

**SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE**

DIPLOMSKI RAD

**RAZVOJ SENZORA ZA PREPOZNAVANJE
OTISKA PRSTA**

Stipe Žeravica

Split, rujan 2020.



SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE



Diplomski studij: **Računarstvo**

Oznaka programa: 250

Akadska godina: 2019./2020.

Ime i prezime: **STIPE ŽERAVICA**

Broj indeksa: 732-2018

ZADATAK DIPLOMSKOG RADA

Naslov: **RAZVOJ SENZORA ZA PREPOZNAVANJE OTISKA PRSTA**

Zadatak: Opisati osnovne karakteristike i funkcionalnost Raspberry Pi razvojne platforme. Detaljnije opisati te analizirati karakteristike dostupnih senzora za otisak prsta. Dati pregled osnovnih tehnika i algoritama za prepoznavanje uzoraka koji se koriste za prepoznavanje otiska prsta. Izraditi sustav na Raspberry Pi platformi koji omogućava prepoznavanje otiska prsta temeljen na odabranim algoritmima prepoznavanja uzoraka i odgovarajućoj senzorskoj tehnologiji. Analizirati funkcionalnost i performanse sustava.

Prijava rada: 02.03.2020.

Rok za predaju rada: 18.09.2020.

Rad predan:

Predsjednik
Odbora za diplomski rad:

Mentor:

prof. dr. sc. Sven Gotovac

izv. prof. dr. sc. Mladen Russo

IZJAVA

Ovom izjavom potvrđujem da sam diplomski rad s naslovom RAZVOJ SENZORA ZA PREPOZNAVANJE OTISKA PRSTA pod mentorstvom izv. prof. dr. sc. Mladen Russo pisao samostalno, primijenivši znanja i vještine stečene tijekom studiranja na Fakultetu elektrotehnike, strojarstva i brodogradnje, kao i metodologiju znanstveno-istraživačkog rada, te uz korištenje literature koja je navedena u radu. Spoznaje, stavove, zaključke, teorije i zakonitosti drugih autora koje sam izravno ili parafrazirajući naveo u diplomskom radu citirao sam i povezao s korištenim bibliografskim jedinicama.

Student

Stipe Žeravica

Sadržaj:

1. UVOD	1
2. RASPBERRY PI PLATFORMA.....	2
2.1. Povijesni razvoj.....	2
2.2. Podjela i osnovne karakteristike Raspberry Pi platforme	4
2.3. Funkcionalnosti Raspberry Pi platforme.....	8
2.3.1. GPIO pinovi	8
3. ANALIZA OTISKA PRSTA.....	10
3.1. Osnovni principi analize otiska prsta	10
3.2. Tipovi senzorskih uređaja za analizu otisaka prsta	11
3.2.1. Senzor otiska prsta sa optičkim skenerom	11
3.2.2. Senzor otiska prsta sa kapacitivnim skenerom.....	13
3.2.3. Senzor za analizu otiska prsta sa ultrazvučnim skenerom	14
3.2.4. Ostali skeneri otiska prsta koji se ne koriste u potrošačkoj elektronici.....	15
3.3. Usporedba gotovih sustava za analizu otiska prsta na tržištu	16
3.4. Karakteristike senzora za analizu otisaka prsta.....	16
3.5. Tehnike analize otiska prsta.....	17
3.5.1. Klasifikacija ključnih točaka.....	19
3.5.2. Algoritmi za klasificiranje	21
4. IZRADA SUSTAVA NA RASPBERRY PI PLATFORMI.....	27
4.1. Komponente korištene i potrebna konfiguracija za izradu sustava.....	27
4.1.1. Raspberry Pi 3 Model B+.....	27
4.1.2. Senzor za analizu otiska prsta ZFM-708.....	28
4.1.3. UART - USB pretvarač adapter CP2102	30
4.1.4. LCD zaslon I2C 16x2 (plavi).....	31
4.1.5. Releji modul 5V	33
4.1.6. Mehanički lokot DC 5/6V sa elektromagnetsko-električnim zaključavanjem.....	34

4.1.7.	Eksperimentalna pločica sa fizičkim botunima.....	35
4.2.	Završna shema spoja sustava	35
4.3.	Postavljanje radnog okruženja	36
4.3.1.	Koristeći daljinski razvoj	36
4.3.2.	Konfiguracija Secured Shell (SSH) veze	37
4.3.3.	Konfiguracija VNC-a.....	39
5.	ANALIZA SUSTAVA	40
5.1.	Funkcionalnost sustava	40
5.1.1.	Kontroliranje korisničkog sučelja	42
5.1.2.	Standardna implementacija za analizu otiska prsta.....	44
5.1.3.	OpenCV implementacija za analizu otiska prsta	45
5.2.	Performanse sustava.....	47
5.3.	Moguća rješenja	50
6.	ZAKLJUČAK.....	51
	LITERATURA	52
	POPIS OZNAKA I KRATICA.....	55
	SAŽETAK	58
	SUMMARY	59

1. UVOD

Biometrija kroz zadnje desetljeće postaje sve popularniji alat za verifikaciju autentičnosti jer ima potencijal da bude dramatično brža, lakša i sigurnija od tradicionalnih načina verifikacije. Biometrija je bazirana na konceptu jedinstvenosti nekih ljudskih karakteristika. Jedna od takvih karakteristika je otisak prsta koja je ujedno i najpopularniji način verifikacije identiteta osobe. Ne postoje dvije osobe koje imaju potpuno iste otiske prstiju. Čak i identični blizanci, sa identičnim DNK (engl. *Deoxyribonucleic acid*, DNA), imaju različite otiske prstiju. Ova jedinstvenost omogućuje da se otisci prstiju koriste na razne načine kao što je za biometrijsku sigurnost, identifikaciju osoba kod masovnih katastrofa, za kriminalističke istrage i brojne druge svrhe.

Razvoj informacijske tehnologije omogućila je bolju pristupačnost uređajima koji mogu iskoristiti princip biometrije da bi se izgradio sustav koji se može iskoristiti u sigurnosne svrhe. Jedan od takvih uređaja bi bio senzor za analizu otiska prsta koji uz pomoć jedinstvenosti uzoraka otiska prsta te mogućnošću korištenja nekakvog uređaja koji može komunicirati sa samim senzorom može ostvariti funkcionalnost verificiranja autentičnosti. Ako se za primjer uređaja koji bi mogao komunicirati sa senzorom koristi uređaj iz *Raspberry Pi* platforme, brzo se može uočiti da se ovakav sustav može poprilično jednostavno realizirati.

Ovaj rad se sastoji od šest poglavlja. Nakon uvoda, u drugom poglavlju biti će izložen povijesni razvoj *Raspberry Pi* platforme te će biti definirane osnovne karakteristike, pojmovi, prednosti te funkcionalnosti vezane za korištenje *Raspberry Pi* platforme. Nakon toga u trećem poglavlju istražit će se karakteristike i funkcionalnosti različitih senzora za detekciju otiska prsta sa naglaskom na tehnike i algoritme prepoznavanja uzoraka. Četvrto poglavlje se bavi izradom *Raspberry Pi* sustava skupa sa senzorom za detekciju otiska prsta koji može uspješno izvršiti akciju očitavanja, spremanja i prepoznavanja otiska prsta te eventualno otključavanje mehaničkog lokota. Takav sustav će se onda upotrijebiti da se pokrene implementacija analize otiska prsta uz pomoć tehnika i algoritama koji se koriste za prepoznavanje uzoraka. Peto poglavlje se fokusira na analizu sustava opisujući moguće funkcionalnosti i performanse između gotove solucije i *OpenCV* solucije, a nakon toga dane su ostale potencijalne solucije koje bi se mogle napraviti koristeći srž ovakvog sustava, odnosno koristeći sposobnost komunikacije *Raspberry Pi* uređaja i verifikacije identiteta senzora za analizu otiska prsta. Na kraju rada je dan zaključak.

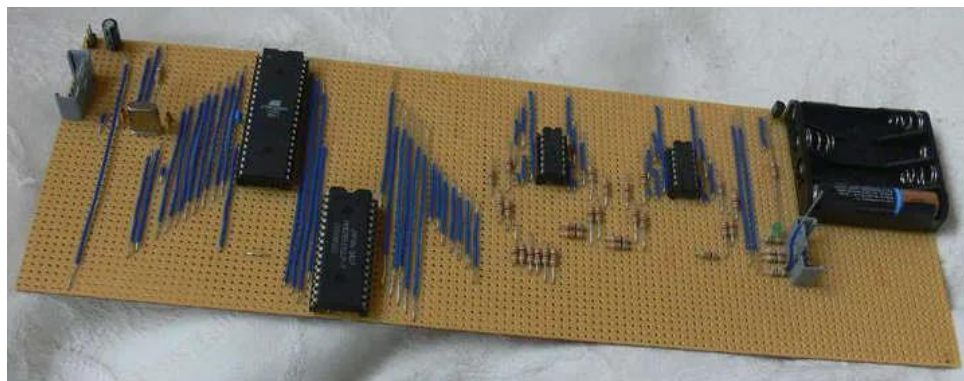
2. RASPBERRY PI PLATFORMA

Raspberry Pi platforma omogućava razna inovativna rješenja koja mogu biti dostupna širokoj skupini korisnika zbog jako pristupačne cijene i fleksibilnosti koja platforma nudi. Upravo zbog tih razloga takva platforma je brzo stekla popularnost a time i još veći izbor mogućih funkcionalnosti korištenjem raznih modula, senzora i uređaja koji su napravljeni da se integriraju na *Raspberry Pi* platformu uz pomoć opće namjenskih ulaznih/izlaznih pinova (engl. *general purpose input/output pins*, GPIO) ili korištenje sličnih uređaja preko univerzalne serijske sabirnice (engl. *Universal Serial Bus*, USB). Popularnost ovakve platforme je uzrokovalo da se razvije kompatibilnost i sa drugim uređajima koji možda početno nisu napravljeni za *Raspberry Pi* platformu, a primjer toga će se moći vidjeti u ovom radu.

U ovom odjeljku biti će popraćen povijesni razvoj *Raspberry Pi* platforme te će biti navedena podjela trenutno dostupnih *Raspberry Pi* uređaja na tržištu zajedno s njihovim karakteristikama i funkcionalnostima koji pružaju korisnicima ovakve platforme. Na kraju će se spomenuti prednosti *Raspberry Pi* platforme u usporedbi sa sličnim solucijama koje postoje trenutno na tržištu.

2.1. Povijesni razvoj

Povijesni razvoj *Raspberry Pi* platforme je započeo 2006. godine u računalnom laboratoriju kod sveučilišta u *Cambridge*-u. Britanski računalni znanstvenik *Ebden Upton*, zajedno s kolegama *Rob Mulins*, *Jack Lang* i *Alan Mycroft* su bili zabrinuti da se nova generacija studenata odvaja od tehničkih aspekata računarstva. To je najvećim dijelom bio rezultat nastavnih planova i programa koji su se više fokusirali na korištenje računala, a ne na njihovo razumijevanje. Pokušavajući nekako riješiti ovaj problem došlo je do formiranja temelja *Raspberry Pi* platforme [1].



Slika 2.1.1 – Primjer prototipa *Raspberry Pi* uređaja baziranog na Atmel ATmega644 mikro kontroleru [2]

Ebden Upton je prvo započeo rad na *Raspberry Pi* prototipu na velikim pločama kao što se može vidjeti na prethodnoj slici 2.1.1. Dizajniranje, testiranje i rješavanje problema je bilo lakše korištenjem velikih pločica ali cilj je na kraju uvijek bio napraviti malo računalo. Tijekom sljedećih šest godina tim je radio na razvoju jeftinog i pristupačnog uređaja koji bi mogao pomoći školama da bolje uče studente o konceptima poput programiranja, te na taj način približio učenicima razumijevanje o principima na kojim računala rade.

Početno komercijalno lansiranje *Raspberry Pi* platforme započelo je u veljači 2012. godine. Od tada, *Raspberry Pi* pločica je prošla kroz niz revizija i razvoj se podijelio na dva modela, a to su *Raspberry Pi* model A i model B. Model A se definirao kao uređaj koji je jeftiniji i jednostavniji dok je model B namijenjen da bude brži, snažniji te uključuje podršku za *Ethernet* vezu.

Ime „*Raspberry Pi*“ je izabrano s „*Raspberry*“ kao oda tradiciji davanja imena ranim računalnim kompanijama po voću, a „*Pi*“ kao referenca na programski jezik *Python* koji se u tom vremenu promovira kao jedno od glavnih programskih jezika na platformi.

Ključni datumi razvoja *Raspberry Pi* platforme [2] [3]:

- 2006: *Ebden Upton* je počeo raditi na prvim konceptima *Raspberry Pi* prototipa
- 2011: Prva alfa i beta izdanja *Raspberry Pi* uređaja tijekom godine
- 2012: Prvih 10 *Raspberry Pi* uređaja prodano na *EBay-u* i službeno lansiranje B modela.
- 2013: Izdanje *Raspberry Pi* Modela A
- 2014: Izdanje *Raspberry Pi* Modela A+ i *Raspberry Pi Compute Module*
- 2015: Izdanje *Raspberry Pi 2* Modela B i *Raspberry Pi Zero*
- 2016: Izdanje *Raspberry Pi 3* Modela B
- 2017: Izdanje *Raspberry Pi Compute Module 3*, *3 Lite* i *Raspberry Pi Zero W*
- 2018: Izdanje *Raspberry Pi 3* Modela B+, A+ i *Zero WH*
- 2019: Izdanje *Raspberry Pi Compute Module 3+*, *3+ Lite* i *Raspberry Pi 4* Modela B
- 2020: Izdanje *Raspberry Pi 4* Modela B sa 8 GB RAM-a

Sa izdavanjem zadnjeg *Raspberry Pi 4* modela službeni OS mijenja svoje ime sa *Raspbian* na *Raspberry Pi OS*, te također službena 64-bitna beta verzija operacijskog sustava postaje dostupna. Službeno broj prodanih *Raspberry Pi* uređaja je došao od 700,000 prve godine do 30 milijuna pri početku 2020 godine sa još ubrzanijim rastom prodaje tijekom 2020. godine zbog potreba računala koja mogu kontrolirati respiratore i 3D printere koji su potrebni tijekom COVID-19 krize [4].

2.2. Podjela i osnovne karakteristike Raspberry Pi platforme

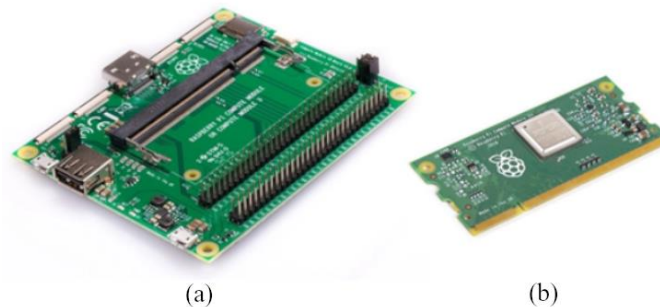
Raspberry Pi platforma je jako napredovala kroz zadnjih 8 godina od začeca u 2012. godine, pa iako je platforma već prošla kroz nekoliko generacija koncept je i dalje ostao isti. Svi modeli platforme koriste *Broadcom* sustav na čipu (engl. *System on a Chip*, SoC) sa integriranom *ARM* kompatibilnom centralom procesnom jedinicom (engl. *Central processing unit*, CPU) i grafičkom procesnom jedinicom (engl. *Graphics processing unit*, GPU).

Ako pogledamo sve uređaje koji su se proizveli kroz taj period onda možemo napraviti jednu glavnu podjelu, a to je podjela na:

- Korisničke uređaje: *Raspberry Pi 1,2,3 i 4 Model A/A+/B/B+*
- Industrijske uređaje: *Compute Module 1,3,3 Lite, 3+ i 3+ Lite*

Može se kazati da se *Raspberry Pi* platforma inicijalno razvila s fokusom na korisničku stranu jer je bio cilj napraviti računalo koje je pristupačno širokoj skupini kupaca za razne svrhe, pogotovo za edukaciju. Ovim načinom razvijanja *Raspberry Pi* platforma je dobila jako veliku popularnost što je uzrokovalo još bržim i većim razvojem same platforme. Malo više od godinu dana nakon izlaska prvog *Raspberry Pi* modula najavljen je prvi računski modul (engl. *compute module*, CM). Glavna razlika između računskog modula i ostalih *Raspberry Pi* modela je to da se CM fokusirao na ugrađene sustave (engl. *Embedded system*), odnosno na industrijske primjene koji nisu trebali razne priključke za zvuk i video te periferiju npr. HDMI, Ethernet, Bluetooth, USB, ...

Ovo je bilo moguće napraviti zbog modularnosti i povećane kompleksnosti koji je računski modul omogućavao, i to tako da je računski modul na sebi sadržavao samo najvažnije komponente kao na primjer CPU, GPU i RAM se modularno uklapao na ulaz/izlaz ploču računskog modula (engl. *Compute module IO Board*, CMIO)



Slika 2.2.1 – (a) CMIO Board; (b) compute module [5]

Na prethodnoj slici se može vidjeti da CMIO ploča koristi sučelje malog obrisnog dualno linijsko memorijskog modula dvostruke brzine prijenosa (engl. *Double data rate 2 small outline dual in-line memory module*, DDR2 SO-DIMM). Prednosti ovakvog načina korištenja sustava omogućava modularnost i fleksibilnost što je jako dobro za industrijske svrhe. Na drugu ruku ako pogledamo *Raspberry Pi* platformu koja je orijentirana prema korisnicima može se primijetiti da iako koriste iste komponente (CPU, RAM, GPU), ostale funkcionalnosti kao HDMI, Ethernet, ... su potrebne kako bi olakšalo korištenje samog uređaja pa time se dovelo do jednostavnije „sve u jednom“ solucije (engl. *all in one solution*).



(a)



(b)

Slika 2.2.2 - Primjer ponude u 2014. god. (a) *Raspberry Pi 1 A+*; (b) *CMIO + CM* [5]

Kako se ide sve dalje i dalje kroz procesni ciklus proizvodnje može se primijetiti kako se *Raspberry Pi* platforma evoluira dok računalni modul ostaje na starijem hardveru, nakon napretka *Raspberry Pi* platforme računski modul se počeo obnavljati sa novijim hardverom od 2017. godine.

U sljedećoj tablici može se vidjeti pregled računskih modula *Raspberry Pi* platforme:

Naziv	Model	Storage	Faktor oblika	GPIO	Lansiran
Compute module 1	1	4 GB eMMC flash memory chip	67.6 × 31 mm	48-pin	2014
Compute module 3	3	4 GB eMMC flash memory chip	67.6 × 31 mm	48-pin	2017
	3 Lite	MicroSDHC	67.6 × 31 mm	48-pin	2017
Compute module 3+	3+	8-32 GB eMMC flash memory chip	67.6 × 31 mm	48-pin	2019
	3+ Lite	MicroSDHC	67.6 × 31 mm	48-pin	2019

Tablica 2.2.1 - Podjela računskih modula kod *Raspberry Pi* platforme [3]

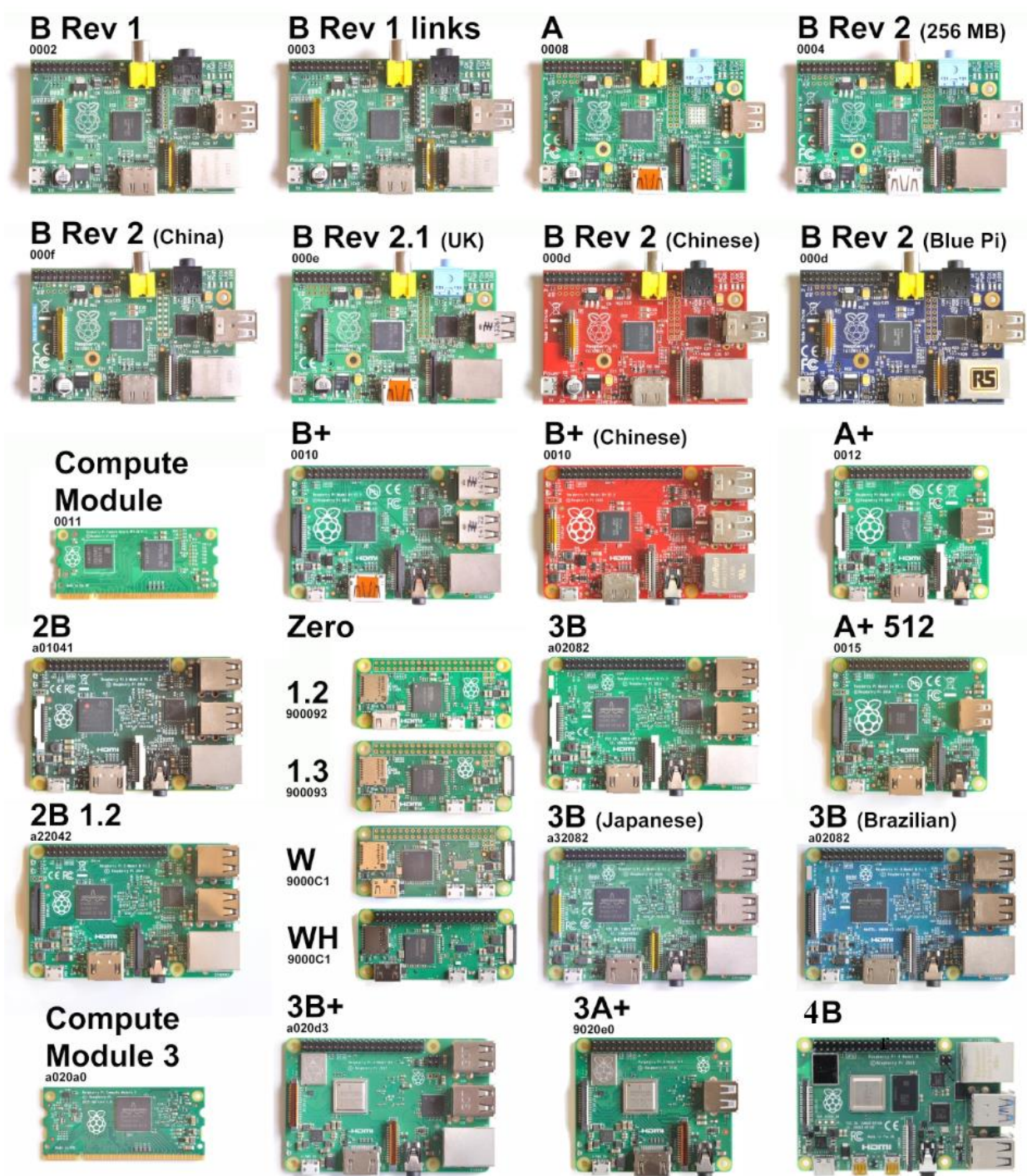
Sami fizički oblik i funkcionalnosti računskih modula se nije promijenio značajno od najave prvog računskog modula, bitne komponente koje su se promijenile su CPU, GPU, RAM i skladištenje podataka (engl. *storage*), što je u 2017. god. dovelo računski modul u razinu sa *Raspberry Pi 3* model 3/3+ pločicom. Jedina razlika između *Lite* i *non-Lite* verzije računskog modula je da umjesto *micro-SD* kartice se koristi *embedded storage*, odnosno čip na samoj pločici.

U sljedećoj tablici može se vidjeti pregled trenutnih modela *Raspberry Pi* platforme:

Naziv	Model	Faktor oblika	Ethernet	Bežičnost	GPIO	Lansiran	Cijena
Raspberry Pi	B	85.6 × 56.5 mm	Da	Ne	26-pin	2012	\$35
	A	85.6 × 56.5 mm	Ne	Ne	26-pin	2013	\$25
	B+	85.6 × 56.5 mm	Da	Ne	40-pin	2014	\$35
	A+	65 × 56.5 mm	Ne	Ne	40-pin	2014	\$20
Raspberry Pi 2	B	85.6 × 56.5 mm	Da	Ne	40-pin	2015	\$35
Raspberry Pi Zero	Zero	65 × 30 mm	Ne	Ne	40-pin	2015	\$5
	WWH	65 × 30 mm	Ne	Da	40-pin	2017	\$10
Raspberry Pi 3	B	85.6 × 56.5 mm	Da	Da	40-pin	2016	\$35
	A+	65 × 56.5 mm	Ne	Da	40-pin	2018	\$25
	B+	85.6 × 56.5 mm	Da	Da	40-pin	2018	\$35
Raspberry Pi 4	B (1 GB)	85.6 × 56.5 mm	Da	Da	40-pin	2019	\$35
	B (2 GB)	85.6 × 56.5 mm	Da	Da	40-pin	2019	\$35-45
	B (4 GB)	85.6 × 56.5 mm	Da	Da	40-pin	2019	\$55
	B (8 GB)	85.6 × 56.5 mm	Da	Da	40-pin	2020	\$75

Tablica 2.2.2 - Podjela *Raspberry Pi* platforme [3]

Korisnička strana *Raspberry Pi* platforme fokusirala se na razvoj funkcionalnosti i povećanje efikasnosti komponenata korištenih u uređaju. Razvijanjem novih modela vidimo kako se cijena nije znatno povećala kroz prvih par godina, dok u nekim slučajevima se znatno smanjila kao na primjer *Raspberry Pi Zero* modeli koji su vrlo efikasni uređaji za cijenu koju se mogu nabaviti. Ako pogledamo kroz zadnjih par godina, poboljšanje raznih funkcionalnosti kao na primjer novi *bluetooth* i WiFi moduli, te noviji procesori imaju posljedicu povećanja cijene samog uređaja i povećanje tehnoloških zahtjeva, a to se može vidjeti sa novijim modelom. Najnoviji model *Raspberry Pi 4* Model B je jako impresivan ako uzmemo u obzir količinu RAM-a koji se može staviti na pločicu, ali povećanje specifikacija nije došlo bez svojih nedostataka, a to je toplina. Maksimalna temperatura prije nego što se CPU započne usporavati je 85 stupnjeva, a najnoviji model *Raspberry Pi* uređaja je testiran za graničnu radnu toplinu i iznosi oko ~75 stupnjeva. Ovo znači da će u budućnosti, ako se nešto ne poduzme u vezi ovog potencijalnog problema, noviji *Raspberry Pi* modeli će morati početi dolaziti sa instaliranim hladnjakom na pločici uređaja.



Slika 2.2.3 – Primjer različitih modela Raspberry Pi platforme [6]

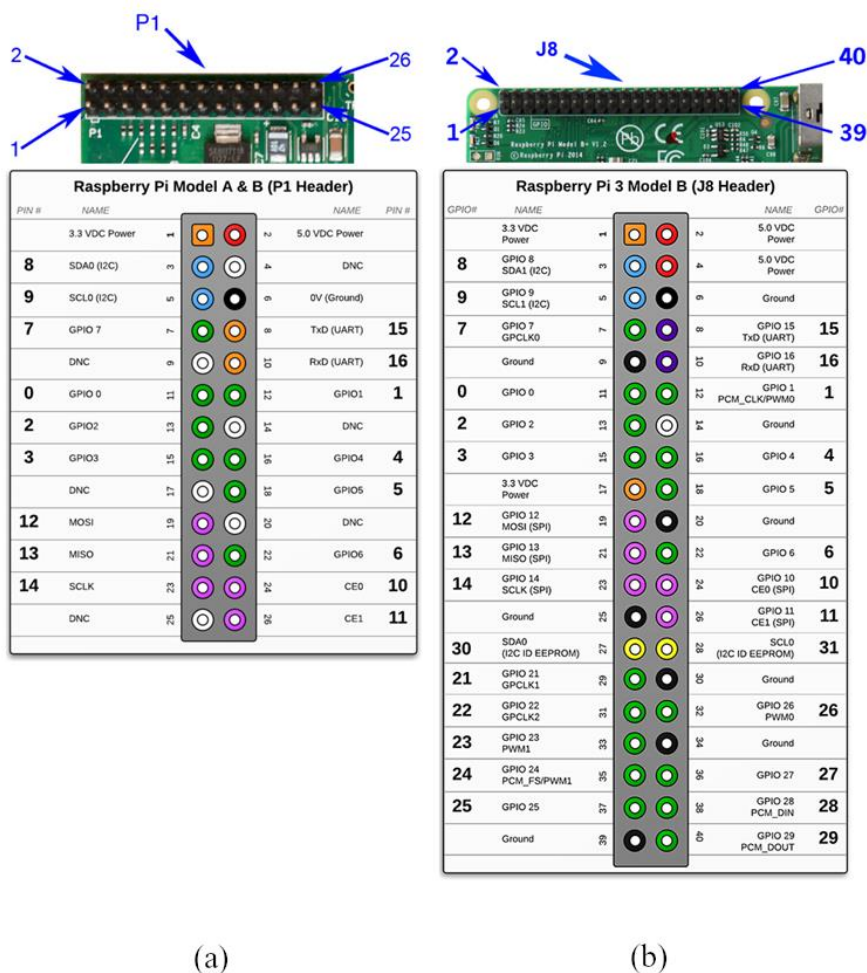
Primjer *Raspberry Pi* uređaja korištenog u ovom radu se može vidjeti na slici 2.2.3. Ovaj rad se fokusira na izgradnju i analizu sigurnosnog sustava na *Raspberry Pi 3 Model B+* uređaju koji je orijentiran prema generalnim korisnicima. Proći će se kroz više detalja o načinu korištenja *Raspberry Pi* uređaja i njegovih karakteristika u jednom od sljedećih poglavlja.

2.3. Funkcionalnosti Raspberry Pi platforme

Raspberry Pi platforma nije samo minijaturno prijenosno računalo koje se može koristiti za razne edukacijske svrhe, sama platforma nudi znatno specijalne i korisne funkcionalnosti kao na primjer mogućnost da ovisno o nekim ulaznim signalima mijenja stanje te šalje izlazne signale koji se mogu iskoristiti za jako mnogo situacija. Takva funkcionalnost je upravo iskorištena u ovom radu, a to je sve moguće zbog načina na kojem je *Raspberry Pi* platforma napravljena.

2.3.1. GPIO pinovi

Opće namjenski ulazni/izlazni (engl. *General Purpose Input/Output*, GPIO) pinovi se nalazu na djelu ploče gdje se nalazi zaglavlje za proširenje. Broj GPIO pinova je narastao od prve *Raspberry Pi* generacije, u prvoj generaciji broj pinova je iznosio 26, dok u trećoj generaciji, broj pinova iznosi 40.



Slika 2.3.1 – Usporedba Raspberry Pi GPIO pinova (a) RP 1 Model B; (b) RP 3 Model B+ [7]

Fizički GPIO pinovi su prikazani na slici 2.3.1. U ovom slučaju vidimo da sa lijeve strane imamo *Raspberry Pi 1 Model B* ploču koja sadrži jedno 26-pinsko zaglavlje za proširenje sa oznakom 'P1' koje pruža pristup prema 17 GPIO pinova. Na desnoj strani imamo *Raspberry Pi 3B+* ploču koja sadrži jedno 40-pinsko (2x20) zaglavlje za proširenje s oznakom 'J8'. Prvi 26 pinova su isti kao i na 'P1' zaglavlju koji je prisutan na ranijim modelima *Raspberry Pi* uređajima s tim da 'J8' zaglavlje pruža 14 dodatnih pinova koji osiguravaju dodatne GPIO pinove te pinove za uzemljenje. Svaki GPIO pin ima svoju svrhu i može se koristiti za specifične svrhe koje je namijenjen.

Kao što je prikazano na *Raspberry Pi 3* modelu B na slici 2.3.1 može se primijetiti da se pinovi 1, 2 i 4 koriste za izvor napajanja, pinovi 6, 9, 14, 20, 25, 30, 34 i 39 koriste za uzemljenje a ostali pinovi se mogu iskoristiti za primanje i slanje raznih signala pomoću promjene napona i toka struje kroz same pinove.

Pin 1 je određen za 3.3 volti istosmjerne struje (engl. *Volts Direct Current*, VDC) a pinovi 2 i 4 su određeni za 5 VDC što omogućuje napajanje spojenog strujnog kruga direktno preko napajanja samog *Raspberry Pi* uređaja. Strujni krug se može zatvoriti tako da se drugi kraj strujnog kruga spoji na jednom od pinova za uzemljenje. Korištenjem ovih dvaju vrsta pinova možemo imati osnovni strujni krug koji može napajati sve što spojimo na njega s obzirom da se poštuju VDC granice. Problem sada nastaje da iako postoji funkcionalni strujni krug, ne postoji način na koji se može koristiti neka vrsta sklopke da palimo ili gasimo strujni krug, a tu upravo dolazimo do treće vrste pinova.

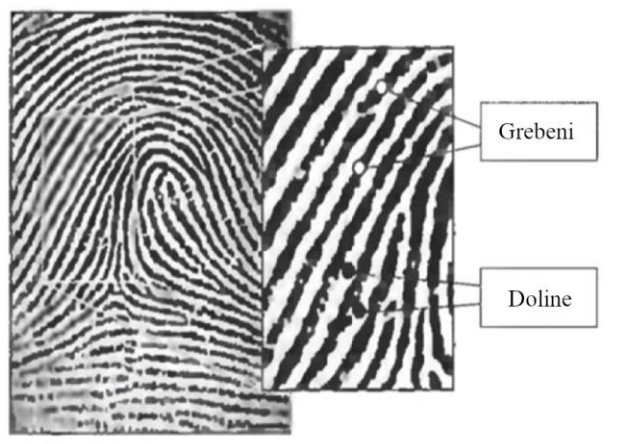
Treća vrsta pinova ima mogućnost mijenjanja napona i time i struje koje prolaze kroz sami pin pa tako daje mogućnost slanja pulsova, odnosno programiranje pinova da vrši interakciju sa raznim komponentama koji mogu biti spojeni u strujni krug. Razine napona koji ovi pinovi mogu podnijeti je 3.3 V i ne podnose 5 V, također ne postoji nikakva zaštita na *Raspberry Pi* ploči koja regulira granice napona na pinovima jer je namjera da osobe koje su zainteresirane za korištenje ovih pinova koriste vanjsku ploču s kojom mogu regulirati napon i koristiti kabele za lako spajanje na pinove umjesto lemljenja izravno na glavnu ploču. Više o ovoj temi će se raspravljati kroz rad u djelu gdje će se fokusirati na izradu praktičkog dijela rada.

3. ANALIZA OTISKA PRSTA

Danas je normalna pomisao potvrditi svoj identitet preko svog otiska prsta, kao na primjer otključavati svoj mobitel. Ne samo normalno nego skoro uzeto zdravo za gotovo. Napredovanje tehnologije je dozvolilo mogućnost korištenja egzotičnih tehnologija kao na primjer senzora za analizu otiska prsta u svakodnevnim uređajima koji koristimo. Kroz zadnje desetljeće ovakvi uređaji koji u sebi sadrže senzore za analizu otisaka prsta su se proširili posvuda, – u policijskim stanicama, zgradama koje zahtijevaju veliku sigurnost, USB-ovima, pametnim mobitelima itd... U ovom djelu proći će se kroz osnove analize otiska prsta, tipove i karakteristike senzora koji obavljaju tu funkcionalnost te tehnike i algoritme koji se koriste za analizu otiske prstiju.

3.1. Osnovni principi analize otiska prsta

Otisci prstiju su jedan od onih bizarnih posljedica prirode. Ljudska bića imaju ugrađene unikatne karakteristike koje možemo smatrati kao lako dostupne osobne iskaznice na svome tijelu a to su kao na primjer maleni rubovi kože na otiscima prstiju kao što se može vidjeti na slici 3.1.1. Uzorak grebena i doline na prstima olakšava držanje stvari na isti način kao i gumeni dizajn sa grebenima i dolinama pomaže gumama od automobila da bolje prijanjaju uz cestu [8].



Slika 3.1.1 – Primjer grebena i dolina na otisku prsta [9]

Druga funkcija otiska prstiju je potpuna slučajnost. Kao i sve u ljudskom tijelu i ovi se grebeni formiraju kombinacijom genetskih i korisnih čimbenika. Genetski kod daje opća naređenja o načinu na koji bi se koža trebala oblikovati u plod koji se razvija, ali specifičan način koji nastaje rezultat je nasumičnih događaja. Točan položaj fetusa u maternici u određenom trenutku i točan sastav i gustoća okoline, te razne tekućine odlučuju kako će se formirati svaki pojedinačni greben.

Dakle, uz bezbroj stvari koje prije svega odlučuju o genetskom sastavu, na formiranje prstiju utječu nebrojni čimbenici okoliša. Baš kao i vremenski uvjeti koji tvore oblake ili obalu neke plaže, čitav razvojni proces je toliko kaotičan da u čitavoj ljudskoj povijesti gotovo da i nema mogućnosti da se dvaput pojavi isti točan obrazac prsta. Prema tome otisci prstiju su jedinstveni marker za osobu, čak i kod identičnih blizanaca, dok dva otiska od svakog blizanca na prvi pogled mogu izgledati isto, iskusni detektivi i napredni softveri mogu otkriti jasne i definirane razlike.

To je osnovna ideja analize otiska prsta, kako u istrazi počinjenog zločina, tako i u sigurnosti. Zadatak senzora otiska prsta je zauzeti mjesto ljudskog analitičara, prikupljanjem uzorka otiska prsta i uspoređivanjem s drugim zabilježenim uzorcima.

3.2. Tipovi senzorskih uređaja za analizu otisaka prsta

Postoje 3 popularna načina na koji se izrađuju senzori za analizu otiska prstiju, svaka sa svojim prednostima i nedostacima. Svaki senzorski uređaj za analizu otiska prsta koristi neku vrstu skenera da uspješno očita otisak prsta kako bi senzor izvršio svoju funkcionalnost.

Podjela senzorskih uređaja za analizu otiska prstiju prema korištenju skenera su:

- Senzor za analizu otiska prsta sa optičkim skenerom
- Senzor za analizu otiska prsta sa kapacitivnim skenerom
- Senzor za analizu otiska prsta sa ultrazvučnim skenerom

Princip rada ostaje isti bez obzira koja vrsta skenera se koristi, bitne stvari što se mijenjaju korištenjem različitih vrsta skenera su količina prostora koji senzor zauzima, način korištenja senzora odnosno limitacija korištenja određenih materijala te brzina izvršavanja funkcionalnosti.

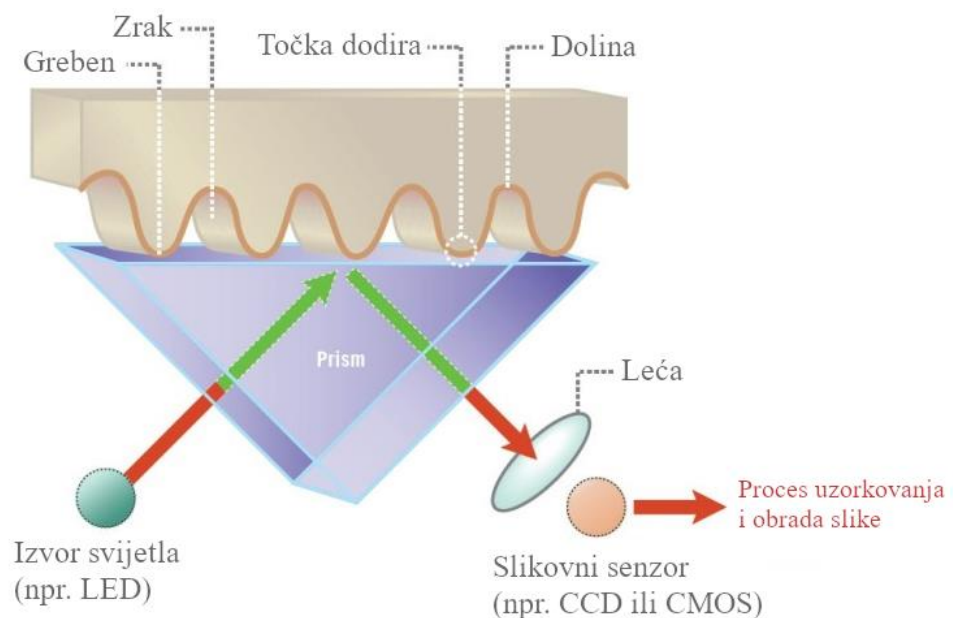
3.2.1. Senzor otiska prsta sa optičkim skenerom

Optički skener je prvi i najstariji način snimanja i usporedbe otisaka prstiju, a to se može lako uočiti jer je način na kojem izvršava svoju funkcionalnost relativno jednostavan za zamisliti i shvatiti. Kao što ime sugerira, ova tehnika se oslanja na snimanje optičke slike otiska prsta tako da koristi nekakav izvor svjetlosti, a to može biti niz dioda koje emitiraju svjetlo (engl. *light emitting diode*, LED) da osvijetli površinu na kojoj bi se stavio prst kako bi se pokrenuo proces snimanja otiska prsta.

Uzimajući refleksiju svjetla od prethodno spomenute površine i pravilnim pozicioniranjem slikovnog senzora sa određenom razlučivosti dobije se dvodimenzionalna slika otiska prsta. Što je razlučivost veća to su finiji detalji koje senzor može razaznati o otisku prsta, tako da povećanjem razlučivosti senzora omogućavamo veću sigurnost.

Važno je spomenuti također da slikovni senzori koji se koriste za ovakve svrhe nisu obični slikovni senzori koji se kao na primjer koriste u običnim aparatima, dok se slikovni senzori u običnim aparatima fokusiraju da obuhvate što veću količinu vidljivog spektra boja, slikovni senzori koji se koriste u sensorima za otiske prstiju rade u području sa sivim tonovima (engl. *greyscale*) jer boja u ovom slučaju nije vrlo bitna dok je kontrast između svijetlog i tamnog područja jako bitna. Ovi slikovni senzori bilježe mnogo veće kontraste u slikama od običnih fotoaparata.

Kad se prst stavi iznad skenera razina svjetla je jako mala pa optički skeneri obično sadrže niz LED dioda kao bljeskalice za osvjetljivanje površine kako bi se bolje uhvatili detalji izbliza. Takav dizajn zna biti poprilično velik što je jedan od nedostataka korištenja ovakve vrste senzora za analizu otiska prsta, pogotovo ako se namjerava koristiti u pametnim mobitelima (engl. *smartphone*).



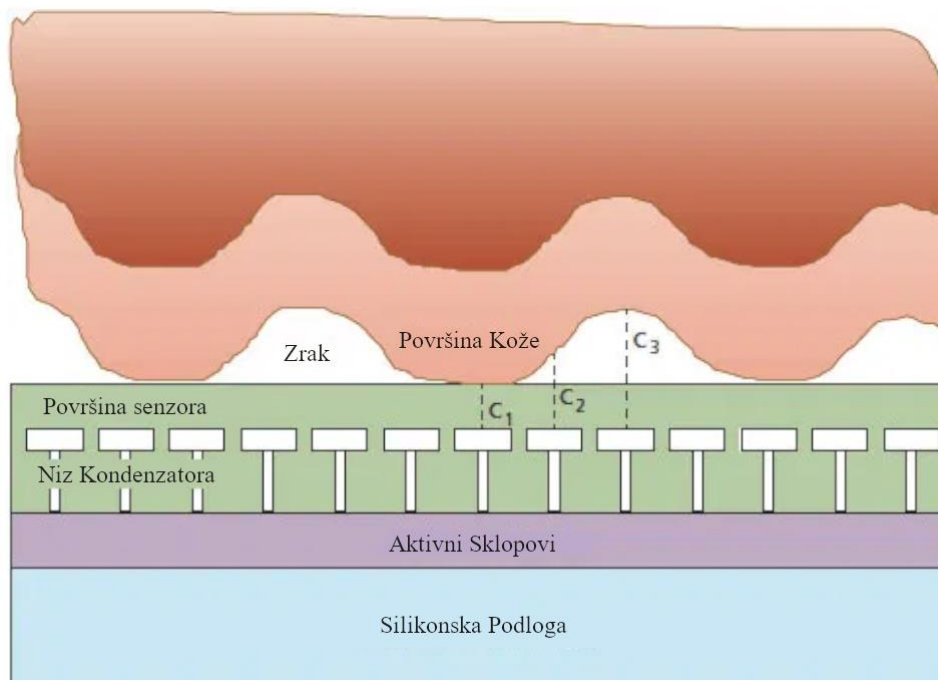
Slika 3.2.1 - Primjer rada senzora za analizu otiska prsta sa optičkim skenerom [10]

Kao što se može vidjeti na slici 3.2.1 proces dobivanja dvodimenzionalne slike je poprilično jednostavan ali sami rezultati mogu varirati po kvaliteti zavisno o uvjetima kada je otisak prsta bio snimljen. Zato postoji dodatan proces koji uzme dobivenu sliku od slikovnog senzora na kojoj se onda dodatno obrađuje slika kako bi se makli svi bez-potrebni podatci koji nisu korisni. Uz pomoć algoritama se ističe kontrast između grebena i dolina kako bi se dobili što bolji uvjeti za provjeru otiska prsta. Detaljniji opis rada principa i tehnika prepoznavanja ključnih elemenata kod otiska prsta će se odraditi u poglavlju pregleda tehnika analize otiska prsta.

Odnosno možemo kazati senzor za analizu otiska prsta sa optičkim skenerom izvršava svoju funkcionalnost uz pomoć osnove fotografije, korištenjem algoritama za otkrivanje jedinstvenih uzoraka na površini poput grebena i dolina ili jedinstvenih tragova i analizom najsvjetlijih i najmračnijih područja slika.

3.2.2. Senzor otiska prsta sa kapacitivnim skenerom

Kapacitivni skener je trenutno jedan od najpopularnijih i najkorištenijih vrsta skenera za spremanje i obradu otiska prstiju. Ovaj tip skenera se danas može pronaći u većini popularnih pametnih mobitela (engl. *smartphone*) jer je najsigurniji i relativno malih dimenzija.



Slika 3.2.2 – Primjer rada senzora za analizu otiska prsta sa kapacitivnim skenerom [11]

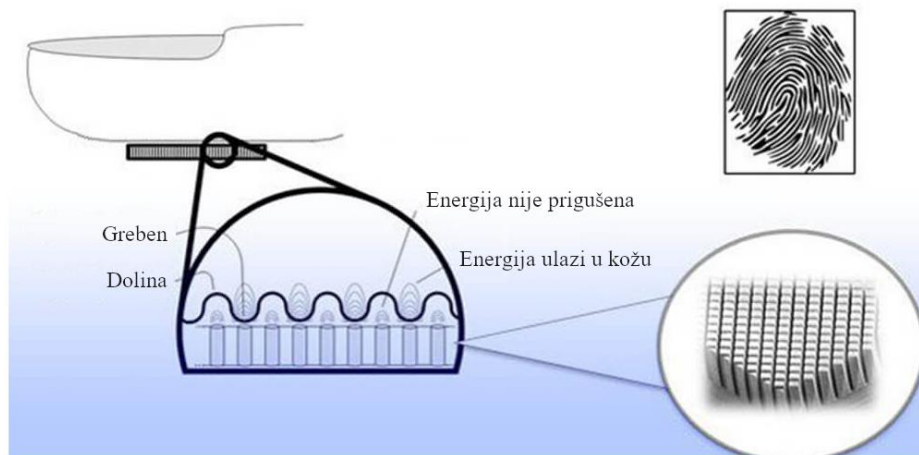
Umjesto stvaranja tradicionalne dvodimenzionalne slike otiska prsta, kapacitivni skener otiska prsta koristi niz malenih kondenzatora za prikupljanje podataka o otisku prsta.

Budući da kondenzatori mogu pohraniti električni naboj, njihovo spajanje na jako malene pločice na površini senzora omogućuje sastavljanje detaljne slike otiska prsta. Naboj koji se pohrani u kondenzatoru će se malo promijeniti kada se na primjer greben na samom otisku prsta pozicionira preko vodljivih pločica na površini senzora, dok se u slučaju doline, naboj na kondenzatoru na toj poziciji ostane relativno nepromijenjen. Korištenjem te informacije zajedno sa analogno-digitalnim pretvaračem se može dobiti slika o otisku prsta koja se može referencirati kada se pokuša usporediti sljedeći otisak prsta sa otiskom prsta koji je prethodno spremljen.

Senzor za analizu otiska prsta koji koristi kapacitivni skener je vrlo robustan jer djeluje samo s kožom. Ne može se prevariti sa lakoćom kao na primjer sa komadom papira ili slike otiska prsta jer skener zahtjeva materijal sa istim oblikom otiska prsta i provodljivosti kao i koža, što je nije nemoguće ali je jako teško za ostvariti. Više o prednostima i manama kapacitivnih skenera će se spomenuti u poglavlju karakteristika senzora [8].

3.2.3. Senzor za analizu otiska prsta sa ultrazvučnim skenerom

Ultrazvučni skeneri su najnoviji dodatak u priči senzora za analizu otiska prstiju. Ova vrsta skenera je sličnih dimenzija kapacitivnih skenera s tim da je osmišljena kako bi riješila problem korištenja senzora za analizu otiska prsta na malo većoj udaljenosti i mogućnošću dohvaćanja otiska prsta kroz materijale kao na primjer staklo što omogućava vrlo zanimljiva rješenja u mobilnoj industriji.



Slika 3.2.3 - Primjer rada senzora za analizu otiska prsta sa ultrazvučnim skenerom [12]

Da bi se uhvatili detalji otiska prsta, hardver se sastoji od ultrazvučnog odašiljača i prijemnika. Ultrazvučni puls se šalje u smjeru otiska prsta koji se postavlja preko skenera. Dio ovog pulsa se apsorbira, a dio se vraća na senzor, ovisno o poziciji grebena, dolina i ostalih detalja koji su jedinstveni za svaki prst. Promatrajući povratni signal uz pomoć senzora koji može detektirati mehanički napon potreban za izračunavanje intenziteta povratnog ultrazvučnog pulsa u različitim točkama skenera kroz neki period vremena omogućava snimanje dodatnih podataka o dubini, što rezultira vrlo detaljnom 3D reprodukcijom skeniranog otiska prsta.

U slici 3.2.3 može se primijetiti kako ultrazvučni valovi ulaze kroz grebene otiska prsta prije nego što se reflektiraju natrag tako da ova tehnologija ima mogućnost raspoznavanja između pravog i lažnog otiska prsta.

Očitavanje otiska prsta u 3D obliku čini ovom tehnologijom još sigurnijom ali dolazi sa cijenom da proces izvršavanja skupljanja otiska prsta te zatim proces usporedbe postojećeg otiska prsta čini sporijom od dosad spomenutih rješenja.

3.2.4. Ostali skeneri otiska prsta koji se ne koriste u potrošačkoj elektronici

Osim 3 popularnih vrsta skenera koja već imaju široku primjenu u svijetu ima još par interesantnih solucija koja nisu vidjela široku primjenu ali ih je vrijedno za spomenuti a to su:

- Senzor za analizu otiska prsta sa termičkim skenerom
- Senzor za analizu otiska prsta sa skenerom pritiska
- Senzor za analizu otiska prsta sa radiofrekvencijom (RF)

Termički skeneri otiska prsta djeluju na sličan način kao i kapacitivni skeneri ali umjesto da mjere električni naboj, termički skener mjeri temperaturnu razliku različitih pozicija na površini senzora. Grebeni otiska prsta stvaraju više topline od dolina i na taj način stvore 2D sliku za analizu.

Skeneri za otiske prsta pomoću pritiska bilježe razliku u tlaku uzrokovanom grebenima i dolinama otiska prsta. Kad se govori o ovakvom skeneru, razlike koje se moraju očitati su jako male tako da je ovaj način bilježenja otiska prsta izuzetno osjetljiv. Loša strana je da će se svaki sloj materijala koji se stavi na vrh senzora kao na primjer staklo, utjecati na osjetljivost skenera, ali ako se ne zaštiti, može doći do oštete samog senzora jer bi mogao biti izložen udarcima i ogrebotinama.

Senzori otiska prsta s radiofrekvencijom (engl. *radiofrequency*, RF) imaju jako malene antene za svaki piksel gotove 2D slike otiska prsta. Ovi senzori mogu proći kroz kožu i očitati otisak prsta ispod kože što pruža više podataka i mogu se izbjeći greške kad se radi sa oštećenim otiskom prsta.

3.3. Usporedba gotovih sustava za analizu otiska prsta na tržištu

Na tržištu ima poprilično dosta gotovih uređaja koji su bazirani na senzorima za analizu otiska prsta, a neki od tih primjera su sljedeći [13]: *Fingerprint SAMSUNG SHS-H700 (400\$+)*, *ULTRALOG U-Bolt Pro Smart Lock (200\$+)*, *Fingerprint Door Lock, Ardwolf A60 (200\$+)*, *TimeMoto TM-626 (400\$+)* itd...

Kako je vrijeme prolazilo, mogućnost kupovanje senzora za analizu otisaka prsta je postalo još pristupačnije, pogotovo sa porastom popularnosti *Raspberry Pi* i *Arduino* platforme. Imaju mogućnost prilagodbe po korisniku jer daje slobodu da korisnik radi što hoće. Može biti malo kompliciranije za konfigurirati ali dosta jeftinije od gotove solucije.

Neki od primjeraka su: *iHaospace AS608 (15\$)*, *R307 Optical Fingerprint Module (15\$)*, *Vaorwne AS608 (9\$)*, *Capacitive fingerprint reader Pzsmocn (44\$)*, *DollaTek Green Light Optical Fingerprint Reader Sensor Module (15\$)*, *Fingerprint Sensor ZFM-708 (40\$+)* te brojni drugi ...

3.4. Karakteristike senzora za analizu otisaka prsta

Do sada smo prošli kroz razne vrste senzora za analizu otiska prsta koji koriste različite skenere da izvrše svoju funkcionalnost. Kao što se moglo primijetiti kroz prijašnja poglavlja, senzori za analizu otiska prsta se baziraju na istim karakteristikama na kojim se bazira bilo koja biometrijska tehnika, a to je ljudsko tijelo. Na ljudskom tijelu se nalazu razni primjeri koji se mogu koristiti za definiranje osobnog identiteta kao na primjer: očna šarenica, otisak prsta, vokalni govor, ljudsko lice... Svaki ovaj primjer nastaje zbog kaotičnosti razvoja ljudskog tijela u ranim fazama začeca.

Kako imamo specijalne karakteristike koje nas definiraju tako su napravljeni uređaji koji se baziraju na njima koji mogu olakšati izvršavanje procesa identifikacije u relativnom brzom vremenu.

Ako uzmemo kao primjer senzor za analizu otiska prsta sa optičkim skenerom, može se kazati da ima dosta prednosti zbog jednostavnosti principa rada senzora, ali ima dosta i nedostataka. Jedan od većih nedostataka je veličina cijelog senzorskog sklopa kada se uzmu u obzir sve komponente koje su zahtjevne da senzor radi na takav način, te lošija sigurnost kad se uzme u obzir da se može

replicirati još jedna 2D slika nečijeg otiska prsta te time „prevariti“ senzor tako da uzme sliku kao otisak prsta umjesto pravog otiska prsta.

Kod senzora za analizu otiska prsta sa kapacitivnim skenerom dobivaju se korisne prednosti. Korištenjem sitnijih kondenzatora izbjegne se situacija sa korištenjem komponenti koji zahtijevaju puno više mjesta. U današnjim tehnologijama sve se više i više pokušava smanjiti veličina uređaja kako bi se što više korisnih funkcionalnosti moglo staviti u istu količinu prostora. Ova vrsta skenera i dalje gradi 2D sliku otiska prsta kao i optički skener tako da generalno ima sličnu brzinu izvršavanja ali neke od loših stvari kod kapacitivnih skenera je da ovise o električnom naboju, tako da ako je površina kože na otisku prsta mokra ili prljava skener će imati poteškoća konstruirati pravilnu sliku otiska prsta te time uzrokovati da se otisak prsta ne provjeri uspješno. Još jedna poteškoća kod senzora sa kapacitivnim skenerima je da rade samo na određenim materijalima. Ako se pokuša staviti senzor na materijal koji ne provodi električki naboj onda rad sa ovakvim senzorima je nemoguće.

Problem sa materijalima rješava se uz pomoć novog senzora sa ultrazvučnim skenerom, ova vrsta skenera ima mogućnost da prolazi kroz pojedine materijale kao na primjer staklo jer ne koristi električni naboj da sagradi sliku otiska prsta već slanjem ultrazvučnih valova ima mogućnost da prolazi kroz materijale koji se često koriste u potrošačkoj industriji. Uz mogućnost korištenja ove vrste skenera također zauzima jako malo prostora kao i kapacitivni senzor što je još jedna dobra karakteristika ovakvog senzora. Jedina loša strana ovakvog skenera je lošija performansa izvršavanja funkcionalnosti analize otiska prsta pošto senzor ne gradi 2D sliku otiska prsta već 3D sliku. Ovom mogućnošću senzor ima puno više podataka s koje mora obrađivati. Iako rad povećanim količinama podataka povećava samu sigurnost procesa analize otiska prsta. Razvijanje tehnika ove vrste senzora za analizu otiska prsta još se i dalje odvija tako da se proces optimizira kako bi se stekla bolja performansa sustava.

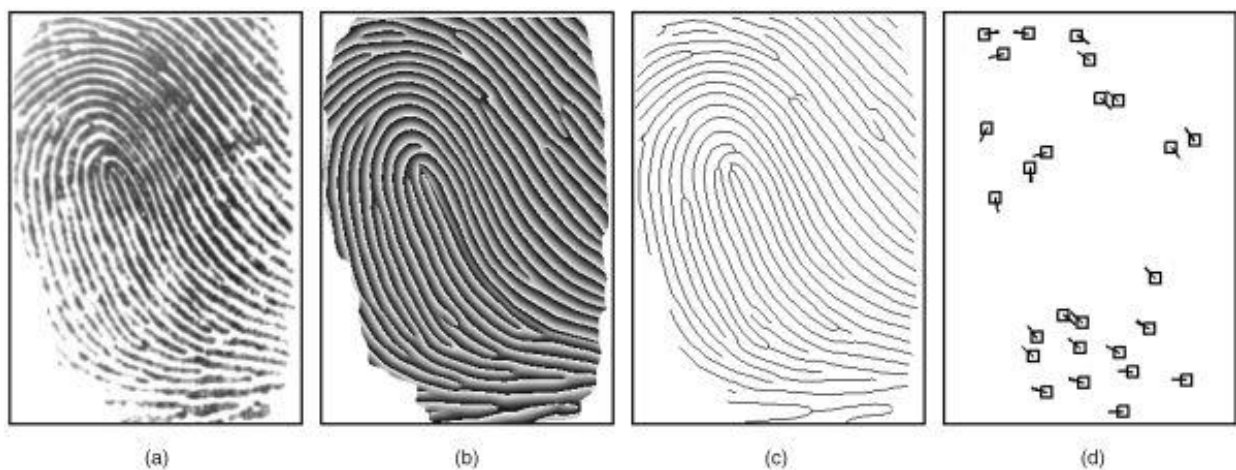
3.5. Tehnike analize otiska prsta

Kao što se dosada moglo vidjeti otisak prsta je uzorak grebena, dolina i pora na površini vrha prsta te svaki pojedinac ima jedinstvene oblike otiska prstiju. Analiza otiska prsta se može vršiti na razne načine koristeći razne tehnike. Jedan od tih načina može biti koristeći tehnike umjetne inteligencije (engl. *artificial intelligence*, AI) kao na primjer: strojno učenje (engl. *machine learning*),

neuronske mreže (engl. *neural networks*), duboko učenje (engl. *deep learning*) itd. Sve ove tehnike koriste principe izvlačenja ključnih elemenata unutar otiska prsta koji čini svaki otisak posebnim.

Ako se krene sa veće slike može se kazati da za analizu i uspješnu prepoznatljivost otiska prsta je dovoljno imati mogućnost prepoznavanja jeli trenutni otisak prsta koji se pokušava prepoznati isti kao i otisak prsta koji je bio prethodno spremljen. Ovaj način razmišljanja dosta simplificira stvari.

Većina sustava za usklađivanje otisaka prstiju temelji se na četiri vrste shema reprezentacije otiska prstiju kao što se može vidjeti na slici 3.5.1. a ti sustavi su: siva slika (engl. *greyscale*), fazna slika (engl. *phase image*), kosturna slika (engl. *skeleton image*), i ključne točke (engl. *minutiae*). Zbog svoje prepoznatljivosti, kompaktnosti i kompatibilnosti s karakteristikama koje koriste stručnjaci za ljudske otiske, reprezentacija na temelju detalja (engl. *minutiae*) postala je najčešće prihvaćena shema predstavljanja otiska prstiju [14].



Slika 3.5.1 - Sheme predstavljanja otiska prstiju (a) siva slika, (b) fazna slika, (c) kosturna slika, (d) minutiae [14]

Raširena upotreba sustava za prepoznavanje otisaka prstiju u raznim aplikacijama uzrokovala je zabrinutost da spremljeni uzorci otisaka prstiju mogu biti kompromitirani pa time se mogu upotrijebiti za izradu lažnih otisaka prstiju, koji bi se zatim mogli upotrijebiti za prevaru svih ostalih sustava koji koriste otiske kompromitirane osobe. Od prije navedenih shema najviši rizik predstavlja „siva slika“ (engl. *greyscale*), a propust fazne slike (engl. *phase image*) ili kosturne slike (engl. *skeleton image*) također je opasno jer je trivijalno rekonstruirati sliku otisaka sive slike s fazne slike ili kosturne slike koja se zatim može koristiti za proboj sigurnosti na svim uređajima sa tim specifičnim otiskom prsta.

Za razliku od gore navedena tri prikaza, kad se radi sa principom vađenja detalja otiska prsta, odnosno ključnih točaka (engl. *minutiae*), slika prsta se ne sprema na uređaj nego samo te ključne točke. Ovo znatno povećava sigurnost jer u slučaju da dođe do propusnosti kao na primjer, ključne točke otiska prsta postanu kompromitirane, smatra se manje ozbiljnim jer nije trivijalno rekonstruirati sliku u sivim tonovima iz ključnih točaka (engl. *minutiae*).

3.5.1. Klasifikacija ključnih točaka

Otisak prsta može imati jako puno različitih vrsta ključnih točaka (engl. *minutiae*). Popisano je osamnaest različitih vrsta obilježja otisaka prstiju. Nadalje, identificirano je 150 različitih lokalnih svojstava grebena (kratki grebeni, ograđeni prostor ili ograda, itd...) [14].

Jedinstvenost otiska prsta se može isključivo određivati sa lokalnim karakteristikama grebena i dolina skupa sa njihovim odnosima. Kao na primjer grebeni i doline u otisku prsta se izmjenjuju gibajući se u lokalnom konstantom smjeru.

Dvije najistaknutije karakteristike lokalnog grebena su:

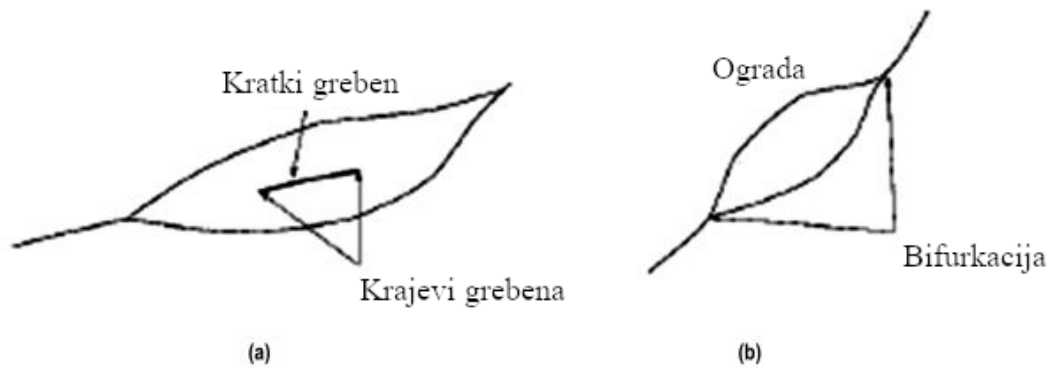
- Završetak grebena (točka na kojoj greben naglo završava)
- Bifurkacija grebena (točka gdje se greben grana ili razilazi u druge smjerove)



Slika 3.5.2 – Dvije najčešće korištene značajke otiska prsta (a) bifurkacija grebena; (b) završetak grebena [14]

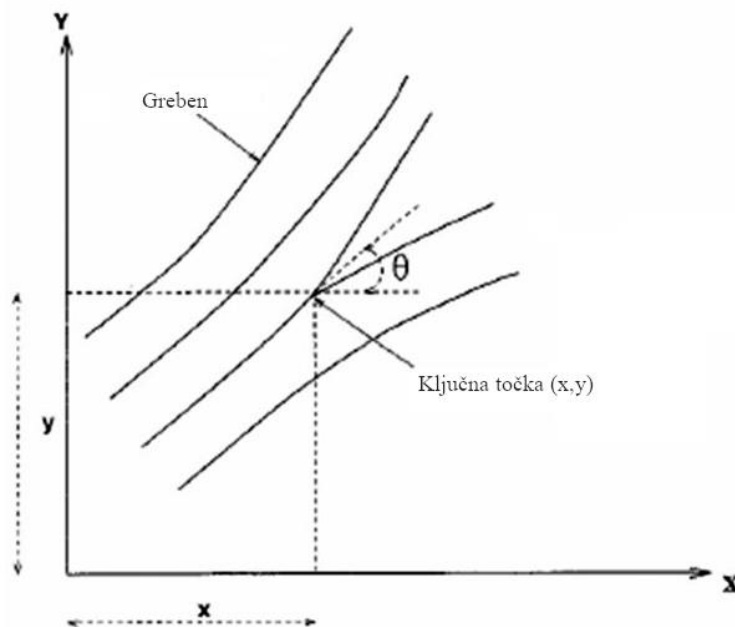
Završetak grebena definira se kao točka na kojoj greben naglo završava dok bifurkacija grebena je definirana kao točka gdje se greben razbija na više smjerova kao što se može vidjeti na slici 3.5.2. Zajedno ove jedinstvene značajke se nazivaju ključne točke (engl. *minutiae*). Većina tehnika koje se koriste za ekstrakciju i vršenja provjere podudarnosti ovih točaka isključivo koriste samo ova dva tipa ključnih točaka, a generalno otisak dobre kvalitete obično sadrži oko 25 - 80 ključnih točaka.

Kod djelomičnog otiska prsta broj ključnih točaka je znatno manji (otprilike 15 - 30). U ovakvom slučaju može doći do neuspješnog podudaranja otiska prsta zbog malog broja ključnih točaka. Složenije značajke otiska prsta se mogu izraziti kombinacijom odvih dviju značajka. Na primjer, ograda (engl, *enclosure*) se može smatrati skupom dvaju bifurkacija, a kratki greben (engl, *short ridge*) se može smatrati parom završetaka grebena kao što je prikazano na slici 3.5.3.



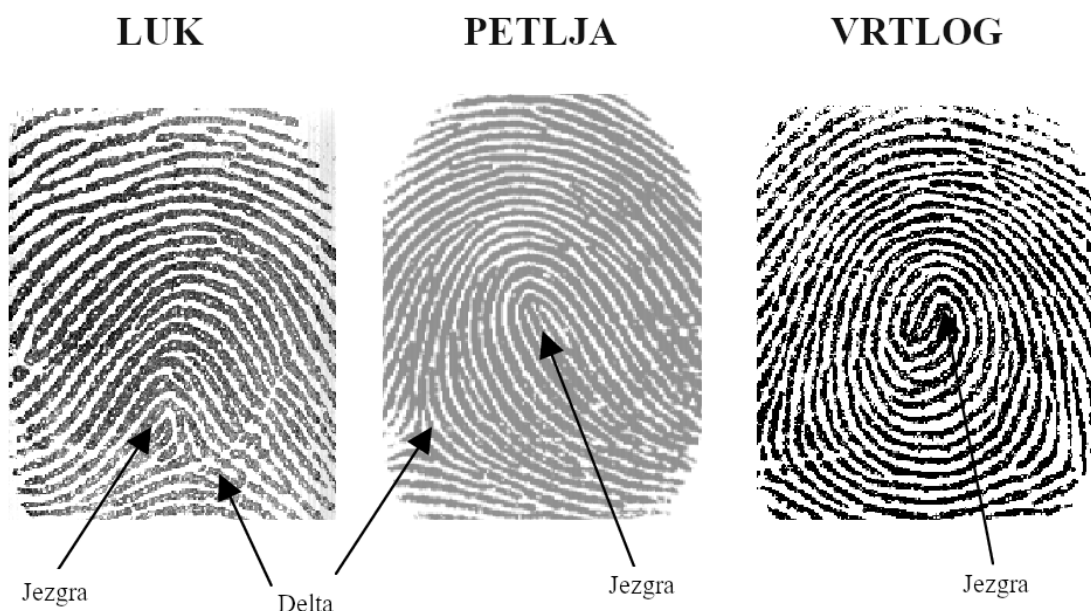
Slika 3.5.3 – Složene značajke kao kombinacija jednostavnih značajki: (a) kratki greben; (b) ograda [14]

Svaki od grebena i bifurkacija vrsta ključnih točaka imaju 3 atributa koji su važni za spremati. To su koordinata x-osi, koordinata y-osi i lokalni smjer (θ) kao što je prikazano na slici 3.5.4.



Slika 3.5.4 – Dijelovi značajka ključnih točaka [14]

Osim ovih detaljnih karakteristika ključnih točaka po potrebi možemo koristiti klasifikaciju otiska prsta prvog sloja (poznato kao i globalne karakteristike) a ta vrsta klasifikacije zasniva se na orijentaciji grebena, te dodatnih oblika koji se mogu naći na otisku prsta. Kao na primjer luk, petlje, vrtlozi i delte koji se mogu vidjeti na slici 3.5.5.



Slika 3.5.5 – Primjer značajka prvog sloja, luk, petlja, vrtlog i delta [15]

Ove značajke se mogu izdvojiti i klasificirati na više načina, ovisno o primijenjenoj metodologiji. Mnogi od ovih algoritama provjereni su za točnost i performanse u odnosu na popularne baze podataka otiska prsta kao što su NIST SD4, SD14 i SD9.

3.5.2. Algoritmi za klasificiranje

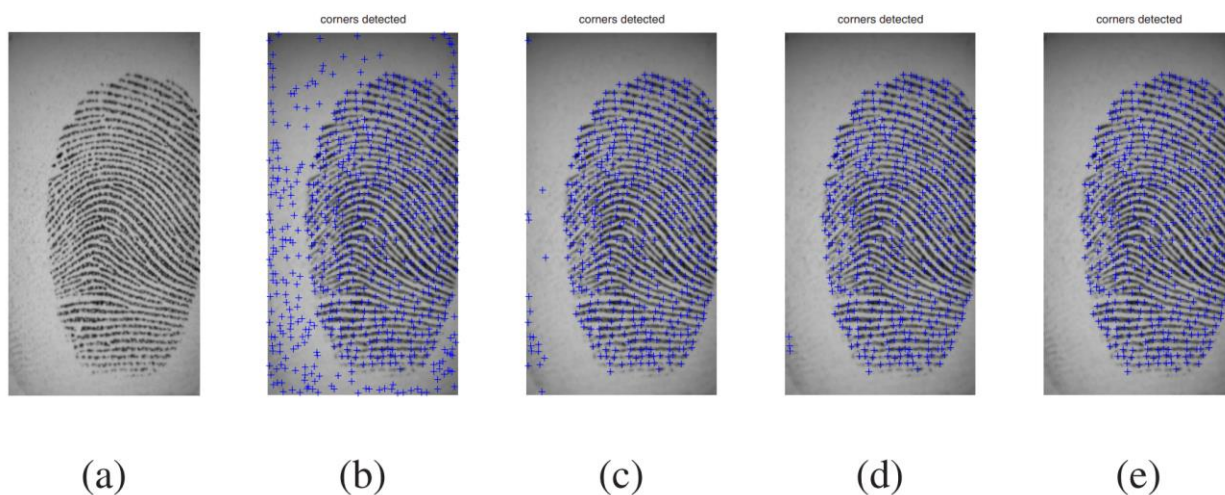
Algoritmi za klasificiranje se služe za ekstrakciju ključnih točaka. Da bi se to moglo izvršiti prvo se mora identificirati regija ključnog interesa što znači da se odbacuju svi nepotrebni dijelovi otiska prsta, kao na primjer prazan prostor u slici koji nastane zbog drugačije pozicije prsta i upotrebe drugačijeg pritiska pri očitavanju otiska prsta. Ovo uzrokuje da je svaka slika otiska drugačija iako se pokušava očitati isti otisak prsta.

Harris-Corner-Points je jedan od algoritama koji se može koristiti za razlikovanje pozadine od otiska prsta. Prvotno je bio razvijen za klasificiranje značajki praćenja kretanja jer može značajno smanjiti količinu izračunavanja u usporedbi sa praćenjem svakog piksela. Za detaljniji pregled *Harris-Corner-Points* algoritma predložio bi da se referencira [16].

Kako bi se segmentirao dio otiska prsta od pozadine koristi se mjera koja se naziva kutna snaga (engl. *corner strength*). Ova mjera se može izračunati na sljedeći način [16]:

$$R = \frac{I_x^2 I_y^2 - I_{xy}^2}{I_x^2 + I_y^2}$$

Snaga *Harris point*-a u području otiska prsta je mnogo veća nego u području pozadine zato što krajevi grebena na rubovima otiska prsta inherentno posjeduju veću kutnu snagu. Većina visokokvalitetnih slika otisaka prsta se može lako segmentirati sa odabirom odgovarajuće granične vrijednosti.



Slika 3.5.6 – Primjer otisaka prsta sa različitim vrijednostima kutne snage (a) 0, (b) 10, (c) 60, (d) 200 i (e) 300

Na slici 3.5.6 odabrana je kutna snaga od 300 kako bi se razlikovale kutne točke iz otiska prsta od kutnih točaka uzrokovanim anomalijama u pozadini. I za kraj može se koristiti *Convex hull* algoritam kako bi se locirala granica otiska prsta povezivanjem rubnih točaka [16].



Slika 3.5.7 – Primjer otiska sa čvršćom vrijednošću kutne snage od (a) 100, (b) 500, (c) 1000, (d) 1500 i (e) 3000

Neke gušće kutne točke se ne mogu u potpunosti filtrirati čak i korištenjem kutne snage od 3000.

Izgleda relativno jednostavno segmentirati slike otiske prstiju za sljedeće procese poput poboljšanja slike, otkrivanja i uspoređivanja ključnih točaka. No potrebno je riješiti dva tehnička problema. Prvo, potrebni su različiti pragovi kutne snage kako bi se postigli dobri rezultati segmentacije za slike različitih kvaliteta na temelju karakteristike slike. Drugo, neke *Harris points* u gustim mrljama mogu imati veću snagu pa se ne može segmentirati odabirom samo jednog praga, jer odabirom jednog praga za sve slike otisaka prstiju može uzrokovati da se ne maknu sve kutne točke iz pozadine u ostalim slikama otisaka prstiju. Ovo se potencijalno može riješiti na način da se implementira heuristički algoritam odabira kao na primjer *Gabor response* [16].

Nakon ovog koraka moguće je raditi poprilično zahtjevnu operaciju na slikama ako se žele dobiti bolji rezultati skupljanja ključnih točaka. Cilj poboljšanja (engl. *enhance*) slike otiska prsta je da se što više eliminiraju faktori koji mogu uzrokovati loša očitavanja prilikom tehnika skupljanja ključnih točaka. Rezultat ovog koraka je slika otiska prsta koja razrjeđuje debljinu grebena i miče nepotrebni šum sa slike. Primjer ovoga se može vidjeti na slici 3.5.1.

Za proces klasifikacije koriste se i globalne karakteristike otiska prstiju, posebno tok grebena, orijentacijska slika i ključne točke. Ekstrakcija ključnih točaka može se provesti na više načina ovisno o vrsti obilježja, kvaliteti slike i točnosti koju zahtijeva sljedeća faza klasifikacije. Cilj faze klasifikacije je spremati lokacije ključnih točaka na temelju očitano otiska kako bi se mogle spremati ili koristiti za uspoređivanje.

Algoritmi za klasifikaciju se mogu kategorizirati na sljedeći način:

- Pravilo: (engl. *rule*): Ovi algoritmi razgrađuju dobiveni unos podataka o otiscima prsta na temelju jedinstvenosti, uzimajući u obzir njihov broj i položaj. Ovdje se pronalaze ključne točke i konačna klasifikacija se vrši na temelju obrasca formiranog tim singularnostima. Dobar primjer prikazan je u ref [17].
- Sintaktički (engl. *Syntactic*): Ovi algoritmi temelje se na općoj gramatici, kao u tom slučaju aplikacija za obradu prirodnog jezika. Ovdje značajke izvađene iz ulaznih podataka pohranjuju se kao simboli u bazu podataka. Simboli su formirani iz grebenastih tokova, koji su zatim klasificirani skupom gramatika. Primjer za ovo može se vidjeti u ref [18].
- Strukturni (engl. *Structural*): Ovi algoritmi razgrađuju unos otisaka prstiju i spremaju ga u određene strukture podataka i formate poput grafova i stabala kako bi bolje razumjeli i pretvorili značajke u prikaz više razine. Primjer toga može se vidjeti u ref [19].

- Statistički (engl. *Statistical*): Ovi algoritmi u potpunosti se temelje na izračunatim statističkim podacima. Ti se podaci općenito izračunavaju korištenjem uobičajenih statističkih aplikacija, kao na primjer jedan od takvih aplikacija je Bayesovo pravilo odlučivanja (engl. *Bayesian decision rule*), podrška vektorskog stroja (engl. *support vector machine*, SVM), i K-najbliži susjed (engl. *K-nearest neighbour*, KNN). Primjer statističkih pristupa u SVM klasifikaciji i poboljšanju provjere otiska prsta može se vidjeti u ref [20].
- Neuronske mreže (engl. *Neural network*): Ovi algoritmi strojnog učenja koriste višeslojne perceptrone. Uzimaju se ulazni podaci, a izvučene značajke se nakon smanjenja dimenzija prosljeđuju kroz algoritam. Izdvojena svojstva su uglavnom singularnosti i orijentacija slike koja se koristi za treniranje perceptrona za poboljšanje njihovih performansi. Te mreže daju izvrsne rezultate klasifikacije. CNN se trenira pomoću preklapajućih blokova raznih veličina kako bi se otkrio otisak. Pristup temeljen na konvolucijskoj neuronskoj mreži (engl. *Convolutional Neural Network*, CNN) predložen je u referenci [21].
- Više-klasifikator (engl. *multi-classifier*): Ovi algoritmi uključuju sve pristupe koji kombiniraju dva ili više klasifikatora [22].

Nakon faze spremanja ključnih točaka dobivenim algoritmima ponavljamo istu proceduru sa drugim otiskom prsta ali ovaj put obavljam proces uspoređivanja. Neki od spomenutih algoritama se također mogu koristiti i u sljedećoj fazi.

Otkrivanje karakteristika i podudaranje slika dva su važna problema u strojnom vidu, a njihove primjene nastavljaju rasti u raznim poljima. Idealna tehnika otkrivanja i uspoređivanja značajki trebala bi biti robusna za transformacije slike kao što su rotacija, promjena veličine slike, promjena osvjetljenja, šum i afine transformacije. Osim toga idealne ključne točke moraju biti vrlo različite kao što u slučaju senzora za otisak prsta na kraju imamo bijelu pozadinu sa crnim grebenima na slici, ovako jedna značajka će se trebati podudarati sa velikom vjerojatnošću.

Jedan od takvih tehnika je inverzibilna značajka transformacije skale (engl. *Scale Invariant Feature Transform*, SIFT) koju je razvio Lowe 2004. godine. SIFT se pokazao vrlo učinkovit za prepoznavanje objekata ali problem je što zahtijeva veliku računalnu složenost koja predstavlja veliki nedostatak, pogotovo ako se koristi u programima koji trebaju raditi u stvarnom vremenu. Postoji nekoliko varijanti i proširenja SIFT-a koji su poboljšali računalnu složenost.

Tehnika ubrzavanja robusne značajke (engl. *Speed up Robust Feature*, SURF) je aproksimacija SIFT-a, a radi brže od SIFT-a bez smanjenja kvaliteta otkrivenih ključnih točaka. Iako SURF ima neke prednosti, SIFT i SURF rade na istom principu deskriptora i detektora.

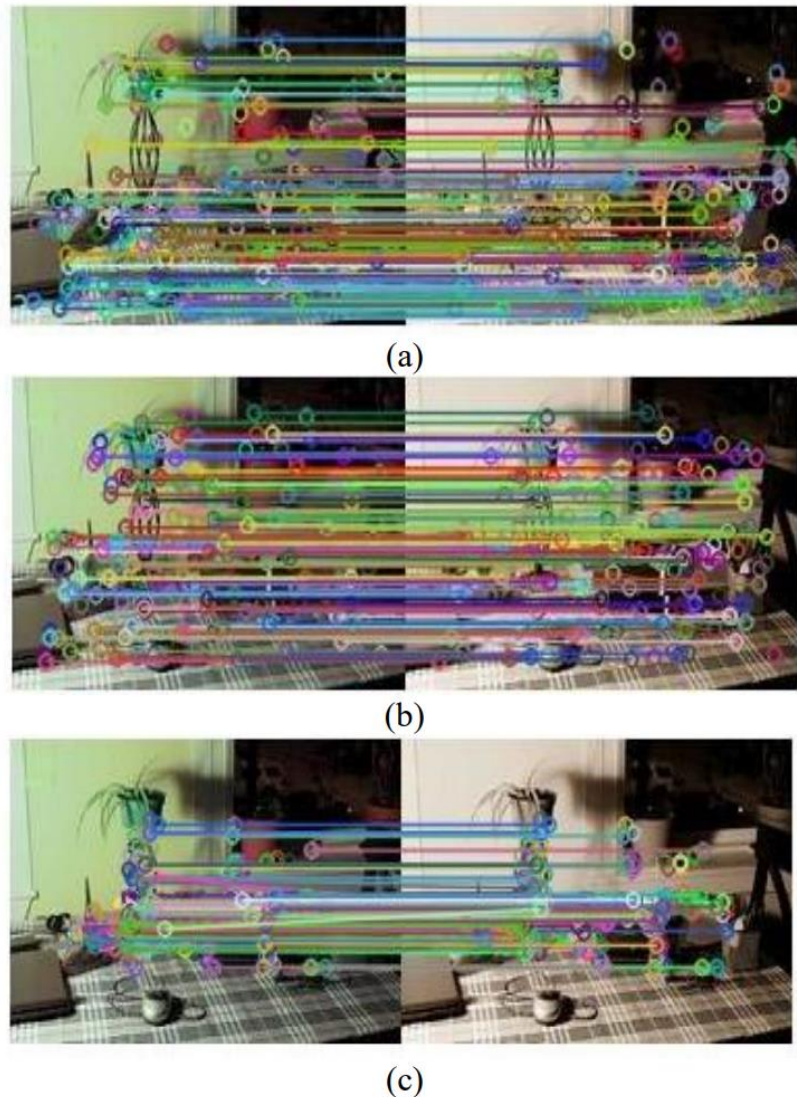
Postoji još jedna tehnika koja je alternativa za SIFT a to je binarno robusne neovisne elementarne značajke (engl. *Binary Robust Independent Elementary Features*, BRIEF) koja zahtjeva manje složenosti od SIFT-a s gotovo sličnim performansama podudaranja. Usmjereni FAST i rotirajući BRIEF (engl. *Oriented FAST and Rotated BRIEF*, ORB) je isto jedna od efikasnijih alternativa za SIFT i SURF.

SIFT koji je predložio Lowe rješava rotaciju slike, affine transformacije, intenzitet i promjenu gledišta u ključnim točkama podudaranja. SIFT algoritam ima 4 osnovna koraka. Prvo se mora procijeniti prostorni ekstrem korištenjem *Gaussove* razlike (engl. *Difference of Gaussian*, DoG). Drugo, ključna točka lokalizacije, gdje su kandidati ključnih točaka lokalizirani i pročišćeni uklanjanjem slabih kontrastnih točaka. Treće, dodjela orijentacije ključnih točaka na temelju lokalnog gradijenta slike i na kraju generator deskriptora za izračunavanje lokalnog slikovnog deskriptora za svaku ključnu točku na temelju veličine i orijentacije gradijenata slike [23].

SURF aproksimira DoG s blokovskim filterima. Umjesto da *Gaussian* izračunava prosječnu vrijednost, koriste se kvadrati aproksimacija jer je skupljanje s kvadratom mnogo brže ako se koristi integralna slika. Također, ovo se može učiniti paralelno za različite skale. SURF koristi BLOB detektor koji se temelji na Hessovoj matrici (engl. *Hessian matrix*) kako bi pronašao ključne točke. Za dodjelu orijentacije koristi valutne odgovore u vodoravnom i okomitom smjeru primjenom odgovarajućih *Gaussovih* utega. Za opis karakteristika SURF-a također koristi valutne odgovore. Susjedstvo oko ključne točke odabire se i dijeli na pod regije, a zatim se za svaku pod regiju uzimaju valutni odgovori i predstavljaju kako bi se dobio deskriptor ključne točke SURF-a. Znak Laplacijana, koji je već izračunat u fazi otkrivanja koristi se za temeljne ključne točke te može razlikovati mrlje na tamnim pozadinama. U slučaju podudaranja, točke se uspoređuju samo ako imaju isti tip kontrasta (na temelju znaka laplacijana) što omogućava brže podudaranje [23].

ORB je fuzija detektora ključne točke FAST i deskriptora BRIEF s nekim modifikacijama. U početku, za utvrđivanje ključnih točaka koristi FAST, onda se primjenjuje Harrisova kutna mjera (engl. *Harris corner measure*) za pronalaženje gornjih N točaka. FAST ne izračunava orijentaciju tako da je ovisna o rotaciji, nego izračunava intenzitet središnje točke sa lociranim kutom u sredini.

Smjer vektora od ove kutne točke do središnje točke daje orijentaciju. Trenutci su izračunati za poboljšanje rotacije. Deskriptor BRIEF loše djeluje ako se dogodila rotacija ravnine. U ORB-u, rotacijska matrica se izračunava pomoću orijentacijskog bloka a zatim se BRIEF deskriptori usmjeravaju prema orijentaciji za uspoređivanje.



Slika 3.5.8 – Primjer uspoređivanja slika sa različitim intenzitetom koristeći (a) SIFT (b) SURF (c) ORB [23]

Generalno, ORB se pokazuje kao najbrži algoritam dok SIFT radi najbolje u većinu scenarija. U posebnim slučajevima kada je kut rotacije proporcionalan 90 stupnjeva, ORB i SURF nadmašuju SIFT, a na šumnim slikama ORB i SIFT pokazuju gotovo slične performanse. U ORB, ključne točke su većinom koncentrirane u objektima u središtu slike, dok u SURF, SIFT i FAST detektorima, ključne točke su raspoređene po ostatku slike [23].

4. IZRADA SUSTAVA NA RASPBERRY PI PLATFORMI

Cilj praktičkog dijela ovog rada je izrada fizičkog sustava koji omogućava efikasan način korištenja postojeće *Raspberry Pi* platforme zajedno sa eksternim senzorom za analizu otiska prsta koji se spaja na sami *Raspberry Pi* uređaj, skupa sa eksternim fizičkim sučeljem. Kao na primjer fizičkim botunima koji bi omogućili potpunu izolaciju sustava na način da nije potrebno koristiti računalo kako bi se kontrolirao tok podataka kroz *Raspberry Pi* i senzor za analizu otiska prsta. Prednost ovakvog sustava je mogućnost implementacije vlastite funkcionalnosti za analizu otiska prsta te ujedno i korištenje cijelog sustava na raznim mjestima gdje bi mogao biti koristan, osobito zbog lake prenosivosti i korisnosti samog sustava.

Kroz ovo poglavlje proći će se kroz komponente korištene u ovom radu te kroz detaljan opis postupka konfiguriranja i realizacije softvera kako bi se sustav uspješno izolirao od potrebe korištenja vanjskih intervencija za normalnu operaciju sustava.

4.1. Komponente korištene i potrebna konfiguracija za izradu sustava

Komponente koje su korištene u ovom radu su:

- *Raspberry Pi* 3 Model B+
- Arduino kompatibilni senzor za otisak prsta, Velleman VMA329 (ZFM-708)
- UART serijski/TTL 3,3/5V USB pretvarač adapter CP2102
- Arduino kompatibilni LCD zaslon I2C 16x2 (plavi)
- Relej modul 5V (Max DC 30V/10A – Max AC 250V/10A)
- Magnetski lokot DC 5V sa elektromagnetsko-električnim zaključavanjem
- Eksperimentalna pločica SD – 120
- Spojne žice 40 pins 30 cm (*male to male* / *male to female* / *female to female*)
- Fizički botuni za pločicu 6x6x7 mm (x5)

4.1.1. Raspberry Pi 3 Model B+

Raspberry Pi model koji se koristio u ovom radu je relativno novija verzija te ima jako zgodne funkcionalnosti koje su se iskoristile, kao na primjer mogućnost konfiguracije *Raspberry Pi*-a na način da se ne more spajati žičano na računalo. Na primjer spajanje periferija kao tipkovnica, miš te korištenje monitora uz pomoć HDMI kabela nije potrebno jer *Raspberry Pi* 3 Model B+ ima integrirani *Bluetooth/WIFI* modul direktno na svojoj ploči.

To znači da je moguće se spojiti na *Raspberry Pi* preko SSH veze te raditi na daljinski način. Ovaj način rada se može još više olakšati ako koristimo kriptiranu vezu pomoću RSA privatnih ključeva tako da nije potrebno pamtit i šifre.



Slika 4.1.1 – Primjer Raspberry Pi Modela korištenog u radu

Jedna bitna stvar koja se može napomenuti je da se u procesu razvoja rada promijenio izvor napajanja samog *Raspberry Pi-a* jer dodavanjem raznih komponenata došlo je do potrebe da se uzme službeni adapter za napajanje kako bi sustav bio stabilan. Izvor starog prijašnjeg napajanja je bio rangiran za 0.85A/5V, a zamjena je bila jedna od službenih napajanja za *Raspberry Pi* koja je bila rangirana za 3.1A/5V. Ovim se osigurala stabilnost rada svih komponenata.

4.1.2. Senzor za analizu otiska prsta ZFM-708

Bio je izazov naći pravi senzor za očitavanje prsta koji je izravno napravljen i kompatibilan sa *Raspberry Pi* platformom. Prvi problem je što se mora pronaći senzor koji je rangiran za 3.3V ako se želi spojiti senzor direktno na *Raspberry Pi* preko GPIO pinova koji imaju funkcionalnost prijenosa podataka preko UART protokola. Drugi problem je što većina GPIO pinova koja se mogu koristiti za ovakve svrhe je ograničena jer većina GPIO pinova su generalni pinovi što se služe za razne stvari dok samo par GPIO pinova su specijalni pinovi koji mogu vršiti komunikaciju sa drugim uređajima preko posebnih protokola. Primjer ovoga se može vidjeti na slici 2.3.1 u jednom od prijašnjih poglavlja.

Specifikacije senzora koji se koristio u ovom radu su [24]:

- Napon: DC 3.8-7V
- Stražnje svjetlo: zeleno
- Brzina prijenosa (engl. *baud rate*): (9600 x N) bps, N = 1-12 (zadano N=6)
- Vrijeme dohvaćanja slike: 1000
- Kapacitet spremanja otisaka prstiju: 1000 otisaka
- Lažna stopa prihvatanja: 0,001% (na razini sigurnosti 3)
- Lažna stopa odbijanja: 1,0% (razina sigurnosti 3)
- Protokolno sučelje: serijski TTL
- Radna struja (Maksimum): 0.95A
- Prosječno vrijeme traženja: 1 sekunda
- Radna temperatura: -20/+50 stupnjeva
- Dimenzije: 54 x 20 x 20,5 mm



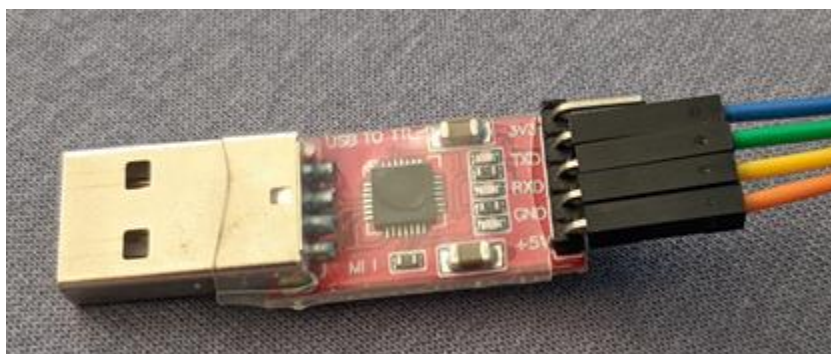
Slika 4.1.2 – Primjer senzora za analizu otiska prsta ZFM-708 korištenog u radu

Kod ovog senzora vidimo u specifikacijama da je rangiran za snagu od 3.8-7V DC što znači da senzor neće ispravno raditi te bi mogli oštetiti *Raspberry Pi* ako bi se spajao direktno preko UART GPIO pinova. Kako bi se riješio ovaj problem kompatibilnosti korištena je dodatna komponenta koja omogućava spajanje senzora na *Raspberry Pi* preko USB sučelja.

Upute za konfiguraciju i primjere koda za upotrebu ZFM-708 senzora se može naći na [25] [26].

4.1.3. UART - USB pretvarač adapter CP2102

Kao što se prethodno spomenulo kako bi se riješio problem kompatibilnosti korišten je serijski UART – USB pretvarač sa kojom postoji mogućnost spajanja senzora koji su rangirani za više od 3.3V tako da se Vin (pin 3) sa senzora koji se koristi za napajanje senzora spoji na 5V pin od UART – USB pretvarača. To se može vidjeti na slici 4.1.3.



Slika 4.1.3 – Primjer UART – USB pretvarača CP2102

Potrebno je spojiti još 3 dodatne veze između senzora i pretvarača. Sljedeće veze koje se moraju spojiti su uzemljenje (GND) i veza za slanje (TD) i primanje (RX) podataka između komponenata.

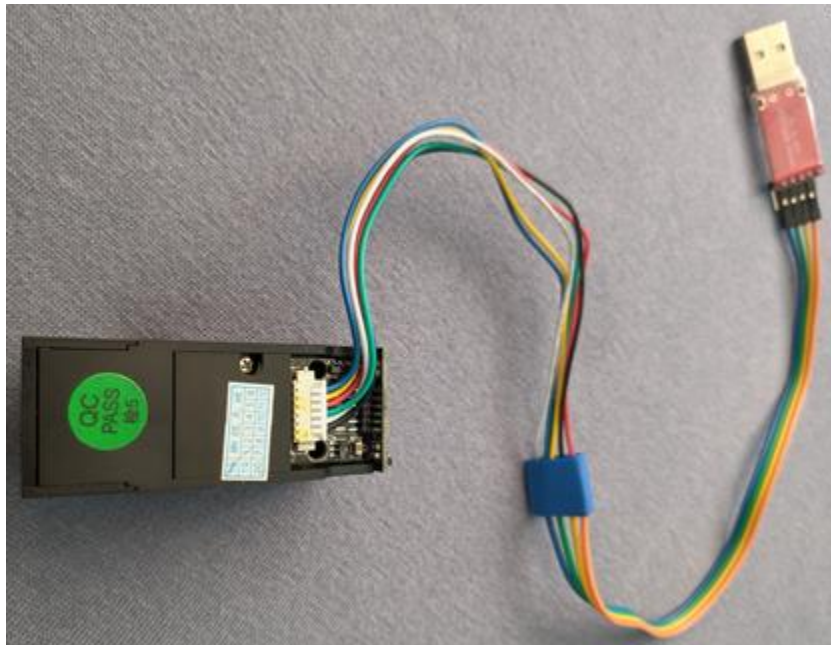


Slika 4.1.4 - Pinovi za konekciju na ZFM 708 senzoru

Pin	Ime	Tip	Boja	Opis	Korišteno
1	Vtouch	In	Plava	Ulaz snage osjetila dodira senzora	Ne
2	Sout	Out	Žuta	Izlaz osjetila dodira senzora	Ne
3	Vin	In	Crvena	Ulaz snage	Da
4	TD	In	Zelena	Izlaz podataka	Da
5	RD	Out	Bijela	Unos podataka	Da
6	GND	-	Crna	Uzemljenje	Da

Tablica 4.1.1 – Opis pinova na ZFM 708 senzoru

Sljedeća konekcija koja se treba spojiti je GND (pin 6) sa senzora na GND pin UART pretvarača. Nakon toga potrebno je još spojiti pinove koji se koriste za prijenos podataka kako bi senzor uspješno mogao vršiti komunikaciju sa *Raspberry Pi* uređajem. Potrebno je TD (pin 4) sa senzora spojiti na RXD pin od pretvarača i RD (pin 5) sa senzora spojiti na TXD pin od pretvarača. Primjer spoja se može vidjeti na slici 4.1.5

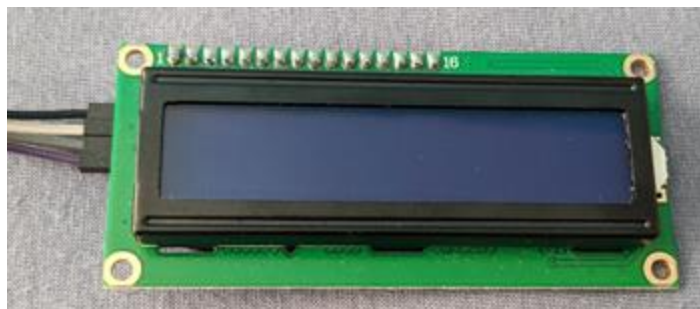


Slika 4.1.5 - Primjer spojenog senzora i UART-USB pretvarača

Prednost korištenja ove komponente je lakše spajanje i od spajanje senzora na *Raspberry Pi* jer se koristi USB sučelje, a poprilično su jeftini za nabaviti i ne moraju se koristiti GPIO pinovi što isto smanjuje kompleksnost.

4.1.4. LCD zaslon I2C 16x2 (plavi)

Postoji više vrsta LCD zaslona koji su kompatibilni sa *Raspberry Pi* platformom. U ovom slučaju koristio se Arduino kompatibilni LCD zaslon sa 2 reda od 16 znakova jer rad nije zahtijevao kompleksne zaslone a i korištenje ovog specifičnog LCD zaslona na *Raspberry Pi* platformi je poprilično jednostavno jer sami LCD zaslon na sebi ima određene čipove i pinove što koriste I2C sučelje/protokol. Ovo je jako korisno jer *Raspberry Pi* ima 2 GPIO pina koji rade preko I2C protokola. Mnogo drugih LCD zaslona koja su napravljeni za *Raspberry Pi* nerade preko I2C protokola pa mogu biti znatno kompleksniji za spajanje u usporedbi sa ovim LCD zaslonom. Primjer i lokaciju I2C GPIO pinova na *Raspberry Pi*-u se može vidjeti na slici 2.3.1



Slika 4.1.6 – Primjer LCD zaslona I2C 2x16 korištenog u radu

Specifikacije LCD zaslona I2C 2x16 [27]:

- Napon: 5V
- Stražnje svjetlo: plavo
- Sučelje: I2C
- Kontrast: Podesivi
- Dimenzije: 82 x 35 x 18 mm



Slika 4.1.7 – Primjer pinova i I2C sučelja LCD zaslona

Na slici 4.1.7 može se vidjeti sličan način spajanja kao i kod UART pretvarača, ali u ovom slučaju spajanje se vrši direktno na *Raspberry Pi* GPIO pin.

Spajanje na *Raspberry Pi* će se napraviti tako da se prvi pin (GND) sa LCD zaslona spoji na jedan od slobodnih GND pinova na *Raspberry Pi*-u, a u ovom slučaju to je bio GND pin broj 6. Sljedeći VCC pin sa LCD zaslona spojen je na jedan od slobodnih 5V pinova na *Raspberry Pi*-u, u ovom slučaju to je bio pin broj 4. Ostalo je spojiti SDA i SCL pinove kako bi omogućili komunikaciju i prijenos podataka između *Raspberry Pi* i LCD zaslona. SDA se treba spojiti na SDA1 pin odnosno GPIO pin 3 a SCL pin LCD zaslona treba spojiti na SCL1 pin (GPIO pin 5).

Upute za konfiguraciju i primjere koda za upotrebu LCD ekrana se može naći na [28].

4.1.5. Releji modul 5V

Releji su jako korisne komponente koje nam omogućuju imitiranje normalne sklopke ali na način da pomoću električnog naboja odnosno signala možemo programirati određenu logiku pa time otvarati i zatvarati neki strujni krug pomoću programskog koda. Ovo je vrlo korisna funkcionalnost jer možemo imati neki strujni krug koji je spojen kao i u ovom radu na *Raspberry Pi* uređaju i pomoću bežične konekcije kontrolirati neki sustav pri velikoj udaljenosti. Releji su nekad potrebni jer se radi sa strujnim krugom velikog napona, a uređaji poput *Raspberry Pi*-a su jako osjetljivi pa visoki napon može uzrokovati veliko oštećenje. Da bi se to izbjeglo koriste se releji koji na jednom kraju imaju spojene potrebne veze kako bi se slali signali za aktivaciju a na drugom kraju se radi sa strujim krugom koji se kontrolira uz pomoć samog releja. U ovom radu primjer strujnog kruga koji je spojen na relej je strujni krug za aktivaciju mehaničkog lokota.



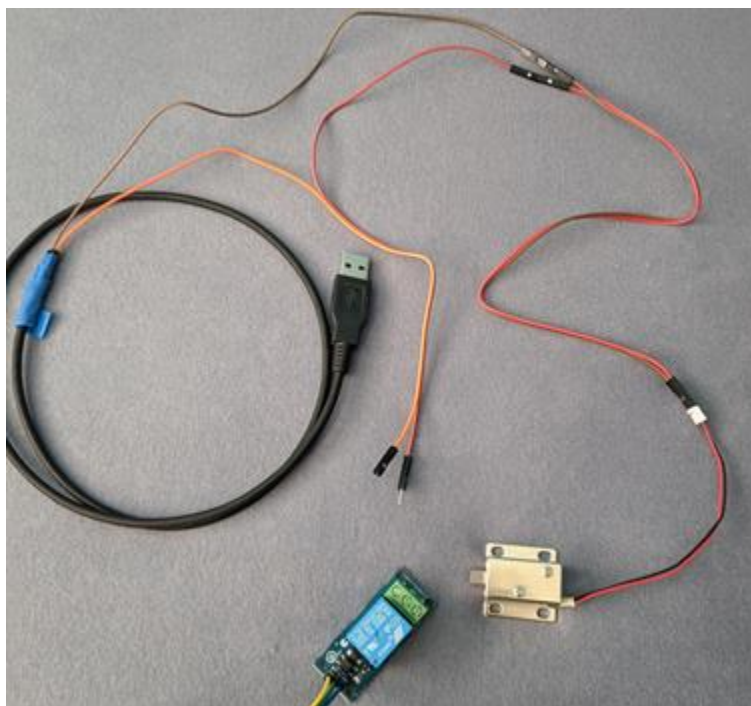
Slika 4.1.8 - Primjer Releji modula

Releji radi na principu elektromagnetizma i to na način da kad se dovede električni naboj na relej stvori se magnetsko polje koje gurne sklopku da zatvori strujni krug, kad električni naboj više nije prisutan ugrađena opruga vraća sklopku u prvobitno stanje pa tako strujni krug postane otvoren. Ovo se može konfigurirati po potrebi na način da se odluči hoće li se spojiti strujni krug na relej tako da prvobitno bude otvoreni strujni krug pa kad se relej aktivira da se zatvori strujni krug ili obratno. Na jednom od krajeva releji obično imaju označeno gdje se može priključiti da bude prvobitno otvoren (engl. *normally open*, NO), prvobitno zatvoren (engl. *normally closed*, NC) i još jednu oznaku što označava da se mora spojiti jedan kraj strujnog kruga (engl. *common*, COM).

Način spajanja releja je sličan proces kao i kod prijašnjih primjera npr. LCD zaslona. GND pin od releja se spoji na jedan od slobodnih GND GPIO pinova na *Raspberry Pi* (pin 9). Sljedeći VCC pin od releja treba se spojiti na jedan od slobodnih 5V pinova za napajanje u ovom slučaju (pin 2). Zadnji pin releja koji se koristi za aktivaciju releja što se označava kao IN pin na releju je spojen na jedan od standardnih GPIO pinova na *Raspberry Pi*-u, a to je u ovom slučaju pin 7.

4.1.6. Mehanički lokot DC 5/6V sa elektromagnetsko-električnim zaključavanjem

Mehanički lokot je jedan od aplikabilnih uređaja koji se može koristiti sa relej modulom koji se koristio u ovom radu kako bi dobili kompletnu funkcionalnost sustava. Uloga lokota je jednostavna. Ukoliko se električni naboj dovede na sami lokot aktivirat će se magnetsko polje koje će povući lokot u samo kućište mehaničkog lokota pa time ostvariti otključano stanje sustava. U slučaju da električni naboj se makne sa mehaničkog lokota onda će se sklopka vratiti u prvobitno stanje i time ostvariti zaključano stanje sustava.

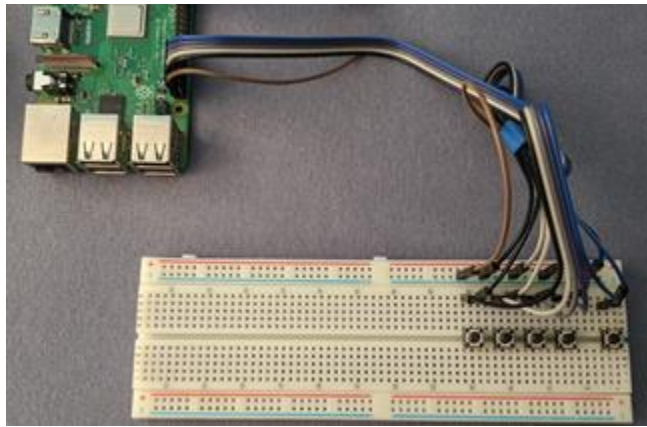


Slika 4.1.9 – Primjer mehaničkog lokota sa elektromagnetsko-električnim zaključavanjem i relej modula

Kako bi se mehanički lokot mogao koristiti u ovom radu mora postojati dodatan strujni krug sa zasebnim izvorom napajanja da bi relej mogao kontrolirati aktivaciju. Ovo se vrši na način da se uzemljenje poveže direktno na lokot a aktivni dio sklopa se odvoji u 2 dijela. Jedan dio se spoji sa lokota na relej a drugi sa releja na izvor napajanja kako bi se zatvorio strujni krug.

4.1.7. Eksperimentalna pločica sa fizičkim botunima

Eksperimentalna *Raspberry Pi* pločica (engl. *breadboard*) omogućava lakše korištenje raznih komponenata. Kao na primjer fizički botuni koji su spojeni na način da kad se jedan od botuna pritisne, strujni krug za taj botun se zatvori. Kad se strujni krug zatvori detektira se promjena električnog napona na pinovima *Raspberry Pi*-a te pri pisanju programa ovo se uzima u obzir kako bi se isprogramirala logika za korisničko sučelje. Svrha ovoga sučelja od 5 botuna je da se izbjegne potreba za korištenjem računala skupa sa tipkovnicom i ostalim periferijama. Primjer ovog korisničkog sučelja može se vidjeti na slici 4.1.10.

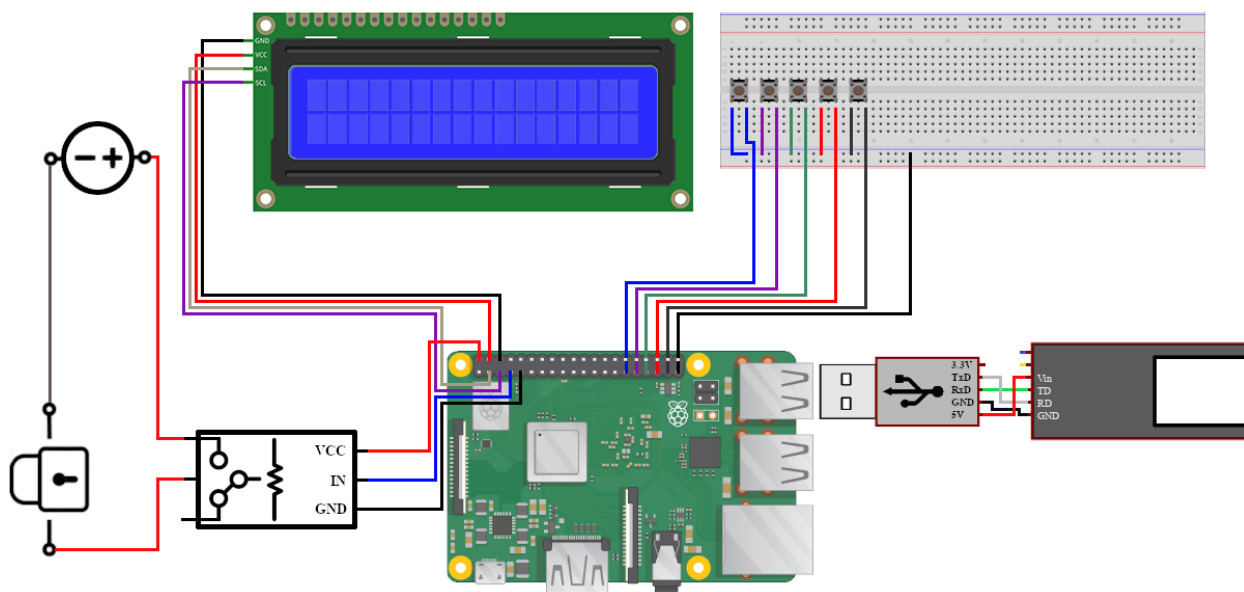


Slika 4.1.10 – Primjer Eksperimentalne pločice sa fizičkim botunima

Bitno za napomenuti je da bi se trebali koristiti povlačni ili vučni otpornici u strujnom krugu svakog botuna kako bi se zaštitila *Raspberry Pi* pločica i osiguralo pravo početno stanje pri korištenju botuna, ali u slučaju ovoga rada to nije bilo potrebno zato jer *Raspberry Pi* u sebi ima ugrađene otpornike koje se koriste za istu namjenu. Zgodna stvar ugrađenih otpornika je što se smanjuje kompleksnost strujnih krugova koji se grade na eksperimentalnoj pločici.

4.2. Završna shema spoja sustava

Na slici 4.2.1 može se vidjeti kako se sve komponente integriraju međusobno sa ciljem spajanja na *Raspberry Pi* 'J8' zaglavlje. Ukupno se iskoristilo 13 od totalno 40 GPIO pinova na zaglavlju *Raspberry Pi*-a. Od tih, koristilo se 2 x 5 VDC pina za izvor napajanja relej modula i LCD zaslona, 3 x GND pina za uzemljenje odnosno zatvaranje strujnog kruga, ostalih 8 za programibilnu logiku od kojih su 2 rezervirana za komunikaciju LCD zaslona preko I2C sučelja/protokola, a 6 za generalnu upotrebu korisničkog sučelja kako bi se mogla vršiti komunikacija sa sustavom. Detaljniji prikaz pinova korištenim u ovom radu se može vidjeti na slici 2.3.1.



Slika 4.2.1 – Kompletna shema sustava za analizu otiska prsta na Raspberry Pi platformi

Shema sustava je napravljena uz pomoć javnog besplatnog softvera „Gimp“ koji radi na principu javnog koda, a resursi koji su se koristili za izgradnju slike mogu se naći na [29] [30].

4.3. Postavljanje radnog okruženja

Kako bi se sustav mogao koristiti prvo se mora konfigurirati i podesiti radno okruženje sustava. Uz pretpostavku da su sve komponente postavljene po tvorničkoj zadanoj konfiguraciji prvi korak bi trebao biti instalirati *Raspbian* operativni sustav na *Raspberry Pi* uređaju. Svaki *Raspberry Pi* uređaj dođe sa SD karticom (engl. *new out of box software*, NOOBS) koja na sebi ima jedno od novijih verzija operacijskog sustava koji se može unaprijediti nakon inicijalne instalacije. Upute za instalaciju i konfiguraciju *Raspbian* operativnog sustava za *Raspberry Pi* platformu koristeći NOOBS može se naći na [31].

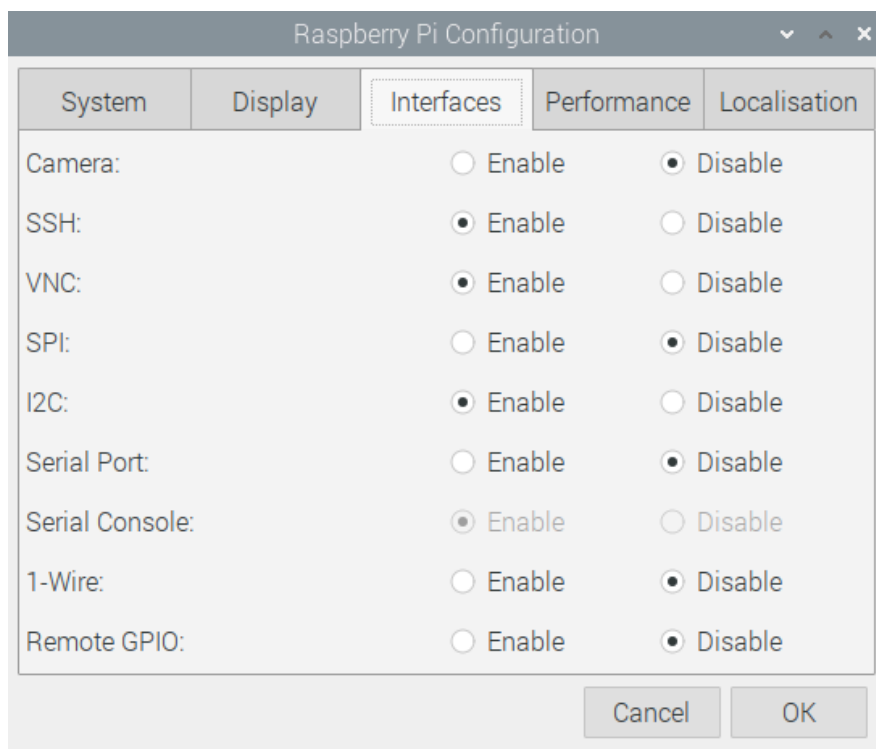
4.3.1. Koristeći daljinski razvoj

Za početnu konfiguraciju *Raspberry Pi*-a potrebno je priključiti *Raspberry Pi* na monitor kako bi se moglo pratiti sučelje za instalaciju operacijskog sustava skupa sa periferijama poput tipkovnice i miša. Tijekom instalacije može se konfigurirati WiFi na *Raspberry Pi* modelima koji u sebi imaju WiFi modul. Korištenjem WiFi-a imamo mogućnost koristiti *Raspberry Pi* uređaj na daljinski način. Uz pomoć daljinskog razvoja može se puno jednostavnije i efikasnije koristiti *Raspberry Pi* uređaj jer više nije potrebno koristiti periferije te spajati se na monitor lokalnog računala.

Za omogućit daljinski razvoj na *Raspberry Pi* uređaju potrebno je otići na sljedeću lokaciju:

Menu -> Preferences -> Raspberry Pi Configuration -> Interfaces

Na isječku konfiguracije 4.3.1 može se vidjeti opcije kao „SSH“, i „VNC“. Kako bi bilo moguće se spojiti na *Raspberry Pi* preko bežične WiFi veze, SSH sučelje mora biti uključeno, a ako se želi daljinski spojiti na *Raspberry Pi* i koristiti radno okruženje *Raspbian* operacijskog sustava preko grafičkog sučelja (engl. *graphical user interface*, GUI) onda se može koristiti VNC.



Isječak konfiguracije 4.3.1 – Primjer potrebne konfiguracije za daljinski razvoj

4.3.2. Konfiguracija Secured Shell (SSH) veze

Kad je SSH opcija osposobljena možemo se spajati na *Raspberry Pi* uređaj na način da sa lokalnog računala koji se nalazi na istoj WiFi mreži pokrenemo terminal ili neki drugi program koji ima mogućnosti korištenja SSH protokola kao na primjer „Putty“ pa zatim korištenje naredbe kao na primjer „ssh user@host“. Kod našeg slučaja ako je ostao podešeni korisnik na *Raspberry Pi* uređaju možemo koristiti komandu. „ssh pi@raspberrypi.local“ ili „ssh pi@192.168.0.32“ jer u slučaju ovog rada *Raspberry Pi* uređaj je poprimio lokalnu IP adresu 192.168.0.32. Nakon što se komanda izvrši SSH protokol će pitati šifru korisnika kako bi se sigurna konekcija mogla uspostaviti. Normalna zadana šifra kod zadanog korisnika na *Raspberry Pi* uređaju je „raspberrypi“.

U slučaju da pisanje šifre postane naporno, postoji dodatna mogućnost korištenja RSA enkripcije preko javnih i privatnih ključeva što je podržano sa SSH protokolom. Ovo se također može lako konfigurirati na način da se na lokalnom računalu koji se koristi za uspostavu veze na *Raspberry Pi* računalo generira privatni i javni RSA ključ. Na isječku konfiguracije 4.3.2 može se vidjeti proces konfiguriranja RSA privatnih i javnih ključeva.

```
[17:27] ~/.ssh ~> ssh-keygen -f ~/.ssh/raspberry-key-ecdsa -t ecdsa -b 521
Generating public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:/Users/ / .ssh/raspberry-key-ecdsa.
Your public key has been saved in C:/Users/ / .ssh/raspberry-key-ecdsa.pub.
The key fingerprint is:
SHA256:zjGd67pd5dWDziJQVtehnAj80JG0ffZZQix0mWC
The key's randomart image is:
+----[ECDSA 521]-----+
| ..o*oo+ |
| o.++o |
| *ooB . |
| .o.=o o |
| .So+ E.oo= |
| ooo ++o +. |
| o....o. |
| o.. |
| ooo |
+----[SHA256]-----+
[17:28] ~/.ssh ~> ll
total 9
-rw-r--r-- 1 197609 54 Jul 16 13:53 config
-rw-r--r-- 1 197609 411 Jul 6 15:48 id_ed25519
-rw-r--r-- 1 197609 183 Jul 6 15:48 id_ed25519.pub
-rw-r--r-- 1 197609 954 Aug 25 17:14 known_hosts
-rw-r--r-- 1 197609 361 Aug 25 17:28 raspberry-key-ecdsa
-rw-r--r-- 1 197609 276 Aug 25 17:28 raspberry-key-ecdsa.pub
[17:28] ~/.ssh ~> ssh-copy-id -i ~/.ssh/raspberry-key-ecdsa pi@192.168.0.32
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/c/Users/ / .ssh/raspberry-key-ecdsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'pi@192.168.0.32'"
and check to make sure that only the key(s) you wanted were added.

[17:28] ~/.ssh ~> rpssh
Linux raspberrypi 5.4.51-v7+ #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Aug 25 17:14:28 2020 from fe80::5174:1f6b:7b9:944b%wlan0

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.
```

Isječak konfiguracije 4.3.2 – Primjer konfiguracije RSA enkripcije preko privatnih i javnih ključeva

Prvi korak je da na lokalnom računalu izvršimo sljedeću naredbu za generaciju privatnog ključa:

```
ssh-keygen -f ~/raspberry-key-ecdsa -t ecdsa -b 521
```

Sa tom naredbom na lokalnom računalu dobijemo 2 nove datoteke. Jedna od datoteka je privatni ključ „raspberry-key-ecdsa“ koji se treba paziti da se ne izgubi jer bi došlo do proboja sigurnosti. Druga datoteka je javni ključ koji se treba kopirati na drugi uređaj, odnosno u ovom slučaju to će biti *Raspberry Pi*. Ovo se može napraviti sljedećom naredbom:

```
ssh-copy-id -i ~/.ssh/raspberry-key-ecdsa user@host
```

Više detalja i uputa o RSA ključevima može se vidjeti na [32].

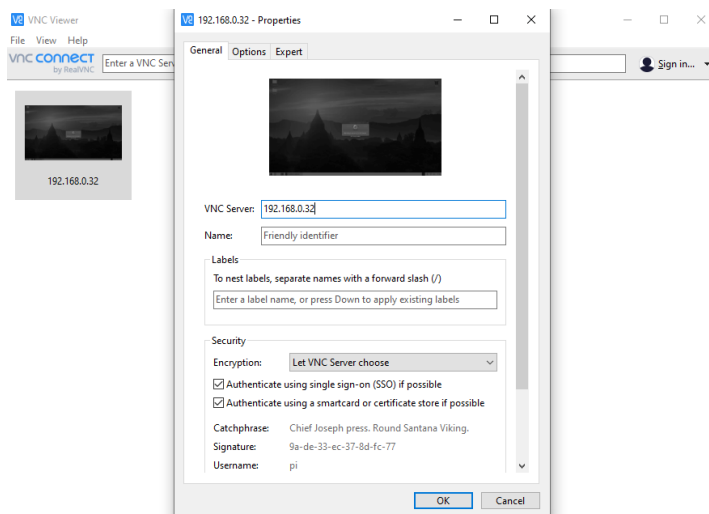
Ako se nakon podešavanja RSA ključeva pokuša spojiti preko SSH konekcije na *Raspberry Pi* može se primijetiti da unos šifre više nije potreban što olakšava prijenos podataka preko SCP protokola te generalno korištenje daljinskog razvoja na *Raspberry Pi* uređaju.

4.3.3. Konfiguracija VNC-a

Virtual Network Computing (VNC) je grafički sustav za dijeljenje radne površine koji koristi protokol udaljenog među-spremnika (engl. *Remote Frame Buffer*, RFB) za daljinsko upravljanjem drugih računala. Funkcionalnosti su kao na primjer prenošenje događaja mišem i tipkovnice sa jednog računala na drugo te mogućnost ažuriranja grafičkog zaslona u drugom smjeru [33].

Konfiguracija VNC aplikacije je isto poprilično jednostavna u usporedbi sa SSH konfiguracijom. Kao i prije što smo vidjeli u isječku konfiguracije 4.3.1. VNC postavka treba bit uključena. Nakon toga na lokalnom računalu preko interneta se preuzme VNC aplikacija te potom instalira na lokalno računalo. Sljedeći korak je pokrenuti aplikaciju i otići na sljedeće:

File -> New Connection...



Isječak konfiguracije 4.3.3 – Primjer konfiguracije VNC aplikacije za daljinsko upravljanje

Kao što se može vidjeti iz isječka konfiguracije 4.3.3. nakon odabira nove konekcije potrebno je unijeti IP adresu VNC servera što je u ovom slučaju to *Raspberry Pi* uređaj a on je pridobio IP adresu 192.168.0.32 kao što je spomenuto prije u koraku korištenja SSH veze. Pod uvjetom da je *Raspberry Pi* uređaj uključen i ima prijestup bežičnoj vezi sada je moguće se spojiti i kontrolirati radno okruženje samog uređaja preko daljinskog upravljanja. Time je završena konfiguracija radnog okruženja koji se koristio u ovom radu.

5. ANALIZA SUSTAVA

Ovo poglavlje će objasniti kako su pojedini dijelovi softverske strane sustava konfigurirani, na koji način rade, kako koriste razne funkcionalnosti *Raspberry Pi* platforme skupa sa svim komponentama korištenim u ovom radu. Osim toga provjerit će se performanse sustava koje su objavljene na službenim stranicama komponenti koje su se koristili u radu, kao na primjer na senzoru za analizu otiska prsta te će se ujedno testirati i zasebna implementacija analize otiska prsta koristeći javnu kodnu biblioteku „otvoreni računalni vid“ (engl. *open computer vision*, OpenCV).

Za stvaranje glavne logike upravljanja sustava, korištenja korisničkog sučelja i zasebne implementacije otiska prsta koristio se programski jezik *Python*. *Python* je dosta popularan jezik jer ima dosta funkcionalnosti ugrađeno u sebi sa dosta fleksibilnosti. Dosta se koristi u svrhe pisanja skripti i u područjima gdje se koristi umjetna inteligencija, strojno učenje itd. Zgodan je za rješavanje problema sa puno računanja ili kalkulacija. Još jedna prednost koja je uzrok visoke popularnosti ovog programskog jezika je količina javnih kodnih biblioteka koje se mogu koristiti za rješavanje raznih solucija. Jedna od tih biblioteka je korištena u ovom radu, a to je *OpenCV* te će se kroz ovo poglavlje detaljnije pogledati.

5.1. Funkcionalnost sustava

Glavna funkcionalnost koja se želi ostvariti sa ovim radom je korištenje gotovog *Raspberry Pi* sustava bez potrebe korištenja drugih vrsta računala kako bi se pokrenuo glavni program sustava. Način na koji se ovo može riješiti je isto poprilično jednostavno za konfigurirati. Nakon što se napiše glavni program koji kontrolira pokretanje i funkcionalnosti sustava mora se napraviti da se taj program izvrši odmah nakon uključivanja *Raspberry Pi* uređaja.

Jedan način na koji je ovo moguće napraviti je da se otvori specifična datoteka u *Raspbian* os-u. Koristeći tekstualni editor po vlastitom izboru izvršite sljedeću naredbu:

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

Ova datoteka je program koji se izvrši svaki put kad se operacijski sustav *Raspberry Pi* uređaja pokrene [34]. Unutar ove datoteke prije kraja programa može se napisati sljedeće:

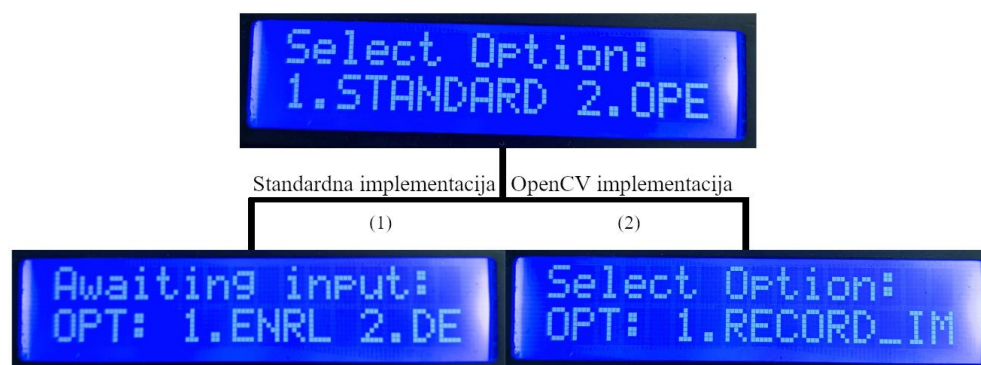
```
sudo bash -c 'python2  
/home/pi/Desktop/Fingerprint/Project/src/pi/main_startup.py >  
/home/pi/Debugging/Fingerprint/fingerprint.log 2>&1' &
```

Ovdje se može napisati bilo koji kod i on će se izvršiti pri pokretanju operativnog sustava *Raspberry Pi* uređaja, u ovom slučaju to je glavni program iz kojeg možemo kontrolirati funkcionalnost sustava. Primjer ovoga bi bio gašenje i ponovno pokretanje, pokretanje programa standardne solucije koja koristi sami senzor za analizu otiska prsta ili vlastite *OpenCV* solucije gdje se senzor koristi samo kao skener za skupljanje slika. Primjer *rc.local* datoteke može se vidjeti na isječku koda 5.1.1 i [34].

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
sudo bash -c 'python2 /home/pi/Desktop/Fingerprint/Project/src/rp/main_startup.py > /home/pi/Debugging/Fingerprint/fingerprint.log 2>&1' &
exit 0
```

Isječak koda 5.1.1 – Primjer *rc.local* datoteke

Prvi dio naredbe „*sudo bash -c*“ izvršava komandu koja je navedena a unutar same naredbe. Pokreće se *python2* program koji se nalazi na specificiranoj lokaciji, a u slučaju kod ovog rada to je na radnoj površini *Raspberry Pi* uređaja. Drugi dio naredbe omogućuje *logiranje* u tekstualnu datoteku kako se program izvršava. Ovo je korisno u slučaju istraživanja neočekivanih problema.



Slika 5.1.1 – Primjer korisničkog sučelja i izbor korištenja implementacija

Na slici 5.1.1 može se vidjeti ispis korisničkog sučelja na LCD zaslonu. Najgornji zaslon je kontrolni program koji se pokreće prilikom pokretanja operacijskog sustava preko prethodno spomenute *rc.local* datoteke. Dok su donji zaslone primjer standardne i *OpenCV* implementacije.

5.1.1. Kontroliranje korisničkog sučelja

Kao što se prethodno spomenulo za kontrolu korisničkog sučelja koriste se fizički botuni koji su spojeni na eksperimentalnoj ploči. Za korištenje ove funkcionalnosti potrebno je napisati kod koji može protumačiti promjenu signala nastalih na GPIO pinovima prilikom korištenja botuna.

Prvo što se napravilo je zasebna klasa zvana *userInput* kako bi se lakše koristio unutar različitih programa jer funkcionalnost korištenja fizičkih botuna za kontroliranje programa je slična bez obzira koji se program pokreće.

```
Project > src > rp > libs > userInput.py > userInput > handleUserInput
Unsaved changes (cannot determine recent change or authors)
1  from time import sleep
2
Unsaved changes (cannot determine recent change or authors)
3  class userInput:
4      def __init__(self):
5          self.user_input = 0
6          self.switcher = {
7              26: 1,
8              19: 2,
9              13: 3,
10             6: 4,
11             5: 5,
12         }
13
14     def setUserInput(self, channel):
15         self.user_input = self.switcher.get(channel)
16
17     def getUserInput(self):
18         return self.user_input
19
20     def handleUserInput(self):
21         print("Input: ")
22         while True:
23             self.user_input = 0
24             sleep(0.2)
25             if self.user_input != 0:
26                 print("User Input: ", self.user_input)
27                 break
28
```

Isječak koda 5.1.2 – Primjer *userInput* klase korištene za kontrolu korisničkog sučelja

Unutar *userInput* klase spremamo zadnju vrijednost botuna uz pomoć *switcher-a* koja predstavlja GPIO pin koji je aktiviran. U ovom slučaju kao što je spomenuto prikazano na shemi spoja sustava za korisničko sučelje koristilo se 6 GPIO pinova gdje su pinovi 26, 19, 13, 6 i 5 pinovi korišteni za aktivirati strujni krug pojedinog botuna i preostali GND pin koji se koristi za zatvaranje spoja. Ova klasa se zatim može pozvati u bilo kojem drugom *Python* programu sa preduvjetom da se kod *importa*. U ovom radu sve potrebne i korištene biblioteke su se stavile u „src/rp/libs“ direktorij. Kako bi se pozvala klasa potrebno je na početak *Python* skripte napisati sljedeće:

```
from libs.userInput import userInput
```


Sljedeća bitna stavka koja se mora softverski konfigurirati je korištenje GPIO pinova tako da se odabere način korištenja pinova. Također je potrebno povezati signale od GPIO pinova sa *callback* funkcijama koje će se aktivirati kada se detektira promjena naboja na pinovima. Primjer ovog a se može vidjeti na isječku koda 5.1.2.

```
11 # User Input - Input control
12 usr = userInput()
13
14 # User Input - Button callbacks (Activate when the button is pressed)
15 def button_callback(channel):
16     usr.setUserInput(channel)
17
18 def GPIO_setup():
19     # Set the GPIO Pin for the relay & push buttons to BMC mode (ex. Using 7th Pin aka BCM GPIO pin 4)
20     GPIO.setmode(GPIO.BCM)
21     GPIO.setup(4, GPIO.OUT, initial = GPIO.LOW)
22     GPIO.setup(5, GPIO.IN, pull_up_down = GPIO.PUD_UP)
23     GPIO.setup(6, GPIO.IN, pull_up_down = GPIO.PUD_UP)
24     GPIO.setup(13, GPIO.IN, pull_up_down = GPIO.PUD_UP)
25     GPIO.setup(19, GPIO.IN, pull_up_down = GPIO.PUD_UP)
26     GPIO.setup(26, GPIO.IN, pull_up_down = GPIO.PUD_UP)
27
28     # Event listeners that activate the callback functions (Bounce controls accidental button triggers)
29     GPIO.add_event_detect(26, GPIO.RISING, callback = button_callback, bouncetime=200)
30     GPIO.add_event_detect(19, GPIO.RISING, callback = button_callback, bouncetime=200)
31     GPIO.add_event_detect(13, GPIO.RISING, callback = button_callback, bouncetime=200)
32     GPIO.add_event_detect(6, GPIO.RISING, callback = button_callback, bouncetime=200)
33     GPIO.add_event_detect(5, GPIO.RISING, callback = button_callback, bouncetime=200)
```

Isječak koda 5.1.3 – Primjer korištenja *userInput* klase za kontrolu korisničkog sučelja

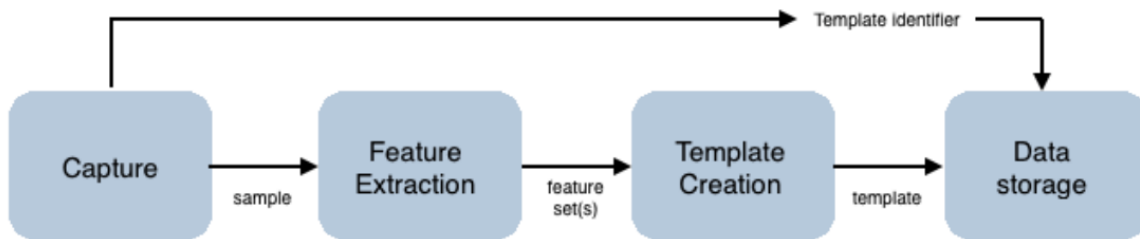
Za korištenje GPIO pinova na *Raspberry Pi* uređaju obavezno je odabrati način konfiguracije pinova, kao na primjer jeli se GPIO pin koristi za ulazne ili izlazne signale. Pošto pinove 5, 6, 13, 19, i 26 koristimo kako bi opisali ulazne signale koristimo argument *GPIO.IN* u koraku konfiguracije. Argument *pull_up_down* predstavlja unutrašnje otpornike koji su se prethodno spomenuli kako bi botuni ispravno radili.

Nakon konfiguracije GPIO pinova u radu se koristi funkcija koja se nalazi u standardnoj GPIO javnoj biblioteci, a to je *GPIO.add_event_detect*. Ova funkcija očitava promjenu električnog napona na pinu. U ovom slučaju kad napon poraste (kada se sklop zatvori pritiskom jednog botuna) promjena se detektira i aktivira se *callback* funkcija koja prima informaciju o kojem se pinu radi kao argument funkcije. Ova informacija se prosljeđuje kroz *userInput* klasu koja onda zatim zadaje određene vrijednosti različitim GPIO pinovima. Ta informacija se zatim sprema u klasi što je potrebno ako se želi očitati koji se botun aktivirao. Zadnja važna stavka je argument *bouncetime*, jer ako se ne podesi pravilno, signal od aktivacije pritiska botuna može se nepravilno očitati zbog vremena koji je potreban da se električni napon stabilizira. Podešavanje vrijednost ovog argumenta na 200 pokazalo se da uklanja sva nepravilna očitavanja aktiviranja fizičkih botuna.

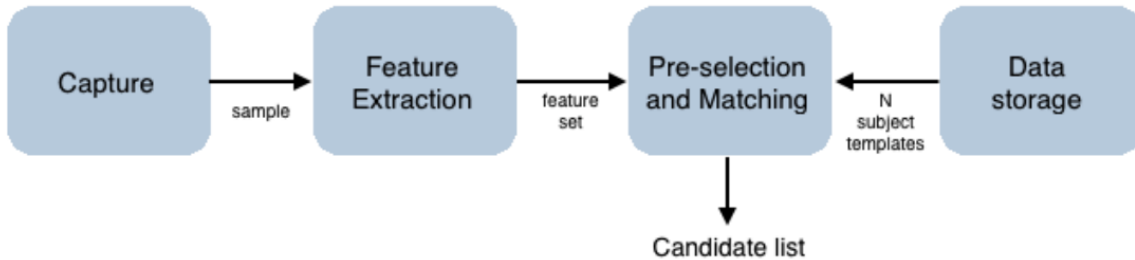
5.1.2. Standardna implementacija za analizu otiska prsta

Standardna implementacija predstavlja normalnu implementaciju gdje se koriste javne kodne biblioteke koje su napisane da rade sa nabavljenim senzorom u ovom radu (ZFM-708). Dostupne biblioteke imaju mogućnosti slanja poruka senzoru tako da pokrene akcije očitavanja otisaka prsta (engl. *read*) što se može koristiti kod spremanja otiska prsta (engl. *enroll*) na senzor sve sa ciljem kako bi se mogao očitati otisak prsta usporediti sa spremljenim otiskom prsta.

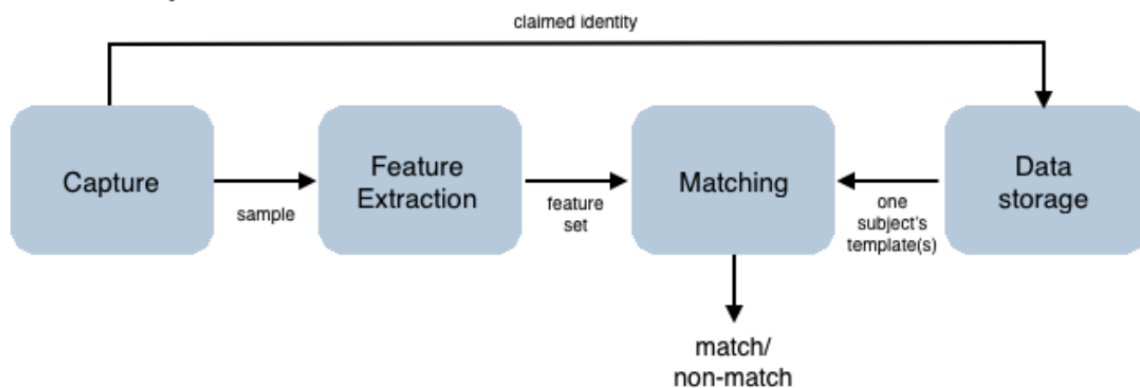
Enrollment process



Identification process



Verification process



Slika 5.1.2 – Pregled procesa spremanja, čitanja i uspoređivanja otisaka prsta [35]

Ovo se izvodi na način da se spremljeni otisak prsta spremi u jedan među-spremnik, a novo očitani otisak prsta se spremi u drugi među spremnik. Nakon toga se pošalje signal na senzor da se pokrene proces uspoređivanja dvaju među-spremnika. Rezultat ove operacije se šalje natrag sa informacijom pozicije spremljenog prsta i brojem koji reprezentira vjerojatnost i preciznost usporedbe u slučaju da je novo očitani otisak prsta uspješno uspoređen, te da odgovara jednom od potencijalno 1000 otisaka prstiju koji su spremljeni na sami senzor.

Program koji kontrolira ovaj dio implementacije se nalazi u „*src/rp/*“ direktoriju pod nazivom *fingerpring.py*. Unutar ove *Python* datoteke može se vidjeti implementacija u kojoj se koristi sam senzor za analizu otiska prsta.

5.1.3. OpenCV implementacija za analizu otiska prsta

OpenCV je javna biblioteka programskih funkcija koja je besplatna za upotrebu i može se koristiti na više platformi te je većinom usmjerena na računalni vid (engl. *computer vision*). U ovom radu su se iskoristili razni primjeri algoritama koji su napisani u *Python* programskom jeziku koristeći ovu javnu biblioteku, a primjer ovakve jedne implementacije se može vidjeti na [36].

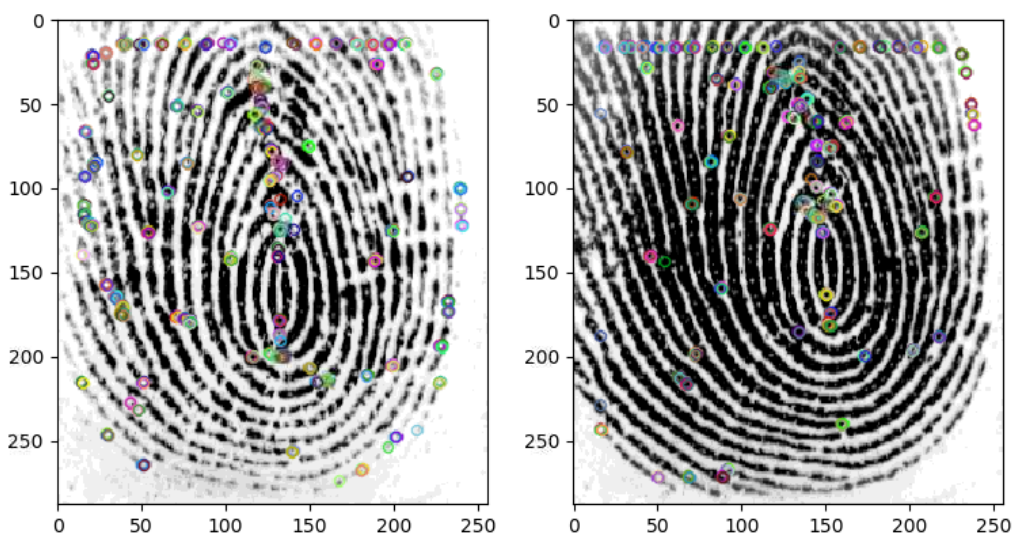
OpenCV implementacija otiska prsta i dalje koristi senzor za izvršavanje funkcionalnosti ali samo kako bi se napravio prvi korak spremanja otisaka prsta u obliku slike na *Raspberry Pi* uređaju. Implementacija ovog sustava može se naći u „*src/rp/*“ i „*src/openCV/*“ direktoriju, a lokacija spremanja slika otisaka prstiju je u sljedećem direktoriju: „*src/openCV/database/*“. Skidanje slika se vrši na način da se pošalje signal senzoru da aktivira proceduru očitavanja prsta, te nakon uspješnog očitavanja sprema sliku prvog otiska kao „*000.bmp*“, a drugog kao „*001.bmp*“.



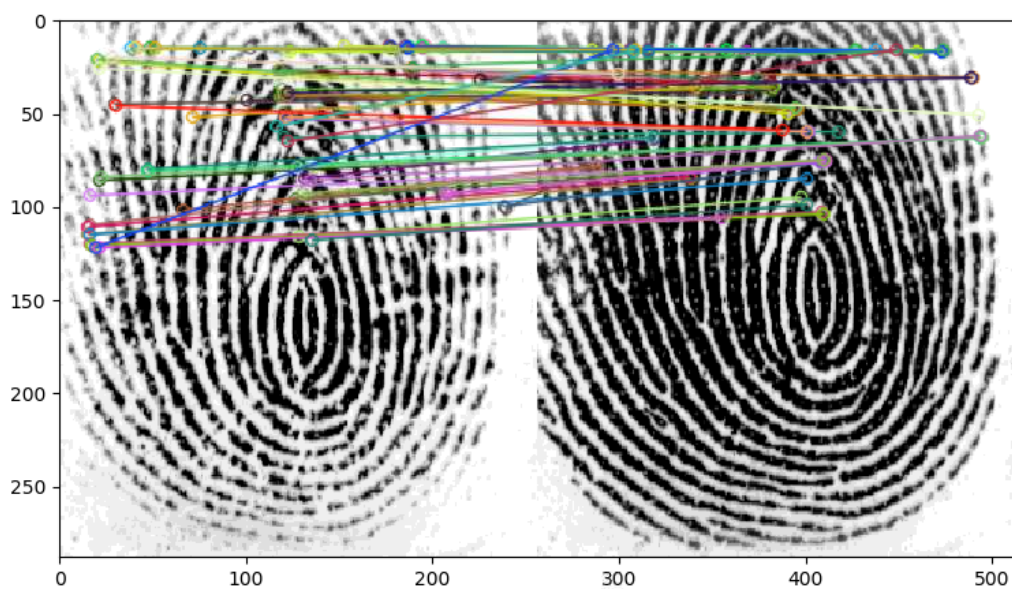
Slika 5.1.3 – Snimljeni otisci prstiju korišteni za usporedbu u *OpenCV* soluciji

Na slici 5.1.3 se može vidjeti 2 različita otiska prsta koja se mogu koristiti za uspoređivanje.

Nakon što se kroz korisničko sučelje pokrenula *OpenCV* implementacija, odabrana se prva opcija *1. RECORD_IMAGES* što prođe kroz proces snimanja dvaju slika otisaka prstiju. Nakon što se slike sprema može se odabrati opcija *3.COMPARE* što pokreće *OpenCV* aplikaciju gdje se korištenjem prethodno spomenutih algoritama segmentiranja (*Harris Corner Detection*) te klasificiranja ključnih točaka i deskriptora (SIFT, ORB) analiziraju koordinate i sve interesantne značajke, te se potom uspoređuju međusobno kako bi se procijenilo jeli se radi o istom otisku prsta.



Slika 5.1.4 – Primjer izvlačenja ključnih točaka u *OpenCV* implementaciji kod usporedbe 1 i 2 otiska prsta



Slika 5.1.5 – Primjer uspoređivanja ključnih točaka u *OpenCV* implementaciji kod usporedbe 1 i 2 otiska prsta

Na slici 5.1.4 može se vidjeti proces klasifikacije svakog otiska prsta, prvi otisak je dosta slabo osvijetljen jer se nije iskoristilo dovoljno pritiska na senzoru pri očitavanju otiska prsta dok je drugi otisak puno kvalitetniji. Algoritmi su i dalje uspješno našli ključne točke kao na primjer bifurkacije i krajeve grebena, te iskoristili tu informaciju kako bi započeli uspoređivanje. Na slici 5.1.5 prikazan je finalni proces uspoređivanja tako da su se usporedili deskriptori svakog otiska prsta da bi se zatim sortirali sve ključne točke. Odlučivanje jeli otisak prsta uspješno uspoređen ili ne obavlja se preko postavljenog praga kojeg računamo na kraju programa. Prag zavisi o sigurnosnoj razini koju želimo postići. Ako želimo imati jako siguran sustav onda se prag može spustiti ali zato kvaliteta otisaka prsta mora biti bolja kako bi se ključne točke preciznije i lakše našle. Ili ako želimo imati manje sigurniji sustav prag se može podići.

```
-----  
Calculating score  
Fingerprint matches.  
Score treshold: 40  
Score: 31
```

Slika 5.1.6 – Primjer rezultata usporedbe 1 i 2 otiska prsta

Na slici 5.1.6 možemo vidjeti izlaz programa koji kazuje vrijednost praga, konačnog rezultata i uspješnost uspoređivanja otisaka prstiju.

5.2. Performanse sustava

Za mjerenje performanse sustava, u radu se mjerilo vrijeme koje je potrebno da se uspješno odradi funkcionalnost senzora, a to znači da se gledalo vrijeme koje je potrebno da se uspoređi očitani otisak prsta sa spremljenim otiskom prsta.

Kod standardne implementacije sustava ovo se napravilo uz pomoć *Python time* biblioteke.

```
f = fp_init()  
start_time = time()  
if not fp_read(f): return  
if fp_authenticate(f)[0]:  
    print("Sucessful Execution time --- %s seconds ---" % (time() - start_time))  
    GPIO.output(4, GPIO.HIGH)  
    sleep(3)  
    GPIO.output(4, GPIO.LOW)  
    sleep(0.1)  
    return  
print("Faliure Execution time --- %s seconds ---" % (time() - start_time))
```

Slika 5.2.1 – Primjer mjerenja vremena izvršavanja uspoređivanja otisaka prstiju u Python-u

Eksperiment se napravio na način da se prst stavio na površinu senzora otiska prsta te se započeo proces usporedbe otiska prsta. Ovo je minimiziralo potrebno vrijeme aktivacije senzora da se dobije što točnija vrijednost vremena izvršavanja funkcionalnosti.

```
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 108
Successful Execution time --- 1.48390078545 seconds ---
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 108
Successful Execution time --- 1.35495901108 seconds ---
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 97
Successful Execution time --- 1.336840868 seconds ---
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 108
Successful Execution time --- 1.4080388546 seconds ---
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 108
Successful Execution time --- 1.43972277641 seconds ---
-----
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 108
Successful Execution time --- 1.40778779984 seconds ---
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 108
Successful Execution time --- 1.38887906075 seconds ---
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 74
Successful Execution time --- 1.29041790962 seconds ---
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 108
Successful Execution time --- 1.41012692451 seconds ---
-----
Input:
('User Input: ', 3)
Waiting for finger...
Found template at position #1
The accuracy score is: 108
Successful Execution time --- 1.38771295547 seconds ---
-----
```

Slika 5.2.2 – Izmjerene performanse standardne implementacije sustava za analizu otiska prsta

Na slici 5.2.2 može se vidjeti 10 pokušaja uspoređivanja otisaka prsta sa spremljenim otiskom prsta koji se nalazio na senzoru. U službenoj specifikaciji ZFM-708 senzorskog uređaja stoji da je srednja vrijednost očitavanja otiska prsta oko ~1 sekunde, dok u eksperimentu vidimo da je srednja vrijednost uspješnog očitavanja otiska prsta oko...

$$t = \frac{13.90838694573}{10} = 1.390838694573$$

Senzor je također testiran za ostale službene specifikacije kao na primjer mogućnost uspješnog očitavanja otisaka prsta uz rotaciju do +/- 45 stupnjeva. Nakon manualnog testiranja vidjelo se da se otisak prsta može uspješno usporediti čak do +/- 100 stupnjeva, uz pretpostavku da je kvaliteta spremljenog otiska prsta relativno dobra.

OpenCV solucija je također testirana kako bi se vidjela performansa sustava na *Raspberry Pi* platformi. Ovaj eksperiment se napravio na način da se senzor koristio samo kao uređaj za skeniranje otisaka prsta kao što je objašnjeno u prethodnim poglavljima. Nakon spremanja otisaka prstiju pokreće se *OpenCV* implementacija uspoređivanja otisaka prstiju pomoću spomenutih algoritama za klasifikaciju i uspoređivanje. Rezultati se mogu vidjeti na slici 5.2.3.

```

-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 35
Execution time --- 51.819562912 seconds ---
-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 33
Execution time --- 47.6804897785 seconds ---
-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 33
Execution time --- 47.6955940723 seconds ---
-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 25
Execution time --- 47.9868628979 seconds ---
-----
Calculating score
Fingerprint does not match.
Score threshold: 40
Score: 41
Execution time --- 47.7760698795 seconds ---
-----
-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 30
Execution time --- 48.1309862137 seconds ---
-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 31
Execution time --- 47.7242569923 seconds ---
-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 33
Execution time --- 47.1467950344 seconds ---
-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 22
Execution time --- 47.6194238663 seconds ---
-----
Calculating score
Fingerprint matches.
Score threshold: 40
Score: 21
Execution time --- 47.0710248947 seconds ---
-----

```

Slika 5.2.3 – Izmjerene performanse *OpenCV* sustava za analizu otiska prsta

Može se primijetiti da ova specifična implementacija uspoređivanja otisaka prstiju je znatno sporija od prijašnje implementacije, a razlog toga može biti zbog načina kako se koristila *OpenCV* implementacija skupa sa slabijim hardverom *Raspberry Pi* uređaja.

$$t = \frac{481.4510673416}{10} = 48.14510673416$$

U ovom slučaju nema već prethodno spremljenih ključnih točaka od jednog otiska prsta pa se slike moraju po redu obraditi kako bi se na kraju napravio proces uspoređivanja. Najviše se vremena potroši na poboljšavanje slike dok je sam proces uspoređivanja poprilično kratak. Uzimajući u obzir kompleksnost procesa poboljšavanja slike s lošijim brzinama *Raspberry Pi* platforme može se doći do zaključka da je potrebno optimizirati kod za bolju performansu sustava.

5.3. Moguća rješenja

Koncept korištenja *Raspberry Pi* uređaja sa senzorom za analizu otiska prsta se može uzeti kao jedinstveni alat koji se može koristiti u razne svrhe. Prvi primjer mogućeg rješenja ovakvog sustava je korišten u ovom radu, a to je bio elektro-magnetski lokot koji se može aktivirati preko relej modula. Aktivacijom releja otključava se lokot, a ovo se izvršava na način da se u kodu promjeni stanje GPIO pina na kojem je relej priključen. Ovo se može vidjeti na slici 5.2.1. Mogućnost korištenja GPIO pinova na ovaj način dozvoljava korištenje bilo kakvog strujnog kruga spojenog na relej, pa time se efektivno omogućava programibilnost svijeta oko samog uređaja.

Druga moguća solucija bi bila korištenje senzora za otključavanje privatne datoteke na udaljenom serveru preko interneta slanjem zahtjeva kroz aplikacije.

Treća moguća solucija bi bila povećanje sigurnosti korištenja 3D printera. 3D printeri mogu biti kontrolirani *Raspberry Pi* platformom. Ovo bi onemogućilo neovlašteno korištenje skupih uređaja.

Četvrta moguća solucija bi bila implementacija ovakvog sustava u pametnoj kući tako da se dozvoljava ovlašteno korištenje računala, Internet rutera, kućanskih aparata, različitih senzora za snimanje kako bi se povećala sigurnost doma.

Peta moguća solucija bi mogla biti implementacija ovakvog sustava u automobilima kako bi se osigurala ovlašteno korištenje vozila te zaštita od krađe.

Generalni koncept je uvijek isti, i to je da u slučaju kad se želi ili mora povećati sigurnost, sustav koji radi na principu provjere identiteta uz pomoć biometrije je puno sigurniji, a pogotovo u slučaju otisaka prsta jer je znatno teže probiti sigurnost sa tehnikama lažiranja prsta od ostalih načina sigurnosti.

6. ZAKLJUČAK

Nekada se cijeli ovakav sustav morao specijalno dizajnirati od početka do kraja kao jedan uređaj kako bi se isporučio na tržište. Potreba za sigurnosti i kompleksnosti ovakvog sustava uzrokovalo je da njegova cijena bude poprilično visoka. Danas, usred strelovitog razvoja IT-a i povećanja opcija na tržištu radi modularnog prijestupa rješavanja problema postoji puno više slobode i mogućnosti za izgradnju sličnog sustava.

Za razliku od prije, danas jedan takav sustav koji može biti namijenjen za rješavanje nekakvog problema može se riješiti uz pomoć programibilne i jako funkcionalne prirode *Raspberry Pi* platforme jer se olakšava pristup mogućnosti izgradnje i nabave komponenata koje se mogu koristiti sa platformom.

Kroz ovaj rad se također uspjelo vidjeti da je moguće napraviti vlastitu implementaciju senzora za analizu otiska prsta, ali treba uzeti u obzir da je dosta truda potrebno uložiti kako bi se povećala efikasnost funkcionalnosti analize otiska prsta. Brzina izvršavanja dosta ovisi o poboljšavanju slike otiska prsta i odabiru algoritama za segmentaciju, klasifikaciju i uspoređivanje, također brzina izvršavanja može ovisiti o načinu spremanja potrebnih podataka i o brzini hardvera.

Mogućnost korištenja *Raspberry Pi* platforme uz pomoć bežične internetske veze omogućava mnogo zgodnih prijenosnih rješenja koja se dosada nisu sa lakoćom mogla izraditi.

Ako se uz navedeno u obzir uzme i otvoren način dizajniranja softvera i mogućnosti da razni sudionici širom svijeta mogu pridonositi inovativna rješenja sa raznim idejama skupa s upotrebom novih tehnologija i tehnika, otvaraju se vrata za buduća poboljšanja i nove funkcionalnosti pa se time može s lakoćom zaključiti da ovoj platformi nema granica.

LITERATURA

- [1] Packt>, »History and background of the Raspberry Pi,« [Mrežno]. Available: https://subscription.packtpub.com/book/hardware_and_creative/9781784397975/1/ch011vl1sec08/history-and-background-of-the-raspberry-pi. [Pokušaj pristupa 27 Kolovoz 2020].
- [2] RaspberryTips, »The awesome story of Raspberry Pi,« [Mrežno]. Available: <https://raspberrytips.com/raspberry-pi-history/>. [Pokušaj pristupa 27 Kolovoz 2020].
- [3] Wikipedia, »Raspberry Pi,« 20 Kolovoz 2020. [Mrežno]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi.
- [4] O. Hughes, »Raspberry Pi sales are rocketing in the middle of the coronavirus outbreak,« 17 Travanj 2020. [Mrežno]. Available: <https://www.techrepublic.com/article/raspberry-pi-why-sales-have-rocketed-in-the-middle-of-the-coronavirus-outbreak/>. [Pokušaj pristupa 27 Kolovoz 2020].
- [5] R. Pi, »Raspberry Pi Products,« 20 Kolovoz 2020. [Mrežno]. Available: <https://www.raspberrypi.org/>.
- [6] B. Nuttall, »What is a Raspberry Pi?,« opensource.com, 19 Veljača 2019. [Mrežno]. Available: <https://opensource.com/resources/raspberry-pi>. [Pokušaj pristupa 28 Kolovoz 2020].
- [7] »The Pi4J Project,« 20 Kolovoz 2020. [Mrežno]. Available: <https://pi4j.com/1.2/pins/model-3b-plus-rev1.html>.
- [8] T. HARRIS, »How Fingerprint Scanners Work,« HowStuffWorks, [Mrežno]. Available: <https://computer.howstuffworks.com/fingerprint-scanner.htm#pt4>. [Pokušaj pristupa 26 Kolovoz 2020].
- [9] S. B. S. V. V.M. Praseetha, »Secure Fingerprint Authentication Using Deep Learning and Minutiae Verification,« De Gruyter, 10 April 2019. [Mrežno]. Available: https://www.degruyter.com/view/journals/jisys/29/1/article-p1379.xml?language=en#j_jisys-2018-0289_ref_013. [Pokušaj pristupa 26 Kolovoz 2020].
- [10] R. Triggs, »How fingerprint scanners work: optical, capacitive, and ultrasonic variants explained,« Android Authority, 28 Ožujak 2019. [Mrežno]. Available: <https://www.androidauthority.com/how-fingerprint-scanners-work-670934/>. [Pokušaj pristupa 26 Kolovoz 2020].
- [11] H. Nguyen, »How Fingerprint Sensors Work,« 08 Studeni 2016. [Mrežno]. Available: <https://www.ubergizmo.com/articles/fingerprint-scanners-how-they-work/>. [Pokušaj pristupa 26 Kolovoz 2020].

- [12] T. Lee, »Sonavation Successfully Embeds Fingerprint Sensor Under A Display,« 22 Srpanj 2015. [Mrežno]. Available: Sonavation Successfully Embeds Fingerprint Sensor Under A Display. [Pokušaj pristupa 26 Kolovoz 2020].
- [13] top5reviews, »The 5 Best Biometric Fingerprint Door Locks,« [Mrežno]. Available: <https://www.top5reviewed.com/biometric-fingerprint-door-lock/>. [Pokušaj pristupa 26 Kolovoz 2020].
- [14] N. Zaeri, »Minutiae-based Fingerprint Extraction and Recognition,« 9 Lipanj 2011. [Mrežno]. Available: <https://www.intechopen.com/books/biometrics/minutiae-based-fingerprint-extraction-and-recognition>. [Pokušaj pristupa 26 Kolovoz 2020].
- [15] P. ENIGMA, »Fingerprint Recognition,« [Mrežno]. Available: <https://prateekvjoshi.com/2012/07/22/fingerprint-recognition/>. [Pokušaj pristupa 26 Kolovoz 2020].
- [16] S. T. V. G. Chaohong Wu, »Robust Point-Based Feature Fingerprint Segmentation,« [Mrežno]. Available: https://www.acsu.buffalo.edu/~tulyakov/papers/Wu_ICB07_Segmentation.pdf. [Pokušaj pristupa 28 Kolovoz 2020].
- [17] M. K. a. A. Tojo, »Fingerprint pattern classification,« u *Pattern Recogn.* 17, 1984, p. 295–303.
- [18] K. R. a. K. Balck, »Type classification of fingerprints: a syntactic approach,« u *IEEE Trans. Pattern Anal. Mach. Intell.* 2, 1980, p. 223–231.
- [19] D. M. a. D. Maltoni, »A structural approach to fingerprint classification,« u in: *Proceedings of the 13th International Conference on Pattern Recognition*, IEEE, 1996, p. 578–585.
- [20] A. R. a. D. S. Palvee, »SVM based classification and improvement of fingerprint verification,« u *Int. J. Sci. Eng. Technol. Res* 3, 2014, p. 879–883.
- [21] X. Y. X. J. a. J. H. Y. Zhu, »Latent fingerprint segmentation based on convolutional neural networks,« u in: *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, IEEE, 2017, p. 1–6.
- [22] Y. G. A.-. D. E. R. S. K. N. a. T. B. M. D. Michelsanti, »Fast fingerprint classification with deep neural network,« u in: *International Conference on Computer Vision Theory and Applications*, 2018.
- [23] S. P. M. S. Ebrahim Karami, »Image Matching Using SIFT, SURF, BRIEF and,« [Mrežno]. Available: <https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf>. [Pokušaj pristupa 28 Kolovoz 2020].
- [24] L. Zhiantec Technologies Co., »ZFM-70 Series Fingerprint Identification Module,« [Mrežno]. Available:

- https://www.velleman.eu/downloads/29/infosheets/vma329_datasheet.pdf. [Pokušaj pristupa 22 Kolovoz 2020].
- [25] P. Meisberger, »ZFM fingerprint sensors,« [Mrežno]. Available: <https://github.com/bastianraschke/pyfingerprint>.
- [26] R. Pi, »Raspberry Pi Tutorials,« [Mrežno]. Available: <https://tutorials-raspberrypi.com/how-to-use-raspberry-pi-fingerprint-sensor-authentication/>. [Pokušaj pristupa 22 Kolovoz 2020].
- [27] R. Pi, »The Pi Hut,« [Mrežno]. Available: <https://thepihut.com/products/i2c-16x2-arduino-lcd-display-module>. [Pokušaj pristupa 22 Kolovoz 2020].
- [28] M. T.-B. (. P. Guy), »LCD (Git Hub),« [Mrežno]. Available: <https://github.com/the-raspberry-pi-guy/lcd>. [Pokušaj pristupa 22 Kolovoz 2020].
- [29] Fritzing, »Slika LCD Zaslona,« [Mrežno]. Available: <https://diyi0t.com/lcd-display-tutorial-for-arduino-and-esp8266/>. [Pokušaj pristupa 24 Kolovoz 2020].
- [30] R. Pi, »Raspperri Pi Stack Exchange,« [Mrežno]. Available: <https://raspberrypi.stackexchange.com/questions/42735/wiring-diagram-software>. [Pokušaj pristupa 24 Kolovoz 2020].
- [31] R. Pi, »Raspberry Pi Installation,« Raspberry Pi, [Mrežno]. Available: <https://www.raspberrypi.org/documentation/installation/>. [Pokušaj pristupa 24 Kolovoz 2020].
- [32] S. Academy, »SSH-keygen - Generate a New SSH Key,« [Mrežno]. Available: <https://www.ssh.com/ssh/keygen/>. [Pokušaj pristupa 26 Kolovoz 2020].
- [33] Wikipedia, »Virtual Network Computing,« [Mrežno]. Available: https://en.wikipedia.org/wiki/Virtual_Network_Computing. [Pokušaj pristupa 26 Kolovoz 2020].
- [34] D. Industries, »Run a Program On Your Raspberry Pi At Startup,« [Mrežno]. Available: <https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/>. [Pokušaj pristupa 27 Kolovoz 2020].
- [35] E. P. Linda Xiang, »1:N fingerprint classification,« Srpanj 2018. [Mrežno]. Available: <https://lup.lub.lu.se/luur/download?func=downloadFile&recordId=8954340&fileId=8954341>. [Pokušaj pristupa 28 Kolovoz 2020].
- [36] K. Jankoski, »python fingerprint recognition with OpenCV - Git Hub,« [Mrežno]. Available: <https://github.com/kjanko/python-fingerprint-recognition>. [Pokušaj pristupa 22 Kolovoz 2020].

POPIS OZNAKA I KRATICA

engl	engleski
DNA	deoxyribonucleic acid
GPIO	general purpose input/output
USB	universal serial bus
HDMI	high-definition multimedia interface
CPU	central processing unit
GPU	graphics processing unit
RAM	random-access memory
OS	operating system
MB	megabyte
GB	gigabyte
SD	secure digital
MMC	multi-media card
eMMC	embedded multi-media card
SDHC	secure digital high capacity
SDIO	secure digital input output
NOOBS	new out of box software
SoC	system on a chip
CM	compute module
CMIO	compute module input output
DDR2	double data rate 2
SO-DIMM	small outline dual in-line memory module

Mhz	megahertz
Ghz	gigahertz
VDC	volts direct current
MOSI	master out slave in
MISO	master in slave out
MOM	master out master in
SCLK	serial clock
SPI	serial peripheral interface
UART	universal asynchronous receiver transmitter
RTC	real time clock
I2C	inter-IC
DPI	display parallel interface
GPCLK	general purpose clock
PWM	pulse width modulation
TxD	transmit data
RxD	receive data
EEPROM	electrically erasable programmable read-only memory
LED	light emitting diode
CCD	charged coupled device
CMOS	complementary metal oxide semiconductor
2D	two dimensional
3D	three dimensional
RF	radio frequency

AI	artificial intelligence
NIST	national institute of standards and technology
SVM	support vector machine
KNN	k-nearest neighbor
CNN	convolutional neural network
FAST	features from accelerated segment Test
BRIEF	binary robust independent elementary features
SIFT	scale invariant feature transform
SURF	speed up robust feature
ORB	oriented FAST and rotated BRIEF
TTL	transistor-transistor logic
WIFI	wireless fidelity
SSH	secure shell
RSA	Rivest–Shamir–Adleman
IT	information technology
VNC	virtual network computing
GUI	graphical user interface
VNC	virtual network computing
RFA	remote frame buffer
DC	direct current
AC	alternating current
SCP	secure copy protocol

SAŽETAK

Sustav za analizu otiska prsta na *Raspberry Pi* platformi sastoji se od tri osnovna elementa. Ti elementi su koncept biometrije, senzora za očitavanje i analizu otiska prsta te *Raspberry Pi* uređaj koji služi kao kontrolna jedinica cijelog sustava.

Biometrija kao koncept se bazira na fizičkim ljudskim karakteristikama koje se mogu iskoristiti za digitalnu identifikaciju i verifikaciju identiteta osobe kako bi se omogućio pristup nekom sigurnom sustavu, uređaju ili podacima. S obzirom na brzi razvoj informacijskih tehnologija kroz prošlo desetljeće, mogućnost korištenja koncepta biometrije u svakodnevne svrhe je postalo mnogo pristupačnije. Zbog toga je u okviru ovog diplomskog rada prikazan proces izrade i korištenja sustava za analizu otiska prsta na *Raspberry Pi* platformi. *Raspberry Pi* platforma nudi dosta prednosti i korisnih funkcionalnosti koje omogućavaju fleksibilno korištenje ovakvog sustava u raznim mjestima na puno različitih načina.

Za očitavanje uzoraka otiska prsta koristi se senzorski modul koji može komunicirati sa *Raspberry Pi* uređajem. Senzor za analizu otiska prsta koristi principe biometrije na način da se kod očitano uzorka otiska prsta provodi proces skupljanja i spremanja jedinstvenih ključnih točaka. Nakon te prve faze, ključne točke od novo očitanih uzoraka se uspoređuju sa prijašnje spremljenim uzorkom kako bi se izvršila verifikacija identiteta osobe. Ovaj proces se izvršava uz pomoć raznih tehnika i algoritama prepoznavanja uzoraka. U sklopu diplomskog rada realiziran je sustav za analizu otiska prsta na *Raspberry Pi* platformi korištenjem svih funkcionalnosti koje su ugrađene kroz sami senzorski modul i samostalne solucije za analizu otiska prsta uz pomoć *OpenCV* javne biblioteke koja se može koristiti za implementaciju spomenutih tehnika i algoritama prepoznavanja uzoraka.

Mogu se vidjeti usporedbe korištenja i izgradnje dvaju pojedinačnih solucija za analizu otiska prsta te povoljniji i pristupačniji način korištenja *Raspberry Pi* platforme u ovakve svrhe, za razliku od nekih već gotovih solucija na tržištu.

Ključne riječi

Biometrija, *Raspberry Pi*, *OpenCV*, Senzor, Otisak prsta

SUMMARY

Title: Development of fingerprint recognition sensor

The fingerprint analysis system on the Raspberry Pi platform consists of three basic elements. These elements are concept of biometrics, sensor for reading and analyzing fingerprints, and the Raspberry Pi device that serves as the control unit of the entire system.

Biometrics as a concept is based on physical human characteristics that can be used to digitally identify and verify a person's identity to allow access to a secure system, device, or data. Given the rapid development of information technology over the past decade, the possibility of using the concept of biometrics for everyday purposes has become much more affordable. Therefore, in this thesis, the process of creating and using a fingerprint analysis system on the Raspberry Pi platform is presented. The Raspberry Pi platform offers a lot of advantages and useful functionalities that allow flexible use of such system in various places in many ways.

A sensor module that can communicate with the Raspberry Pi is used to read fingerprint samples. The fingerprint analysis sensor uses the principles of biometrics in such a way that the process of collecting and storing unique key points is performed on the initial fingerprint sample. After this first phase, the key points from the newly read fingerprint samples are compared with the previously saved sample to verify if they match and complete the verification process. This process is performed using various pattern recognition techniques and algorithms. As part of the thesis, a fingerprint analysis system was implemented on the Raspberry Pi platform using all functionalities built in the sensor module, and standalone fingerprint analysis solution with the help of OpenCV public library that can be used to implement previously mentioned techniques and pattern recognition algorithms.

Comparisons of the use and construction of two individual fingerprint analysis solutions can be seen, as well as a more favorable and affordable way of using the Raspberry Pi platform for such purposes as opposed to some ready-made solutions on the market.

Key words

Biometrics, Raspberry Pi, OpenCV, Fingerprint, Sensor