hex(openssl_random_pseudo_bytes(16));  
$_SESSION['token']=$token;
```

Naredba **bin2hex(openssl\_random\_pseudo\_bytes(16))** generirati će 32 nasumična znaka za koje se garantira iznimna nasumičnost i neponovljivost. **token** i **id** korisnika spremamo u tablicu **sesije**.

Time smo stvorili novi varijabilni kontrolni podatak u sesiji i bazi podataka, koji koristimo umjesto dosada korištenog **id-a**. Prilikom odjave, osim brisanja podataka sesije biti će potrebno i izbrisati podatke iz baze unutar tablice **sesije**.

Dodati ćemo sljedeći kod u **odjava.php** između **session\_start()** i **session\_unset()**:

```
include("db_connect.php");  
$token=$_SESSION['token'];  
$query="DELETE FROM sesije WHERE id=:token";  
$stmt=$con->prepare($query);  
$stmt->bindParam(':token', $token);  
$stmt->execute();
```

Postigli smo određenu vrstu zaštite protiv otimanja sesije. Iako ne možemo utjecati na povremene upade upornih hakera, ovim načinom možemo smanjiti njihovo vrijeme djelovanja.

#### 4.2.2.1. SSL

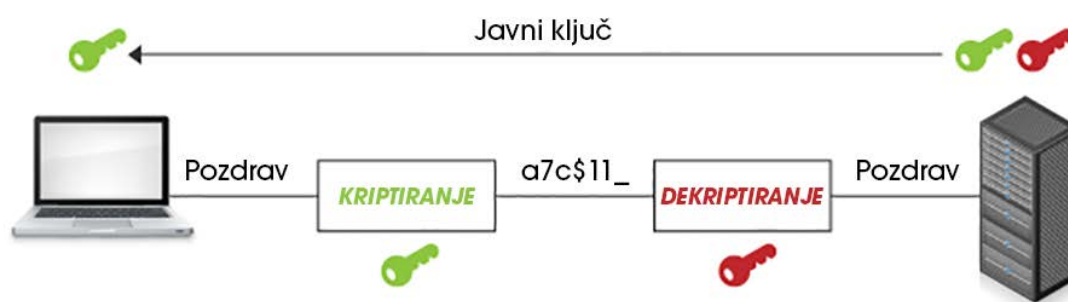
Za potpunu zaštitu ovog problema, potrebno je koristiti SSL (Secure Socket Layer).

SSL je metoda kriptiranja web prometa, kojom se sprečava neovlaštenim osobama pristup do web paketa. To uključuje sesije, kolačiće te slanje podataka preko POST metode.

Točnije, hakeri i dalje mogu doći do svih web paketa, ali s obzirom da su paketi kriptirani neće moći doznati što je u njima.

Web stranice koje koriste SSL lako je primijetiti uočavanjem **https://** umjesto običajnih **http://** na početku web adrese.

Kako bi web stranica koristila SSL mora dobiti SSL certifikat od treće komercijalne strane, koji se sastoji od javnog i privatnog ključa za kriptiranje i dekriptiranje.



Slika 4.1 SSL protokol

Kako je prikazano na slici 4.1, server dijeli javni ključ kako bi korisnici sve podatke slali kodirane. Samo server ima ključ za dekriptiranje, te samo on može znati što je korisnik poslao.

Na taj način osiguravamo da poruka bude dostavljena samo namijenjenom sugovorniku, bez razlike koliko puta bila presrećena.

S obzirom da u ovom radu, web stranica neće biti dostupna preko Interneta, nema smisla platiti treću stranu za izradu SSL certifikata.

Samostalna izrada certifikata i konfiguracija je moguća, ali nadilazi spektar ovo diplomskog rada te se neće obrađivati.

#### 4.2.3. Fizički napad

Najlakša metoda krađe podataka je fizički pristup serveru.

Ukoliko je korisnik web stranice, nakon dugog napornog dana, otišao kući bez odjave sa web stranice, pristup je još uvijek ostao aktivan.

Neovlaštenoj osobi, u ovom slučaju lopovu, bilo bi dovoljno ušuljati se na uobičajeno radno mjesto i korištenjem računala umornog radnika, dobiti direktan pristup bazi podataka.

Kako bi to spriječili, dodati ćemo automatsko odjavljivanje u našu kontrolnu **lock.php** datoteku.

Najprije moramo proširiti tablicu **sesije** iz prethodnog poglavlja sa poljem **vrijeme**. Unutar tog polja spremati ćemo zadnje vrijeme pristupa nekoj od zaštićenih web stranica, tj. onih stranica koje u sebi imaju **lock.php**.

Kako bi inicijalno vrijeme početka sesije bilo spremljeno u istoimenoj tablici, potrebno je dodati varijablu **vrijeme** u upit unutar **provjera.php**, dodjeljujući joj funkciju **NOW()**.

```
INSERT INTO seanse SET id=:token, userid=:userid, vrijeme=NOW()
```

Odlaskom na tajne stranice aktivira se **lock.php**. Ukoliko razlika zadnjeg vremena pristupa i trenutnog vremena prelazi određeni iznos potrebno je korisnika automatski odjaviti pozivom **odjava.php**.

Tu provjeru vršimo dodajući sljedeći kod u **lock.php**:

```
#dohvaćanje zadnjeg vremena
$query="SELECT UNIX_TIMESTAMP(vrijeme) FROM sesije WHERE id=:token";
$stmt=$con->prepare($query);
$stmt->bindParam(':token', $token);
$stmt->execute();
```

```
$baza=$stmt->fetch(PDO::FETCH_ASSOC);  
$vrijeme=$baza['UNIX_TIMESTAMP(activity)'];  
#Da li se prešao limit od 4 sata  
if (isset($vrijeme) && (time() - $vrijeme > (4*60*60))) {  
    include("odjava.php");  
#Ako se nije prešao limit, postavi novo vrijeme  
else{  
    $query="UPDATE sesije SET vrijeme=NOW() WHERE id=:token";  
    $stmt=$con->prepare($query);  
    $stmt->bindParam(':token', $token);  
    $stmt->execute();}
```

SQL-ova naredba **NOW()** i PHP naredba **time()** rade istu stvar. Dohvaćaju trenutno vrijeme, ali u drugačijim formatima.

**time()** dohvaća UNIX zapis vremena, tj. koliko je sekundi prošlo od 00:00:00 1. Siječnja 1970., dok **NOW()** dohvaća vrijeme u ISO zapisu npr. 2016-11-14 04:25:52. SQL naredba **UNIX\_TIMESTAMP()** pretvara ISO zapis u UNIX zapis.

Kako bi naš kod u potpunosti radio kako treba i za specificirano vrijeme koje smo zadali, potrebno je editirati konfiguracijsku datoteku **/etc/php5/apache2/php.ini** i promijeniti broj u retku 1510:

```
session.gc_maxlifetime=1440
```

Broj 1440 označava sekunde. Po trenutnim postavkama nakon 24 minute sesije postaju dostupne operaciji čišćenja memorije. To ne znači da će odmah po isteku 24. minute biti izbrisane, ali od tada postoji mogućnost da se to dogodi.

#### 4.2.4. Hashiranje lozinke

Hashiranje je uobičajena praksa spremanja lozinke koju bi svaka baza podataka trebala imati za svoje korisnike. Kako nijedna sigurnost nije stopostotna, pitanje je samo vremena kada će netko upasti u sustav. Kada se to dogodi, hashiranje omogućava spremanje korisničkih lozinke na bazu, u obliku koji ni vlasnik baze ne može razaznati od koje je riječi zapravo nastalo, a kamoli haker.

Hashiranje je posebna vrsta nepovratnog kriptiranja koja pretvara običnu riječ u dugačak niz nasumičnih znakova. Posebnost hashiranja, naspram običnog kriptiranja, je što možemo

hashiranu riječ usporediti sa običnom i doznati da li su „iste“ tj. da li je pogođena originalna lozinka. Ne postoji način dobivanja originalne riječi iz hashirane riječi, osim pukim pogađanjem.

Na taj način, u slučaju neovlaštenog ulaza u bazu, možemo zaštititi mnogobrojne korisnike koji koriste iste lozinke na svim svojim korisničkim računima diljem Interneta.

Hashiranje riječi provodimo PHP naredbama na sljedeći način:

```
$lozinka="12345";  
$skrivenalozinka=password_hash($lozinka, PASSWORD_DEFAULT);
```

Lozinka „12345“, je hashiranjem postala sljedećim nizom znakova:

```
$2y$10$eVGUDGp5Orz/qTNMn1P4eOhr8tlgRzFLtXgp/5U318nUM4qMZRN7K
```

A usporedbu sa originalnim riječju radimo na sljedeći način:

```
if (password_verify($lozinka , $skrivena_lozinka)){  
    #provodi naredbe ukoliko je pogođena lozinka  
}  
else {  
    #provodi naredbe ukoliko nije pogođena lozinka  
}
```

Implementacijom hashiranja ne povećavamo sigurnost naše stranice, ali povećavamo sigurnost naših korisnika, što je uvijek poželjno.

#### **4.2.5. Brute Force napad**

Brute Force napad podrazumijeva napad kojim haker pokušava upasti u sustav konstantnim pokušajima različitih kombinacija korisničkog imena i lozinke. To najčešće ne vrši sam haker, već posjeduje specijaliziran alat koji automatski prolazi kroz specijalizirane rječnike za najčešće korištene riječi i kombinacije. To zna potrajati od nekoliko minuta do nekoliko dana, ovisno o duljini lozinke.

Najlakši način za riješiti ovaj problem je dodavanje limitiranog broja pokušaja prijave, nakon čega se mora pričekati određeno za otključavanje novog broja pokušaja.

Svaki uređaj, kojim se pristupa web stranici, ima svoju jedinstvenu IP adresu.

Do adrese možemo doći upisivanjem sljedeće PHP naredbe:

```
$ip=$_SERVER[ 'REMOTE_ADDR' ] ;
```

Kako bi smo pratili broj prijava potrebno je napraviti tablicu **pokusaji** koja će nositi polja **ipadresa, vrijeme i pokusaj**.

Za implementaciju samo je potrebno unutar **provjera.php** dodati naredbe koje će u odgovarajućim trenucima popunjavati i ažurirati tablicu **pokusaji**, te brisati podatke starije od zadanog vremena npr. 5 minuta ili prilikom uspješne prijave.

#### **4.2.6. DDOS napad**

DDOS (Distributed Denial of Service) napad je glorificirani hakerski napad koji kao jedinu funkciju ima rušenje stranice, tj. prestanak rada servera.

DDOS napad nipošto ne može doći do podataka niti išta promijeniti, nego jednom vrstom vandalizma prekinuti rad servera za vrijeme trajanje napada.

To postiže koordiniranim dolaskom na našu stranicu sa milijuna IP adresa istovremeno unutar kratkog vremenskog perioda često radeći istu funkciju na toj stranici, koju smatraju da bi popunili RAM-e, disk ili preopteretili CPU servera do trenutka neizdržljivosti.

U prethodnom poglavlju dodali smo zaštitu za Brute Force napad. Iako smo se zaštitili od forsirane prijave, u sustav smo uveli popunjavanje SQL tablice prilikom pogrešne prijave. Iako je trajanje tih podataka samo 5 minuta, to daje mogućnost popunjavanja SQL tablice do maksimalnog kapaciteta diska, ukoliko bude dovoljno velik broj IP adresa s kojih se stranica napada.

S obzirom da DDOS napad ne ugrožava same podatke na računalu, bolji izbor je bila zaštita za Brute Force napad.

##### **4.2.6.1. Zaštita DNS-om**

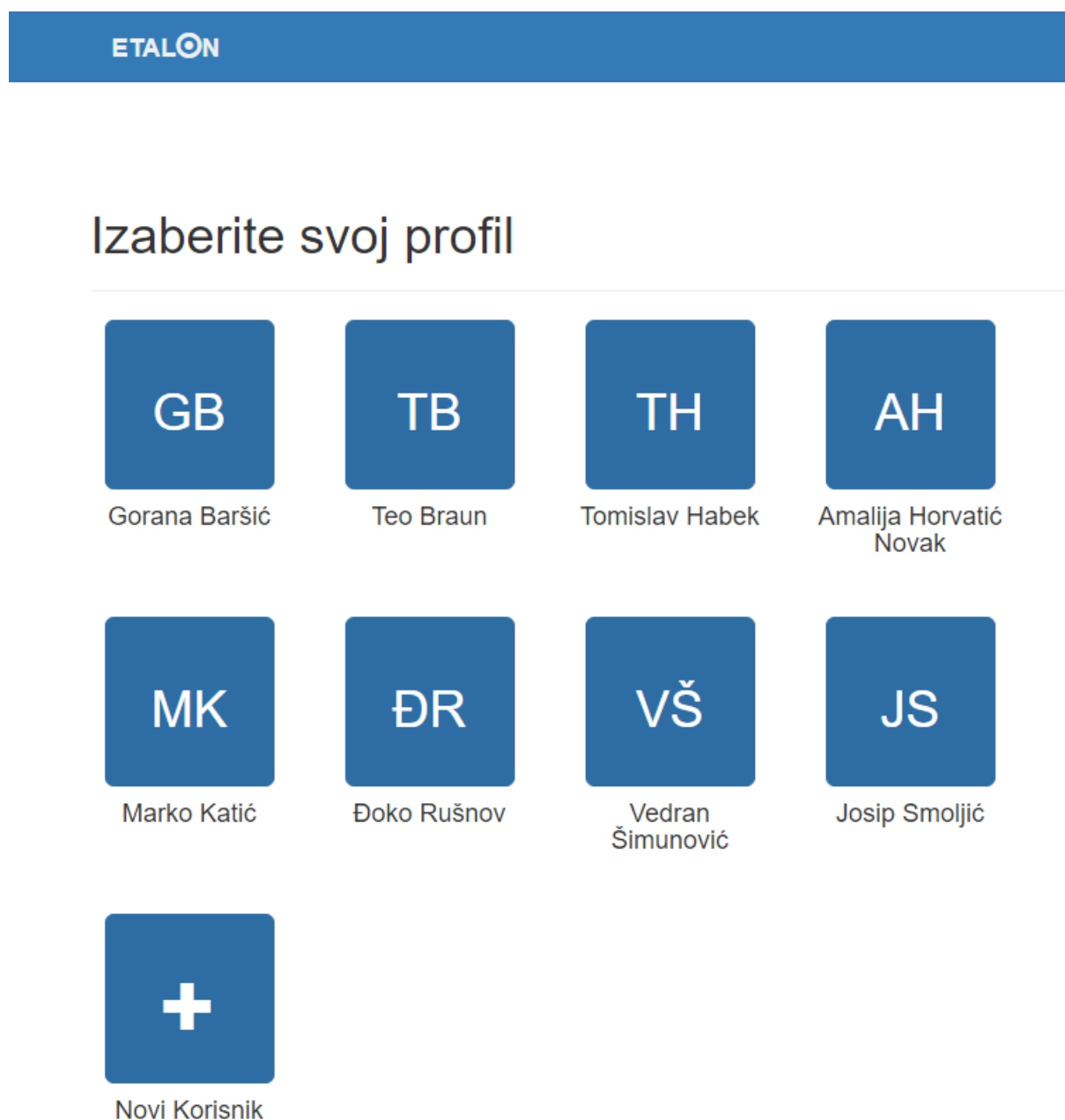
Jedino rješenje za DDOS napade je korištenje komercijalnog DNS servera.

DNS server je uređaj koji preusmjerava promet prema našoj web stranici. U slučaju komercijalnog DNS servera, on bi radio kao filter prema našem serveru, te ukoliko detektira DDOS napad, sprječava prolazak dodatnog prometa prema nama, time štiteći naš server.



### 4.3. Primjena

Sve spomenute načine zaštite u poglavlju 4, iskoristili smo u ovome diplomskom radu izuzevši komercijalna rješenja za SSL protokol i DNS zaštitu.



Slika 4.2 Početno sučelje

Početno sučelje (slika 4.2) čini niz tipki, a svaka tipka predstavlja jednog korisnika. S obzirom da će mali broj korisnika koristiti web stranicu, iskorištena je prilika da se korisnicima olakša prijava automatskim ispunjavanjem korisničkog imena na pritisak tipke.

ETALON

## Unesite lozinku

Teo Braun

teo.braun@zoho.com

.....

Prijava

ETALON

Laboratorij za precizna mjerenja dužina

Ivana Lučića 1, 10 000 Zagreb

tel: +385 01 61 68 327 fax: +385 01 61 68 599

Fakultet strojarstva i brodogradnje

izradio: teo.braun@zoho.com

**Slika 4.3 Sučelje za unos lozinke**

Pritiskom na jednu od tipki na početnom sučelju, stranica nas vodi na sučelje za unos lozinke (slika 4.3) gdje je korisničko ime već uneseno. U ovom stadiju može se promijeniti i korisničko ime, ne bi li koristili skrivene korisničke račune kao što je administrator.

ETALON

Unesite lozinku za registraciju

.....

Novi Korisnik

Ime i Prezime

Teo

Braun

Vrsta korisnika

Mjeritelj

Provjeritelj

Email adresa

teo.braun@zoho.com

teo.braun@zoho.com

Lozinka

.....

.....

Ponovljena lozinka nije ista

Registriraj se

**Slika 4.4** Sučelje za registraciju novih korisnika

Ukoliko želimo dodati novog korisnika, to činimo na početnoj stranici pritiskom na tipku „Novi Korisnik“. Biti ćemo dovedeni na stranicu za registraciju novih korisnika (slika 4.4). Novi korisnici se mogu dodati ukoliko znaju lozinku za registraciju, dok su ostala polja proizvoljna.

Kao dodatnu mjeru sigurnosti za ispravan unos podataka zahtijevamo korisnika za dvostruko upisivanje email adrese i vlastite lozinke, te javascript-om upozoravamo korisnika ukoliko to nije istina.



## Zapisi

Veza sa Certifikatom umjeravanja br.	Naslovnik	Mjerno područje	Broj komada	Klasa	Datum i Vrijeme	Stanje	Akcije
1234	123	2.00 do 23.00 mm	23	00	2016-11-16 04:11:07	0/23	
123456	789	1.00 do 22.00 mm	12	00	2016-11-16 03:13:41	1/12	
456	123	1.00 do 5.00 mm	22	1	2016-11-16 02:56:44	0/22	
123456789	789	10.00 do 30.00 mm	10	00	2016-11-16 01:24:56	1/10	
123456	neznam	20.00 do 40.00 mm	17	1	2016-11-15 23:24:24	0/17	
1337/15	Star Labs	4.00 do 5.00 mm	6	1	2015-02-14 18:45:00	2/6	
1456/18	Stark Enterprise	1.00 do 9.00 mm	14	1	2015-02-14 15:03:22	13/14	
0666/15	Paklenica	1.00 do 3.00 mm	3	0	2015-02-12 18:42:04	2/3	

Slika 4.5 Unutar web stranice – pregled zapisa

Nakon uspješne prijave na stranicu, alatna traka se popunjava korisničkim imenom te tipkama za mjerenja, pregled i odjavu korisnika.

## 5. KOMUNIKACIJA

Komunikacija mjernog uređaja s korisničkim računalom prolazi kroz puno slojeva za njeno ostvarivanje i spremanje u bazu podataka:

1. Mjerni uređaj Mahr Millitron 1240
2. Serijskim sučeljem šalju se mjerni podaci do Raspberry Pi-a
3. Raspberry Pi
4. Python skriptom očitavaju se mjerni podaci te šalju u RAM memoriju **/dev/shm**
5. **/dev/shm** – lokacija na Raspberry Pi-u za virtualne datoteke smještene u RAM-u
6. AJAX metodom, PHP dohvaća podatke iz **/dev/shm** lokacije
7. AJAX metodom, javascript dohvaćene podatke iz PHP-a prikazuje u HTML elementu
8. Javascript-om populiraju se potrebne forme
9. PHP PDO naredbama mjerni podaci se spremaju u SQL bazu podataka
10. Iz SQL baze podataka generira se PDF certifikat

Slojeve 1 do 3 pokrili smo u poglavlju 3.4, dok smo slojeve 8 do 10 pokrili u Završnome radu [1], što nam ostavlja slojeve 4 do 7.

### 5.1. Sloj 4 - Python

Sloj 4, riješen je sljedećom python skriptom:

**rs232devshm.py**

```
import time
import serial
port = "/dev/ttyAMA0"
baudrate = 9600
ser = serial.Serial(port,
baudrate,bytesize=serial.SEVENBITS,parity=serial.PARITY_EVEN,
stopbits=serial.STOPBITS_ONE, timeout=1, xonxoff=True)
if ser.isOpen():
    start=1;
```

```
while (True):
    out=''
    while (out==''):
        time.sleep(10)
        cmd="I"
        ser.write((cmd +'\r\n').encode())
        out = ''
        time.sleep(0.08)
        while ser.inWaiting() > 0:
            out += (ser.read(1)).decode()
    pokusaj=0
    while (True and (pokusaj<50)):
        cmd="M"
        ser.write((cmd +'\r\n').encode())
        out = ''
        time.sleep(0.08)
        while ser.inWaiting() > 0:
            out += (ser.read(1)).decode()
        if (out != ''):
            my_string=out
            new=[x.strip() for x in my_string.split(',') ]
            izlaz=new[-1]
            f = open("/dev/shm/serial_data.txt", 'w')
            f.write(str(izlaz))
            f.close()
            pokusaj=0
        else:
            pokusaj+=1
ser.close()
```

Ova skripta nakon inicijalizacije serijskog sučelja ulazi u beskonačnu petlju **while(True)**. To smo napravili jer je skripta osmišljena kao servis koji se pokreće čim se Raspberry Pi upali.

Beskonačna **while** petlja sastoji se od dvije **while** podpetlje.

Prva inicijalizacijska **while** petlja, svakih 10 sekundi šalje inicijalizacijski znak „I“ za početak komunikacija sa Mahr Millitron-om 1240. To traje sve dok ne dobije odgovor.

Ukoliko dobije odgovor za inicijalizaciju, ulazimo u drugu mjeriteljsku **while** petlju. Unutar petlje šalje se upit „M“ svakih 80 milisekundi, nakon čega se vraćaju mjeriteljski podaci.

Ukoliko nešto pođe po krivu, i mjeriteljski podaci prestanu dolaziti, pokušaj traženja mjeriteljskih podataka će se izvršiti još barem 50 puta. Ukoliko je i dalje neuspješno dohvaćanje mjeriteljskih podataka, vraćamo se u inicijalizacijsku **while** petlju.

Sam dio koda koji sprema podatke u **txt** datoteku je:

```
f = open("/dev/shm/serial_data.txt", 'w')
f.write(str(izlaz))
f.close()
```

Gdje **open** priprema lokaciju i ime datoteke, te akciju koja će se izvršavati, „**w**“ označavajući pisanje. Ukoliko datoteka već postoji, skripta će pisati preko nje. Naredba **write()** označava sam čin pisanja varijable **izlaz** unutar **txt** datoteke, dok **close()** finalizira postupak.

Kako bi se python skripta pokrenula zajedno s Raspberry Pi-om editirati ćemo **rc.local** datoteku kao i u poglavlju o tipki za sigurno gašenje 3.5:

```
sudo nano /etc/rc.local
```

dodajući lokaciju python skripte u predzadnji red:

```
python3 /home/pi/rs232devshm.py
```

## 5.2. Sloj 5 - /dev/shm/

Raspberry Pi, tj. svi Linux sustavi, posebni su po tome što imaju vrlo jednostavan pristup RAM memoriji, korištenjem virtualne lokacije **/dev/shm/**.

Ta lokacija vidljiva je unutar strukture mapa kao i svaka normalna lokacija koja se nalazi na disku.

Sve datoteke ili mape spremljene unutar **/dev/shm/** lokacije postoje samo u RAM-ima, time noseći i sve pogodnosti korištenja RAM-a.

S obzirom da je disk Raspberry Pi-a SD kartica, a koja ima konačan broj brisanja i pisanja, nedopustivo je da na njoj vršimo pisanje i brisanje datoteke svakih 80 milisekundi.

Ovim načinom, osim što produžujemo vijek trajanja SD kartici, postizemo deseterostruko veće brzine pisanja i brisanja, što osigurava bolji prijenos podataka između Python-a i PHP-a.

### 5.3. Sloj 6 – AJAX, PHP

#### rs232devshm.php

```
<?php
$data = file_get_contents('/dev/shm/serial_data.txt');
$niz=array("x1" => $data);
echo json_encode($niz);
?>
```

Naredbom **file\_get\_contents()** dohvaćamo mjerni podatak iz **/dev/shm** lokacije, te je **json\_encode()** naredbom pakiramo za javascript.

### 5.4. Sloj 7 – AJAX, Javascript

```
<script type="text/javascript">
$(function() {
    var repeat= function(){
        $.ajax({
            type: "GET",
            //dataType: "json",
            url: "config/rs232devshm.php",
            success: function(data) {
                var mjerenje=JSON.parse(data);
                document.getElementById("millitron").innerHTML=mjerenje.x1;
            }
        });
    }
    setInterval(function(){repeat()}, 250);
});
</script>
```

Za rad ove skripte koristi se jquery biblioteka [VI]. Ova skripta nalazi se na dnu svake stranice u kojoj želimo prikazivati mjerne podatke, te dohvaća podatke iz **rs232devshm.php** svakih 250 milisekundi i prikazuje mjerenje unutar HTML elementa sa **id** oznakom **millitron**.



## 5.5. Sučelje

Za korištenje mjernih podataka osmislio se sljedeće sučelje prikazano na slici 5.1.

The interface is titled "Mjerenje etalona br. 1 / 10" with a link to "Veza sa certifikatom br. 123456789". It includes a header with "ETALON", "Novo mjerenje", "Pregled", and user information "teo.braun@zoho.com".

Below the header, there are settings for "Oštećenja na mjernim površinama" (radio buttons for -, M, K), "Ln" (20.000 mm), and "Serijski broj" (123456).

The main display shows a large value of **-0.344** with four green checkmarks and two blue plus signs.

Below the display are three columns of measurement data:

Mjerenje 1, $\mu\text{m}$		Mjerenje 2, $\mu\text{m}$		Mjerenje 3, $\mu\text{m}$	
1	-0.400	1	-0.331	1	-0.399
2	-0.377	2	-0.331	2	-0.371
3	-0.390	3	-0.329	3	-0.371
4	-0.375	4	-0.342	4	-0.346
5	-0.345	5	-0.383	5	-0.384
$\delta\text{Li}$	-0.377	$\delta\text{Li}$	-0.343	$\delta\text{Li}$	-0.374

Slika 5.1 Sučelje za očitavanje mjerenja

Mjerni se podaci mogu unositi na tri načina. Ručnim utipkavanjem, pritiskom na „+“ tipku pokraj pojedinog mjernog polja ili na kružne tipke unutar „ekrana“ mjernog uređaja.

Svaka kružna tipka predstavlja lokaciju područja etalona kojeg mjerimo, pa se svakim pritiskom popunjavaju korespondirajuća polja za tu poziciju počevši od mjerenja jedan sve do pet.

## 6. ZAKLJUČAK

Informacijski sustav umjeravanja etalona duljine započet je završnim radom [1] 2015. godine u suradnji sa Nacionalnim Laboratorijom za Duljinu na FSB-u. Rješavanjem njihovog problema papirologije, potrebnih za svrhe certificiranja etalona duljine, nastala je web stranica koja ubrzava veliki dio tog posla.

Uvođenjem web stanice veliki dio ljudskih pogrešaka, u samom procesu izrade certifikata, uvelike se smanjio, ali se nije u potpunosti uklonio. Zadnji izazov u cijelom procesu ostala je komunikacija mjernog uređaja sa web stranicom, ne bi li se uklonila potreba ručnog upisivanja mjernih podataka.

Diplomskim radom nastavljamo započeti posao, te obrađujemo zadnji problem u procesu certificiranja etalona,. Problemom komunikacije sa mjernim uređajem.

Kako je Mahr Millitron 1240 za komunikaciju koristio serijsko sučelje, najteže je bilo usmjeriti naše napore u obećavajuće sklopovlje i softvere koji bi serijsko sučelje uspješno povezali sa PHP kodom naše stranice.

Za potrebe sklopovlja izabrali smo računalo veličine kreditne kartice, Raspberry Pi 3 i modul za serijsku komunikaciju, dok smo za softver utonuli duboko u svijet Linux operativnih sustava i softvera otvorenog koda.

Koristeći Apache2, PHP 5, MariaDB i phpMyAdmin pretvorili smo Raspberry Pi u server, dok smo softverskim paketima DNSmasq i HostAPD pretvorili Raspberry Pi u bežičnu pristupnu točku.

Sama komunikacija sa mjernim uređajem osmišljena je slojevito u smislu više programskih jezika (Python, PHP i Javascript) koji međusobno razmjenjuju varijable korištenjem virtualne memorije.

Kako bi naš diplomski rad postao proizvodom, spremnim za rad u industriji, implementirali smo autentifikaciju korisnika, te brojne sigurnosne protokole koje rješavaju hakerske napade kao što su SQL injektiranje, otimanje sesija, fizičke i brute force napade.

Time smo stvorili proizvod koji uz sitne preinake može, ne samo rješavati probleme unutar Nacionalnog laboratorija, već može živjeti na Internetu i rješavati probleme svim laboratorijima u toj industriji.

## LITERATURA

- [1] Teo Braun, Izrada web aplikacije za analizu rezultata usporedbenih mjerenja etalona duljine, Završni rad - FSB, Zagreb, 2015.
- [2] B. Runje, Istraživanje mjernih nesigurnosti u postupcima umjeravanja etalona duljine, Zagreb, 2002.
- [3] Mahr Federal Millitron 1240 Korisničke upute, 2001.
- [4] <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [5] MAX3232 tehničke specifikacije
- [6] <https://www.python.org/>
- [7] [http://pyserial.readthedocs.io/en/latest/pyserial\\_api.html](http://pyserial.readthedocs.io/en/latest/pyserial_api.html)
- [8] <https://httpd.apache.org/>
- [9] <https://nginx.org/en/>
- [10] <https://secure.php.net>
- [11] <https://mariadb.org/>
- [12] <https://www.phpmyadmin.net/>
- [13] <http://phpsec.org/projects/guide/>

## RESURSI

- I. Raspbian Jessie Lite - <https://www.raspberrypi.org/downloads/raspbian/>
- II. SD Card Formatter 4.0 - [https://www.sdcard.org/downloads/formatter\\_4/](https://www.sdcard.org/downloads/formatter_4/)
- III. Win32 Disk Imager - <https://sourceforge.net/projects/win32diskimager/>
- IV. Putty - <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- V. Filezilla - <https://filezilla-project.org/download.php?type=client>
- VI. jQuery - <https://jquery.com/>