

SKLOPOVSKA PLATFORMA ZA KONTROLU PRISTUPA

Čerina, Ivan

Master's thesis / Specijalistički diplomski stručni

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:228:654130>

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

Download date / Datum preuzimanja: **2024-11-28**



Repository / Repozitorij:

[Repository of University Department of Professional Studies](#)



UNIVERSITY OF SPLIT



SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Elektrotehnike

IVAN ČERINA

ZAVRŠNI RAD

**SKLOPOVSKA PLATFORMA ZA KONTROLU
PRISTUPA**

Split, rujan 2019

SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Elektrotehnike

Predmet: Senzorske mreže

ZAVRŠNI RAD

Kandidat: Ivan Čerina

Naslov rada: Sklopovska platforma za kontrolu pristupa

Mentor: dr.sc. Tonko Kovačević, profesor V. Š.

Split, rujan, 2019.

Sažetak

Sklopovska platforma za kontrolu pristupa

U ovom radu opisan je princip otključavanja električne brave koristeći otisak prsta te sigurnosnu lozinku. Kako bi se postigao takav princip otključavanja potreban je senzor otiska prsta te tipkovnica za upisivanje sigurnosne lozinke. Za izradu sustava uz senzor otiska prsta, tipkovnicu i električnu bravu, koriste se: Arduino Uno, Arduino sučelje, LCD zaslon te napajanje za električnu bravu od 12 volta. Definirani su protokoli korišteni u sustavu, objašnjen pojam biometrije i uzimanja otiska prsta, te je opisan je proces izrade Arduino sučelja te testiranje funkcionalnosti. Za potrebe ovog rada sustava izrađena je minijaturna maketa koja demonstrira rad sustava u cijelosti.

Summary

Hardware platform for access control

This paper describes the principle of unlocking an electric lock using a fingerprint and a security password. To achieve this unlocking principle, a fingerprint sensor and a keyboard are required to enter a security password. To build the system besides a fingerprint sensor, keyboard and power lock, the following parts are used: Arduino Uno, Arduino shield, LCD screen and 12 volt power supply for electric lock. The protocols used in the system were defined, the concept of biometrics and fingerprint acquisition was explained, and the process of creating an Arduino shield and testing the functionality were described. For the purpose of this system operation, a miniature model has been created that demonstrates the operation of the system as a whole.

SADRŽAJ

1.	UVOD	1
2.	KOMPONENTE SUSTAVA.....	2
2.1.	Arduino	2
2.1.1.	Tehničke karakteristike	3
2.2.	Tipkovnica	3
2.3.	LCD zaslon	7
2.4.	Električna brava	9
2.5.	Senzor otiska prsta	10
2.5.1.	Specifikacije senzora.....	11
2.6.	Jedinica za napajanje.....	12
3.	KOMUNIKACIJSKI PROTOKOLI.....	13
3.1.	Serijski komunikacijski protokol	13
3.2.	SPI (engl. Serial Peripheral Interface)	16
3.3.	I ² C protokol (engl. Inter-Integrated Circuit).....	19
4.	TEHNOLOGIJE ZA KONTROLU PRISTUPA.....	23
4.1.	Biometrika.....	23
4.2.	Princip rada biometrije.....	24
4.2.1.	Digitalizacija	24
4.2.2.	Fourierova transformacija	25
4.2.3.	Umjetna inteligencija	25
4.3.	Biometrijske tehnike	26
4.3.1.	Fizička biometrija.....	26
4.3.2.	Biometrija ponašanja.....	27
4.3.3.	Multimodalna biometrija.....	27

4.3.4.	Usporedba biometrijskih tehnika	28
4.4.	Otisak prsta	31
4.4.1.	Sustavi i algoritmi za raspoznavanje otiska prsta.....	33
4.4.2.	AFIS(<i>Sustav automatizirane identifikacije otiska prsta</i>)	34
4.4.3.	Prednosti i nedostaci AFIS-a.....	35
5.	ARHITEKTURA SUSTAVA	37
6.	ARDUINO SUČELJE.....	40
6.1.	Dizajn Arduino sučelja	42
6.2.	Testiranje Arduino sučelja	43
7.	MAKETA SUSTAVA I TESTIRANJE KOMPONENTI	45
7.1.	Maketa sustava.....	45
7.2.	Kodovi za testiranje komponenti	46
7.2.1.	Kod za testiranje Arduino sučelja	46
7.2.2.	Kod za testiranje tipkovnice.....	47
7.2.3.	Kod za testiranje senzora otiska prsta	51
8.	ZAKLJUČAK	58
9.	LITERATURA.....	59
10.	POPIS SLIKA I TABLICA	60

1. UVOD

Razvojem tehnologije dolazi do potrebe da se „uobičajene“ radnje digitaliziraju. Uređaji se sve više primjenjuju u svrhu olakšavanja života. Jedan od primjera implementacije novih tehnologija su tzv. „pametne kuće“, ili „sistem kućne automatike“. Pod tim nazivom podrazumijeva se integracija raznih električnih kućanskih uređaja kojima se može upravljati na različite načine (bluetooth, wi-fi, itd.). Ovaj rad se temelji na jednoj od tehnologija koja se koristi u pametnim kućama, a to je otključavanje vrata pomoću električne brave. Električna brava se može otključati na razne načine, koristeći senzor otiska prsta, tipkovnicom, RFID karticom, itd. RFID (engl. *Radio frequency identification*) je tehnologija koja pomoću magnetskog polja komunicira između prijemnika i predajnika bežičnim putem. RFID tehnologija ima široku primjenu u današnjem svijetu, počevši od bankarstva do jednostavnih izvedbi kao što su već spomenuto otključavanje električne brave pomoću kartice. No, tu se javlja problem sigurnosti, što ako se izgubi kartica? Dakle onaj tko je nađe ima otvoren pristup svemu onom što ona može otvoriti i javlja se problem sigurnosti. Ovaj rad se temelji na izradi sustava pametnog otključavanja uz pomoć senzora otiska prsta i tipkovnice. Cilj je osposobiti komponente potrebne za rad sustava, testirati njihovu ispravnost te spojiti u funkcionalnu cjelinu. Komponente potrebne za izradu sustava su: senzor otiska prsta, tipkovnica tipa 3x4, LCD (engl. *Liquid crystal display*) ekran, Arduino Uno, električna brava, Arduino sučelje, napajanje za električnu bravu i Arduino Uno te maketa koji demonstrira rad cjeline. Sustav također teži jednostavnošću, te onom najbitnijem a to je sigurnost.

2. KOMPONENTE SUSTAVA

2.1. Arduino

Arduino je tehnologija otvorenog koda čija je platforma zasnovana na fleksibilnim, te jednostavnim za korištenje, sklopovlju i programskoj podršci. Namijenjen je za umjetnike, dizajnere, i sve zainteresirane koji žele stvarati interaktivne objekte ili okruženje. Arduino platforma konstantno se razvija te danas postoji veliki broj različitih Arduino mikrokontrolerskih pločica koje se koriste za različite svrhe. Osnovni model je Arduino Uno R3 prikazan je na slici 1, često se nadograđuje, te danas je aktivna njegova treća izvedba. Jedna od glavnih odlika Arduino proizvoda temelji se na jednostavnosti korištenja. Za programiranje Arduina koristi se C++ programski jezik koji je prilagođen Arduino platformama.



Slika 1 - Arduino Uno

Arduino Uno je mikrokontrolerska ploča koja radi na principu Atmega328 mikrokontrolera. Ima 14 digitalnih izlazno /ulaznih pinova od kojih se 6 može koristiti kao PWM izlazi, 6 analognih ulaza, 16 MHz keramički otpornik, USB priključak, priključak za napajanje, ICSP zaglavlje, i tipku za reset. Sadržava sve što je potrebno kao podrška mikrokontroleru; spaja se na računalo pomoću USB kabela te ima mogućnost napajanja sa AC/DC adapterom ili baterijama. Za razliku od prethodnih Arduino ploča Arduino Uno ne koristi FTDI USB čip za serijsku komunikaciju. Umjesto toga Atmega 16u2 je programiran kao USB-serijski pretvarač. Atmega 328 ima kapacitet memorije 32 KB (sa

0,5 KB iskoristivih pri podizanju). Također još ima i 2 KB SRAM memorije i 1 KB EEPROM memorije (može biti čitana i zapisivana sa EEPROM bibliotekom).

2.1.1. Tehničke karakteristike

U tablici 1 prikazane su tehničke karakteristike Arduina.

Tablica 1 - Tehničke karakteristike Arduina

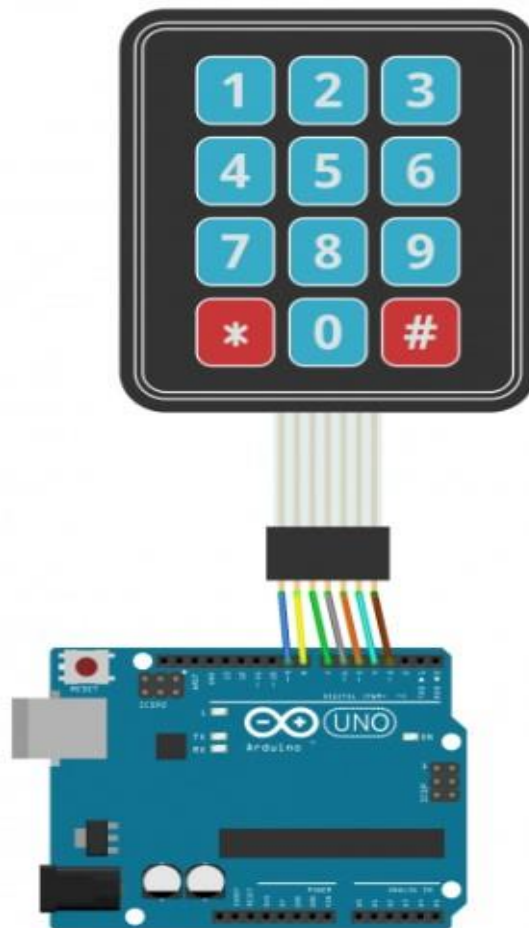
Parametar	Vrijednost
Mikrokontroler	Atmega328
Radni napon	5V
Ulazni napon (preporučeno)	7 - 12V
Ulazni napon (ograničenje)	6-20V
Digitalni I/O Pinovi	14 (od čega 6 omogućava PWN izlaz)
Analogni Input Pinovi	6
DC Struja za I/O Pin	40 mA
DC Struja za 3.3V Pin	50 mA
Flah Memorija	32 KB (Atmega328) od čega se 0.5 KB koristi pri učitavanju i podizanju sustava
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Brzina	16 MHz

2.2. Tipkovnica

Tipkovnica služi kao potpora senzoru. Ukoliko senzor nije u funkciji tipkovnica se koristi kao dodatna mogućnost otključavanja električne brave. U ovom projektu koristi se 3x4 tipkovnica koja se također spaja s Arduinom. Ispod svake tipke nalazi se membranski

prekidač. Svaki prekidač u nizu povezan je s drugim prekidačima u redu provodljivim tragom ispod podloge. Svaki prekidač u stupcu je spojen na isti način - jedna strana prekidača je spojena na sve ostale prekidače u tom stupcu vodljivim tragom. Svaki redak i stupac prikazuju se na jednom pinu, ukupno 7 pinova na 3X4 tipkovnici.

Tipkovnica ima 7 pinova koje je potrebno spojiti na Arduino, što je prikazano na slici 2.

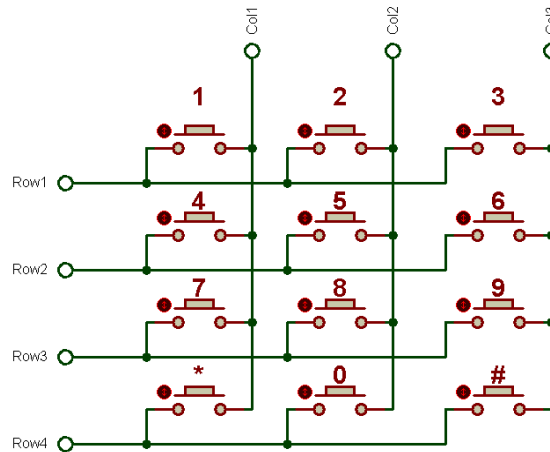


Slika 2 - Spajanje tipkovnice s Arduinoom

Princip spajanja tipkovnice s Arduinoom nalazi se na adresi:

<http://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/>

Slika 3 prikazuje shemu tipkovnice.

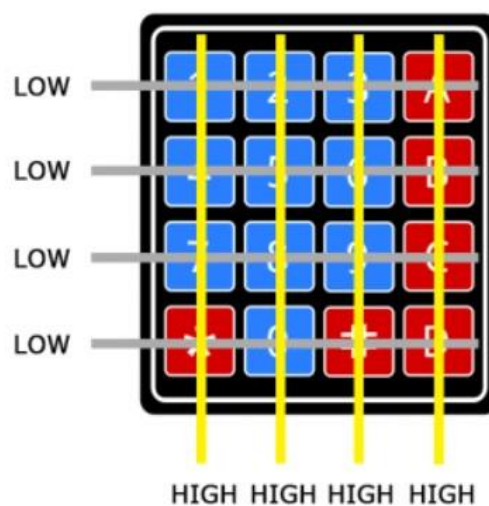


Slika 3 - Shema tipkovnice

Pritiskom na gumb zatvara se prekidač između stupca i retka te dopušta da struja teče između njihovih pinova. Arduino otkriva koja je tipka pritisnuta tako što nađe pin retka i stupca koji je povezan. To je objašnjeno u 4 koraka:

1.KORAK

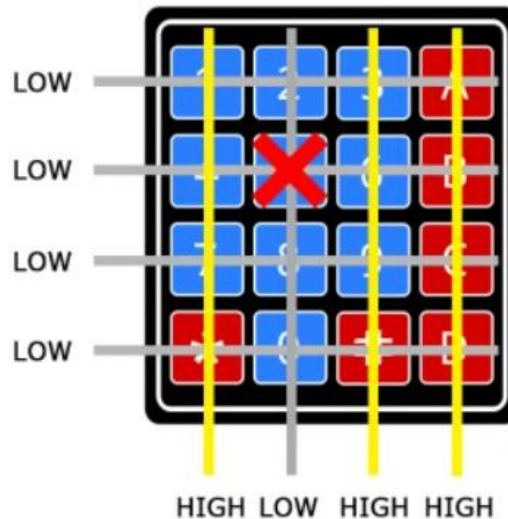
Kad nije pritisnuta nijedna tipka svi pinovi stupca su visoke razine dok su pinovi redaka niske razine, što je prikazano na slici 4.



Slika 4 - Izgled tipkovnice prije pritiska tipke

2. KORAK

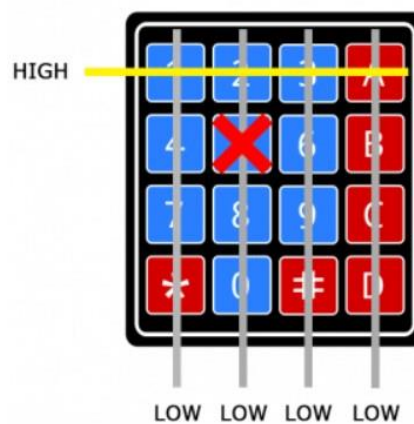
Na slici 5, kada se pritisne gumb, pin stupca se povlači u nisku razinu jer struja iz stupca visoke razine ulazi u red niske razine.



Slika 5 - Pritisak tipke postavlja stupac tipke u nisko stanje

3.KORAK

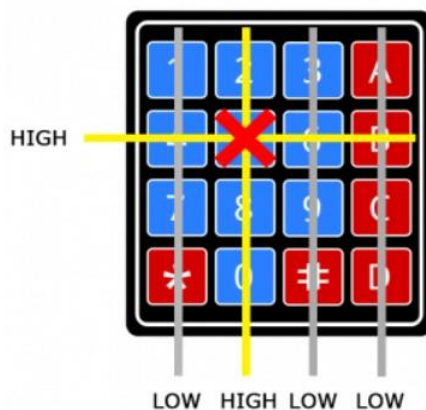
Arduino sada zna u kojem se stupcu nalazi tipka, tako da samo treba pronaći red. To čini prebacivanjem svakog od pinova retka u visoku razinu te istovremeno čita sve pinove stupca dok ne detektira koji se od njih vraća u visoku razinu, što prikazuje slika 6.



Slika 6 - Traženje retka tipkovnice

4.KORAK

Na slici 7, kada se pin stupca ponovno vrati u visoku razinu, Arduino je pronašao red koji je spojen na dugme.



Slika 7 - Kombinacijom drugog retka i drugog stupca dobije se broj

2.3. LCD zaslon

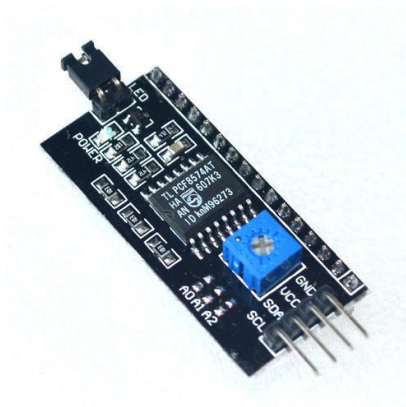
LCD zaslon 20x4 prikazan na slici 8, ima 20 znakova u svakom redu sa maksimalno 4 reda. Koristi napon od 5V DC te sadrži 2 registra: upravljački i podatkovni. Upravljački registar pohranjuje isključivo naredbe koje dobije (brisanje trenutnog sadržaja na zaslonu, postavljanje kursora, kontrola zaslona itd). Podatkovni registar pohranjuje podatke koji se moraju prikazati na LCD zaslonu. Podatak je u ASCII kodu tj. podaci se šalju u 8 bitnoj kombinaciji te je potrebno 8 digitalnih pinova. Budući da je broj pinova ograničen radi lakšeg korištenja, LCD zaslon se povezuje I²C pretvaračem. I²C pretvarač se koristi radi jednostavne primjene unutar samog programskog koda te broja pinova koji su potrebni za komunikaciju. U programskom kodu koji upravlja mikrokontrolerom potrebno je uključiti biblioteku za rad sa I²C pretvaračem, što je realizirano korištenjem naredbe `#include`, a definicija trenutnog LCD zaslona naredbom `LiquidCrystal_I2C lcd(0x27, 16, 2);`. Ostale naredbe definirane su unutar funkcija.



Slika 8- LCD zaslon

I²C pretvarač sastoji se od 4 ulazna pina od kojih dva služe za napajanje (VCC i GND), a preostala dva su podatkovna (SDA i SCL), također ima 16 izlaznih pinova od kojih se 8 koriste za komunikaciju s LCD zaslonom. Ovisno o brzini rada procesora SCL mijenja svoja logička stanja (0,1). SDA prelazi iz logičke 0 u logičku 1 kada SCL poprimi vrijednost logičke 1, a prijenos podatkovnog bita s Arduinoom započinje kada su SDA i SCL u stanju logičke 1. Za svaki prelazak SDA iz logičke 0 u logičku 1, a da je pritom SCL u stanju logičke 1 šalje se po 1 podatkovni bit. I²C protokol objašnjen je u poglavlju 3.

Na slici 9 prikazan je I²C pretvarač.



Slika 9 - I²C pretvarač

2.4. Električna brava

Električna brava serije 802 prikazana na slici 10, koristi 12V DC za napajanje. Veličine 8.9 centimetara i širine od 2 centimetra. Teži svega 200 grama te je jednostavna za korištenje i ugradnju.



Slika 10 - Električna brava 802 serije

Da sustav električne brave sigurno otključava ili zaključava, dodaje se relejni modul. Relejni modul u sebi sadrži tranzistorsku sklopku, relej te ostale elemente potrebne za rad. Upravljački krug radi na 5V DC dok izvršni strujni krug ima mogućnost podnošenja 220V AC. Za upravljanje se koristi jedan od digitalnih pinova. Za definiciju u programskom kodu nije potrebna biblioteka već je dovoljno da se definira pin na koji će se spojiti upravljačka žica naredbom `int relay=13;`, dok se izlazni pin definira naredbom `pinMode(relay, OUTPUT);`. Radi jednostavnijeg korištenja za potrebe ovog rada izrađeno je Arduino sučelje koje sadrži relejni modul. Postupak izrade i rada sučelja objašnjen je u poglavlju 6.

2.5. Senzor otiska prsta

Pomoću modula senzora otiska prsta olakšano je uzimanje uzoraka te njihovo prepoznavanje, te je sve češća potražnja u svrhu izrade raznih projekata. Veoma jednostavno se uzimaju uzorci, pohranjuju u memoriju te se zatim pretražuju. Ovakvi moduli većinom imaju svoju vlastitu memoriju u koju mogu pohraniti velike količine otisaka prstiju, također mogu raditi s mikrokontrolerom ili TTL(engl. *transistor-transistor logic*) serijskom komunikacijom. Za potrebe ovog rada korišten je ZFM-708 senzor otiska prsta prikazan na slici 11.



Slika 11 - Senzor za otisk prsta ZFM-708

Optički senzor prikazan na slici 11 sastoji se od DSP (eng. *Digital Signal Processing*) procesora iznimno velike brzine, senzora otiska prsta koji je optički, te *FLASH* memorije u koju se može pohraniti do 1000 otisaka. *ZFM-708* troši vrlo malo energije što je korisno ukoliko radi u sustavu koji za napajanje koristi baterije. Senzor pruža širok spektar instrukcija za upravljanje te je fleksibilan za podešavanje razine sigurnosti. Radi poboljšanja performansi i kako bi se osigurala trajnost senzora pozadinsko svijetlo na senzoru je promijenjeno iz crvenog u zeleno.

Spajanje senzora s Arduinoom te potreban kod može se pronaći na adresama:

<http://www.theorycircuit.com/fingerprint-sensor-scanner-arduino/>

https://www.electrooobs.com/eng_arduino_tut41.php

2.5.1. Specifikacije senzora

Tablica 3 prikazuje specifikacije senzora za otisak prsta.

Tablica 2 - Specifikacije senzora otiska prsta

Vrsta modula	ZFM-708
Napon napajanja	3.8V ~ 7.0 V(direct 3.3V)
Radna struja	65 mA
Kapacitet predložaka	1000 predložaka
Brzina prijenosa podataka	9600,19200,28800,38400,57600 bps(default 57600 bps)
Sučelje	UART (TTL Serial)/USB 1.1
Temperatura rada	-20°C~+60°C
Razine sigurnosti	5 (min to max:1,2,3,4,5)
Usporedna metoda	1:1
Metoda pretraživanja	1:N
FRR(False Recognition rate)	< 1 %
FAR(False Acceptance rate)	< 0.001 %
Prosječna brzina traženja otisaka	1s (1:500)

2.6. Jedinica za napajanje

Za potrebe rada koristi se V-Tac VT-20025 jedinica za napajanje prikazana na slici 12. Služi za potrebe napajanja električne brave koja radi na 12V DC, te za napajanje Arduino Uno sučelja.



Slika 12 - V-Tac VT-20025 jedinica za napajanje

Tablica 3 prikazuje specifikacije jedinice za napajanje

Tablica 3 - Specifikacije V-Tac jedinice za napajanje

Izlaz	DC napon	12V	24V				
	Jakost struje	2.1A	1A				
	Raspon struje	0-2.1A	0-1A				
	Snaga	25W	24W				
	Maksimalan šum ²	150mVp-p	240mVp-p				
	Podešavanje napona	10.8-13.2V	22-27.6V				
	Točnost napona ³	±1%	±1%				
	Linearno podešavanje ⁴	±0.5%	±0.5%				
	Podešavanje tereta ⁵	±0.5%	±0.5%				
	Vrijeme porasta	1000ms,30ms/230VAC 1000ms,30ms/110V					
Vrijeme zadržavanja (Typ)	50ms/230VAC 10ms/115AC						
Ulaz	Raspon napona	90-264VAC/120-370VDC					
	Raspon frekvencije	47-63HZ					
	Efikasnost (Typ)	70%	75%	80%	81%	82%	82%
	AC struja(Typ)	0.57A/90V 0.37A/220V					
	Udar struje (Typ)	Cold start: 65A/230VAC					
	Gubitak struje	<2mA/240VAC					

3. KOMUNIKACIJSKI PROTOKOLI

Komunikacijski protokoli korišteni u radu su:

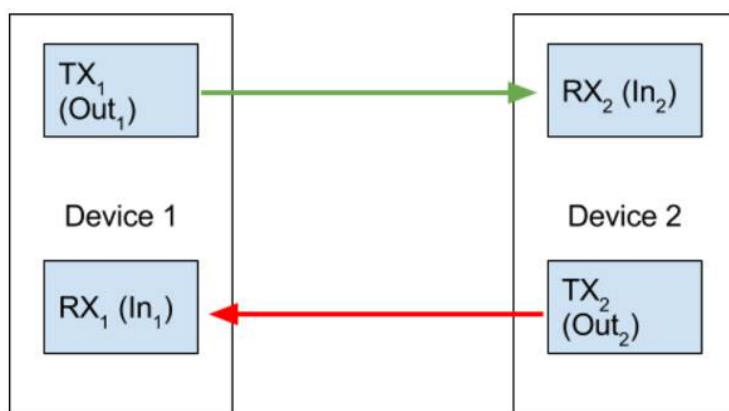
- Serijski komunikacijski protokol,
- SPI(engl. *Serial Peripheral Interface*),
- I²C(engl. *Inter-Integrated Circuit*).

Izvor: [1]

<https://www.deviceplus.com/how-tos/arduino-guide/arduino-communication-protocols-tutorial/>

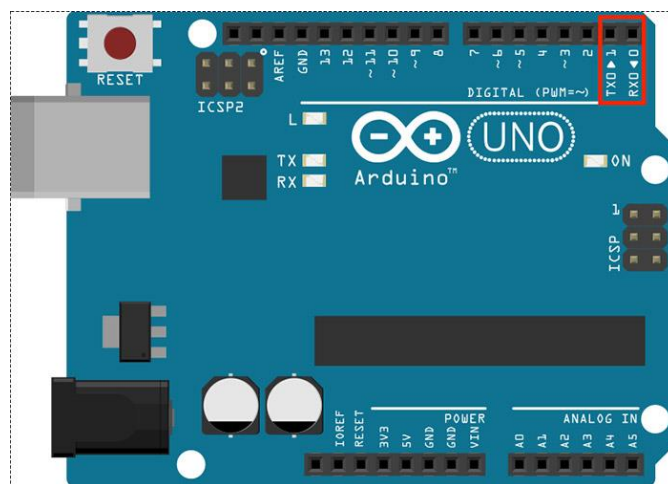
3.1. Serijski komunikacijski protokol

Serijski komunikacijski protokol ili kraće SKP serijski je oblik komunikacije. Kod SKP podaci se prenose kao sekvencijski bitovi. Ožičenje potrebno za postavljanje SKP komunikacije je veoma jednostavno. Jedna linija služi za prijenos podataka (TX), dok druga linija koristi za primanje podataka (RX), što prikazuje slika 13. Može se pretpostaviti da TX linija služi za slanje podataka s uređaja, dok RX prima podatke koji dolaze na uređaj. TX i RX linije zajedno koriste serijsku komunikaciju te čine serijski ulaz koji služi za uspostavu komunikacije.



Slika 13 - Dijagram sklopovnog spajanja za SKP

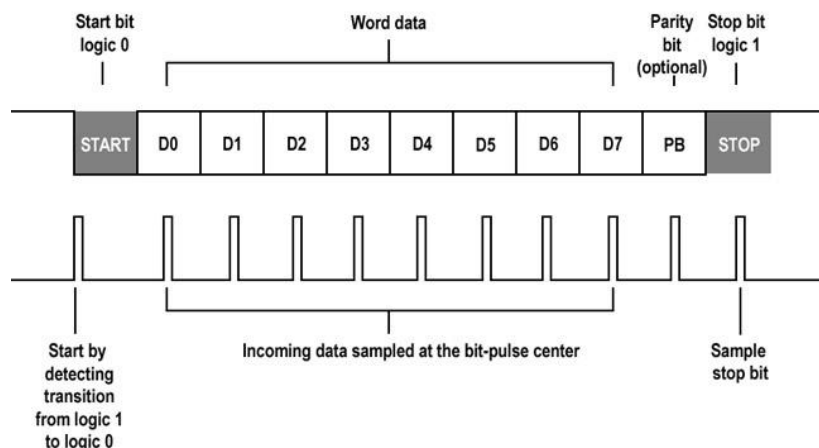
UART (engl. *Asynchronous Receiver/Transmitter*) odnosi se na ugrađeno sklopovlje koje upravlja pakiranjem i prijevodom serijskih podataka. Da bi uređaj mogao komunicirati putem SKP protokola, potrebno je imati navedeno sklopovlje. Na Arduino Uno-u postoji jedan serijski ulaz namijenjen komunikaciji s računalom s kojim je Arduino spojena, a to je USB(engl. *Universal Serial Bus*), tj. serijski ulaz. Na Arduino Uno-u, ova USB veza ostvarena je putem ugrađenog sklopovlja u dva digitalna kontakta, GPIO 0 i GPIO 1, koji se mogu koristiti u projektima koji uključuju serijsku komunikaciju s raznom elektronikom a ne samo računalima. Prikaz GPIO 0 i GPIO 1 nalazi se na slici 14.



Slika 14 - GPIO 0 i GPIO1 su serijski RX i TX. Bilo koji GPIO pin može se koristiti kao serijski RX i TX pomoću `SoftwareSerial` biblioteke

Također, može se koristiti bilo koji drugi GPIO pin kao serijski RX i TX uz pomoć `SoftwareSerial` Arduino biblioteke (`SoftwareSerial.h`).

SKP se naziva asinkronim jer komunikacija ne ovisi o sinkroniziranom signalu takta između dva uređaja koji pokušavaju međusobno komunicirati. Budući da brzina komunikacije nije definirana putem stabilnog signala, uređaj "pošiljalac" ne može biti siguran da "prijemnik" dobiva točne podatke. Zbog toga, uređaji prekidaju podatke u dijelove fiksne veličine kako bi osigurali da su zaprimljeni podaci isti kao podaci koji su poslani. Slika 15 prikazuje vizualizaciju SKP podatkovnih paketa.



Slika 15 - Vizualizacija SKP podatkovnih paketa

Uređaji koji komuniciraju putem SKP šalju pakete unaprijed određene veličine koji sadrže dodatne informacije koje se odnose na početak i kraj poruke, te potvrdu je li poruka primljena ispravno. Na primjer, za početak komunikacije, uređaj za odašiljanje povlači nisku predajnu liniju, ukazujući na početak podatkovnog paketa. Međutim, dugoročno, to znači da je SKP sporiji u usporedbi sa sinkroniziranim oblikom komunikacije jer je samo dio prenesenih podataka namijenjen aplikacijama uređaja (ostalo je za samu komunikaciju).

Prilikom implementacije SKP serijske komunikacije na većini ugrađenih platformi kao što je Arduino, korisnik se ne mora baviti komunikacijom na razini bita. Umjesto toga, platforma često nudi više razine softverskih biblioteka koje su jedini aspekt komunikacijskog procesa s kojim se korisnik mora suočiti. U Arduino platformi korisnici mogu koristiti serijske i *SoftwareSerial* biblioteke za implementaciju UART komunikacije za svoje projekte.

Važan aspekt za komunikaciju s SKP-om je da je dizajniran za komunikaciju između samo dva uređaja od jednom. Budući da protokol šalje samo bitove koji označavaju početak poruke, sadržaj poruke i kraj poruke, ne postoji metoda razlikovanja višestrukih prijenosnih i prijemnih uređaja na istoj liniji. Ako više od jednog uređaja pokuša prenijeti podatke na istoj liniji, dolazi do sukoba na sabirnici, a prijemni uređaji će najvjerojatnije dobiti neupotrebljive podatke.

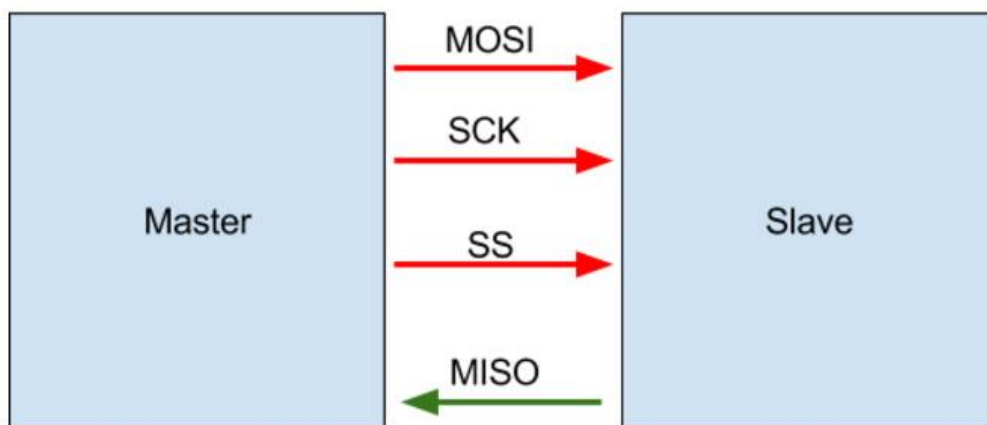
Nadalje, SKP je *half-duplexs*, što znači da, iako se komunikacija može odvijati dvosmjerno, oba uređaja ne mogu prenositi podatke jedni drugima u isto vrijeme. U projektu u kojem dva Arduina komuniciraju jedni s drugima preko serijske veze, na primjer, to bi značilo da u određenom trenutku samo jedan Arduino može "razgovarati" s drugim. Za većinu primjena, međutim, ova činjenica je relativno nevažna i nije ni na koji način nepovoljna.

3.2. SPI (engl. Serial Peripheral Interface)

Sljedeći komunikacijski protokol koji se koristi je SPI (engl. *Serial Peripheral Interface*). SPI se razlikuje od SKP-a u nekoliko ključnih stavki:

- Sinkron je,
- Koristi model upravljač-izvršitelj (engl. *master-slave*), no može biti više izvršitelja dok je upravljač samo jedan,
- Za implementaciju je potrebno više od dvije linije.

Dijagram spajanja hardvera kod SPI-a prikazan na slici 16, složeniji je za razliku od SKP:



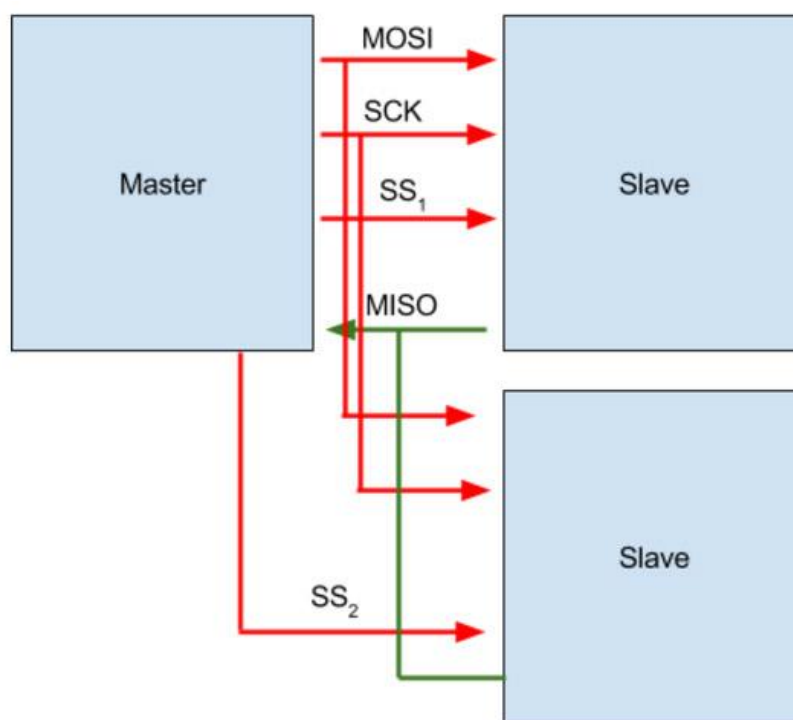
Slika 16 - Dijagram spajanja hardvera za SPI

- **MOSI („Master Out Slave In“):** Linija za prijenos podataka od upravljača do izvršitelja,
- **SCK („CLOCK“):** Definira brzinu prijenosa te prijenosne karakteristike,

- **SS(„Slave Select“)**: Linija koja omogućuje upravitelju s kojim će izvršiteljem komunicirati,
- **MISO(„Master In Slave Out“)**: Linija za prijenos podataka od izvršitelja do upravitelja.

Prva značajka SPI-a je da slijedi upravitelj-izvršitelj (engl. *master-slave*) model. To znači da se komunikacija oslanja na definiranje jednog uređaja kao glavnog, a ostalih kao pomoćnih. To stvara hijerarhiju između uređaja koja definira koji je uređaj zapravo „u kontroli“ drugih.

Također, postoji mogućnost spajanja više izvršitelja na jedan upravitelj prikazano na slici 17.



Slika 17 - Više izvršitelja spojeno na jedan upravitelj

SPI ne zahtijeva posebnu liniju za slanje i primanje za svaki izvršitelj koji je povezan s upraviteljem. Jedna zajednička linija prijema (MISO) i jedna zajednička linija za prijenos (MOSI), s jednom zajedničkom linijom takta (SCK) su povezani između svih izvršitelja i jednog upravitelja. Upravitelj odlučuje s kojim izvršiteljem komunicira putem zasebne SS linije za svakog izvršitelja. To znači da svaki dodatni izvršitelj s kojim upravitelj komunicira zahtijeva drugi GPIO pin na strani upravitelja.

SPI je sinkron, što znači da je komunikacija između upravitelja i izvršitelja usko vezana za taktni signal (kvadratni val fiksne frekvencije) definiran upraviteljem. Ovdje vidimo jedan od izravnih učinaka modela upravitelj-izvršitelj. Upravitelj pokreće komunikaciju definiranjem brzine komunikacije pomoću takta signala, a izvršitelji reaguju komuniciranjem pri toj brzini. Ta definicija brzine radi za bilo koju brzinu komunikacije koju upravitelj odlučuje (unutar maksimalne dopuštene brzine od strane izvršitelja).

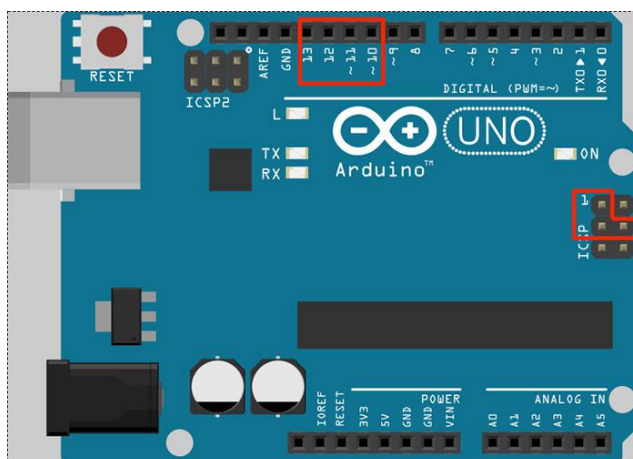
U SPI-u, dvije specifične značajke taktnog signala određuju početak i kraj prijenosa podataka: **taktni polaritet** (CPOL) i **taktna faza** (CPHA). CPOL se odnosi na stanje mirovanja (bilo nisko ili visoko) taktnog signala. Kako bi uštedjeli energiju, uređaji će postaviti liniju takta u stanje mirovanja kada ne komuniciraju s izvršiteljem, a dvije opcije dostupne za ovo stanje mirovanja su niske ili visoke. CPHA se odnosi na rub takta signala za koji se snimaju podaci. Kvadratni val ima dva ruba (uzlazni rub i padajući rub), a ovisno o postavkama CPHA-a, podaci se bilježe na rastućem rubu ili padajućem rubu.

Pri implementaciji SPI-ja na Arduino koristeći Arduino kao glavni uređaj (*SPI.h*). SPI digitalni pin priključci za SCK, MOSI i MISO su unaprijed definirani na Arduino pločama.

Za Arduino Uno, veze su sljedeće:

- SCK: GPIO 13 ili ICSP 3
- MOSI: GPIO 11 ili ICSP 4
- MISO: GPIO 12 ili ICSP 1
- SS: GPIO 10

Bilo koji digitalni pin može se također koristiti kao SS pin. Da bi se odabrao određeni uređaj, SS digitalni pin mora biti u niskom stanju. Raspored pinova prikazan je na slici 18.



Slika 18 - Raspored MOSI, MISO SCK pinova na Arduino Uno pločici

SPI je *full-duplex*, što znači da se komunikacija uvijek javlja dvosmjerno, čak i ako aplikacija zahtijeva samo prijenos podataka u jednom smjeru. SPI *full-duplex* prijenos podataka obično se provodi s registrom pomaka. To znači dok se svaki bit čita, bitovi koji su prije učitani pomiču se za jedno mjesto. Bit se naknadno šalje natrag kroz SPI vezu drugom uređaju.

SPI se ne koristi u ovom radu ali je objašnjen zbog buduće nadogradnje sustava na sučelja koja to zahtijevaju.

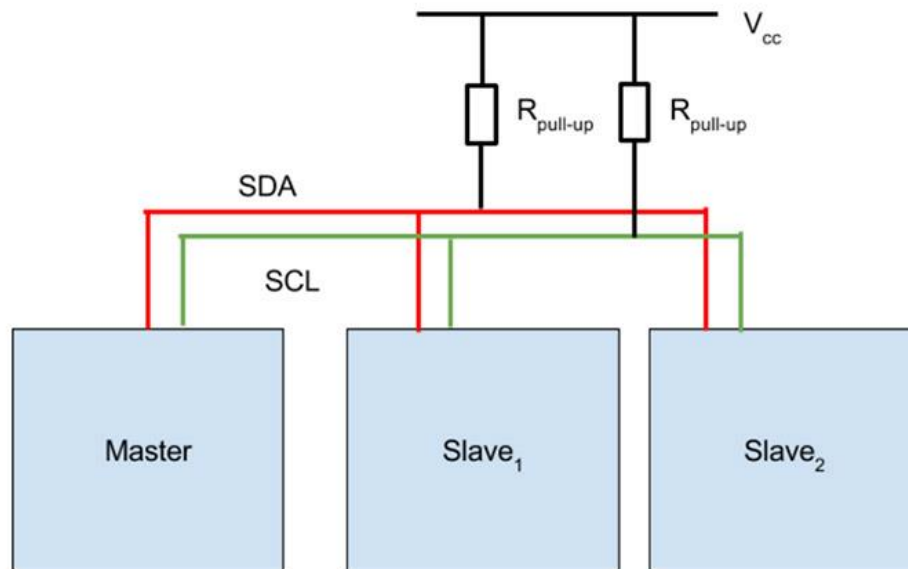
3.3. I²C protokol (engl. Inter-Integrated Circuit)

Inter-integrirani krug (I²C), koji se izgovara ili "i-na kvadrat-c" ili "i-dva-c", konačni je komunikacijski protokol koji se koristi. Iako je njegova implementacija najkompliciranija od ova tri protokola, I²C rješava nekoliko nedostataka u drugim komunikacijskim protokolima, dajući mu prednost u odnosu na druge u nekim aplikacijama. To uključuje:

- Mogućnost povezivanja više upravitelja s više izvršitelja ,
- Sinkronost (baš kao i SPI), što znači komuniciranje s višim brzinama,
- Jednostavnost: implementacija zahtijeva samo dvije žice i nekoliko otpornika.

Na razini hardvera, I²C je sučelje s dvije žice. Jedine žice potrebne za I²C vezu su podatkovna linija (nazvana SDA) i linija takta (koja se zove SCL). Podaci i takt linije su u visokom stanju dok je sustav u stanju mirovanja, međutim kad se podaci moraju poslati

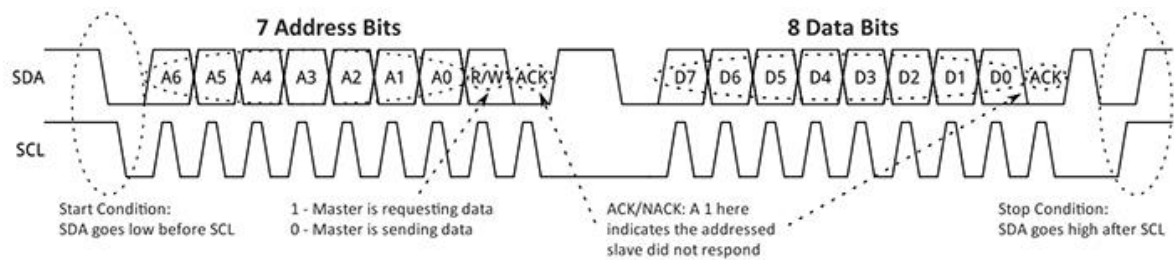
preko veze, linije su u niskom stanju što se postiže MOSFET krugovima. Važna činjenica koju treba uzeti u obzir je da se koristi sustav otvorenog odvoda, što znači da linije mogu biti u niskom stanju samo uz pomoć uređaja. Dakle, kad se koristi I²C presudno je koristiti *pull-up* otpornike (obično 4.7kΩ) kako bi se osiguralo da je linija u visokom stanju dok traje stanju mirovanja. Dijagram spajanja I²C sustava prikazan je na slici 19.



Slika 19 - Hardverski dijagram spajanja I²C sustava

I²C je jedinstven jer rješava pitanje povezivanja s višestrukim izvršiteljskim uređajima putem adresiranja. Baš kao u SPI komunikaciji, I²C koristi upravitelj-izvršitelj model za uspostavljanje "hijerarhije" komunikacije. Međutim, umjesto odabira izvršitelja putem zasebnih digitalnih linija, upravitelji biraju izvršitelje kroz svoje jedinstvene adrese bajtova. Ove adrese mogu izgledati ovako: 0x1B. To znači da povezivanje više izvršitelja s upraviteljem neće zahtijevati dodatne digitalne linije. Sve dok svaki izvršitelj ima jedinstvenu adresu, aplikacija će i dalje moći razlikovati izvršitelje samo na tim adresama. Za pozivanje izvršiteljeve funkcije, upravitelj samo poziva ime izvršitelja te samo on ima pravo odgovara.

Adresni i podatkovni dijelovi komunikacijske linije I²C prikazani su na slici 20.



Slika 20 - Izgled adresne i podatkovne komunikacijske linije

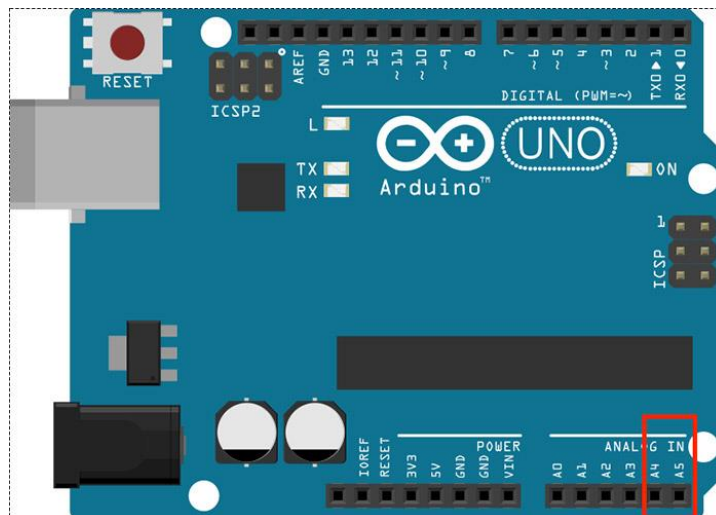
Primijetite ACK i NACK bitove na komunikacijskoj liniji. Ti bitovi označavaju da li adresirani izvršitelj odgovara na komunikaciju - način periodičnog provjeravanja govori da li se komunikacija odvija kako se očekuje. Ovi bitovi, naravno, nemaju veze s adresom ili podatkovnim bitovima koji se šalju, ali u velikoj shemi komunikacije oni dodaju malo više vremena komunikaciji u usporedbi s mnogim početnim i završnim bitovima i pauzama koje se pojavljuju u protokolima kao što je SKP.

I²C daje izvršiteljima slobodu odlučivanja o tome kakvi bi komunikacijski zahtjevi mogli izgledati. Različiti izvršitelji zahtijevaju pisanje različitih bajtova u različitim nalozima preko SDA linije za pisanje i traženje informacija od njih. U nekim modulima akcelerometra, na primjer, bajtovi su napisani da označe koji hardverski registar upravitelj želi pročitati prije nego što se naprave bilo kakvi zahtjevi za čitanje. Za ove specifikacije korisnik se mora pozvati na tablice izvršitelja za adrese uređaja, adrese registara i postavke uređaja.

Na Arduinou se implementacija I²C odvija putem biblioteke *Wire* (*wire.h*). Arduino se može konfigurirati kao upravitelj ili izvršitelj I²C. Na Arduino Uno, veze su sljedeće:

- **SDA:** Analogni pin 4,
- **SCL:** Analogni pin 5.

Raspored pinova prikazan je na slici 21.



Slika 21 – I²C (Wire) SDA je analogni pin 4(A4). SCL je analogni pin 5 (A5)

Višestruki upravitelji mogu biti spojeni preko I²C sabirnice samo spajanjem svojih SDA i SCL linija na linije sabirnice. Međutim, samo jedan upravitelj može komunicirati s izvršiteljem odjednom, jer bi višestruki uređaji koji pokušavaju međusobno komunicirati doveli do sukoba na sabirnici. Slično tome, komunikacija se ne može vršiti dvosmjerno od upravitelja do izvršitelja i obratno u isto vrijeme jer to bi također dovelo do sukoba na sabirnici. To čini I2C *half-duplexs*, baš kao i SKP.

Konačno, višestruki upravljački uređaji ne mogu međusobno komunicirati preko iste I2C sabirnice. U aplikacijama gdje su višestruki upravitelji povezani s izvršiteljima, upravitelji mogu međusobno komunicirati preko zasebne sabirnice ili putem zasebnog komunikacijskog protokola.

4. TEHNOLOGIJE ZA KONTROLU PRISTUPA

Svaki sustav iziskuje način pristupa i kontrole koji se postiže na različite načine kombinacijom sučelja i programske podrške. Za potrebe ovog diplomskog rada koristi se tipkovnica spojena na Arduino sučelje. Tipkovnica daje nekoliko mogućnosti u sustavu. Pruža mogućnost otključavanja električne brave pomoću unosa lozinke, koja je prethodno definirana u programskom kodu ovog sustava. Pritiskom na tipku „*“ pruža se mogućnost otključavanja brave pomoću otiska prsta, te pritiskom na tipku „#“ postiže se unos novog otiska prsta.

Neke od tehnologija koje se koriste za kontrolu pristupa su:

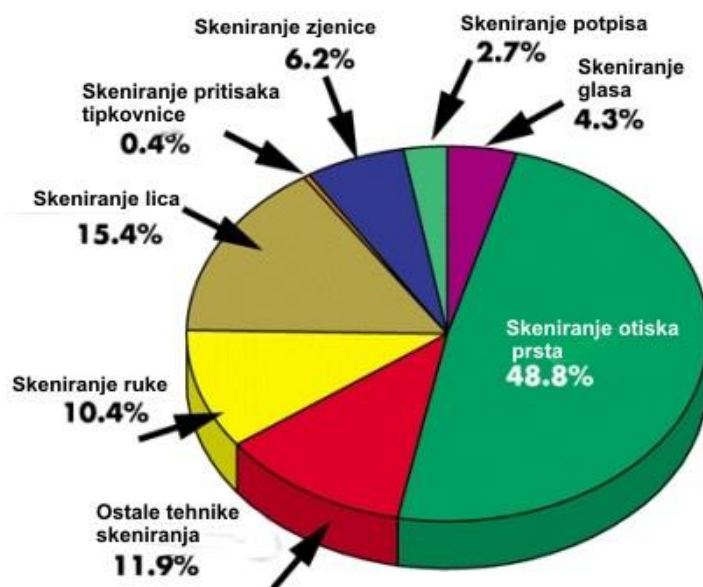
- RFID (engl. *Radio Frequency Identification*),
- Kontrola pomoću svjetla,
- Biometrika.,

U ovom radu pored tipkovnice koristi se biometrijska metoda temeljena na gore spomenutom otisku prsta.

4.1. Biometrika

Prema literaturi [5], biometrija (grč. bios – život, metron – mjera) predstavlja jedinstven skup metoda koje su automatizirane a čiji je cilj prepoznavanje ljudi na jedinstven način. Prepoznavanje se može temeljiti na jednoj ili na više karakteristika (fizičkih ili ponašajnih). Biometrijska autentifikacija, u informatičkoj tehnologiji, odnosi se na fizičke karakteristike (npr. prepoznavanje lica, otisak prsta, prepoznavanje rožnice oka itd.), te na ponašajne karakteristike (uzorak kretanja, jedinstven rukopis itd.) pojedine osobe. Biometrija se primarno koristi za autentificiranje osobe, ali može se primjenjivati u različitim područjima jedno od tih područja je prepoznavanje govora čija je svrha brže pisanje. Kako bi se mogućnosti biometrije iskoristile potrebno je koristiti specijalne uređaje koji služe kako bi pratili određene fizičke ili ponašajne karakteristike te je također potrebna uporaba programa koji analiziraju dobivene informacije. Sastavni elementi

biometrije su: **uzorkovanje** (pretvaranje analognog signala u digitalni) te **umjetna inteligencija**. Krajnje informacije se obrađuju pomoću računala, nastaje umjetna inteligencija, dobiveni uzorci prepoznaju se računalom te se uspoređuje način na koji računalo uči s načinom rada ljudskog mozga. Krajnji cilj je upotrijebiti računalo kao posrednika u uzorkovanju, dok u međuvremenu programski paket odlučuje koje će radnje poduzeti s primljenim digitaliziranim uzorcima. Na taj način je obuhvaćen cijeli proces digitalizacije, prepoznavanje određenih uzoraka, te princip rada umjetne inteligencije, a sve to sa krajnjim ciljem unaprijeđenja procesa učenja te samostalnog pamćenja i korištenja uzoraka u radu računala. Slika 22 prikazuje učestalost korištenja biometrijske tehnologije.



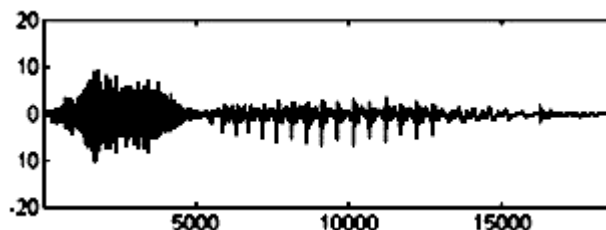
Slika 22 - Učestalost korištenja biometrijske tehnologije

4.2. Princip rada biometrije

4.2.1. Digitalizacija

Digitalizacija je najvažniji trenutak u procesu prepoznavanja uzorka. Sve dobivene podatke (skeniranje lica, otiska prsta, tijela itd.) potrebno je prevesti u računalu poznati

format. Taj proces dobiva se tako što se analogni signal pretvara u digitalni, te se prepoznaje potrebnom programskom opremom. Korištenjem DAC (engl. *Digital audio-video converter*) uređaja analogni se signal pretvara u digitalni. Proces se sastoji od niza Fourierovih transformacija, kvantizacija i ostalih pojmova koji služe da što vjernije opišu ulazni signal. Slika 23 prikazuje izgled digitaliziranog uzorka.



Slika 23 - Izgled digitaliziranog uzorka

4.2.2. Fourierova transformacija

Brza Fourierova transformacija (engl. *Fast Fourier Transformation*) jedna je od najkorištenijih a samim time i najlakših tehnika za koje se koriste pri opisivanju i obradi signala. Rezolucija s kojom radi je relativno mala, dok se opisivanje i obrade signala obavljaju u digitalnom signalnom procesoru (engl. *Digital Signal Processor*) korištenjem Fourierovih transformacija. Fourierovim transformacijama opisuje se krivulja uzorka i harmonika te se prevode u nekoliko sinusnih funkcija.

4.2.3. Umjetna inteligencija

Umjetna inteligencija pripada računalnoj znanosti čija je svrha rad sa zahtjevima za obavljanje raznih poslova iz domene percepcije, shvaćanja i učenja uzoraka. Današnji tipovi umjetne inteligencije svi spadaju u grupu umjetne inteligencije koja je ograničena svojim korištenjem, jer rješavaju samo određene probleme, a uzrok tome je ograničena svijet.

Jedan od osnovnih ciljeva umjetne inteligencije je odrediti koje će se vrste zadataka rješavati te kako će i na koji način ocjenjivati napredak. Umjetna inteligencija potrebna je

kao podloga za „neuronske mreže“ koje se primjenjuju u analizi slike i signala. Cilj je spojiti sposobnosti ljudi (prepoznavanje oblika, lica i glasova) i sposobnost računala da izvršava proračune i radi s ogromnim količinama podataka.

4.3. Biometrijske tehnike

Postoje tri različite vrste biometrijskih tehnika, a to su:

- **Fizička biometrija,**
- **Biometrija ponašanja,**
- **Multimodalna biometrija.**

U samom začetku biometrijskih sustava, prednost se izražavala fizičkim karakteristikama. Smatralo se da fizičke karakteristike imaju manje razlike unutar grupa u odnosu na ponašajne karakteristike.

4.3.1. Fizička biometrija

Fizička biometrija se bavi uzorkovanjem fizičkih osobina ljudskog tijela, te njegovim jedinstvenim karakteristikama. S obzirom da je svaki čovjek jedinstven po svojim fizičkim osobinama lako ga je prepoznati koristeći odgovarajuće opise uzoraka za prepoznavanje istih. Postoje različite vrste fizičke biometrije kao što su:

- Čitanje DNK zapisa,
- Skeniranje rožnice,
- Prepoznavanje lica,
- Geometrija šake,
- Provjera vena,
- Otisak prsta.

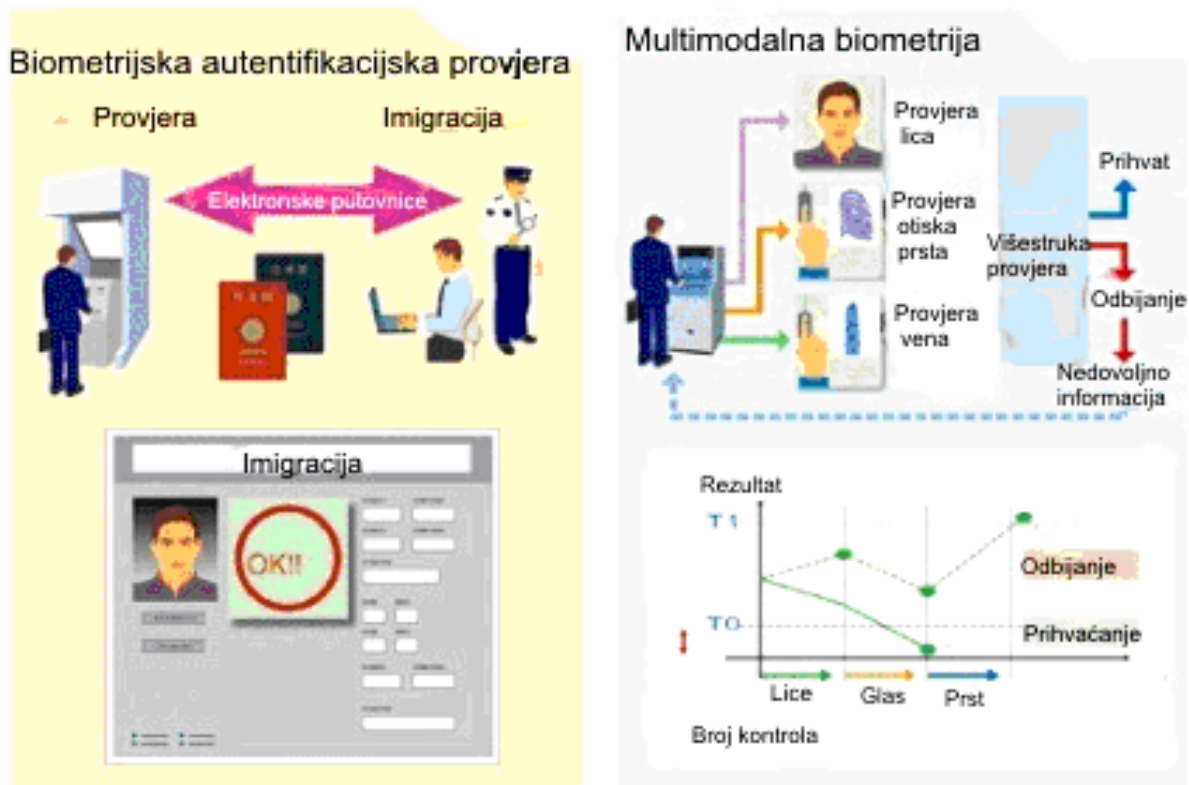
4.3.2. Biometrija ponašanja

Biometrija ponašanja koristi fizikalne karakteristike (glas, pokreti, vanjski izgled...) ljudskog tijela koje su većinom jedinstvene za određenu osobu. Ponašanje čovjeka se opisuje tako što se dobiveni uzorci ponašanja opisuju krivuljama koje služe za raspoznavanje različitih ljudi i njihovih fizikalnih osobina. Vrste biometrije ponašanja su:

- Prepoznavanje glasa,
- Prepoznavanje rukopisa ili potpisa,
- Dinamika tipkanja,
- Dinamika hoda,
- Dinamika mirisa.

4.3.3. Multimodalna biometrija

Multimodalna biometrija radi na principu da kombinira fizičku biometriju i biometriju ponašanja. Kombinacijom nekoliko tehnika stvara se sigurniji informacijski sustav te je manja mogućnost lažnog predstavljanja ili identificiranja. Primjerice pri očitavanju otiska prsta postoji mogućnost da osoba koristi lažni otisak prsta te može dovesti do krivog utvrđivanja osobe. No ako se koristi dodatna tehnika raspoznavanja (verifikacija lica, rožnice ili provjera vena) tada se s velikom mogućnošću može utvrditi pravi identitet osobe. Slika 24 prikazuje multimodalnu biometriju u carinskoj kontroli.



Slika 24 - Multimodalna biometrija u carinskoj kontroli

4.3.4. Usporedba biometrijskih tehnika

U sljedećim tablicama prikazana je paralelna usporedba više biometrijskih tehnologija s obzirom na sljedeće karakteristike:

- univerzalnost – koliko se tehnika može primijeniti u svakodnevnom životu;
- jedinstvenost – koliki je postotak jedinstvenosti navedene kategorije uzimajući u obzir pojedinca;
- trajnost – kolika je promjenjivost s obzirom na vrijeme tj. koliko pojedina osoba zadržava određenu karakteristiku;
- prikupljivost – kolika je lakoća dobivanja uzorka navedene kategorije;
- izvedivost - koliko je moguće u praksi iskoristiti navedene biometrijske metode i
- prihvatljivost – koliko je moguća implementacija a da se pritom ne narušavaju ljudska prava.

Tablica 4 prikazuje usporedbu biometrijskih tehnika prema univerzalnosti, jedinstvenosti i trajnosti.

Tablica 4 - Usporedba biometrijskih tehnika prema univerzalnosti, jedinstvenosti i trajnosti [5]

METODE BIOMETRIJE / KARAKTERISTIKE	Univerzalnost	Jedinstvenost	Trajnost
Lice	Visoka	Niska	Srednja
Otisak prsta	Srednja	Visoka	Visoka
Geometrija dlana	Visoka	Srednja	Srednja
Šarenica	Visoka	Visoka	Visoka
Mrežnica	Visoka	Visoka	Srednja
Termogram	Visoka	Visoka	Niska

Uho	Srednja	Srednja	Visoka
DNK	Visoka	Visoka	Visoka
Potpis	Niska	Niska	Niska
Glas	Srednja	Niska	Niska
Dinamika tipkanja	Niska	Niska	Niska
Miris	Visoka	Visoka	Visoka
Hod	Srednja	Niska	Niska

Tablica 5 prikazuje usporedbu biometrijskih tehnika prema prikupljivosti, izvedivosti i prihvatljivosti.

Tablica 5 - Usporedba biometrijskih tehnika prema prikupljivosti, izvedivosti i prihvatljivosti [5]

METODE BIOMETRIJE / KARAKTERISTIKE	Prikupljivost	Izvedivost	Prihvatljivost
Lice	Visoka	Niska	Visoka
Otisak prsta	Srednja	Visoka	Srednja
Geometrija dlana	Visoka	Srednja	Srednja
Šarenica	Srednja	Visoka	Niska
Mrežnica	Niska	Visoka	Niska
Termogram	Visoka	Srednja	Niska

Uho	Srednja	Srednja	Visoka
DNK	Niska	Visoka	Niska
Potpis	Visoka	Niska	Visoka
Glas	Srednja	Niska	Visoka
Dinamika tipkanja	Srednja	Niska	Srednja
Miris	Niska	Niska	Srednja
Hod	Srednja	Niska	Niska

4.4. Otisak prsta

Slika 25 prikazuje otisak prsta.



Slika 25 - Otisak prsta

U ovom radu koristi se tehnika otiska prsta, opisana u literaturi [6], kako bi se identificirala osoba i omogućio joj se pristup otključavanju električne brave. Otisak prsta najčešće se upotrebljuje u biometrijskoj tehnologiji, zauzima čak 48.8% uporabe što je gotovo polovica uporabe biometrijske tehnologije. Velika upotreba otiska prsta se temelji na tome da je otisak prsta jedinstven za svaku osobu, uključujući čak i jednojajčane

blizance. Najveću primjenu ima u forenzičkim i kriminalističkim istragama gdje služi za identificiranje žrtve ili samog počinitelja.

Razlozi široke uporabe otiska prsta su:

- velika brzina prepoznavanja,
- relativno velika brzina računalnog hardvera,
- cijena i veličina uređaja su male,
- jednostavna uporaba,
- transakcije putem interneta,
- sigurnost.

Površina kože na određenim dijelovima ljudskog tijela (dlanovi, prsti i stopala) je prekrivena brazdama koje se nazivaju papilarne linije. Njihova uloga je da omogućavanje boljeg hvata i lakšeg kretanja.

Osobine dobivene iz otiska prsta dijele se u dvije skupine:

1. **Globalne osobine** – nisu dovoljne za identificiranje, međutim dovoljne su da se otisci mogu grupirati. Identifikacija se provodi na principu poklapanja lokalnih osobina s prototipom.
2. **Lokalne osobine** – čine minucijske točke. Minucijeske točke ili minucije su prekidi tokova papilarnih linija (nepredviđen prekid, grananje itd.) i upravo su one nositelji jednoznačnih informacija na temelju kojih se vrši identifikacija.

Slika 26 prikazuje osnovne uzorke papilarnih linija.

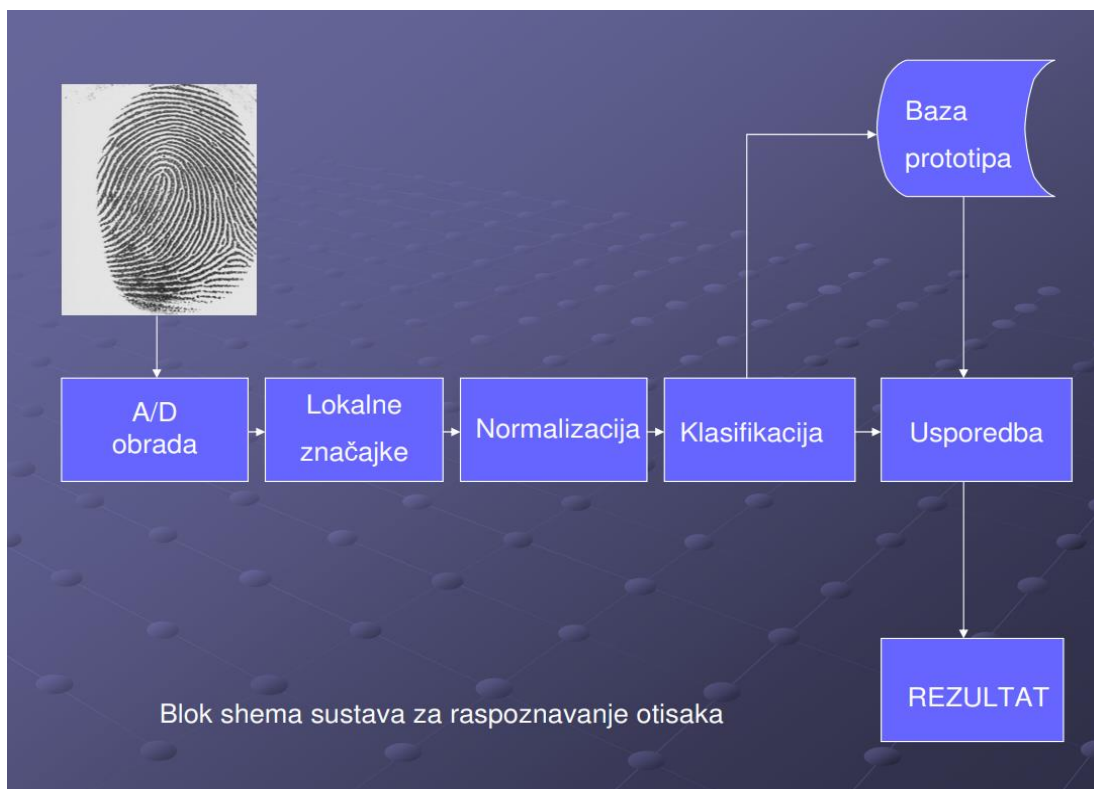


Slika 26 - Osnovni uzorci papilarnih linija su: petlja (engl. *Loop*), luk(engl. *Arch*) i vrtlog (engl. *Whorl*)

4.4.1. Sustavi i algoritmi za raspoznavanje otiska prsta

Svi sustavi identificiranja temelje se na usporedbi provjerenih karakteristika s prije pohranjenim prototipom. Prototip je prije potrebno pohraniti na određenom mjestu (baza podataka ili tvrdi disk) ili ovlaštena osoba mora imati uz sebe svoj prototip (magnetska ili čip kartica).

Slika 27 prikazuje sustav za raspoznavanje otiska prsta.



Slika 27 - Blok shema sustava za raspoznavanje otiska prsta

4.4.2. AFIS(*Sustav automatizirane identifikacije otiska prsta*)

Prema literaturi [7], najpoznatiji sustav za raspoznavanje otiska prsta je AFIS(engl. *Automated fingerprint identification system*) ili Sustav automatizirane identifikacije otisaka prstiju.

AFIS je računalni sustav koji automatski obrađuje papilarne linije prstiju i dlanova. U bazi podataka sadrži otiske iz deseteroprstne DKT zbirke, otiske dlanova i sporne tragove papilarnih linija, uključujući i određene fragmente otisaka prstiju. AFIS znatno skraćuje vrijeme koje je potrebno za unos podataka.

Princip rada sustava se temelji na sljedećim provjerama:

1. Provjera deseteroprstnog s deseteroprstnim obrascima – služi za sravnjivanje baze izbjegavanjem duplikata u bazi podataka, za utvrđivanje lažnih podataka, lažnih imena te zamjenjuje dosadašnji način ručne klasifikacije.

2. Provjera deseteroprstnog obrasca s bazom spornih tragova – uspoređivanje novoevidentirane osobe osobe s bazom spornih otisaka, odnosno tragova koji pripadaju dotad nepoznatim osobama.
3. Provjera spornog traga s bazom spornih tragova – ova provjera omogućuje povezivanje počinitelja sa serijama kaznenih djela.
4. Usporedba spornog traga s bazom deseteroprstih obrazaca – usporedba sa svim do tada daktiloskopiranim osobama.
5. Mogućnost različitih obrada (povećanje kontrasta, svjetlosti, itd.) spornih tragova u svrhu poboljšanja kvalitete-olakšava identifikaciju.

AFIS-ov princip rada je sljedeći: otiske prsta prvo je potrebno digitalizirati, tj. unijeti ih u bazu podataka tako da se otisak prsta skenira na mjestu događaja (od osoba ili mrtvih tijela ili tragova otisaka papilarnih linija), na način da se trag ili otisak, skenira na mjestu događaja tako da se izuzme na foliju ili fotografira. Radi kompatibilnosti i mogućnosti korištenja otiska, fotografije se izrađuju u rezoluciji od 500 dpi, razlog tome je jer se 500 dpi koristi u čitačima za otisak prsta, uređajima za usporedbu otisaka i algoritmima za kompresiju otisaka. Nakon što se unese slika otiska papilarnih linija u bazu podataka, potrebno ju je analizirati na princip da se analiziraju mjesta spajanja i završetaka zavijutaka papilarnih linija ili analizirati cijeli pravac svake linije pojedinačno. Znači pri analiziranju otisaka papilarnih linija AFIS ne koristi klasifikaciju temeljenu na tri osnovna uzorka luka, zamke ili kruga, kao što je to slučaj u daktiloskopskim zbirkama.

4.4.3. Prednosti i nedostaci AFIS-a

Prednosti sustava su:

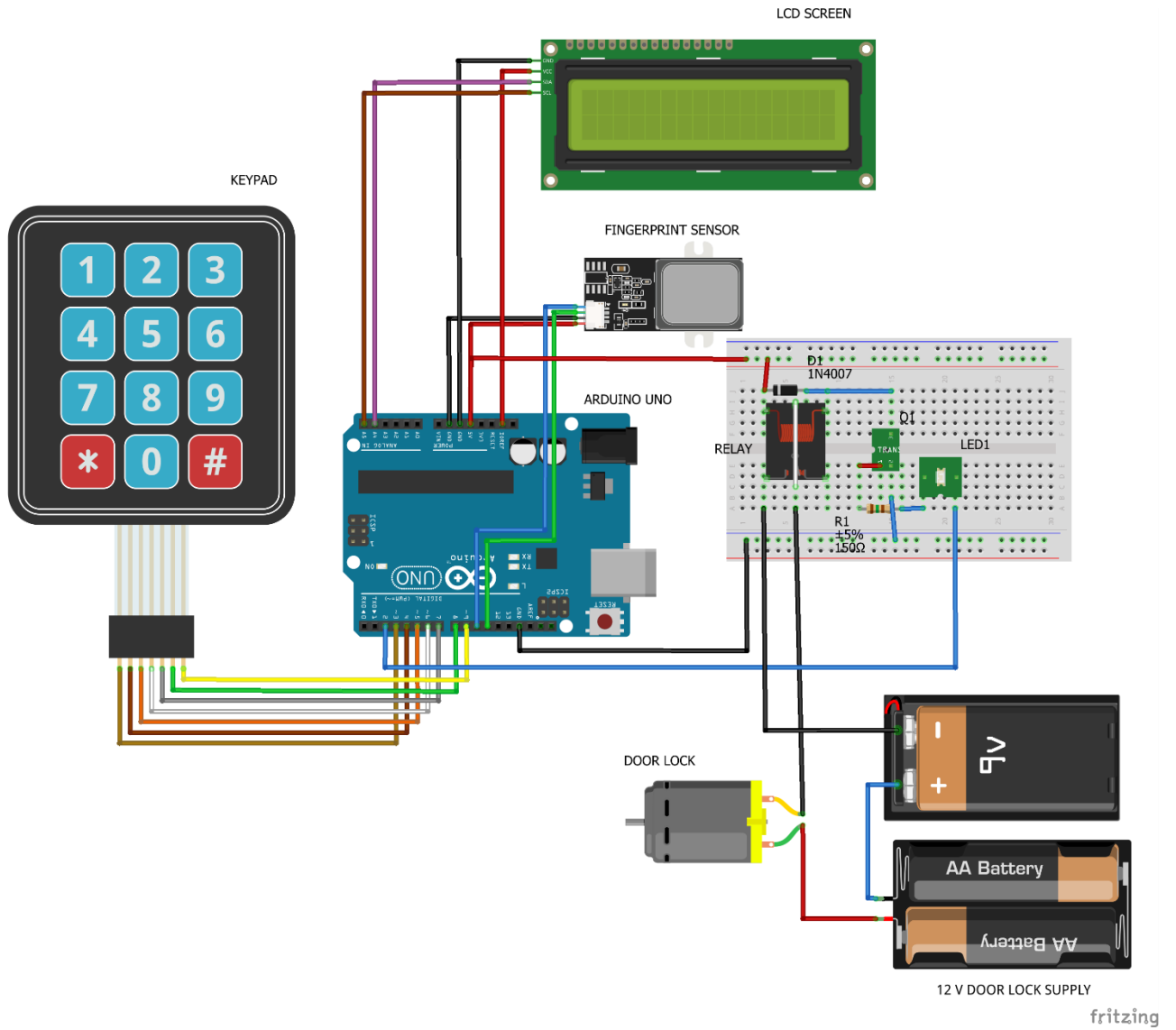
- Prihvatljiv je – ljudi su većinom upoznati sa principom na koji radi uzimanje otisaka te je prihvatljiv u društvu;
- točan – sustavi za uzimanje otisaka prsta u većem dijelu su vrlo točni te je mogućnost pogreške reducirana na minimum;
- jednostavan za korištenje – nije potrebno puno vremena kako bi se uzeo otisak, nije potrebna koncentracija i sl.;
- jedinstven – svaki otisak je jedinstveni identifikator pojedine osobe.

Nedostaci sustava su:

- sigurnost – postoji velika nepovjerljivost prema snimanju otisaka zbog mogućnosti uporabe u druge svrhe;
- velik broj različitih proizvođača i uređaja pa često dolazi do problema pri implementaciji u sustav.

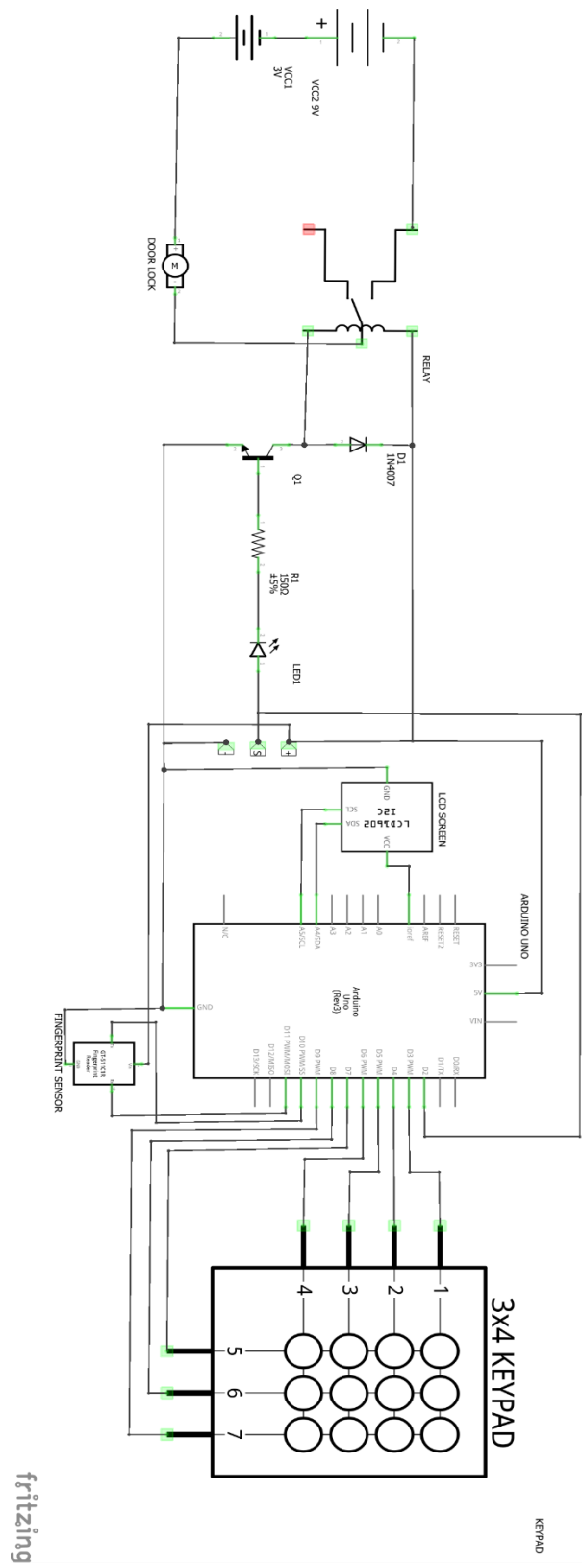
5. ARHITEKTURA SUSTAVA

Slika 28 prikazuje arhitekturu cjelokupnog sustava.



Slika 28 - Shema sustava

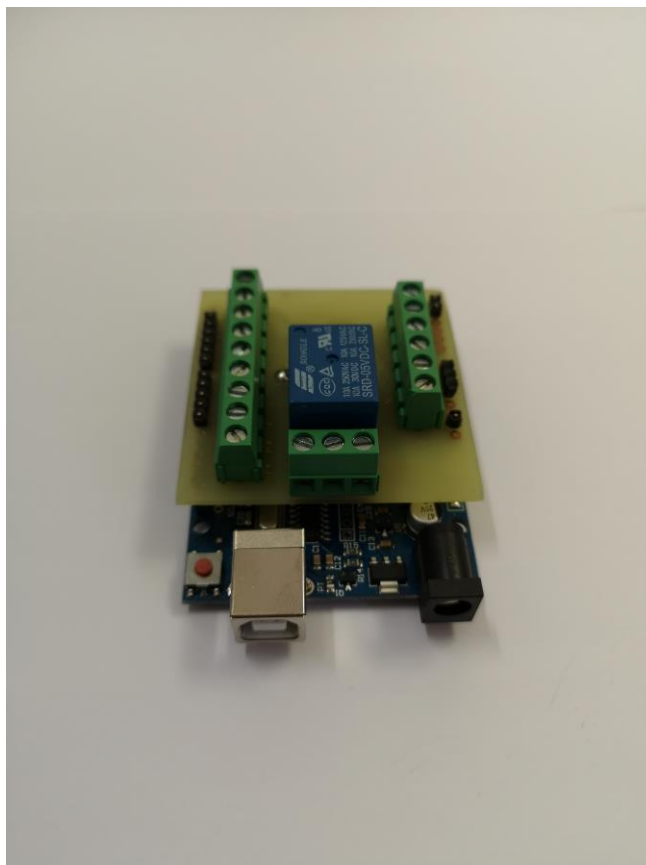
Sustav radi na način da se otključavanje brave vrši pomoću otiska prsta ili sekundarnim načinom, lozinkom koja se unosi pomoću tipkovnice. Prvo korisnik stavlja prst na senzor za otisak prsta koji komunicira s Arduinoom. Nakon što Arduino primi zahtjev za provjeru otiska prsta, u bazi podataka koja se nalazi u samom senzoru provjerava se postoji li spremljen upravo taj otisak prsta. Ukoliko u bazi postoji taj otisak prsta potvrda se šalje Arduinou te se digitalni izlaz 2 na Arduinou postavlja u visoko (engl. HIGH) stanje te se pali relej koji otključava bravu i drži je otključanu nekoliko sekundi. Međutim ako u bazi ne postoji otisak prsta Arduino dobiva negativan odgovor te električna brava ostaje zaključana, a od korisnika se traži da unese sigurnosnu lozinku pomoću tipkovnice. Postupak otključavanja brave koristeći tipkovnicu sličan je postupku otključavanja brave sa senzorom za otisak prsta, samo što se unosi kombinacija od 8 brojeva na tipkovnici. Ukoliko je lozinka točna Arduino pali relej i otključava se brava.



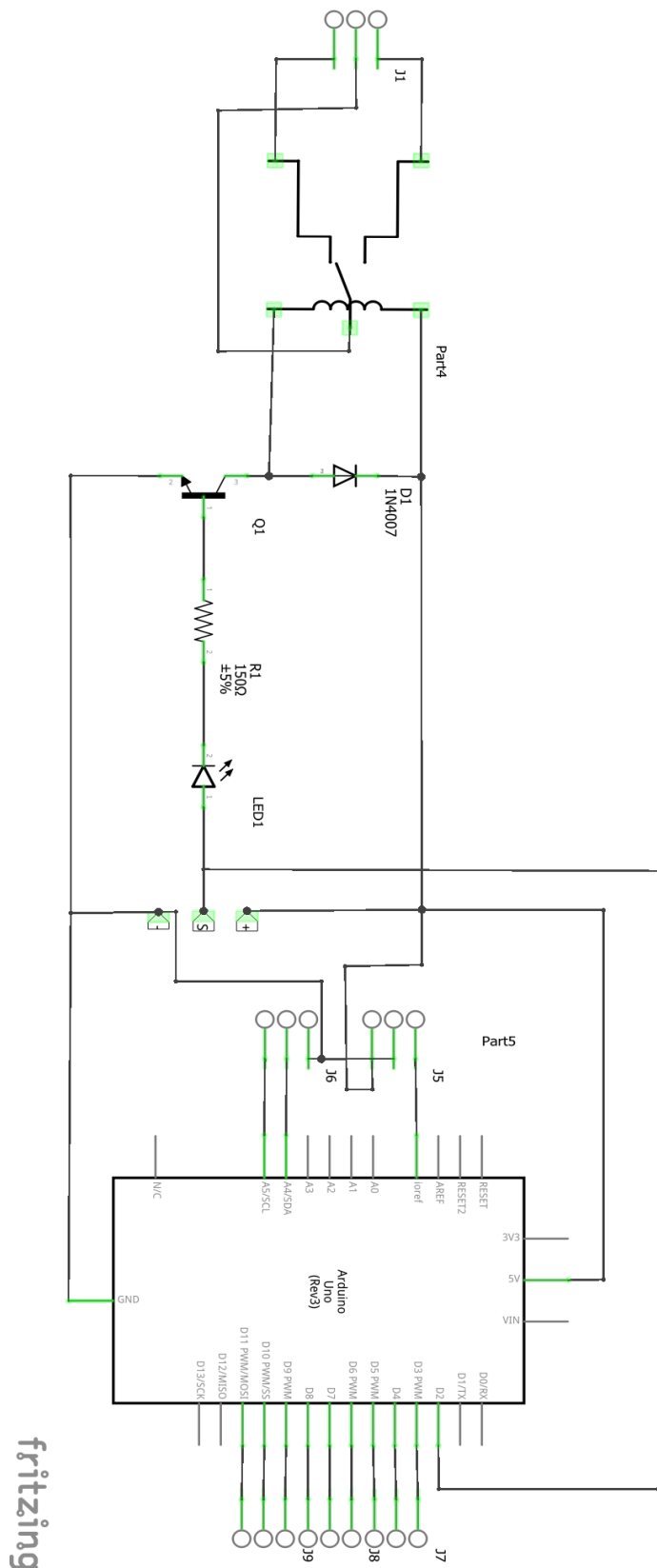
Slika 29 - Električna shema sustava

6. ARDUINO SUČELJE

Elektronička brava koja se koristi ima radni napon od 10V do 12V, pa je nije moguće spojiti direktno na Arduino čiji je radni napon 5V. U tu svrhu potrebno je izraditi Arduino sučelje prikazano na slici 30, kako ne bi došlo do pregaranja samog Arduina. Za izradu sučelja koristi se shema na slici 31.



Slika 30 - Sučelje spojeno na Arduino Uno

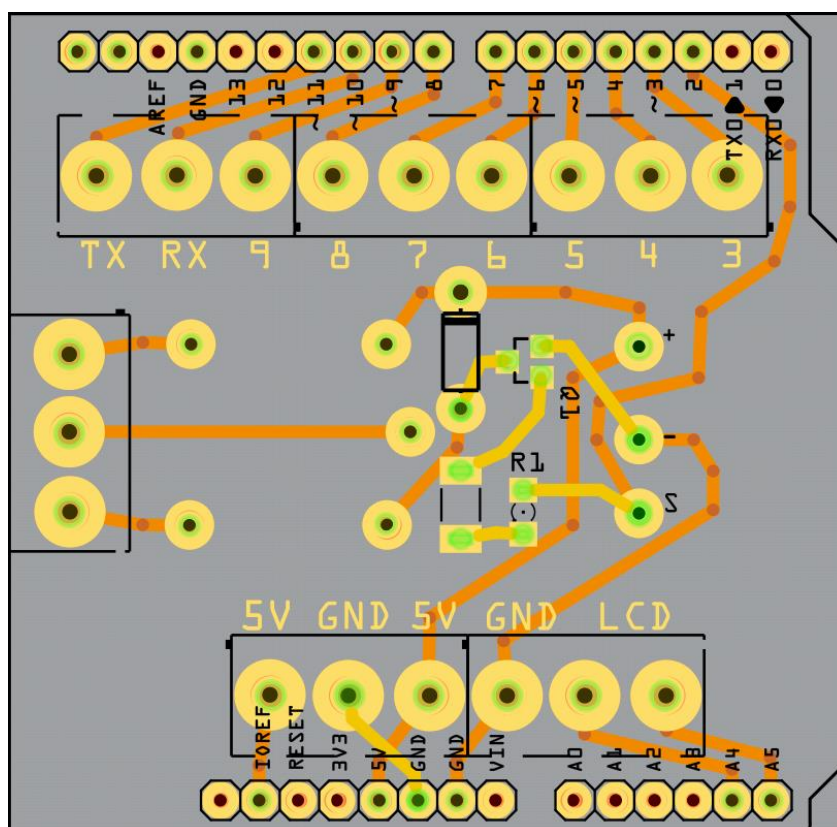


fritzing

Slika 31 - Shema Arduino sučelja

6.1. Dizajn Arduino sučelja

Dizajn pločice prikazane na slici 32 izrađuje se u programskom paketu Fritzing. Dizajn pločice odgovara dimenzijama Arduino Una kako bi se sama pločica mogla jednostavno staviti na Arduino. Pinovi su povezani na stezaljke zbog jednostavnijeg povezivanja sa žicama. Vodovi označeni smeđom bojom predstavljaju vodove na donjoj strani pločice dok su žuti na gornjoj strani. Potpuni dizajn pločice prikazan je na slici 32.

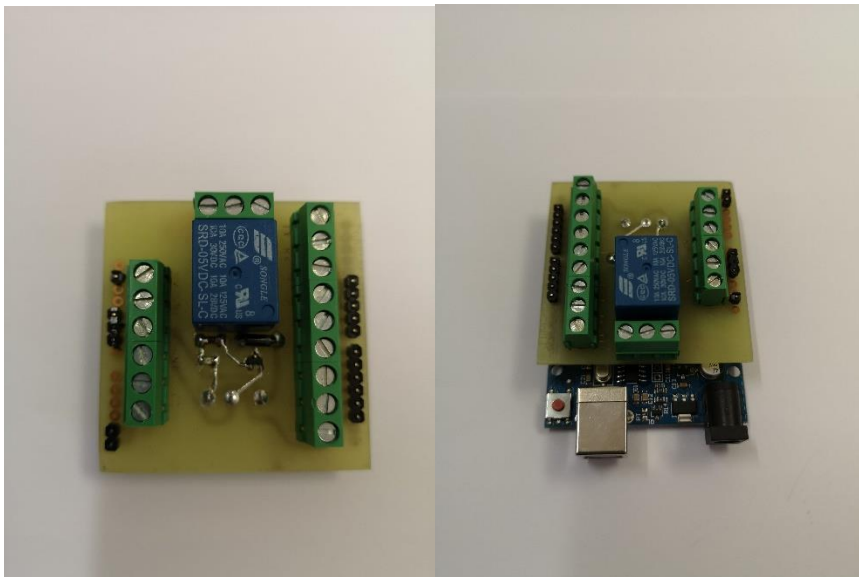


Slika 32 - Dizajn sučelja u programskom paketu Fritzing

Kako bi se moglo pristupiti izradi pločice potrebno je isprintati na prozirnoj foliji vodove prethodno izrađene u programskom paketu Fritzing. Uzorak dobiven na prozirnoj foliji stavlja se na pločicu prekrivenu tankim bakrenim filmom te se postupkom osvjetljavanja ultraljubičastim svjetlom vodovi otiskavaju na pločici. Posebnom otopinom solne kiseline i hidrogen peroksida odstranjuje se višak bakrenog filma te ostaju vodovi potrebni za izradu pločice. Nakon što su dobiveni potrebni vodovi na pločici, potrebno je probušiti rupe kako bi se postavili elementi, te postupkom lemljenja spojili na pločicu.

Nakon spajanja postupak izrade pločice je završen te se dolazi do završnog proizvoda spremnog za testiranje.

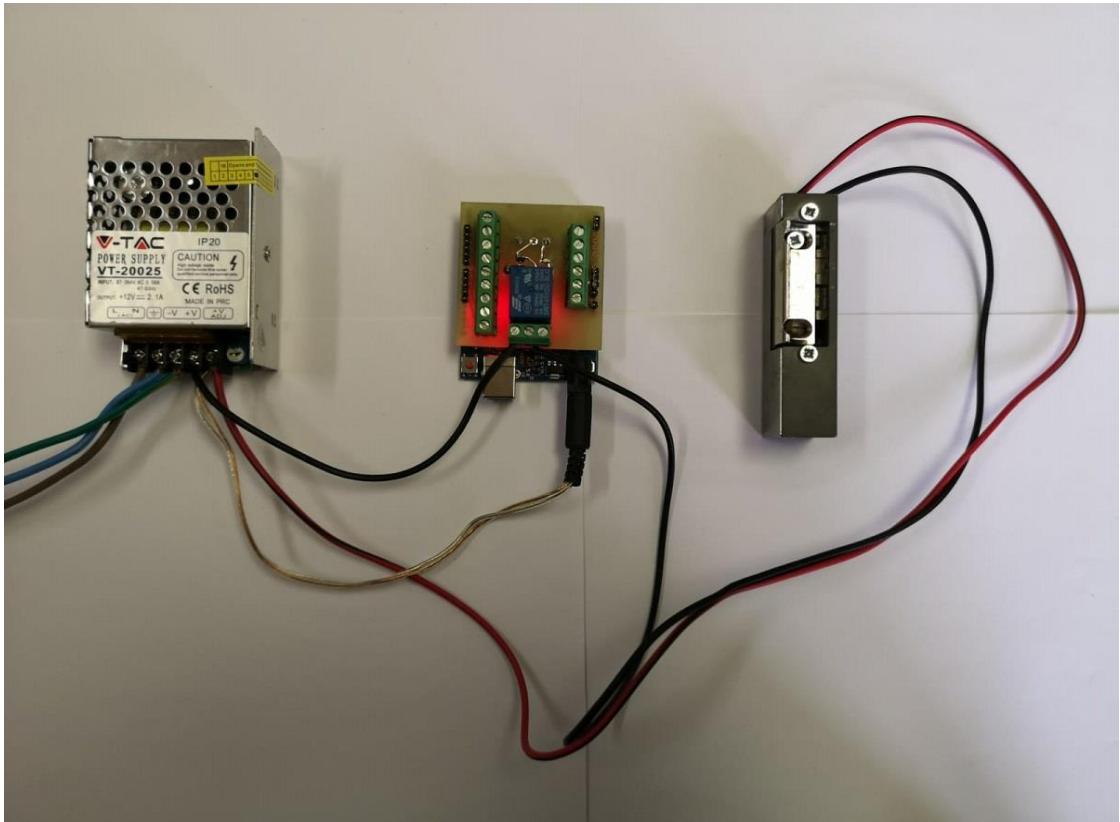
Slika 33 prikazuje završni proizvod Arduino sučelja.



Slika 33 - Završni proizvod Arduino sučelja

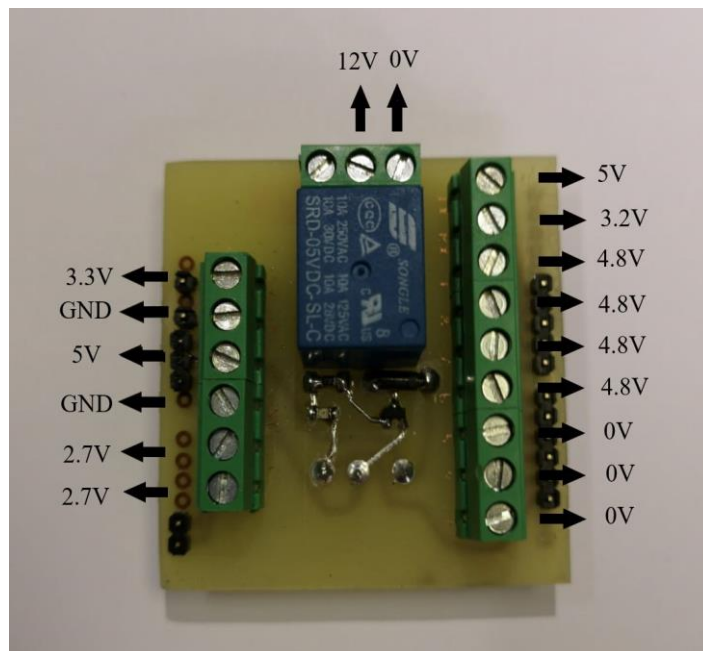
6.2. Testiranje Arduino sučelja

Kako bi testiranje bilo uspješno potrebno je spojiti sučelje s Arduino Uno-m, te na sučelje spojiti električnu bravu kojoj se dovodi napon od 12V. U Arduino Uno se učitava jednostavni programski kod koji služi kao testni primjer otvaranja brave (kod se nalazi u poglavlju 8, pod nazivom „Kod za testiranje Arduino sučelja“). Test se smatra uspješno izvršenim i ispravnim ukoliko se brava otključa te ostane otključana onoliko sekundi koliko je zadano u programskom kodu. Primjer testiranja Arduino sučelja prikazan je na slici 34.



Slika 34 - Testiranje Arduino sučelja

Također, na sučelju su testirane naponske razine na pojedinom pinu prikazane na slici 35.



Slika 35 - Naponske razine na pojedinom pinu

7. MAKETA SUSTAVA I TESTIRANJE KOMPONENTI

7.1. Maketa sustava

U svrhu ovog rada također je izrađena maketa. Maketa se sastoji iz dva dijela. Prvi dio makete su minijaturna vrata izrađena od drveta, koja simuliraju rad električne brave u stvarnim uvjetima.

Drugi dio makete je elektronika u kojoj se nalazi sustav za upravljanje električnom bravom. Sva elektronika spomenuta u prvom poglavlju (izuzev električne brave), smještena je unutar plastične kutije.

Na slici 36 može se viditi kompletan prikaz makete sustava za otključavanje električne brave.



Slika 36 - Maketa sustava

7.2. Kodovi za testiranje komponenti

Prije nego se pristupi spajanju potpunog sustava potrebno je testirati svaku komponentu koja se koristi. U tu svrhu izrađeni su testni programski kodovi koji provjeravaju ispravnost svake komponente pojedinačno. Testirani su redom: Arduino sučelje, tipkovnica i senzor otiska prsta.

7.2.1. Kod za testiranje Arduino sučelja

```
int relay=2;

void otvori() {
    digitalWrite(2, HIGH);
    delay(5000);
    digitalWrite(2, LOW);
    delay(2000);
}

void setup() {
    pinMode(relay, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    Serial.print("vrata otvorena \n");
```

```
otvori();
```

```
}
```

7.2.2. Kod za testiranje tipkovnice

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

//definiranje duzine lozine
#define Password_Length 9

//inicijalizacija i deklaracija potrebnih varijabli
char Data[Password_Length];
char Master[Password_Length] = "24033103";
byte data_brojac = 0;
byte master_brojac = 0;
bool Pass_is_good;
char customKey;
int relay = 2;

void otvori() {
    digitalWrite(2, HIGH);
    delay(5000);
```

```

digitalWrite(2, LOW);

}

//broj redova i stupaca kod tipkovnice
const byte REDAK = 4;
const byte STUPAC = 3;

//kreiranje tipkovnice
char hexaKeys[REDAK][STUPAC] = {
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'}
};

//pinovi na koje je spojena tipkovnica
byte rPins[REDAK] = {9, 8, 7, 6};
byte sPins[STUPAC] = {5, 4, 3};

//kreiranje virtualne tipkovnice
Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rPins, sPins, REDAK,
STUPAC);

//postavljanje adrese LCD-a te njegov broj kolona i redaka
LiquidCrystal_I2C lcd(0x3F, 20,4);

void setup() {
pinMode(relay, OUTPUT);

```

```

Serial.begin(9600);

//inicijalizacija zaslona i početna poruka

lcd.init();

lcd.backlight();

lcd.setCursor (0,0);

lcd.print ("Dobrodosli");

delay(2000);

lcd.clear();

}

void loop() {

lcd.setCursor(0,0);

lcd.print("Unesi lozinku:");

//unos sa tipkovnice i ispis unosa na zaslon

customKey = customKeypad.getKey();

if (customKey){

    Data[data_brojac] = customKey;

    lcd.setCursor(data_brojac,2);

    lcd.print("*");

    data_brojac++;

}

//ako je duzina unesene lozinke ista kao i izvorna

if(data_brojac == Password_Length-1){

    lcd.clear();

//uspoređuje se unos i postavljena lozinka

```

```

if(!strcmp(Data, Master)){

    //ukoliko se podudaraju

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Lozinka je točna");

    otvori();

    delay(2000);

}

else{

    //ukoliko se lozinke ne podudaraju

    lcd.print("Lozinka nije točna");

    delay(1000);

}

lcd.clear();

clearData();

}

}

//funkcija vraćanja brojača unesenih karaktera

void clearData(){

    while(data_brojac !=0){

        Data[data_brojac--] = 0;

```



```
    }  
    return;  
}
```

7.2.3. Kod za testiranje senzora otiska prsta

```
#include <LiquidCrystal_I2C.h>  
  
#include <Adafruit_Fingerprint.h>  
  
#include <Wire.h>  
  
int getFingerprintIDez();  
  
//postavljanje serijske komunikacije senzora i arduina  
SoftwareSerial mySerial(10,11);  
  
//postavljanje adrese LCD zaslona  
LiquidCrystal_I2C lcd (0x3F, 20,4);  
  
//izvodi se objekt finger iz klase Fingerprint i dodjeljuje mu se vrijednost koju senzor  
pošlje serijskom komunikacijom  
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);  
  
int relay = 2;  
  
void otvori() {  
    digitalWrite(2, HIGH);  
    delay(5000);  
    digitalWrite(2, LOW);  
}  
  
void setup(){
```

```

pinMode(relay, OUTPUT);

Serial.begin(9600);

//inicijalizacija parametara LCD zaslona

lcd.init();

//gasi pozadinsko svijetlo na 2s

lcd.noBacklight();

delay(2000);

//pali pozadinsko svijetlo i ispisuje poruku

lcd.backlight();

lcd.begin(20,4);

lcd.setCursor(0,0);

lcd.print("Skeniraj svoj otisak ");

//postavlja se brzina komunikacije senzora i arduina

finger.begin(57600);

}

void loop(){

//funkcija koja čeka unos otiska prsta

getFingerprintID();

delay(500);

digitalWrite(13,HIGH);

}

//deklaracija funkcije uzimanja otiska prsta

uint8_t getFingerprintID(){

//spremanje slike otiska u varijablu

```

```

int8_t p = finger.getImage();

switch(p){

    case FINGERPRINT_OK:

        lcd.clear();

        lcd.print(" Otisak je ocitan... ");

        delay(2000);

        break;

    case FINGERPRINT_NOFINGER:

        return p;

    case FINGERPRINT_PACKETRECEIVEERR:

        return p;

    case FINGERPRINT_IMAGEFAIL:

        return p;

    default:

        return p;

}

//ako je otisak uredno ucitan

//generira datoteku znakova sa izvorne slike otiska i sprema u buffer

p = finger.image2Tz();

switch(p) {

```

```

case FINGERPRINT_OK:

break;

case FINGERPRINT_IMAGEMESS:

return p;

case FINGERPRINT_PACKETRECEIVEERR:

return p;

case FINGERPRINT_FEATUREFAIL:

return p;

case FINGERPRINT_INVALIDIMAGE:

return p;

default:

return p;

}

//Ako je slika kovertirana

//funkcija za traženje otiska prsta u FLASH memoriji senzora

p = finger.fingerFastSearch();

//ako je otisak pronađen ispisuje sljedeće

if (p == FINGERPRINT_OK){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Otisak pronaden");

lcd.setCursor(0,1);

lcd.print(" u bazi.");

delay(2000);

```

```

}
else if(p==FINGERPRINT_NOTFOUND){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Otisak nije");
    lcd.setCursor(0,1);
    lcd.print(" pronaden.");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Skeniraj otisak");
    return p;
}
else{
    return p;
}

//Ako je nađen otisak u memoriji senzora
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Pronaden ID ");
lcd.print(finger.fingerID);
lcd.setCursor(0,1);
lcd.print("sa uvjerljivosti ");
lcd.print(finger.confidence);
lcd.setCursor(0,2);

```

```
lcd.print("na skali od 0-255.");  
delay(2000);  
otvori();  
  lcd.clear();  
  lcd.setCursor(0,0);  
  lcd.print("Skeniraj otisak");  
}
```

//funkcija vraća -1 ukoliko je neuspješno izvedena,u protivnom slučaju vraća ID

```
int getFingerprintIDez() {  
  uint8_t p = finger.getImage();  
  
  if (p != FINGERPRINT_OK) return -1;  
  p = finger.image2Tz();  
  if (p != FINGERPRINT_OK) return -1;  
  p = finger.fingerFastSearch();  
  if (p != FINGERPRINT_OK) return -1;
```

//nađeno poklapanje otisaka

```
digitalWrite(13, LOW);  
delay(10);  
digitalWrite(13, HIGH);  
delay(10);  
lcd.clear();  
lcd.setCursor(0, 0);
```

```
lcd.print("Pronaden ID ");  
lcd.print( finger.fingerID);  
lcd.setCursor(0, 1);  
lcd.print("sa uvjerljivosti ");  
lcd.print(finger.confidence);  
lcd.setCursor(0,2);  
lcd.print("na skali od 0-255");  
  
return finger.fingerID;  
delay(2000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Skeniraj otisak");  
  
}
```

8. ZAKLJUČAK

U ovom radu opisane su komponente koje tvore cjelinu sustava pametnog zaključavanja, također razni protokoli koji se koriste u radu Arduino Uno pločice (UART, SPI, I²C). Svaka komponenta sustava pojedinačno je spojena sa Arduino mikrokontrolerom, provjeren je njihov rad i testirani su kodovi napisani C++ programskim jezikom prilagođenom Arduino platformi, čija je svrha upravljanje pojedinim komponentama. Princip rada sustava je takav da se može pristupiti otključavanju električne brave na dva načina: unosom lozinke i otiskom prsta. U tu svrhu korišteni su senzor za otisak prsta i tipkovnica spojeni na Arduino Uno preko Arduino sučelja, koji je prethodno bilo potrebno izraditi i testirati. Također na sučelje se spaja električna brava kako bi se osigurala sigurnost sustava od pregaranja. Sustav je naknadno ispravno testiran u cjelosti. Prednost sustava je ta što su komponente za izradu jeftine i dostupne na tržištu. Sklopovlje i aplikacija sustava jednostavni su za korištenje. Također postoji mogućnost nadogradnje kako sklopovlja, tako i dodatnog ažuriranja aplikacije. Npr. pomoću Arduino Mega može se koristiti više digitalnih izlaza koji daju mogućnost dodatnih načina otključavanja električne brave što daje dodatnu sigurnost sustavu, može se koristiti RFID sustav za otključavanje, također pomoću Bluetooth-a te pomoću svjetla koje će se u budućnosti dodatno razvijati.

9. LITERATURA

- [1] Rahul, I. Arduino Communication Protocols Tutorial, 2016
<https://www.deviceplus.com/how-tos/arduino-guide/arduino-communication-protocols-tutorial/> [20.04.2019.]
- [2] Theory circuit,
<http://www.theorycircuit.com/fingerprint-sensor-scanner-arduino/> [17.05.2019.]
- [3] Circuit basics,
[http://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/?](http://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/)
[18.05.2019.]
- [4] Electronoobs, https://www.electronoobs.com/eng_arduino_tut41.php [22.05.2019.]
- [5] CARNet, Biometrija,
<https://www.cert.hr/wp-content/uploads/2006/09/CCERT-PUBDOC-2006-09-167.pdf> [18.06.2019.]
- [6] Hip, O., Idlbek, R. Biometričke tehnologije u zaštiti informacijskih sustava
[12.07.2019.]
- [7] Želimir Radmilović: Biometrijska identifikacija
<https://hrcak.srce.hr/file/117825> [24.07.2019.]

10. POPIS SLIKA I TABLICA

Popis tablica

Tablica 1 - Tehničke karakteristike Arduina	3
Tablica 2 - Specifikacije senzora otiska prsta.....	11
Tablica 3 - Specifikacije V-Tac jedinice za napajanje	12
Tablica 4 - Usporedba biometrijskih tehnika prema univerzalnosti, jedinstvenosti i trajnosti [5].....	29
Tablica 5 - Usporedba biometrijskih tehnika prema prikupljivosti, izvedivosti i prihvatljivosti [5]	30

Popis slika

Slika 1 - Arduino Uno.....	2
Slika 2 - Spajanje tipkovnice s Arduinoom	4
Slika 3 - Shema tipkovnice	5
Slika 4 - Izgled tipkovnice prije pritiska tipke.....	5
Slika 5 - Pritisak tipke postavlja stupac tipke u nisko stanje	6
Slika 6 - Traženje retka tipkovnice	6
Slika 7 - Kombinacijom drugog retka i drugog stupca dobije se broj	7
Slika 8- LCD zaslon.....	8
Slika 9 - I ² C pretvarač.....	8
Slika 10 - Električna brava 802 serije	9
Slika 11 - Senzor za otisk prsta ZFM-708	10
Slika 12 - V-Tac VT-20025 jedinica za napajanje	12
Slika 13 - Dijagram sklopovnog spajanja za SKP	13
Slika 14 - GPIO 0 i GPIO1 su serijski RX i TX. Bilo koji GPIO pin može se koristiti kao serijski RX i TX pomoću SoftwareSerial biblioteke	14
Slika 15 - Vizualizacija SKP podatkovnih paketa	15
Slika 16 - Dijagram spajanja hardvera za SPI	16
Slika 17 - Više izvršitelja spojeno na jedan upravitelj.....	17
Slika 18 - Raspored MOSI, MISO SCK pinova na Arduino Uno pločici	19
Slika 19 - Hardverski dijagram spajanja I ² C sustava.....	20
Slika 20 - Izgled adresne i podatkovne komunikacijske linije	21
Slika 21 – I ² C (Wire) SDA je analogni pin 4(A4). SCL je analogni pin 5 (A5)	22
Slika 22 - Učestalost korištenja biometrijske tehnologije	24

Slika 23 - Izgled digitaliziranog uzorka.....	25
Slika 24 - Multimodalna biometrija u carinskoj kontroli	28
Slika 25 - Otisak prsta.....	31
Slika 26 - Osnovni uzorci papilarnih linija su: petlja (engl. <i>Loop</i>), luk(engl. <i>Arch</i>) i vrtlog (engl. <i>Whorl</i>)	33
Slika 27 - Blok shema sustava za raspoznavanje otiska prsta	34
Slika 28 - Shema sustava	37
Slika 29 - Električna shema sustava	39
Slika 30 - Sučelje spojeno na Arduino Uno.....	40
Slika 31 - Shema Arduino sučelja.....	41
Slika 32 - Dizajn sučelja u programskom paketu Fritzing.....	42
Slika 33 - Završni proizvod Arduino sučelja	43
Slika 34 - Testiranje Arduino sučelja	44
Slika 35 - Naponske razine na pojedinom pinu	44
Slika 36 - Maketa sustava	45