



Št. naloge: 00433/2009

Datum: 05.04.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **UROŠ AHAČIČ**

Naslov: **RADIOFREKVENČNA IDENTIFIKACIJA IN SLEDENJE OBJEKTOV**
RADIOFREQUENCY IDENTIFICATION AND OBJECT TRACKING

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Kandidat naj v diplomski nalogi izdelava programsko aplikacijo za pregledovanje in sledenje opreme in knjig v podanem prostorskem okolju, ki vključuje tudi identifikacijo oseb in njihovo povezanost z možnostjo izposoje knjig ali opreme. Za izdelavo aplikacije naj uporabi sistem RFID, ki deluje v visokofrekvenčnem območju 13.56MHz in je z računalnikom povezan preko standardne TCP/IP povezave. Analizira in uporabi naj systemske knjižnice proizvajalca opreme, ustrezna programska orodja in izdelava program za komunikacijo s sistemom RFID ter grafični vmesnik za njegovo upravljanje in povezavo s podatkovnimi bazami. Delovanje aplikacije in zahtevanih funkcionalnosti sistema naj kandidat izvede v laboratoriju in na osnovi testiranja ovrednoti prednosti in slabosti sistema.

Mentor:


doc. dr. Mira Trebar



Dekan:


prof. dr. Franc Solina

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Uroš Ahačič

RADIOFREKVENČNA IDENTIFIKACIJA IN
SLEDENJE OBJEKTOV

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Mira Trebar

Ljubljana, 2009

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Uroš Ahačič,

z vpisno številko 63050227,

sem avtor diplomskega dela z naslovom:

RADIOFREKVENČNA IDENTIFIKACIJA IN SLEDENJE OBJEKTOV

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mire Trebar
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____ Podpis avtorja/-ice: _____

Zahvala

Zahvaljujem se mentorici, doc. dr. Miri Trebar, za pomoč in usmerjanje pri pisanju diplomske naloge. Zahvaljujem se tudi staršem in bratu za potrpežljivost in vzpodbude v času pisanja diplomske naloge.

Kazalo

Povzetek.....	1
Abstract.....	3
1. Uvod.....	5
2. Uporabljene tehnologije in orodja.....	7
2.1. Tehnologija RFID.....	7
2.1.1. Vloga frekvence pri sistemih RFID.....	8
2.1.1.1. Nizkofrekvenčni sistemi RFID.....	9
2.1.1.2. Visokofrekvenčni sistemi RFID.....	9
2.1.1.3. Ultra visokofrekvenčni RFID sistemi.....	9
2.1.2. Elementi sistema RFID.....	10
2.1.2.1. Značke RFID.....	10
2.1.2.2. Čitalci RFID.....	11
2.1.3. Uporaba RFID.....	12
2.1.3.1. Sledenje predmetom.....	12
2.1.3.2. Elektronsko plačevanje.....	12
2.1.3.3. Kontrola dostopa.....	12
2.2. Operacijski sistem.....	13
2.3. Razvojno ogrodje.....	13
2.4. Razvojno orodje.....	14
2.5. Podatkovna baza.....	14
3. Strojni vidik sistema RFID.....	15
3.1. Čitalec QuasarMR1.....	17
3.2. Multiplekser hfMUX-4-MP.....	18
3.3. Antene MaxiPCB.....	19
3.4. Značke RFID.....	19
4. Programski vidik sistema RFID.....	21
4.1. Protokol za komunikacijo s čitalcem QuasarMR1.....	21
4.1.1. Zahteva ASCII.....	22
4.1.2. Odgovor ASCII.....	26
4.1.3. Organizacija pomnilnika v čitalcu.....	28
4.1.4. Primeri komunikacije s čitalcem.....	29
4.1.4.1. Branje TID kartic v doseg, neodvisno od tipa kartice.....	29
4.1.4.2. Nastavljanje RID čitalca.....	30
4.1.4.3. Branje podatkov iz pomnilnika na znački.....	31
5. Aplikacija.....	33
5.1. Zanka za dostop.....	34
5.2. Zanka za komunikacijo s čitalcem.....	35
5.2.1. Delovanje prioritete vrste.....	38

5.3. Branje z več antenami.....	40
5.4. Model podatkovne baze.....	41
5.5. Uporabniški vmesnik.....	42
5.5.1. Izposoja in vračilo knjig ali opreme.....	43
5.5.2. Urejanje značk.....	45
5.5.3. Podatkovna baza.....	47
5.5.4. Nastavitve.....	48
5.6. Namestitev aplikacije.....	50
6. Zaključek.....	51
7. Literatura.....	53

Kratice

ASCII	American Standard Code for Information Interchange
CR	Carriage Return
EEPROM	Electrically Erasable Programmable Read-Only Memory
GNU	Gnu's Not Unix
GPL	General Public License
HF	High Frequency
IDE	Integrated Development Environment
ISM	Industrial-Scientific-Medical
LF	Low Frequency
LF	Line Feed
MUX	Multiplexer
ODBC	Open Database Connectivity
ORDBMS	Object-Relational Database Management System
PC	Personal Computer
RFID	Radio-frequency Identification
TID	Tag Identifier
UHF	Ultra High Frequency
UID	Unique Identifier
XML	Extensible Markup Language

Povzetek

Namen diplomskega dela je predstaviti sistem RFID, ki je postavljen v laboratoriju, in je uporabljen za sledenje knjigam in opremi v laboratoriju. Vsaka oseba ima za dostop do laboratorija svojo kartico, ki se uporablja za beleženje vstopov in izstopov. Sistem omogoča tudi beleženje izposoje knjig in opreme ter njihovo pregledovanje v podatkovni bazi.

V delu bodo predstavljene osnove tehnologije RFID in njene uporabe danes. Opisano bo, kako lahko preko omrežja Ethernet upravljamo s čitalcem RFID QuasarMR1. Čitalec z uporabo štirih anten, ki so priključene na multiplekser, sledi značkam v prostoru. Predstavljen bo razvoj in delovanje aplikacije, ki teče na operacijskem sistemu Linux in krmili delovanje celotnega sistema RFID. Aplikacija je bila razvita s programskim ogrodjem Qt. Za hranjenje podatkov o osebah, knjigah in opremi skrbi podatkovna baza PostgreSQL. Na koncu je predstavljenih nekaj idej, kako bi lahko sistem še nadaljno izboljšali.

Ključne besede: radiofrekvenčna identifikacija, čitalec RFID, značka RFID, sledenje objektov

Abstract

The purpose of this thesis is to present an RFID system that is set up in a laboratory and used to track books and equipment inside. Every person uses an access card which is used to log any entries and exits. The system can also be used to lend and return books and equipment in the laboratory and for browsing through all the tagged items that are stored in the database.

The thesis will present the basics of RFID systems and their use today. It will describe how Ethernet network can be used to communicate with the QuasarMR1 RFID reader. The reader uses four antennas that are connected using a multiplexer to track RFID tags in the room. The development and use of an application, executing on a Linux operating system will be presented. The application controls the RFID system in the laboratory and was developed using the Qt framework. To store the necessary information about persons, books and equipment the PostgreSQL database is used. In the end, a few ideas will be presented on further possibilities of expanding and improving the system.

Keywords: radiofrequency identification, RFID reader, RFID tag, object tracking

1. Uvod

Radio frekvenčna identifikacija (ang. Radio-frequency identification), opisuje tehnologijo, ki uporablja radijske valove za identifikacijo predmetov in oseb [1]. V praksi to pomeni oddajanje določene oblike identifikacijske številke z uporabo radijskega valovanja. Predmeti, ki oddajo svojo identifikacijsko številko, se imenujejo značke RFID. Naprave, ki preberejo oddano identifikacijsko številko, se imenujejo čitalci RFID.

RFID hitro postaja ena izmed najbolj kontroverznih tehnologij našega časa. Zagovorniki tehnologije trdijo, da lahko te majhne značke pripomorejo k preprečevanju prodaje ponarejenih zdravil, pomagajo pri boju proti terorizmu ob enem pa skrbijo, da so police v trgovskih centrih ustrezno založene. Na drugi strani pa se večina kritik RFID nanaša na zasebnost oz. pomanjkanje le-te. Ker se lastnik predmeta morda ne zaveda, da predmet vsebuje značko RFID, je možno z razdalje prebrati njeno vsebino in tako zbirati občutljive podatke o uporabniku, brez njegove vednosti. Ena izmed težav je lahko tudi dejstvo, da so značke na kupljenem predmetu uporabne tudi po tem, ko predmet kupimo in odnesemo domov. V tem primeru se lahko uporabijo za namen nadzorovanja ali druge stvari, ki ne spadajo v obseg funkcij, za katere je bila značka prvotno namenjena. Tako bi lahko sledili predmetom, ki vsebujejo značko, s tem pa tudi ljudem, ki te predmete prenašajo [2].

Namen diplomskega dela je z uporabo odprtih orodij načrtovati in implementirati manjši sistem RFID, ki je postavljen v laboratoriju in se uporablja za sledljivost oseb, knjig in opreme v laboratoriju. Cilji postavljenega sistema so:

- Izposoja in vračanje knjig in opreme v laboratoriju. Vsaka knjiga ali oprema mora imeti svojo značko, oseba pa mora imeti svojo kartico za dostop.
- Urejanje značk, torej možnost, da poljubni znački priredimo poljubno knjigo, opremo ali osebo.
- Pregled in urejanje tabel s podatki o osebah, knjigah in opremi v podatkovni bazi.
- Beleženje oseb s kartico za dostop, ki vstopijo in izstopijo iz laboratorija.

Za dosego ciljev je potrebno rešiti sledeče probleme:

- Izbrati ustrezna orodja za razvoj aplikacije sledenja objektov z uporabo RFID.
- Izbrati ustrezno podatkovno bazo in načrtovati relacijski model tabel, ki je primeren za hranjenje podatkov o osebah, knjigah in opremi.
- Preučiti protokole za komunikacijo z uporabljenno strojno opremo ter načrtovati in izdelati aplikacijo, ki zna te protokole ustrezno implementirati.
- Izdelati uporabniški grafični vmesnik, ki omogoča preprosto uporabno vseh funkcij, ki jih sistem ponuja.

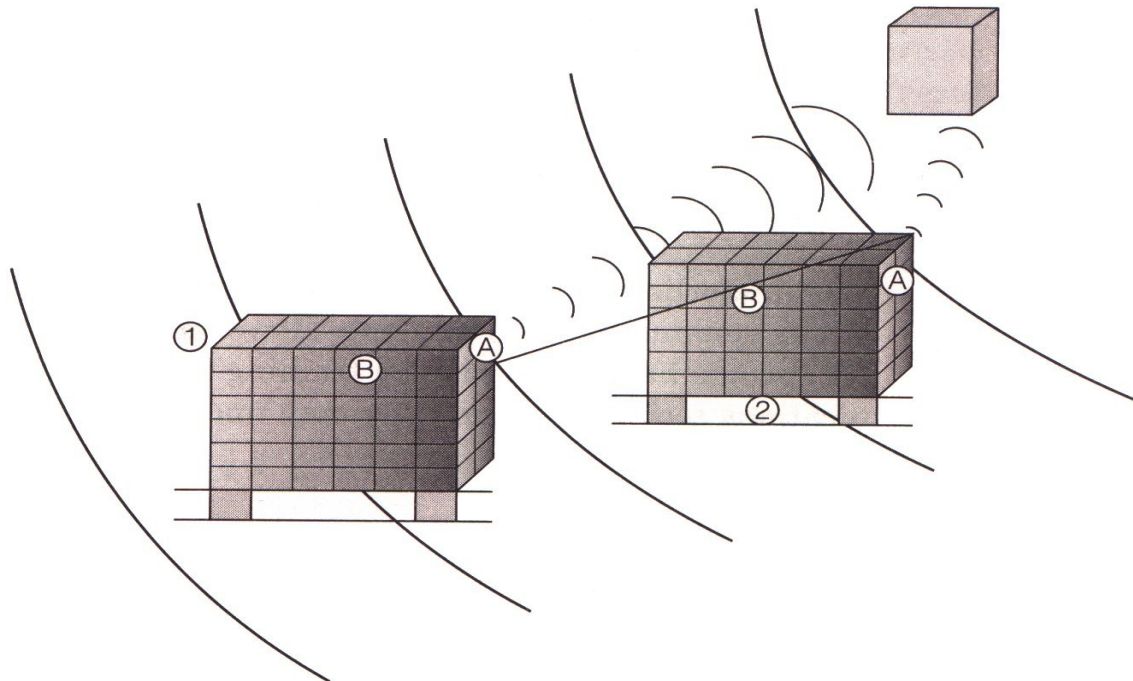
2. Uporabljene tehnologije in orodja

2.1. Tehnologija RFID

Tehnologija RFID temelji na radijskem valovanju. Za razliko od svetlobnega valovanja lahko radijski valovi potujejo skozi večino materialov. Brez težav lahko potujejo skozi papir, karton, les, blago, plastiko, usnje in koruzne kosmiče. Kljub temu, da radijski valovi lahko potujejo skozi večino materialov, skozi nekatere materiale potujejo manj uspešno. Taki materiali so npr. kovina, grafit in tekočine ter se imenujejo neprozorni materiali. Radijski valovi imajo dve lastnosti, ki vplivata na zmožnost prodiranja valov skozi neprozorne materiale: frekvenca/valovna dolžina in moč. Pri načrtovanju sistema RFID moramo biti pozorni na to, s kakšnimi materiali imamo opravka in glede na to ustrezno načrtovati sistem. Nekateri radijski valovi imajo težave pri potovanju skozi neprozorne materiale. Nizko frekvenčni in visoko frekvenčni valovi gredo lažje skozi neprozorne materiale ali jih obidejo vendar so nizko frekvenčne in visoko frekvenčne značke večje in dražje. Ultra visoko frekvenčne značke so občutno cenejše za izdelavo, njihova cena pa z vedno večjo uporabo še pada. Kljub svojim pomanjkljivostim pri potovanju skozi neprozorne materiale je njihova cena botrovala k temu, da so postale standard v primerih sistemov RFID, kjer je potrebno večje število značk.

Sposobnost radijskega valovanja, da lahko prodira skozi nekatere materiale pomeni, da za branje značk RFID ne potrebujemo vidne linije med čitalcem in značkami RFID. To je ena izmed ključnih prednosti, ki jih imajo značke RFID pred črtno kodo. Črtno kodo morajo biti prebrane vsaka posebej, ponavadi pa mora biti vključena tudi oseba, ki jo moramo plačati za to, da s čitalcem prebere vsako črtno kodo posebej. Značko RFID v dobro načrtovanem sistemu lahko preberemo, ko gre mimo čitalca, brez da bi bil v proces vključen človek.

Slika 1 prikazuje dve paleti, ki vsebujeta artikla z značkami RFID. Na paleti 1 so pločevinke. Značka (A) na paleti 1 je obrnjena proti čitalcu zato od njega dobi signal in se na zahtevo odzove. Značka (B) na paleti 1 je na drugi strani palete. Ker je kovina eden izmed neprozornih materialov, signal ne doseže značke, zato se ta ne odzove na zahtevo. Na paleti 2 pa so papirnati prtički. Radijski valovi brez težav potujejo skozi papir, zato obe znački (A) in (B) na paleti 2 dobita signal in se odzoveta na zahtevo.



Slika 1. (1) Paleta pločevink (2) Paleta papirnatih prtičkov.

2.1.1. Vloga frekvence pri sistemih RFID

Delovanje sistema RFID je v veliki meri odvisno od frekvenčnega območja, ki vpliva na hitrost branja podatkov, razdaljo, s katere lahko preberemo značko, občutljivost na neprozorne materiale in drugo. Sistemi RFID se tako delijo na tri vrste [1]: nizkofrekvenčne (LF – ang. Low-frequency), visokofrekvenčne (HF – ang. High-frequency) in ultra visokofrekvenčne (UHF – ang. Ultra high-frequency) sisteme RFID glede na to, na katerem frekvenčnem pasu delujejo. Tabela 1 povzema osnovne lastnosti in značilnosti posameznega sistema RFID.

	LF	HF	UHF
Frekvenca	9 – 135 kHz	13,56 – 15,56 kHz	860 – 930 MHz
Standardi	ISO/IEC 18000-2	ISO/IEC 18000-3, ISO 15693, ISO 14443	ISO/IEC 18000-6, EPCglobal
Uporaba v RFID	Označevanje živali, kontrola dostopa	Označevanje knjig v knjižnicah, biometrija	Preskrbovalne verige, upravljanje s sredstvi
Neprozorni materiali	Neobčutljivi	Malo občutljivi	Zelo občutljivi
Hitrost branja	Počasna	Srednje hitra	Hitra
Razdalja branja	½ metra	1 meter	4 – 5 metrov

Tabela 1. Lastnosti sistemov RFID.

2.1.1.1. Nizkofrekvenčni sistemi RFID

LF pas predstavlja območje 9 – 135 kHz. Ta frekvenčni pas opisujejo dolgi valovi in se uporablja pri komunikaciji na dolge razdalje. Tipične radijske storitve za to frekvenčno območje so npr. aeronavtična komunikacija in radijska komunikacija za vojsko. Da bi preprečila poseg v ta frekvenčni pas je evropska oblast določila pas med 70 in 119 kHz v katerem so sistemi RFID prepovedani. LF sistemi RFID uporabljajo frekvence v pasu 125 – 134 kHz, kjer je valovna dolžina okoli 2300 m.

Taki sistemi RFID se uporabljajo pretežno za označevanje živali, kontrolo dostopa, preprečevanje kraje vozil in niso preveč občutljivi na neprozorne materiale. To je najstarejša oblika sistemov RFID, hitrosti branja so relativno počasne, antene, potrebne pri uporabi takega sistema, pa so velike in dokaj drage. LF značke lahko beremo z razdalje do pol metra, so komercialno dostopne in nanje lahko shranimo do 60 znakov. Taki sistemi RFID so idealni, če okolje, v katerem jih uporabljamo, vsebuje umazanijo, sneg ali blato.

2.1.1.2. Visokofrekvenčni sistemi RFID

HF pas predstavlja sredino tako imenovanega kratkega frekvenčnega pasu, ki je primeren za medcelinsko komunikacijo in se uporablja v telekomunikacijah. HF sistemi RFID uporabljajo območje 13,56 – 15,56 MHz. Valovna dolžina pri 13,56 MHz je približno 22 m. Taki sistemi se uporabljajo pri pametnih karticah, kontroli dostopa, kontroli prtljage, biometrični identifikaciji in v knjižnicah.

2.1.1.3. Ultra visokofrekvenčni RFID sistemi

UHF sistemi RFID uporabljajo območje 860 – 930 MHz. Valovna dolžina je približno 33 cm. Ta frekvenčni pas je rezerviran za ISM (ang. Industrial-Scientific-Medical). Z UHF sistemi upravlja specifikacija ISO/IEC 18000-6 in nova Gen-2 specifikacija EPCglobal. UHF značke so daleč najcenejše in se veliko uporabljajo pri preskrbovalnih verigah za sledenje paletam in označevanje škatel. Uporabljajo se tudi za elektronsko cestninjenje nekatera podjetja pa jih uporabljajo tudi za upravljanje sredstev. UHF značke, lahko shranijo do 8000 znakov in so berljive z razdalje 4 – 5 metrov. Ena izmed slabosti UHF značk je to, da na različnih celinah

delujejo na različnih frekvenčnih pasovih.

2.1.2. Elementi sistema RFID

Sistem RFID v osnovi sestavljajo štiri elementi: značke RFID, čitalec RFID, antene lahko pa še računalniško omrežje, za povezavo več čitalcev [2].

2.1.2.1. Značke RFID

Značka je osnovni gradnik RFID. Sestavljata jo antena in majhen čip iz silicija, ki vsebuje radijski sprejemnik, radijski modulator za pošiljanje odgovora nazaj čitalcu, logiko za upravljanje, določena količina pomnilnika in sistem za napajanje. Sistem za napajanje se je zmožen napajati izključno s prejetim radijskim signalom. V tem primeru se značka imenuje pasivna značka. Če sistem za napajanje vsebuje tudi baterijo, se značka imenuje aktivna značka.

Glavni prednosti aktivnih značk sta njihov doseg in zanesljivost. Z ustrežno anteno na čitalcu in znački lahko na frekvenci 915 MHz značko preberemo iz razdalje 30 m ali več. Aktivne značke so ponavadi tudi bolj zanesljive, ker ne potrebujejo neprekinjenega radijskega signala za napajanje. Baterije na aktivni znački zdržijo nekaj let.

Pasivne značke so lahko precej manjše in cenejše od aktivnih, ker ne vsebujejo baterije. Še ena prednost je daljša življenjska doba. Značko lahko uporabljamo desetletja po tem, ko je bil čip izdelan.

Obstajajo tudi polpasivne značke. Te značke imajo baterijo, kot aktivne značke, ampak za oddajanje signala nazaj čitalcu še vedno uporabljajo napajanje radijskega signala. Imajo zanesljivost branja aktivnih značk in doseg branja pasivnih značk. Imajo tudi daljšo življenjsko dobo kot aktivne značke.

Obstajajo značke vseh oblik in velikosti. Najmanjša izdelana značka je Hitachijev mu-chip, ki v dolžino meri manj kot 0,4 mm. Zasnovan je bil za vgradnjo v list papirja za sledljivost dokumentov natisnjenih v pisarni. mu-chip je pasivna značka in jo je možno brati z razdalje

nekaj centimetrov [2].

Najpreprostejše značke RFID vsebujejo samo UID (ang. Unique Identifier), tj. 64 ali 96 bitni blok podatkov samo za branje. UID je lahko vprogramiran na čip s strani proizvajalca lahko pa ga vprogramira tudi uporabnik. Nekatere značke podpirajo samo eno serijsko številko, pri drugih pa jo lahko spremenimo po tem, ko je bila že vprogramirana. Bolj napredni čipi vsebujejo tudi pomnilnik za branje/pisanje, v katerega lahko piše čitalec [2].

Značke RFID lahko motijo ena drugo. Če je v dosegu čitalca več značk naenkrat, se lahko zgodi, da čitalec ne bo mogel dešifrirati signalov značk. V mnogih primerih, kot na primer dviganje garažnih vrat, to ni problem. Taki sistemi so optimizirani tako, da je v dosegu samo ena značka naenkrat. V drugih primerih pa je pomembno, da lahko preberemo več značk, ki so v dosegu čitalca v istem trenutku. V takih primerih antikolizijski protokol čitalcu omogoča, da zazna, da je v njegovem dosegu več značk in jih lahko po vrsti prebere. Značke morajo upoštevati uporabo protokola in izmenično odgovarjati na zahteve čitalca.

2.1.2.2. Čitalci RFID

Čitalec RFID pošlje pulz radijske energije znački in posluša njen odgovor. Značka zazna poslano energijo in pošlje nazaj odgovor, ki vsebuje serijsko številko značke lahko pa tudi kakšne dodatne informacije. Pri enostavnih sistemih RFID deluje pulz energije kot stikalo za vklop/izklop. Pri naprednejših sistemih pa lahko vsebuje ukaze znački, ukaze za branje ali pisanje v pomnilnik, ki ga ima značka, ali celo gesla. Čitalci so bili v osnovi zasnovani tako, da lahko berejo samo eno vrsto značk, toda vse bolj priljubljeni so čitalci, ki znajo brati več vrst značk. Čitalci RFID so ponavadi ves čas vklopljeni in ves čas oddajajo radijsko energijo ter čakajo, da značka vstopi v njihov doseg branja. V določenih primerih, npr. ko je treba iz kakršnih koli razlogov varčevati z energijo, je tako obnašanje nezaželeno zato nekateri čitalci pošljejo radijski pulz šele ob nekem zunanjem dogodku [2].

Obstajajo čitalci različnih velikosti. Večji čitalec bi lahko obsegal osebni računalnik s posebno kartico in priklopljenimi več antenami. Tak računalnik bi bil verjetno priklopljen tudi na omrežje, da bi lahko informacije o prebranih značkah pošiljal drugim računalnikom.

Najmanjši čitalci so velikosti poštnih znamk primerni za vgradnjo v mobilne telefone.

2.1.3. Uporaba RFID

Uporaba tehnologije RFID se v splošnem deli na štiri področja, ki se razlikujejo po osnovnem načinu delovanja. Področja so:

- Sledenje predmetom
- Elektronsko plačevanje
- Kontrola dostopa
- Telematika

2.1.3.1. Sledenje predmetom

Na tem področju se izkorišča prednost tehnologije, da lahko enolično identificiramo predmet in posredujemo identifikacijo skupaj z npr. časom, lokacijo in informacijami pridobljenimi s senzorji za temperaturo, vlago in drugo. Zmožnost povezovanja identificiranega predmeta z zapisi v podatkovni bazi zelo poveča uporabnost sistema. Primeri predmetov, ki jim sledimo so: prtljaga na letališču, artikli v zalogi, oprema, pacienti v bolnišnicah, zaporniki, knjige v knjižnicah in drugo.

2.1.3.2. Elektronsko plačevanje

Uporablja se na avtocestah v obliki elektronskega cestninjenja. Uporabnik kupi značko z enolično identifikacijsko številko in jo napolni z določenim zneskom. Ko se voznik pripelje mimo cestninske postaje, značka odda identifikacijsko številko in iz računa uporabnika se odšteje ustrezen znesek. Primer elektronskega plačevanja je tudi nova kartica Urbana, ki se uporablja za javni prevoz v Ljubljani.

2.1.3.3. Kontrola dostopa

Gre za dostopne kartice, ki imajo vgrajeno RFID značko. Značko približamo čitalcu, ki je

povezan na varnostni sistem. Ko se prebere značka sistem preveri ali ima uporabnik pravico do dostopa in v tem primeru odklene elektronsko ključavnico ali premična vrata. Sistem zabeleži tudi čas dostopa in identifikacijsko številko značke.

2.2. Operacijski sistem

Pri izdelavi diplomske naloge je bil za razvoj in namestitvev aplikacije uporabljen operacijski sistem Linux. To je splošen izraz, ki opisuje Unixu podoben operacijski sistem, ki temelji na jedru Linux. Gre za odprt operacijski sistem, kar pomeni, da je njegova izvorna koda prosto dostopna in zaščitena z licenco GNU GPL (ang. GNU General Public License) [3], [4]. Linux je znan pretežno po njegovi uporabi na področju strežniških sistemov, čeprav ga je mogoče namestiti na širok izbor strojne opreme, od vseh vrst vgrajenih naprav, mobilnih telefonov in celo nekaterih ročnih ur do superračunalnikov. V zadnjih letih pa se je povečala tudi uporaba Linuxa na namiznih in prenosnih računalnikih. Ker je Linux samo jedro operacijskega sistema, skrbi predvsem za upravljanje s strojno opremo in ne vsebuje niti uporabniškega vmesnika. Zato se Linux uporablja v paketu, ki poleg jedra vsebuje vrsto orodij, sistemskih in uporabniških programov, uporabniški grafični vmesnik in drugo. Vsi programi in orodja, ki obdajajo jedro, so prav tako odprti in je njihova izvorna koda prosto dostopna. Tak paket jedra skupaj s programi imenujemo distribucija Linuxa. Obstajajo različne distribucije, ki so namenjene različnim ciljnim skupinam uporabnikov in tako poleg jedra, ki ga vsebuje vsaka distribucija, vsebujejo različne izbore programov in orodij, ki so namenjeni točno določenim uporabnikom.

2.3. Razvojno ogrodje

Za razvojno ogrodje (ang. framework) je bil uporabljen Qt podjetja Nokia, odprto razvojno ogrodje za aplikacije z grafičnim uporabniškim vmesnikom uporablja pa se tudi za aplikacije brez grafičnega vmesnika, npr. za orodja v tekstovnem načinu ali za strežnike. Qt ponuja tudi dodatna orodja, npr. orodje za enostavno gradnjo uporabniškega vmesnika in orodje za poenostavitev pisanja programske dokumentacije [6]. Uporablja standardni jezik C++ vendar veliko uporablja predprocesor C, za obogatitev in dodajanje funkcionalnosti v jezik. Qt teče na vseh glavnih operacijskih sistemih kot so Linux, Windows, Mac OS X in platforma S60 za

telefone z operacijskim sistemom Symbian. Ima tudi obširno podporo za prevajanje aplikacij v več jezikov.

Ponuja vrsto standardnih kontrol, ki se uporabljajo za grajenje grafičnega uporabniškega vmesnika, npr. gumbе, vnosna polja, drsnike, sezname elementov, tabele, in drugo. Z vidika aplikacij brez grafičnega vmesnika pa ponuja podporo za delo z podatkovnimi bazami, XML analizo, upravljanje z nitmi, podporo za omrežno komunikacijo in podporo za upravljanje z datotekami [7].

2.4. Razvojno orodje

Aplikacija je razvita z razvojnim orodjem Eclipse CDT (ang. C/C++ Development Toolkit), ki je integrirano razvojno okolje (ang. IDE – Integrated Development Environment), ki združuje množico orodij, ki so potrebna za razvoj aplikacij, in jih ponuja v obliki grafičnega vmesnika. To razvijalcu omogoča, da porabi manj časa pri preklapljanju med različnimi orodji in se osredotoči na kvaliteto rešitve, ki jo razvija. Omogoča razvoj aplikacij v različnih jezikih, kot so npr. C/C++, Java, in PHP. Eclipse CDT omogoča razvoj aplikacij v programskem jeziku C/C++. Okolje je zelo razširljivo. Z uporabo dodatnih vtičnikov (ang. plug-in) je okolju mogoče dodati funkcionalnosti, ki jih osnovno okolje ne podpira. Uporabil sem vtičnik za Qt aplikacije, ki je del dodatnih orodij v razvojnem okolju Qt.

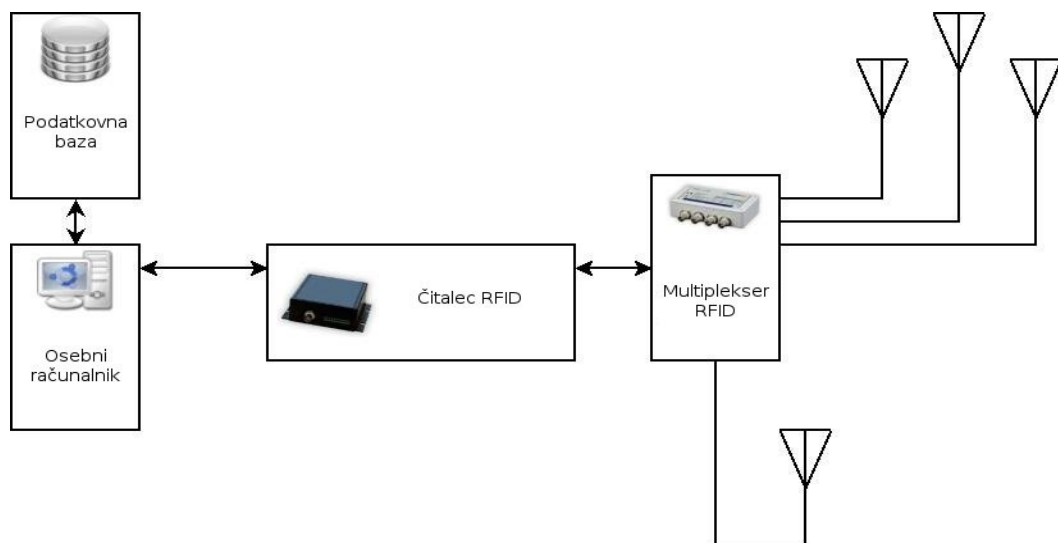
2.5. Podatkovna baza

Aplikacija uporablja podatkovno bazo PostgreSQL. To je odprt objektno-relacijski sistem za upravljanje podatkovnih baz (ang. ORDBMS). Aktivno je v razvoju že 15 let in je v tem času pridobil na slovesu kot podatkovna baza, ki je zanesljiva in zagotavlja integriteto in pravilnost podatkov. PostgreSQL teče na vseh glavnih operacijskih sistemih kot so UNIX, Linux in Windows. Razvijalci dajejo velik poudarek na standarde zato se PostgreSQL tesno prilagaja standardoma ANSI SQL 92/99. Ima veliko domorodnih (ang. native) vmesnikov za programiranje, med drugim C/C++, Java, ODBC in odlično dokumentacijo [5].

3. Strojni vidik sistema RFID

Za izdelavo programske aplikacije, ki omogoča identifikacijo oseb ob vstopu v prostor, identifikacijo objektov (knjige, oprema) in njihovo sledenje, je bil uporabljen sistem RFID, ki ga sestavljajo elementi (slika 2):

- Osebni računalnik
- Podatkovna baza
- Čitalec RFID
- Multiplekser RFID
- Antene
- Značke RFID



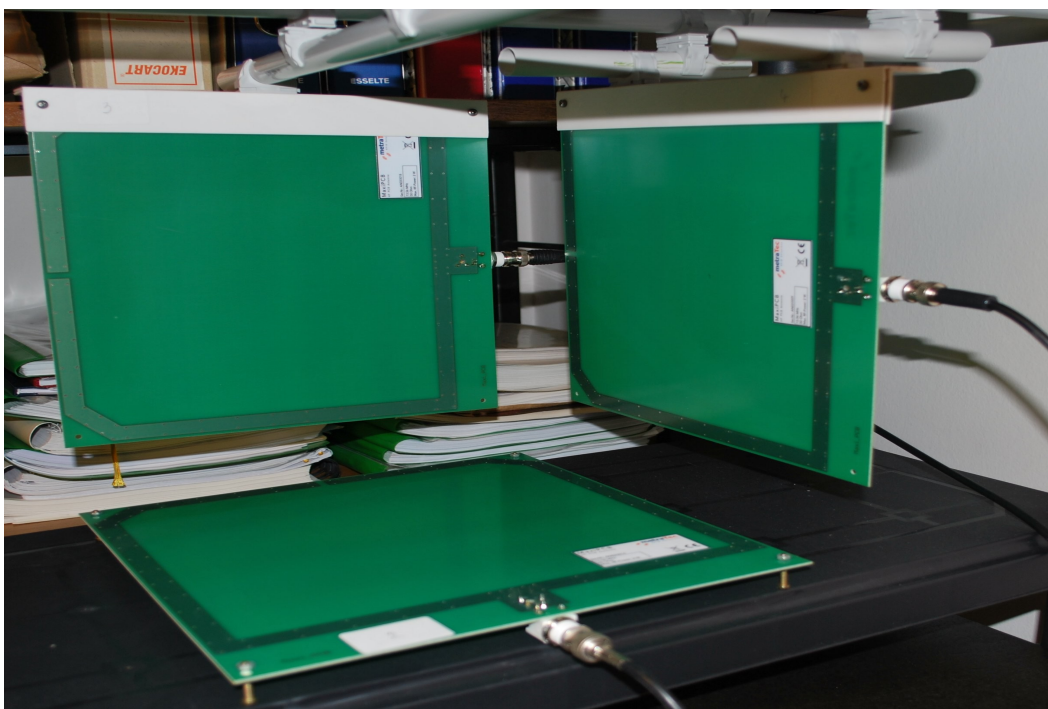
Slika 2. Shema sistema RFID v laboratoriju.

Jedro sistema je osebni računalnik, na katerem teče aplikacija, ki vodi in usmerja delovanje. Vse informacije, ki se v sistemu prenašajo, pridejo v nekem trenutku do računalnika, ta pa jih ustrezno usmerja naprej.

Računalnik je povezan s podatkovno bazo, ki skrbi za trajno hranjenje podatkov. Sistem za upravljanje podatkovnih baz lahko teče na istem računalniku, kjer se izvaja aplikacija, lahko pa je dostopen preko omrežja, npr. preko lokalnega omrežja ali preko interneta.

Čitalec RFID je z računalnikom povezan preko omrežja Ethernet. Lahko je s kablom povezan neposredno z računalnikom, lahko pa je med računalnikom in čitalcem tudi omrežje, npr. lokalno omrežje s stikali in drugimi napravami ali pa je povezava vzpostavljena kar preko interneta. Čitalec sam od sebe ne dela nič in sam ne bo nikoli sprožil komunikacije z drugimi napravami. Vedno čaka na zahtevo od računalnika in šele nato sproži ustrezno akcijo. Ko obdela zahtevo računalniku vrne odgovor in čaka na naslednjo zahtevo.

Čitalec je povezan z multiplekserjem preko koaksialnega kabla, preko katerega se prenašajo signali, ki jih ujame antena. Poleg koaksialnega kabla sta čitalec in multiplekser povezana z dvema dodatnima žicama za naslavljanje anten, ki so povezane na multiplekser. Vsaka izmed dveh dodatnih žic lahko prenaša logično stanje 0 ali 1, kar pomeni, da lahko naslavljamo štiri antene. Čitalec lahko v danem trenutku bere le iz ene antene. Če želimo npr. dobiti podatke o značkah od vseh anten moramo zaporedno nasloviti eno za drugo in za vsako posebej sprožiti zahtevo za branje značk.



Slika 3. Simulacija zaznavanja dostopov z uporabo 3D branja.

Multiplekser je preko štirih koaksialnih kablov povezan s štirimi antenami. Tri od teh anten so uporabljene za simulacijo zaznavanja oseb z dostopno kartico (slika 3), ko te vstopijo ali izstopijo iz laboratorija. Preostala, četrta antena, pa je uporabljena za izposajo in vračanje knjig in opreme ter za urejanje značk. Uporaba treh anten za zaznavanje vstopa / izstopa, nam omogoča 3D branje, ko lahko značko preberemo neodvisno od polarizacije. V praksi to pomeni, da lahko značko preberemo, ne glede na njeno orientacijo.

3.1. Čitalec QuasarMR1

QuasarMR1 je čitalec podjetja metraTec RFID Solutions (slika 4). To je čitalec srednjega dosega, ki deluje v HF frekvenčnem pasu (13,56 MHz). Z ustrežno kombinacijo anten in značk je zmožen brati značke z razdalje do 40 cm. Čitalec podpira različne protokole značk, med njimi tudi ISO15693, ISO14443A in ISO-180003 podpira pa tudi različne lastniške protokole, npr. značke I·Code1 podjetja Philips ali značke Tag-It HF podjetja Texas Instruments.

Za povezavo z ostalimi napravami čitalec omogoča povezavo preko USB kabla, in povezavo preko Ethernet omrežja. V diplomski nalogi sem uporabil povezavo preko Ethernet omrežja, kjer se čitalec obnaša kot običajna mrežna naprava, kar pomeni da ima svoj IP naslov in na določenih vratih posluša TCP zahteve. Prednosti takšne uporabe čitalca so: (i) med napravo in čitalcem imamo lahko daljši kabel (100 m po standardu Ethernet in 5 m po standardu USB 2.0); (ii) čitalec lahko priklopimo tudi v stikalo in je tako dostopen vsem računalnikom, ki so v istem omrežju, ali pa do njega lahko dostopamo kar preko internetne povezave; (iii) čitalec manj odvisen od operacijskega sistema. Podjetje metraTec ponuja USB gonilnik za čitalec samo za operacijske sisteme Windows. Če bi želeli čitalec preko USB povezave uporabiti na drugih operacijskih sistemih, bi morali zanj napisati USB gonilnik.

Poleg USB in Ethernet priključka ima čitalec še 8 splošno namenskih V/I pinov, ki jih je mogoče programsko krmiliti. Ti pini se lahko uporabijo za krmiljenje zunanjih naprav, kot so multiplekserji, alarmi, itd.



Slika 4. Čitalec QuasarMR1, priključen na napajanje in povezan s kablom UTP.

3.2. Multiplekser hfMUX-4-MP

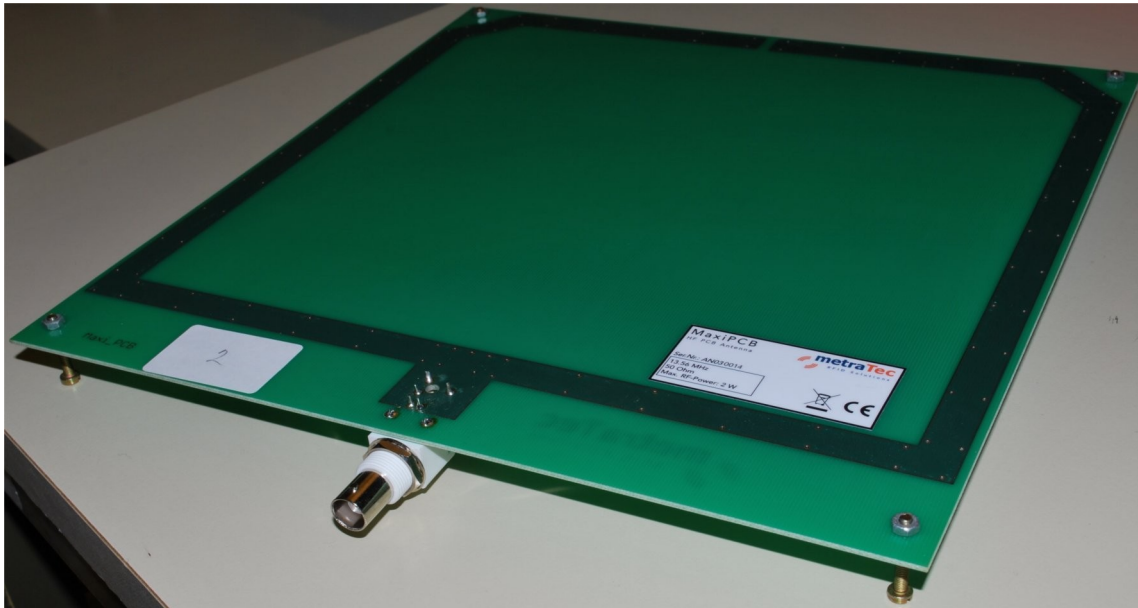
hfMUX-4-MP je multiplekser podjetja metraTec RFID Solutions (slika 5), ki je primeren za HF sisteme, ki delujejo na frekvenci 13,56 MHz z največjo močjo 2 W. Multiplekser omogoča istočasen priklop največ štirih anten na en čitalec RFID. To pomeni, da lahko zmanjšamo število čitalcev v našem sistemu in s tem zmanjšamo zapletenost sistema kot tudi stroške postavitve sistema.



Slika 5. Multiplekser hfMUX-4-MP povezan s štirimi antenami.

3.3. Antene MaxiPCB

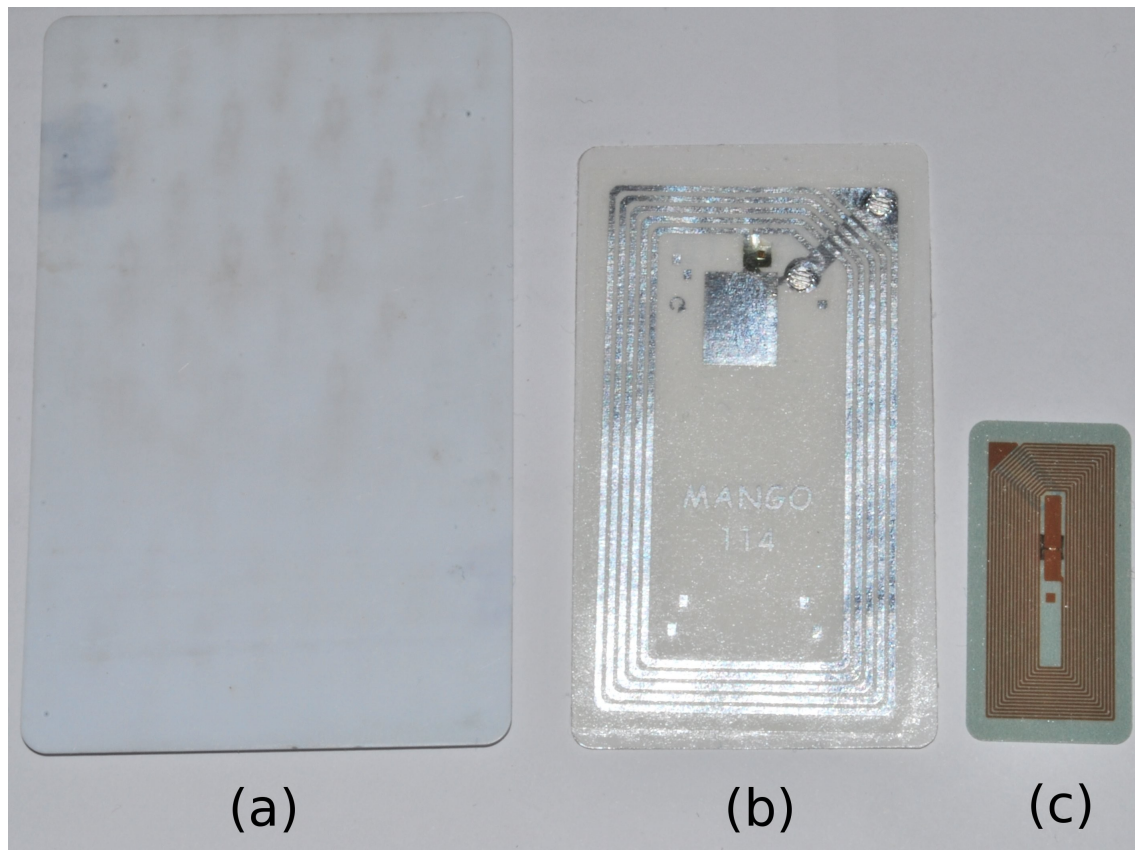
MaxiPCB (slika 6) so antene podjetja metraTec RFID Solutions, dimenzije 25 x 25 cm, ki so prirejene za delo na frekvenci 13,56 MHz. Z uporabo ustreznega čitalca in kartic omogočajo branje z razdalje do 40 cm.



Slika 6. Antena MaxiPCB.

3.4. Značke RFID

Za značke RFID sem uporabil značke Philips I-CODE SLI SL2 ICS20 (slika 7), ki podpirajo standard ISO15693. Značke so pasivne kar pomeni, da jih napaja energija radijskih valov, ki jih pošlje čitalec, in ne potrebujejo lastne baterije. Delujejo na frekvenci 13,56 MHz in jih je mogoče brati z razdalje do 1,5 m, z ustrežno anteno. Podpirajo hitrost prenosa podatkov do 53 kbit/s in podpirajo upoštevanje antikolizijskega protokola. Če imamo v dosegu več značk hkrati nam antikolizijska funkcija omogoča, da komuniciramo z eno samo, točno določeno značko, pri tem pa ostale značke v dosegu ne motijo komunikacije. Vsaka značka vsebuje enolični identifikator (ang. UID), ki je zapisan v proizvodnem procesu značke in ga ne moremo spremeniti. Zagotavlja nam, da je vsaka značka enolično razpoznavna. Značke se pojavljajo tako v obliki plastificiranih kartic kot tudi v obliki nalepk različnih velikosti.



Slika 7. Značke Philips I-CODE SLI.

- (a) značka v obliki plastificirane kartice*
- (b) hrbtna stran značke v obliki nalepke*
- (c) hrbtna stran značke v obliki manjše nalepke*

Značke Philips I-CODE SLI imajo v čipu 1024 bitov pomnilnika. Proizvajalec navaja obstojnost podatkov 10 let in vzdržljivost 100.000 pisalnih ciklov. Pomnilnik na značkah je obstojni EEPROM pomnilnik, ki je organiziran je v 32 blokov po 4 zloge. Prvi štirje bloki vsebujejo podatke, ki so na voljo za branje, npr. enolični identifikator značke in kontrolo uporabe značke. Ostalih 28 blokov pa je na voljo za branje in pisanje uporabniških podatkov. Značke nam omogočajo tudi zaklepanje podatkov za pisanje. Vsakega izmed 28 uporabniških blokov lahko zaklenemo za pisanje – po zaklepu pisanje v zaklenjeni blok ni več mogoče in ga ne moremo nikoli več odkleniti.

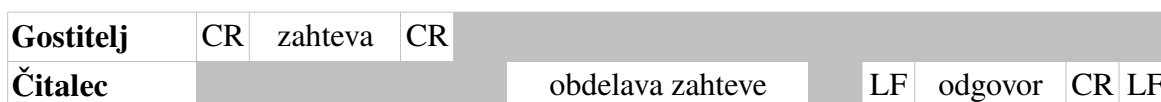
4. Programski vidik sistema RFID

4.1. Protokol za komunikacijo s čitalcem QuasarMR1

Čitalec QuasarMR1 za komunikacijo podpira protokol AURA podjetja SkyeTek, ki določa način izmenjave podatkov med gostiteljem (npr. osebni računalnik) in čitalcem RFID. Opisuje, kako gostitelj komunicira s čitalcem, kako ga lahko konfigurira in kako lahko na njem sproži akcije kot so branje in pisanje na značke RFID. Protokol opisuje izmenjavo podatkov le na podatkovnem nivoju. Komunikacija na strojnem nivoju je odvisna od vrste povezave čitalca z gostiteljem in naprej od operacijskega sistema na gostitelju.

Izmenjavo podatkov med gostiteljem in čitalcem poteka na dva načina: v formatu ASCII in v binarnem formatu. Komunikacijo vedno začne gostitelj s tem, da pošlje zahtevo čitalcu. Čitalec na vsako zahtevo pošlje odgovor. Format podatkov poslanih zahtev se vedno ujema s formatom podatkov odgovora (ASCII – ASCII in binarni – binarni).

Zahteva in odgovor ASCII

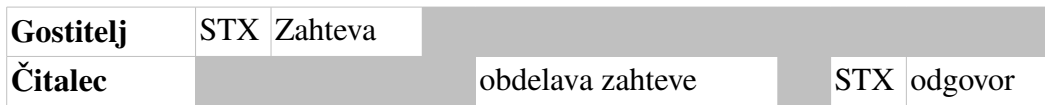


Slika 8. Časovni potek ASCII zahteve in odgovora.

Slika 8 prikazuje časovni potek zahteve in odgovora ASCII. Gostitelj čitalcu pošlje zahtevo obdano z znakoma CR (ang. Carriage Return) s kodo ASCII 13₍₁₀₎ oz. 0D₍₁₆₎. Čitalec prejme zahtevo, jo obdela in gostitelju vrne odgovor, ki ga obda z znakom LF (ang. Line Feed) na začetku in znakoma CR in LF na koncu. Znak LF ima v tabeli ASCII kodo 10₍₁₀₎ oz. 0A₍₁₆₎.

Vsak zlog (ang. byte) v zahtevi mora biti čitalcu poslan v obliki dveh šestnajstiških vrednosti: najprej spodnja četvorka (ang. nibble) zloga in nato še zgornja četvorka zloga. Npr. ko čitalcu pošiljamo znak CR, ki je šestnajstiško predstavljen kot 0D, pošljemo najprej prvi znak (0) kot kodo ASCII in nato še drugi znak (D) kot kodo ASCII. To pomeni, da za vsak zlog v zahtevi čitalcu pošljemo dva zloga.

Binarna zahteva in odgovor



Slika 9. Časovni potek binarne zahteve in odgovora.

Slika 9 prikazuje časovni potek binarne zahteve in odgovora. Gostitelj čitalcu pošlje zahtevo, ki jo naznani s kontrolnim znakom STX (ang. Start Transmission) s kodo ASCII 02. Ko čitalec zahtevo obdela pošlje gostitelju kontrolni znak STX, temu pa sledi odgovor.

V primeru binarnega prenosa mora biti vsak naslednji zlog oddan v roku 10 ms od oddaje prejšnjega zloga. Če več kot 10 ms ni prenesenega nobenega zloga se to smatra kot konec oddajanja.

Pri povezavi čitalca preko protokola Ethernet ni mogoče vedno zagotoviti, da bo naslednji zlog prišel na cilj v roku 10 ms od prihoda prejšnjega zloga, še posebej, če do čitalca dostopamo preko interneta, zato ta rešitev ni najbolj primerna. Zato je pri komunikaciji s čitalcem uporabljen protokol ASCII, kjer je začetek in konec okvirja določen s kontrolnimi znaki in ne s časovnim intervalom.

4.1.1. Zahteva ASCII

Vsaka zahteva, poslana čitalcu, ima format, kot ga prikazuje slika 10. Polja obarvana s svetlejšo barvo predstavljajo obvezna polja, ki morajo biti prisotna v vsaki zahtevi. Polja obarvana s temnejšo barvo so prisotna samo v določenih vrstah zahtev ali pa so opcиска, kar pomeni, da sami izbiramo ali bomo polje vključili v zahtevo ali ne.

zastavice	zahteva	RID	tip značke	TID	AFI	začetni blok	št. blokov	podatki	CRC
8 bitov	8 bitov	8 bitov	8 bitov	64 bitov	8 bitov	8 bitov	8 bitov	n*8 bitov	16 bitov

Slika 10. Format zahteve ASCII.

Polje zastavice

Polje z zastavicami vsebuje podatek o tem, katera od neobveznih polj bodo prisotna v zahtevi čitalcu. Vsak izmed osmih bitov predstavlja eno polje v zahtevi (slika 11). Če je zastavica postavljena (vrednost bita je 1), potem je polje, ki ga zastavica predstavlja, vključeno v zahtevo. Če zastavica ni postavljena (vrednost bita je 0), potem polja v zahtevi ni.

RID_F	TID_F	CRC_F	AFI_F	RF_F	LOCK_F	INV_F	LOOP_F
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Slika 11. Format polja z zastavicami.

- **RID_F**

Zastavica pove ali je polje RID (Reader ID) del zahteve. Uporablja se v primeru, ko je na gostitelja priključenih več čitalcev in bi želeli nasloviti točno določenega med njimi. Če zastavica ni postavljena, potem se polje RID ne pošlje kot del zahteve čitalcu.

- **TID_F**

Zastavica pove ali je polje TID (Tag ID) del zahteve. Ker ima vsaka značka svoj enoličen TID, nam to omogoča, da naslavljammo točno določeno značko. Zastavica je smiselna le pri zahtevah, ki se nanašajo na točno določeno značko, npr. branje ali pisanje na značko.

- **CRC_F**

Zastavica pove ali je polje CRC del zahteve. CRC nam omogoča zaznavanje napak pri prenosu podatkov. Če je zastavica ni postavljena, potem se CRC ne pošlje skupaj z zahtevo in se preverjanje pravilnosti prenosa ne izvaja razen v primeru zahtev za pisanje, kjer moramo CRC obvezno vključiti v zahtevo.

- **RF_F**

Zastavica pove ali naj čitalec ostane prižgan tudi po tem, ko se zahteva izvede in čitalec vrne odgovor. To je pomembno v primerih, ko želimo, da si značke zapomnijo informacijo o trenutnem stanju (npr. pri ukazu „ostani tiho“). Če je zastavica postavljena na 0, potem se čitalec po izvedeni zahtevi ugasne in vse značke, ki so sicer

v njegovem dosegu, izgubijo napajanje.

- **LOCK_F**

Zastavica nam omogoča zaklepanje podatkovnih blokov na značkah. Če na znački zaklenemo določen blok za pisanje, potem na ta blok ne moremo nikoli več zapisati informacije. Omogočeno je samo še branje tega bloka.

- **INV_F**

Zastavica nam omogoča vklop preprečevanja kolizije. V primeru, da zastavica ni postavljena in je v dosegu čitalca več značk bo čitalec prebral TID samo prve značke, ki bo odgovorila na zahtevo. Če pa je zastavica INV_F postavljena, bo čitalec poleg zahteve za oddajo TID poslal še zahtevo „ostani tiho“. V tem primeru bo značka oddala svoj TID na naslednje zahteve čitalca pa se ne bo več odzivala. Čitalec tako lahko zaporedno prebere vse značke, ki so v njegovem dosegu. Da pa si značka zapomni, da mora ostati tiho, mora med posameznimi zahtevami čitalec ostati prižgan, da ostanejo značke v njegovem dosegu napajane. Uporaba zastavice INV_F ima tako smisel le, če zraven nastavimo še zastavico RF_F.

Polje zahteva

Vsako zahtevo sestavljata natanko ena vrsta zahteve in en naslovljen pomnilnik. Polje zahteve je tako sestavljeno iz dveh skupin bitov – zgornja četvorka predstavlja vrsto zahteve, spodnja pa pomnilnik, ki ga bomo z zahtevo naslovili (slika 12).

Vrsta zahteve				Pomnilnik, ki ga z zahtevo naslavljamo			
Rezervirano (vedno 0)	Write_Bit	Read_Bit	Sel_Bit	Rezervirano (vedno 0)	Tag_Bit	Sys_Bit	Mem_Bit
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Slika 12. Format polja z zahtevo

Tabela 2 prikazuje zahteve, ki jih podpira protokol.

Šestnajstiška vrednost	Ime	Opis
14	SELECT_TAG	Zahteva za TID vseh značk, ki so v dosegu čitalca
21	READ_MEM	Branje iz obstojnega pomnilnika na čitalcu
22	READ_SYS	Branje iz neobstojnega pomnilnika na čitalcu
24	READ_TAG	Branje iz pomnilnika na znački
41	WRITE_MEM	Pisanje v obstojni pomnilnik na čitalcu
42	WRITE_SYS	Pisanje v neobstojni pomnilnik na čitalcu
44	WRITE_TAG	Pisanje v pomnilnik na znački

Tabela 2. Podprte zahteve.

Polje tip značke

To polje mora biti nujno prisotno v zahtevi, kadar se zahteva nanaša na značko (kadar je Tag_Bit=1). Polje določa tip značke RFID s katero bo čitalec poskušal komunicirati.

Polje RID

Polje se uporablja za naslavljanje čitalca s katerim bo gostitelj komuniciral v primeru, če imamo na gostitelja priklopljenih več čitalcev hkrati. Polje je v zahtevi prisotno le, če postavljena zastavica RID_F.

Polje TID

Polje vsebuje TID (serijsko številko) značke v primerih, ko mora čitalec komunicirati s točno določeno značko. Polje je smiselno le v primeru, ko se zahteva nanaša na značko. Polje je vključeno v zahtevo, samo če je TID_F=1.

Polje začetni blok

Polje predstavlja začetni naslov pri operacijah branja in pisanja. Polje se ne uporablja pri SELECT zahtevah. Kadar se zahteva nanaša na čitalec blok predstavlja en zlog podatkov. Kadar pa se zahteva nanaša na značke pa je število zlogov, ki ga predstavlja en blok, odvisno od tipa kartice, ki jo naslavljamo.

Polje podatki

Polje predstavlja podatke, ki bodo zapisani v okviru zahteve za pisanje. Polje se ne uporablja pri zahtevah SELECT in READ.

Polje CRC

Polje CRC je pri ASCII zahtevah poljubno, razen ko je zahteva pisanje. Uporablja se za preverjanje pravilnosti prenosa podatkov. Gre za 16 bitni CRC, ki se izračuna na gostitelju na podlagi vseh polj, ki jih zahteva vključuje (razen kontrolnih znakov).

4.1.2. Odgovor ASCII

Vsak odgovor, ki ga čitalec vrne gostitelju ima format, kot ga kaže slika 13.

odgovor	RID	tip značke	podatki	CRC
8 bitov	8 bitov	8 bitov	n*8 bitov	16 bitov

Slika 13. Format odgovora ASCII.

Polje odgovor

Polje vsebuje rezultat zahteve, ki jo je obdelal čitalec. Rezultat je lahko uspešen ali neuspešen. Odgovori in njihove kode so navedeni v tabeli 3.

Polje RID

Polje je v odgovoru prisotno samo če je bila pri zahtevi postavljena zastavica RID_F. Polje vsebuje RID čitalca, od katerega je prišel odgovor.

Polje tip značke

Polje je v odgovoru prisotno samo če je bila poslana zahteva SELECT_TAG in je bilo polje tip značke postavljeno na Auto, kar pomeni, da smo dali zahtevo za branje značk poljubnega tipa.

Polje podatki

Polje vsebuje podatke, ki so bili zahtevani. V primeru zahteve SELECT_TAG ti podatki predstavljajo TID značke. V primeru branja pomnilnika na čitalcu ali na značkah ti podatki predstavljajo prebrane podatke.

Polje CRC

Polje je v odgovoru prisotno, če je bila v zahtevi postavljena zastavica CRC_F. Gre za 16 bitni CRC izveden nad vsemi polji, ki so v odgovoru (razen kontrolnih znakov).

Koda odgovora (šestnajstiško)	Odgovor	Opis
14	SELECT_TAG Success	Uspešno zaznana značka v dosegu čitalca
94	SELECT_TAG Failure	Ni več nobene značke v dosegu čitalca
21	READ_MEM Success	Branje iz obstojnega pomnilnika v čitalcu je bilo uspešno
22	READ_SYS Success	Branje iz neobstojnega pomnilnika v čitalcu je bilo uspešno
24	READ_TAG Success	Branje iz pomnilnika na znački je bilo uspešno
A1	READ_MEM Failure	Branje iz obstojnega pomnilnika v čitalcu ni bilo uspešno
A2	READ_SYS Failure	Branje iz neobstojnega pomnilnika v čitalcu ni bilo uspešno
A4	READ_TAG Failure	Branje iz pomnilnika na znački ni bilo uspešno
41	WRITE_MEM Success	Zapisovanje v obstojni pomnilnik v čitalcu je bilo uspešno
42	WRITE_SYS Success	Zapisovanje v neobstojni pomnilnik v čitalcu je bilo uspešno
44	WRITE_TAG Success	Zapisovanje v pomnilnik na znački je bilo uspešno

Koda odgovora (šestnajstiško)	Odgovor	Opis
C1	WRITE_MEM Failure	Zapisovanje v obstojni pomnilnik v čitalcu ni bilo uspešno
C2	WRITE_SYS Failure	Zapisovanje v neobstojni pomnilnik v čitalcu ni bilo uspešno
C4	WRITE_TAG Failure	Zapisovanje v pomnilnik na znački ni bilo uspešno
80	Non-ASCII Character in Request	Zahteva vsebuje neveljaven znak
81	CRC not valid	Neveljaven CRC
82	Flags don't fit with Request	Zastavice ne ustrezajo zahtevi
83	Flags don't fit with Tag Type	Zastavice ne ustrezajo tipu značke
84	Unknown Request	Neznana zahteva
85	Unknown Tag Type	Neznan tip značke
86	Invalid Starting Block	Napačen začetni naslov
87	Invalid Number of Blocks	Napačno število blokov
88	Invalid Request Length	Napačna dolžina zahteve

Tabela 3. Vsi možni odgovori čitalca QuasarMR1 s pripadajočimi kodami

4.1.3. Organizacija pomnilnika v čitalcu

Čitalec QuasarMR1 ima 256 zlogov EEPROM pomnilnika. Del tega pomnilnika je namenjen sistemskim parametrom. Čitalec omogoča branje in nastavljanje teh parametrov in s tem nastavljanje obnašanja čitalca. Parametre lahko spreminjamo na dva načina:

- Spremenimo jih v obstojnem pomnilniku na čitalcu. Spremembe, ki jih naredimo v obstojnem pomnilniku začnejo veljati ob zagonu čitalca. Čitalec prebere parametre iz obstojnega pomnilnika in jih prepíše v neobstojni pomnilnik.
- Spremenimo jih direktno v neobstojnem pomnilniku. Spremembe so veljavne takoj, ko spremenimo parametre vendar se izgubijo, ko čitalec izklopimo iz napajanja.

Organizacija pomnilnika čitalca je prikazana v tabeli 4.

Ime	Naslov (šestnajstiško)
Serijska številka	00
Verzija firmwarea	01
RID	02
Baudna hitrost	03
Status čitalca	04
Rezervirano	05 – 06
Smer V/I pinov	07
Vrednost na V/I pinih	08

Tabela 4. Organizacija pomnilnika v čitalcu. Vrednosti na ostalih naslovih so rezervirane in jih ne moremo brati ali vanje pisati.

4.1.4. Primeri komunikacije s čitalcem

4.1.4.1. Branje TID kartic v dosegu, neodvisno od tipa kartice

Če želimo prebrati vse kartice, ki so v dosegu branja čitalca, moramo postaviti zastavico INV_F. To omogoča, da se značka začasno deaktivira, ko se prebere njen TID, in dá možnost drugim značkam, da odgovorijo na zahtevo in pri tem ni prekinitev. Ker protokol podpira branje različnih tipov značk, lahko v zahtevi povemo, naj se berejo značke poljubnega tipa. V tem primeru bo v odgovoru vsebovan tudi tip prebrane značke.

	zastavice	zahteva	tip značke	
<CR>	02	14	00	<CR>

Tabela 5. Zahteva za branje značk poljubnega tipa.

V zahtevi (tabela 5) je postavljena zastavica INV_F (02₍₁₆₎). Zahteva je 14₍₁₆₎, kar pomeni zahtevo SELECT_TAG (tabela 2). Tip značke je postavljen na Auto (00₍₁₆₎). Zahteva je oklenjena s kontrolnima znakoma CR.

	odgovor	tip značke	podatki (TID)	
<LF>	14	01	E007 0000 0164 5E37	<CR><LF>
<LF>	14	01	E007 0000 0154 6531	<CR><LF>
<LF>	14	01	E007 0000 0154 4132	<CR><LF>
<LF>	14	02	0100 0000 33B1 DF8E	<CR><LF>
<LF>	14	02	0100 0000 025D CAD2	<CR><LF>
<LF>	94	<CR><LF>		

Tabela 6. Odgovor čitalca v petih blokih na zahtevo v tabeli 5.

Vsi bloki v odgovoru (tabela 6), razen zadnjega, imajo v polju za odgovor $14_{(16)}$ kar je koda za SELECT_TAG Success (tabela 3). Zadnji blok ima odgovor $94_{(16)}$, kar pomeni SELECT_TAG Failure (tabela 3), kar pomeni, da v dosegu ni več nobene druge značke. Čitalec je uspel prebrati 3 značke tipa ISO15693 in dve znački tipa I-Code SL1, kar je razvidno iz polja tip značke. V polju podatki, se za vsako značko nahaja TID značke.

4.1.4.2. Nastavljanje RID čitalca

RID čitalca je shranjen na dveh mestih: v obstojnem in v neobstoječem pomnilniku čitalca. V tem primeru bomo nastavili RID čitalca v obstojnem pomnilniku. Nastavitev bo začela veljati ob naslednjem zagonu čitalca.

	zastavice	zahteva	začetni blok	število blokov	podatki	CRC	
<CR>	20	41	02	01	FF	C985	<CR>

Tabela 7. Zahteva za nastavitev RID čitalca na vrednost $FF_{(16)}$.

Zastavice imajo v zahtevi (tabela 7) postavljen bit CRC_F. CRC je pri vseh ukazih za pisanje obvezen. Zahteva $41_{(16)}$ je WRITE_MEM (tabela 2). Na naslov 02, kjer se po shemi organizacije pomnilnika nahaja RID, bomo zapisali en zlog podatkov z vsebino $FF_{(16)}$.

	odgovor	CRC	
<CR>	41	538D	<CR><LF>

Tabela 8. Odgovor čitalca na zahtevo v tabeli 7.

Čitalec pošlje odgovor 41₍₁₆₎, WRITE_MEM Success (tabela 3) – zapisovanje je uspelo.

4.1.4.3. Branje podatkov iz pomnilnika na znački

V zahtevi za branje podatkov (tabela 9) vključimo TID značke, ki ga poznamo že od prej, npr. ga pridobimo z izvedbo ukaza SELECT_TAG (tabela 2). V polju zastavic povemo, da bo zahteva vsebovala TID (postavimo zastavico TID_F). Zahteva je 24₍₁₆₎, kar pomeni ukaz READ_TAG (tabela 2). V polju tip značke povemo, da bomo brali z ISO 15693 značke. Prebrali bomo dva bloka podatkov iz naslova 0A₍₁₆₎.

	zastavice	zahteva	tip značke	TID	začetni blok	število blokov	
<CR>	40	24	1	E007 0000 0164 5E37	0A	02	<CR>

Tabela 9. Zahteva za branje podatkov iz značke s TID E007000001645E37.

Čitalec pošlje odgovor (tabela 10) 24₍₁₆₎, kar pomeni READ_TAG Success (tabela 3). Čitalec je vrnil dva bloka podatkov (2 x 4 zloge) z vsebino 01010000000002AA₍₁₆₎.

	odgovor	podatki	
<CR>	24	01010000 000002AA	<CR><LF>

Tabela 10. Odgovor čitalca na zahtevo v tabeli 9.

5. Aplikacija

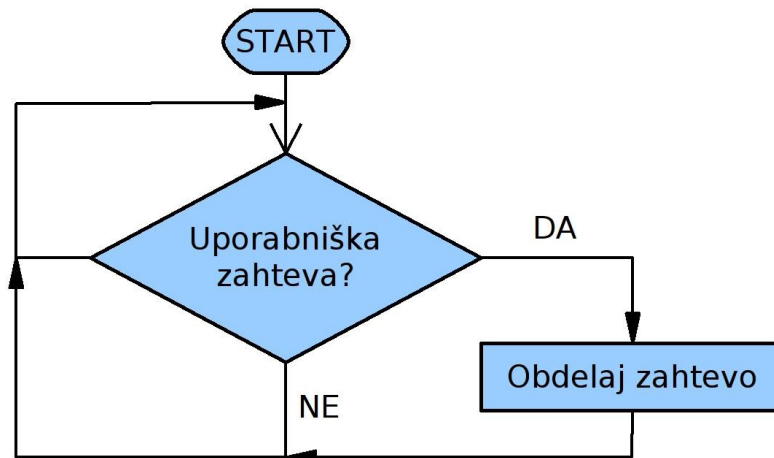
Aplikacija v celoti vodi delovanje sistema. Ko se aplikacija zažene se vzpostavi povezava s čitalcem in podatkovno bazo, uporabnik pa lahko aplikacijo uporablja preko uporabniškega vmesnika. Ker komunikacija med čitalcem in računalnikom poteka preko TCP protokola aplikacija ne potrebuje administratorskih pravic za delovanje. Za uporabo čitalca ne potrebujemo gonilnika in vsa logika, ki je potrebna za komunikacijo s čitalcem, je implementirana v sami aplikaciji.

Aplikacija se izvaja v treh nitih naenkrat. To pomeni, da se istočasno izvajajo tri zanke, vsaka s svojo nalogo. Zanke tečejo neodvisno ena od druge, vendar si stalno izmenjujejo podatke. Izmenjava podatkov je potrebna za pošiljanje zahtev in prejemanje odgovorov čitalca (slika 14).



- **Glavna zanka** (slika 15)

Skrbi za obdelavo zahtev, ki jih uporabnik pošlje programu preko grafičnega uporabniškega vmesnika. Prejema dogodke tipkovnice in miške in obdeluje zahteve, kot so kliki na gumbe, izbira elementa iz seznama, itd. Skrbi tudi za izrisovanje uporabniškega vmesnika na ekran. Za izvajanje glavne zanke skrbi ogrodje Qt, zato ni možno neposredno prilagajati njenega delovanja.



- **Zanka za dostop** (slika 16)

Stalno preverja stanje na treh antenah, ki so namenjene zaznavanju vstopov in izstopov iz laboratorija. Ko na eni izmed anten zazna kartico ugotovi ali gre za vstop ali za izstop iz laboratorija.

- **Zanka za komunikacijo s čitalcem** (slika 16)

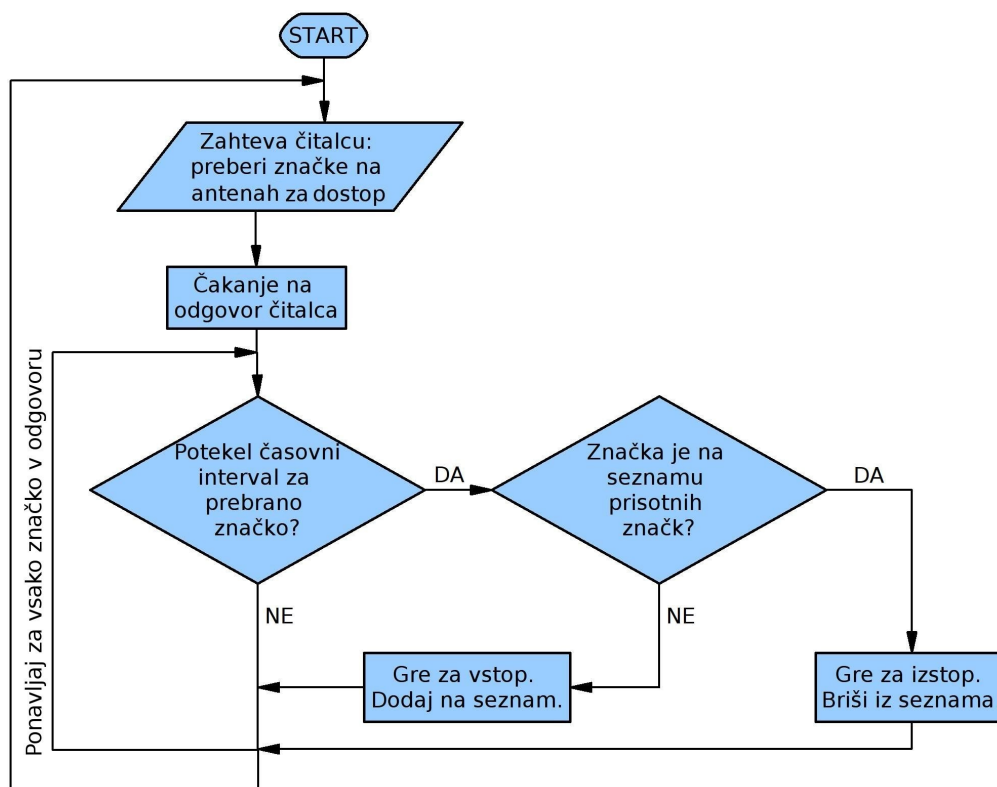
Stalno preverja, če čaka kakšna zahteva za čitalca in v tem primeru zahtevo pošlje čitalcu. Počaka na odgovor čitalca in vrne odgovor na zahtevo.

5.1. Zanka za dostop

Zanka skrbi, da aplikacija zazna, ko oseba z dostopno kartico vstopi ali izstopi iz laboratorija. Ves čas mora pregledovati, če se je v dosegu kakšne izmed treh anten za dostop pojavila značka. Naloga zanke je, da čitalcu odda zahtevo, naj preveri stanje na antenah za dostop. Ko odda zahtevo čaka na odgovor od čitalca. Če je odgovor, da v dosegu ni nobene značke, takoj pošlje novo enako zahtevo. Če pa odgovor vsebuje značke, ki so v dosegu, je za vsako značko treba ugotoviti ali gre za vstop ali izstop iz laboratorija. Ko je to ugotovljeno se ponovno pošlje zahteva za stanje na antenah za dostop. Ta zanka se ves čas izvaja.

Da lahko ugotovimo ali gre za vstop ali izstop, ko zaznamo značko na eni izmed anten,

moramo voditi evidenco značk (tj. oseb, ki imajo dostopno kartico), ki so trenutno v laboratoriju. Ko se v dosegu anten pojavi značka pregledamo seznam trenutno prisotnih značk; če je značka na tem seznamu, kar pomeni, da je trenutno v laboratoriju, potem gre za izstop iz laboratorija. Če pa prebrane značke ni na seznamu pomeni, da je oseba vstopila v laboratorij. Pri tem pa je treba paziti na dejstvo, da se zanka hitro vrti in da v času, ko oseba prestopi prag laboratorija lahko značko zaznamo večkrat, čeprav gre samo za en vstop ali izstop. Potrebno je uvesti nek časovni interval v katerem se bo prebrana značka ignorirala. Šele po preteku intervala prebrano značko obravnavamo kot nov vstop ali izstop. Delovanje zanke prikazuje diagram na sliki 16.



Slika 16. Diagram zanke za dostop.

5.2. Zanka za komunikacijo s čitalcem

Naloga zanke za komunikacijo s čitalcem je, da stalno preverja, če obstaja kakšna zahteva za čitalec, da to zahtevo čitalcu pošlje, počaka na njegov odgovor in rezultate odgovora pošlje

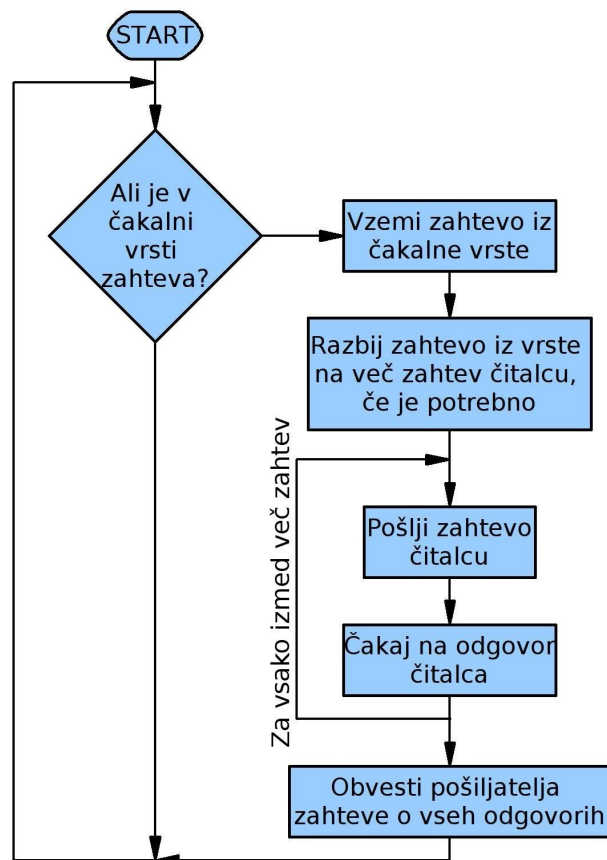
nazaj pošiljatelju zahteve. Zahteve za čitalec prihajajo iz glavne zanke, to so zahteve za branje značk z anteno, namenjeno izposoji in vračilu knjig ter opreme, in iz zanke za dostop, to so zahteve za branje značk s tremi antenami za dostop. Glede na to, da zahteve za čitalec prihajajo iz dveh niti, ki se izvajata istočasno (slika 14), hitro nastane situacija, kjer pride nova zahteva ali celo več novih zahtev, medtem ko se izvaja še prejšnja. V taki situaciji mora nit, ki oddaja novo zahtevo, čakati, da čitalec obdela staro zahtevo. Šele po tem čitalec v obdelavo vzame novo zahtevo, nit pa mora še naprej čakati, da se obdela zahteva, ki jo je sama poslala, preden dobi odgovor in nadaljuje z delom. Medtem ko nit čaka na odgovor čitalca ne more početi ničesar drugega. V primeru glavne zanke, katere naloga je tudi izrisovanje grafičnega vmesnika, to pomeni, da ko glavna zanka čaka na odgovor čitalca ne osvežuje grafičnega vmesnika. Z vidika uporabnika to pomeni, da je vmesnik neodziven.

Da se temu izognemo moramo uvesti dve stvari:

- Čakalno vrsto za zahteve – Vse zahteve iz vseh niti, se odlagajo v čakalno vrsto. Zanka za komunikacijo s čitalcem pobere prvo zahtevo, ki je v čakalni vrsti, jo pošlje čitalcu in počaka na odgovor. Ko se odgovor odpošlje iz vrste vzame naslednjo zahtevo, vse dokler vrsta ni prazna. Ko se vrsta izprazni zanka stalno preverja, če je v vrsti že kakšna nova zahteva. Določene zahteve bi radi obravnavali z višjo prioriteto kot ostale. To pomeni, da bi zahteva z višjo prioriteto od ostalih v čakalni vrsti, takoj pristala na prvem mestu, kljub temu, da so bile zahteve z nižjo prioriteto oddane prej. Zanka za dostop stalno pošilja zahteve za branje značk. V čakalni vrsti je tako lahko več zahtev te zanke, ki čakajo na obdelavo. Medtem pa lahko uporabnik zahteva seznam vseh knjig, ki so na voljo za izposajo. Če ima uporabnikova zahteva enako prioriteto kot zahteve zanke za dostop, bo moral uporabnik čakati, da se izvedejo vse zahteve, ki so že v čakalni vrsti. Uporabniški vmesnik sicer lahko ostane odziven, vseeno pa mora uporabnik čakati na seznam knjig. Temu se izognemo tako, da uporabniški zahtevi priredimo višjo prioriteto. Uporabnik bo takoj, ko odda zahtevo, dobil seznam knjig, ki so na voljo za izposajo, zahteve, ki so bile v vrsti že prej, pa se bodo naprej izvajale.
- Obveščanje o odgovorih čitalca – Če želimo, da je uporabniški vmesnik odziven, nit ne sme čakati na odgovor čitalca ampak mora naprej izvajati ostale naloge. Kljub temu pa

mora vedeti, ko je pripravljen odgovor na njeno zahtevo. To lahko rešimo z uporabo čakalnih listkov. Ko nit odda zahtevo v čakalno vrsto dobi čakalni listek (neko unikatno število), nato pa nadaljuje z izvajanjem. Ko bo čitalec obdelal zahtevo bo nit za komunikacijo s čitalcem oddala signal, da je pripravljen odgovor, skupaj s številko čakalnega listka. Nit, ki je oddala zahtevo, prepozna številko svojega listka in prebere odgovor.

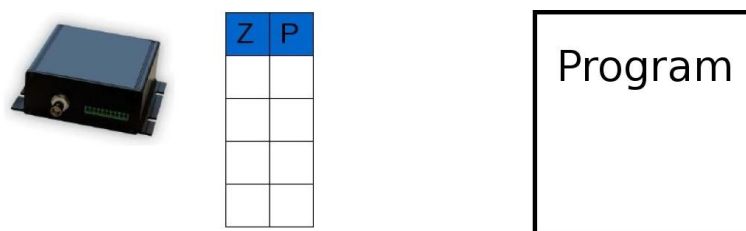
Zahteve, ki se vstavljajo v prioriteto vrsto niso enake ASCII zahtevam, ki so opisane v poglavju 4.1.1, v smislu enakih podatkov, ampak predstavljajo nekakšno ovojnico za ASCII zahteve. Vsebujejo namreč dodane podatke, npr. prioriteto zahteve in seznam anten, na katerih naj se zahteva izvede. Čitalec sam ne razume teh dodatnih podatkov – namenjeni so niti za komunikacijo s čitalcem. Naloga te niti je, da z upoštevanjem teh dodatnik podatkov sestavi več ASCII ukazov in jih pošlje čitalcu (npr. pri branju več anten, glej pogl. 5.3). En ukaz v prioriteto vrsti se torej lahko prevede v več ukazov čitalcu. Slika 17 prikazuje diagram poteka zanke za komunikacijo s čitalcem.



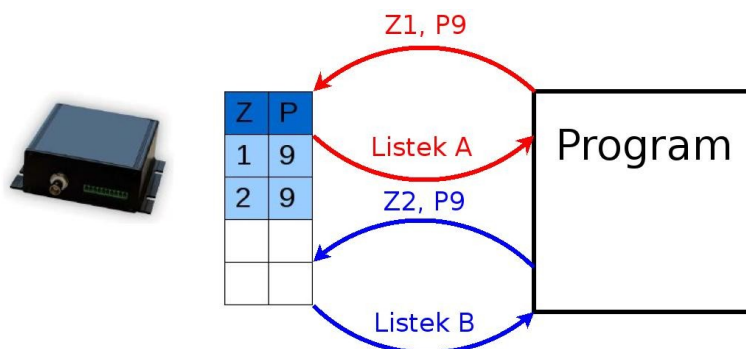
5.2.1. Delovanje prioritetne vrste

V nadaljevanju je prikazan primer delovanja prioritetne vrste. Zanka za komunikacijo s čitalcem je zadolžena za upravljanje s čitalcem, ki je na slikah prikazan kot čitalec QuasarMR1, in z vstavljanjem in jemanjem zahtev iz prioritetne vrste, ki je predstavljena v obliki tabele. Tabela vsebuje stolpec Z za zahtevo in stolpec P za prioriteto (manjša številka pomeni večjo prioriteto). Čitalec in tabela na sliki predstavljata zanko za komunikacijo s čitalcem. Pravokotnik program pa predstavlja obe ostali zanki, glavno zanko in zanko za dostop.

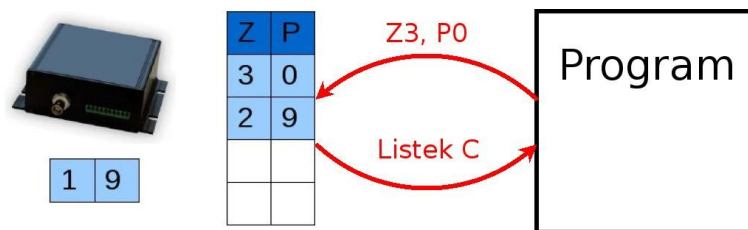
Na sliki 18 je čakalna vrsta prazna. Čitalec nima nobenih zahtev in zato čaka. Zanka za komunikacijo s čitalcem stalno preverja stanje čakalne vrste, da v primeru, ko se pojavi nova zahteva takoj lahko ukrepa.



Pojavita se dve novi zahtevi (slika 19), Z1 in Z2. Obe zahtevi imata enako prioriteto, P9. Nit za komunikacijo s čitalcem ju razporedi v čakalno vrsto – ker imata obe isto prioriteto ima prednost tista, ki je prišla prej, v tem primeru Z1. Takoj ko se zahteva uvrsti v čakalno vrsto, dobi program kot odgovor svoj unikatni čakalni listek. Po prejemu lista lahko nadaljuje s svojim delom, o odgovoru pa bo obveščen, ko bo le-ta pripravljen.

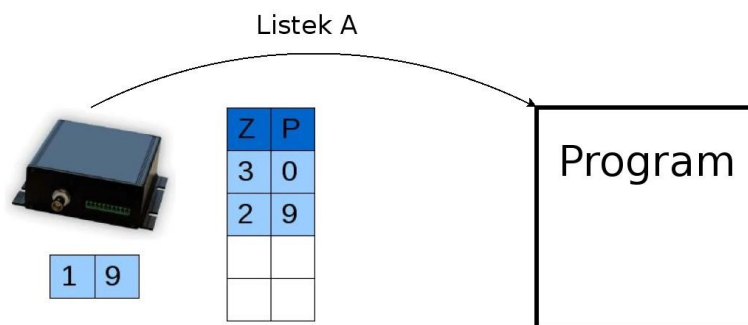


Takoj ko se je v čakalni vrsti pojavila prva zahteva, Z1, je nit poskrbela, da je bila zahteva vzeta iz vrste in poslana čitalcu. Ko se zahteva izvaja na čitalcu, je bila v čakalni vrsti samo še zahteva Z2, dokler od programa ni prišla nova zahteva Z3 (slika 20). Z3 ima prioriteto P0, kar je višja prioriteta kot P9. Zahteva je zato v čakalni vrsti postavljena na prvo mesto, kljub temu, da je zahteva Z2 prispela pred zahtevo Z3 in je bila že prej v čakalni vrsti. Medtem pa se zahteva Z1 še vedno izvaja na čitalcu.



Slika 20. Prihod nove zahteve z višjo prioriteto.

Ko se izvajanje zahteve Z1 na čitalcu konča, nit za komunikacijo s čitalcem prejme odgovor na Z1. Naloga niti je, da obvesti program, da je pripravljen odgovor za čakalni listek A, zato pošlje signal programu, ki sedaj lahko prebere odgovor na svojo zahtevo (slika 21).



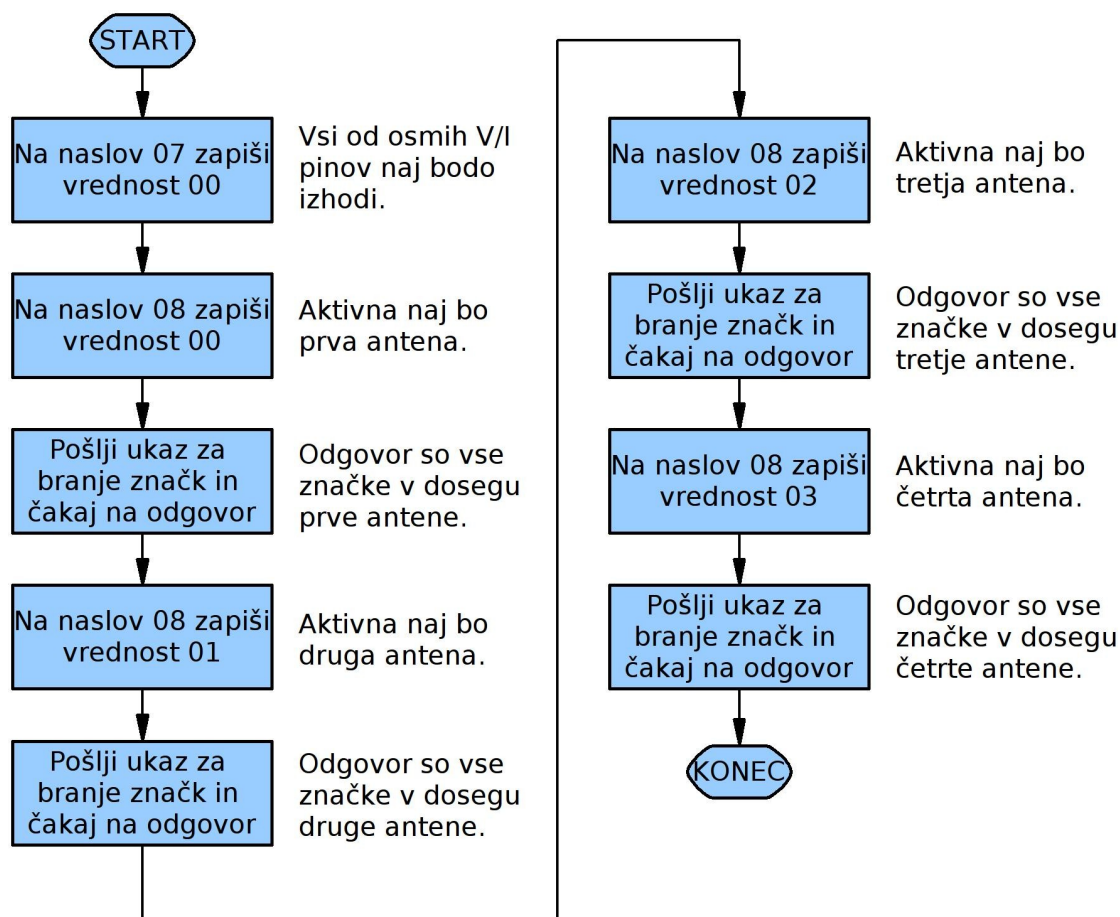
Slika 21. Zahteva Z1 se je izvedla, program je obveščen o odgovoru.

Ko je program obveščen o odgovoru na zahtevo Z1, se iz čakalne vrste vzame naslednja zahteva Z3. Po uspešni obdelavi zahteve je program ponovno obveščen o pripravljenem odgovoru, tokrat za čakalni listek C. Nazadnje se iz čakalne vrste vzame še zahteva Z2 in se

kot ostale izvede, program pa je obveščen o odgovoru. Po izvedeni zahtevi Z2 je čakalna vrsta ponovno prazna, niz za komunikacijo s čitalcem pa čaka na novo zahtevo.

5.3. Branje z več antenami

Branje z več antenami je omogočeno z uporabo multiplekserja. Čitalec sam ne zna upravljati z multiplekserjem. Čitalec se niti ne zaveda, da je povezan z multiplekserjem ali katerokoli drugo napravo, povezano preko splošno namenskih V/I pinov. Naloga aplikacije, ki upravlja s čitalcem, je, da preko V/I pinov na čitalcu krmili delovanje multiplekserja. V/I pine na čitalcu krmilimo s pisanjem in branjem v neobstojni pomnilnik čitalca na naslovih $07_{(16)}$ in $08_{(16)}$ (tabela 4). Ker čitalec ne zna delati z multiplekserjem mu ne moremo poslati ukaza kot je npr. „beri značke z vseh anten“. Za izvedbo take operacije moramo čitalcu poslati zaporedje ukazov.

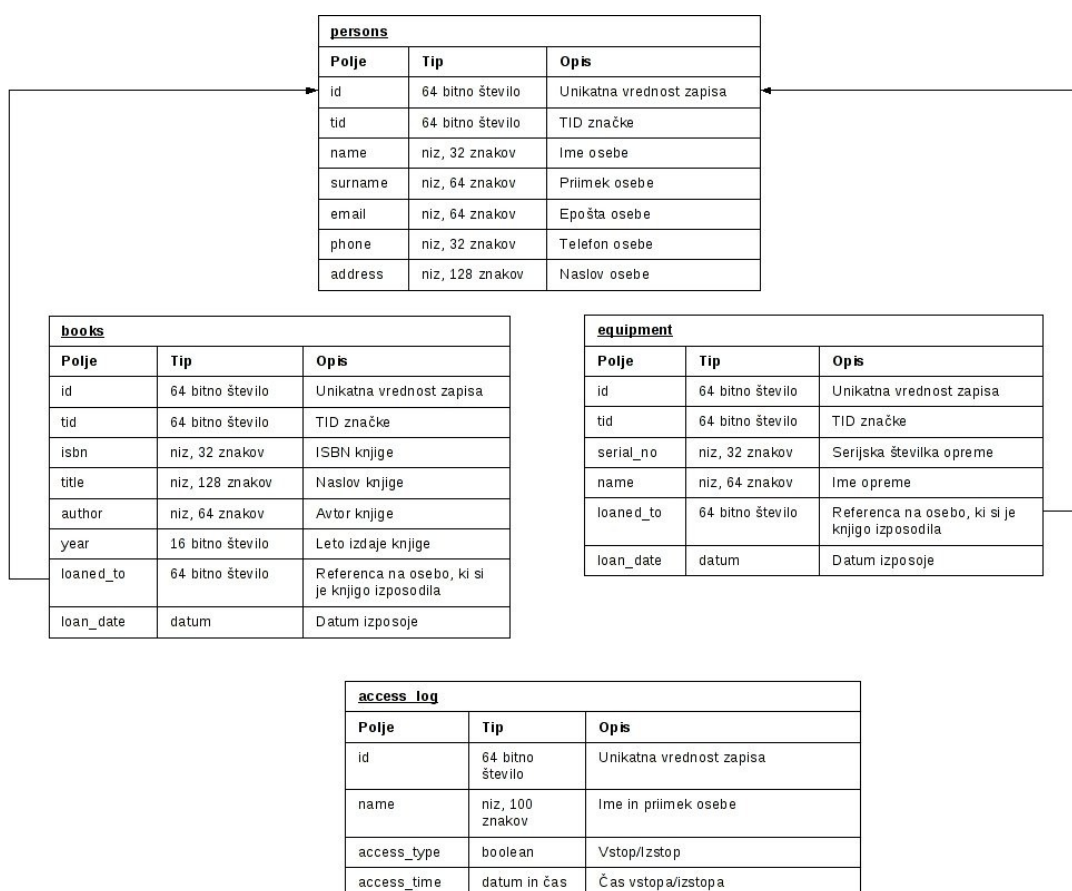


Slika 22. Branje značk z vseh štirih anten na multiplekserju.

Diagram na sliki 22 prikazuje izvedbo ukaza „beri značke z vseh anten“, v primeru, da imamo čitalec povezan z multiplekserjem za štiri antene. V tem primeru dobimo od čitalca štiri odgovore, za vsako anteno svojega. Pri tem pa moramo upoštevati, da v primeru, da se obseg anten prekriva, lahko več anten prebere isto značko. Ista značka tako lahko nastopi v več odgovorih čitalca in naloga aplikacije, ki upravlja s čitalcem, je, da to upošteva in ustrezno rešuje.

5.4. Model podatkovne baze

Sistem v podatkovni bazi hrani podatke o osebah, knjigah, opremi in o vstopih in izstopih iz laboratorija. Podatki se v podatkovni bazi nahajajo v štirih tabelah: persons, books, equipment in access_log, kjer tabela persons hrani podatke o osebah, tabela books podatke o knjigah, tabela equipment podatke o opremi in access_log podatke o vstopih in izstopih iz laboratorija.



Vse tabele vsebujejo polje id. Polje je 64 bitno število in za vsak naslednji vstavljeni zapis v

tabeli, polje samodejno dobi vrednost, ki je za 1 večje od vrednosti polja pri prejšnjem zapisu. Tako zagotovimo, da ima vsak zapis v posamezni tabeli v tem polju unikatno vrednost, po katerem ga lahko prepoznamo. Ker se osebe, knjige in oprema navezujejo na značke RFID imajo tabele persons, books in equipment polje tid. Polje je 64 bitno število in hrani 64 bitni TID oz. UID značke, ki pripada osebi, knjigi ali opremi. Oseba si lahko izposodi knjige ali opremo, zato tabeli books in equipment vsebujeta še dve polji: loaned_to in loan_date. Polje loan_to je referenca na polje id v tabeli persons. Če polje vsebuje vrednost, pomeni da si je knjigo ali opremo izposodila oseba, na katero kaže referenca. V tem primeru polje loan_date vsebuje datum izposoje. Če knjiga ali oprema ni izposojena sta polju loaned_to in loan_date prazni. Poleg že omenjenih polj pa vsaka tabela vsebuje še polja, ki opisujejo predmet, o katerem tabela hrani podatke. Slika 23 prikazuje model podatkovne baze.

5.5. Uporabniški vmesnik

Uporabniški vmesnik sestavljata dve okni:

- Glavno okno (RFID Lab) omogoča uporabo vseh funkcij, ki jih aplikacija ponuja;
- Dostopno okno (Persons) pa je odprto ves čas in prikazuje trenutno prisotne osebe v laboratoriju in čas, ko so osebe vstopile v laboratorij.

Glavno okno na levi strani pod napisom RFID Lab vsebuje glavni meni, ki prikazuje ikone štirih glavnih opravil, ki jih aplikacija omogoča uporabniku. Od zgoraj navzdol si sledijo v naslednjem vrstnem redu:

- Izposoja / Vračilo (Loan / Return)
- Urejanje značk (Edit tags)
- Podatkovna baza (Database)
- Nastavitve (Preferences)

Ob kliku na eno izmed ikon se v desnem delu glavnega okna prikaže vsebina, povezana z

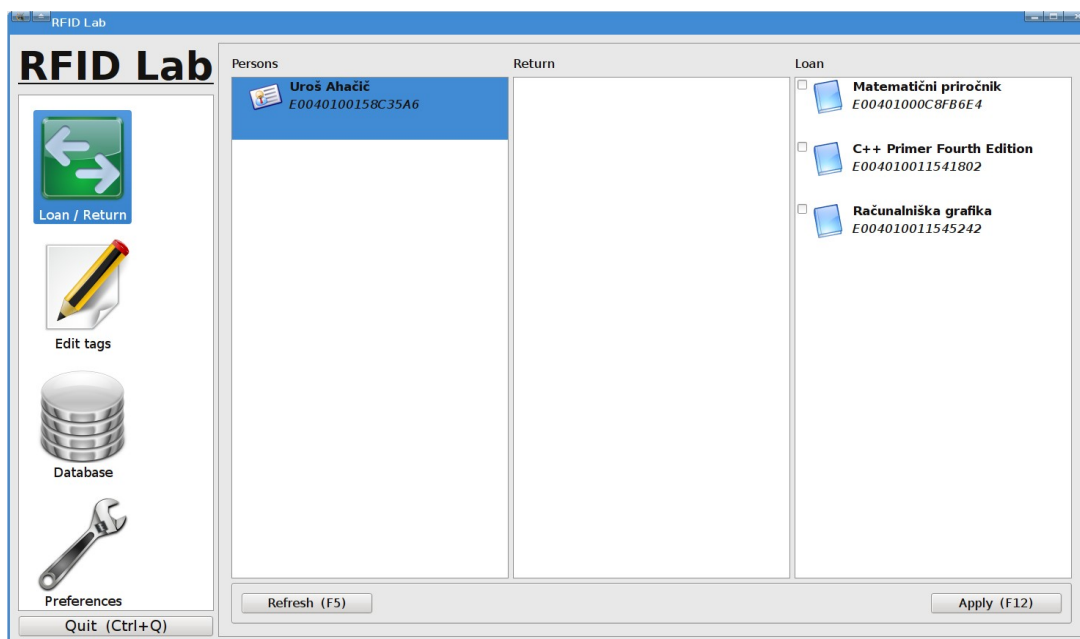
izbranim opravilom. Na sliki 24 je izbrano opravilo Urejanje značk.



Slika 24. Aplikacija, z vsemi odprtimi okni.

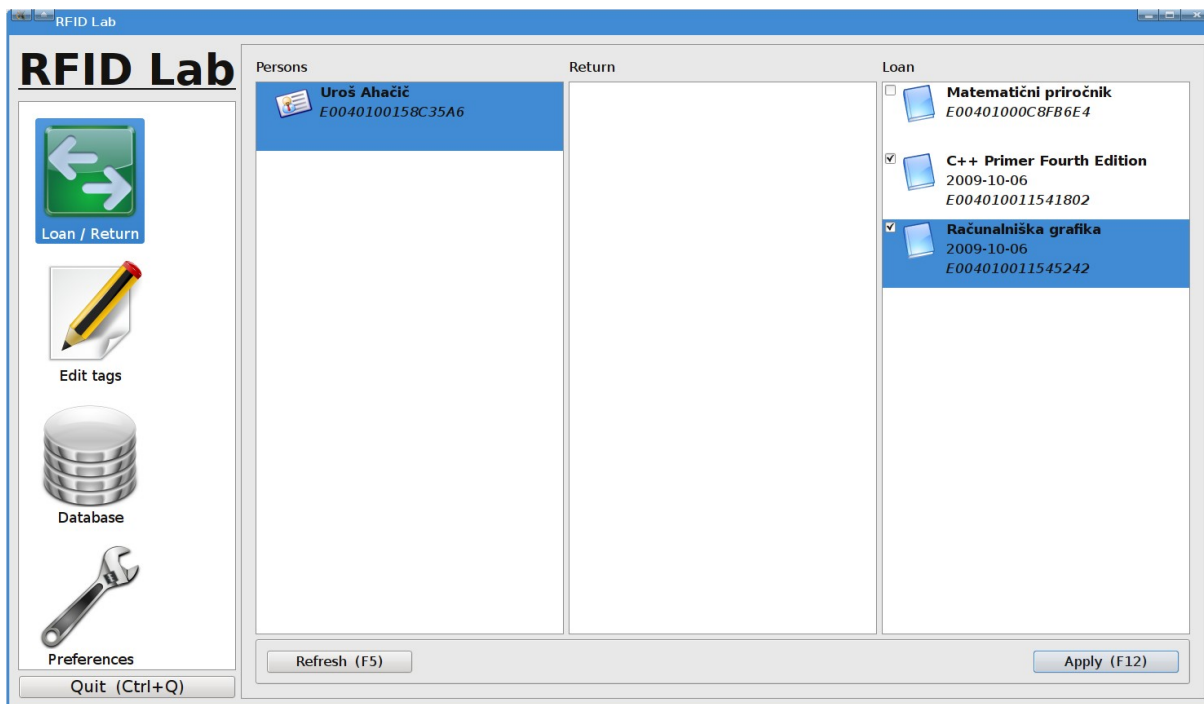
5.5.1. Izposoja in vračilo knjig ali opreme

Do opravila pridemo s klikom na ikono Loan / Return v meniju v glavnem oknu. Omogoča nam izposajo in vračanje knjig ali opreme. Preden knjige ali opremo odnesemo iz laboratorija nam aplikacija omogoča, da z uporabo kartice za dostop zabeležimo, kdo si je knjige ali opremo izposodil.

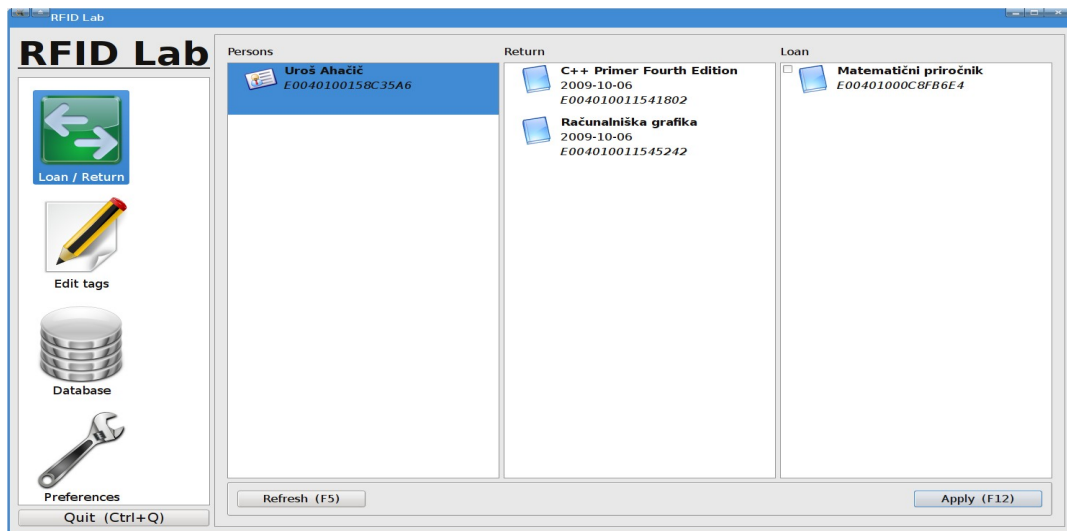


Slika 25. Čitalec je prebral kartico za dostop in tri knjige, ki so na voljo za izposajo.

Z bližnjico F5 na tipkovnici ali s klikom na gumb Refresh sprožimo, da čitalec prebere značke, ki so v njegovem dosegju branja. Slika 25 prikazuje primer, ko so v dosegju čitalca kartica za dostop, oseba Uroš Ahačič, in tri knjige, ki so na voljo za izposojlo. Ob vsaki sličici knjige je na levi strani kvadrateg, ki ga lahko označimo. S tem povemo, da si izbrana oseba želeli izposoditi izbrane knjige.



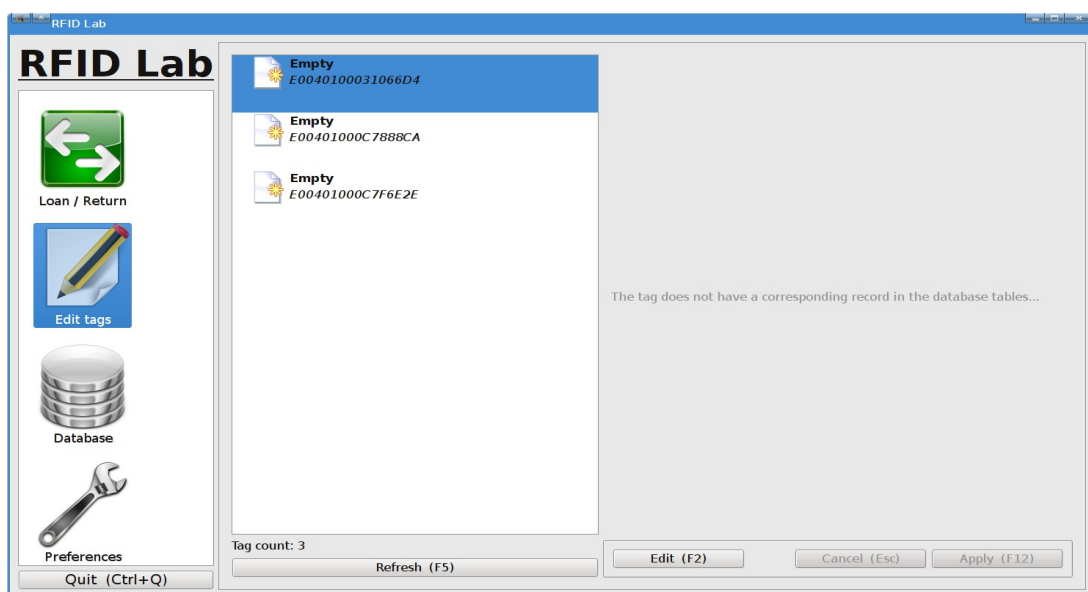
Na sliki 26 si bo oseba Uroš Ahačič izposodila knjigi C++ Primer Fourth Edition in Računalniška grafika. Izposojlo izbranih knjig potrdimo s klikom na gumb Apply ali bližnjico F12 na tipkovnici. S potrditvijo se v bazo podatkov zapiše, da si je oseba Uroš Ahačič izposodila obe knjigi in datum izposoje. Slika 27 prikazuje stanje po potrditvi. Obe knjigi, ki sta bili izposojeni, nista več na voljo za izposojlo in se zato pojavita v razdelku Return, kar pomeni, da ju oseba lahko vrne.



Slika 27. Oseba si je izposodila dve knjigi.

5.5.2. Urejanje značk

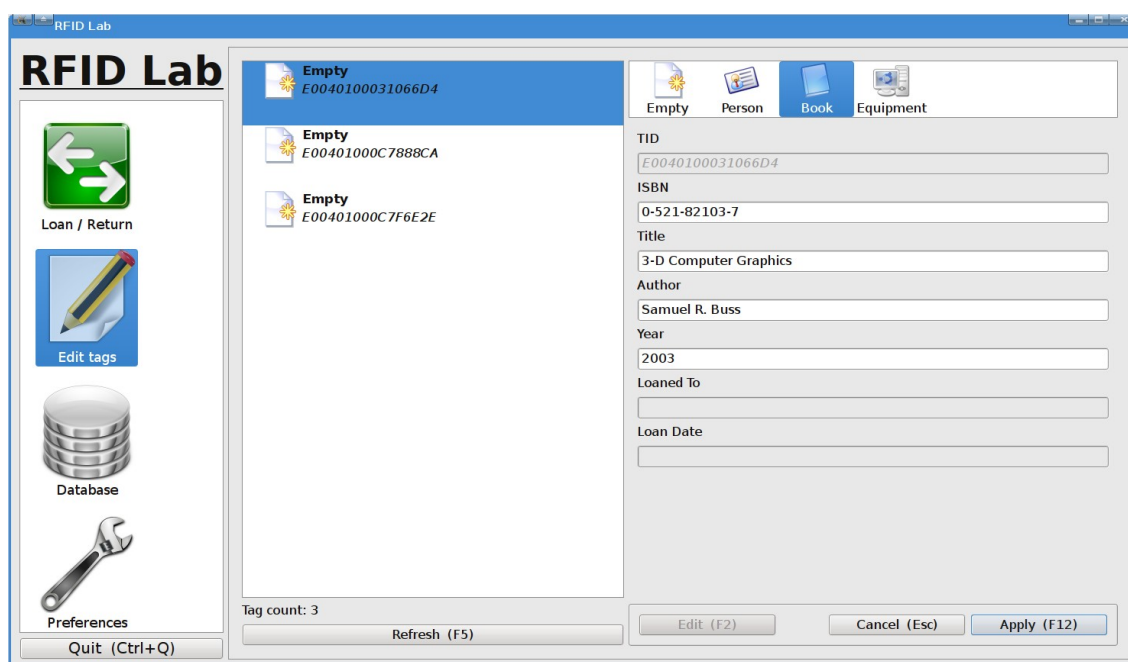
Do opravila pridemo s klikom na ikono Edit tags v meniju v glavnem oknu. Urejanje značk nam omogoča, da določeni znački priredimo poljubno knjigo, opremo ali osebo. Imamo možnost, da knjigo, opremo ali osebo priredimo prazni znački ali pa znački, ki že ima povezavo, v primeru, ko želimo posodobiti vsebino, uporabiti značko za drug namen ali jo zbrisati. V podatkovni bazi se ustvari zapis, ki povezuje TID značke s podatki o knjigi, opremi ali osebi.



Slika 28. Čitalec je prebral tri prazne značke, ki jih lahko urejamo.

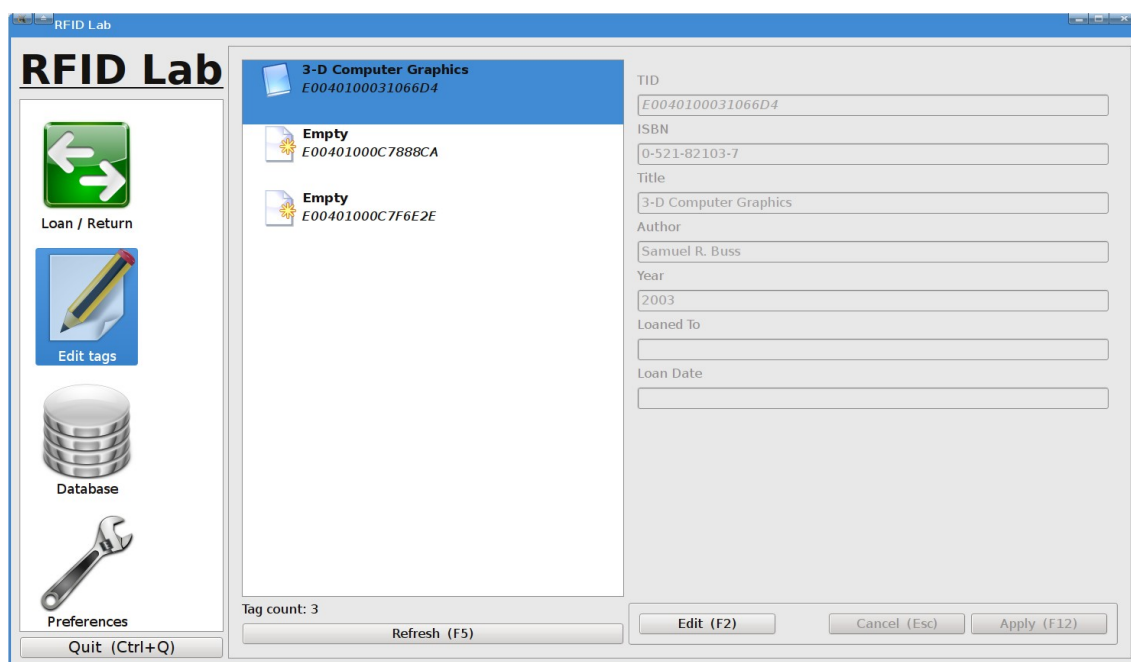
V meniju glavnega okna izberemo drugo ikono Edit tags in s klikom na gumb Refresh ali z bližnjico F5 sprožimo branje značk v doseg čitalca. Na sliki 28 vidimo, da je čitalec prebral tri značke, ki še nimajo povezave s knjigo, opremo ali osebo, torej so prazne. Edina informacija, ki jo imamo o znački na tej točki je TID značke.

S klikom na gumb Edit ali z bližnjico F2 na tipkovnici sprožimo urejanje izbrane značke. Znački lahko priredimo vlogo kartice za dostop, nalepke za knjigo ali nalepke za opremo. To lahko določimo s klikom na eno izmed ikon v zgornjem desnem področju okna. Na voljo imamo: prazna značka, oseba, knjiga in oprema v tem vrstnem redu od leve proti desni. V primeru na sliki 29 je izbrana prva izmed treh praznih značk v doseg čitalca. S klikom na ikono Book smo izbrali, da bo značka uporabljena kot nalepka za knjigo. V prazna polja vpišemo vse potrebne podatke o knjigi in nato s klikom na gumb Apply ali z bližnjico F12 na tipkovnici potrdimo vnos.



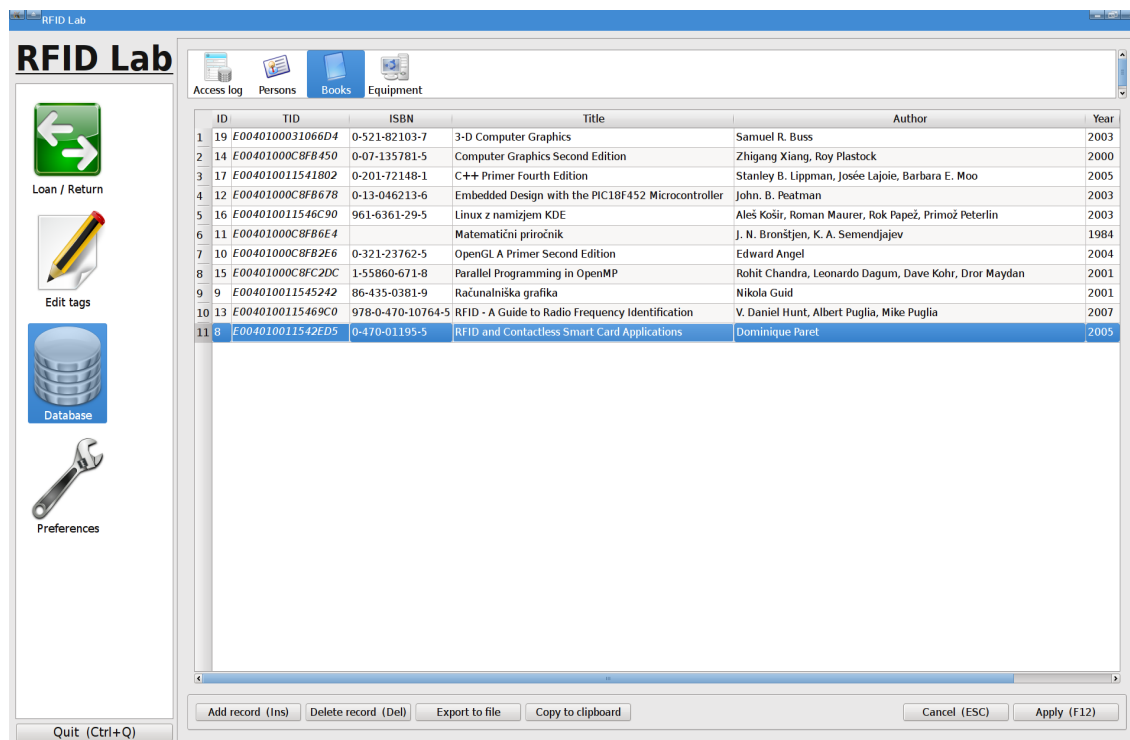
Slika 29. Za izbrano značko so vneseni podatki, ki jih bo značka predstavljala.

Slika 30 prikazuje stanje po potrditvi. Prva značka ni več prazna ampak pripada knjigi 3-D Computer Graphics. Ostali dve znački sta še vedno prazni, lahko pa ju uredimo na enak način kot prvo.



5.5.3. Podatkovna baza

S klikom na ikono Database v meniju v glavnem oknu lahko pregledujemo in urejamo tabele v podatkovni bazi. Po kliku na ikono Database imamo v zgornjem delu okna možnost izbire



id	tid	name	su rname
45	E0040100158C35A6	Uroš	Ahačič
46	E0040100158C2714	Mira	Trebar
47	E0040100158C4BE6	Patricio	Bulić
48	E00401000310B9F2	Veselko	Guštin

Slika 32. Del vsebine izvožene datoteke.

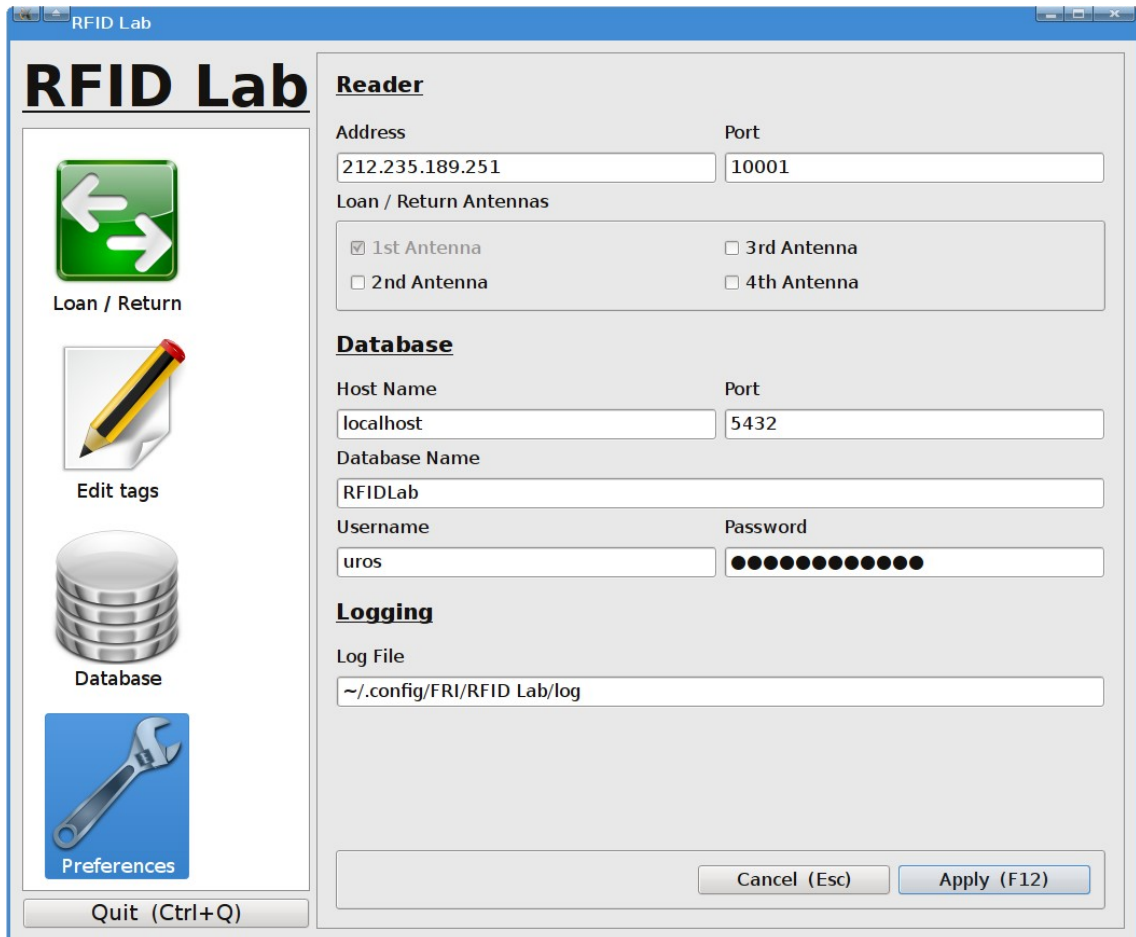
tabele, ki jo želimo pregledovati ali urejati. Na voljo imamo tabelo dostopov, tabelo oseb, tabelo knjig in tabelo opreme. V primeru na sliki 31 je izbrana tabela knjig. S klikom na naslov stolpca lahko tabelo sortiramo po tistem stolpcu. V spodnjem delu okna so gumbi za upravljanje s tabelo. V tabelo lahko vstavimo nov zapis, popravimo starega, tako da se enostavno postavimo na zeleno polje in vpišemo novo vrednost ali brišemo že obstoječi zapis. Vsakršno urejanje tabele moramo potrditi s pritiskom na gumb Apply ali s pritiskom na bližnjico F12 na tipkovnici. Na voljo imamo tudi dve posebni operaciji. Prva je izvoz vsebine izbrane tabele v tekstovno datoteko ki jo lahko sprožimo s klikom na gumb Export to file. Slika 32 prikazuje del vsebine take tekstovne datoteke. Druga operacija pa je kopiranje vsebine tabele v odložišče. Vsebina polj se kopirajo ločena z ASCII znakom TAB. Kopirano vsebino tako enostavno prilepimo v katerikoli program za preglednice, kot sta npr. OpenOffice.org Calc ali Microsoft Excel. Programi znajo vsebino avtomatsko pravilno razporediti v stolpce v tabeli.

5.5.4. Nastavitve

Do nastavitve pridemo s klikom na ikono Preferences v meniju v glavnem oknu. V nastavitvah (slika 33) lahko urejamo nastavitve v zvezi z delovanjem programa. Nastavitve so razdeljene v tri skupine:

- **Nastavitve čitalca**
Lahko nastavimo IP naslov in port čitalca. Nastavimo lahko tudi kako bomo uporabili štiri antene, ki so povezane na multiplekser. Označimo antene, ki jih želimo uporabiti za izposajo in vračanje knjig in opreme ter za urejanje značk. Vse ostale antene, torej tiste, ki niso označene, bodo uporabljene za beleženje prisotnosti oseb v laboratoriju. V primeru na sliki 33 bo prva antena uporabljena za izposajo in vračanje, ostale tri pa bodo uporabljene za beleženje prisotnosti.
- **Nastavitve baze**
Nastavimo lahko IP naslov in port strežnika in podamo vse potrebne podatke za povezavo, to so ime podatkovne baze, uporabniško ime in geslo za povezavo.
- **Beleženje**

Tukaj lahko določimo ime datoteke, kamor se bodo beležili dogodki v programu, kot so napake, opozorila in razna druga sporočila z informacijami, ki lahko pomagajo pri odkrivanju napak v programu, ko pride do nepredvidenih dogodkov.



Slika 33. Okno z nastavitvami aplikacije.

5.6. Namestitev aplikacije

Za razliko od operacijskega sistema Windows, kjer za namestitev aplikacije dobimo namestitveni program v obliki izvršljive datoteke, je v operacijskem sistemu Linux praksa, da program dobimo v obliki izvorne kode, ki jo moramo prevesti, da lahko aplikacijo uporabljamo.

Izvorna koda aplikacije RFID Lab se nahaja v datoteki rfid-lab.tar.gz. Gre za stisnjeno datoteko. Ko datoteko razširimo se ustvari imenik (ang. directory) z imenom RFID Lab, ki

vsebuje:

- imenik `release`, kjer se bodo pri izvajanju ustvarile t. i. objektne datoteke;
- imenik `src`, kjer se nahaja vsa izvorna koda aplikacije;
- imenik `resources`, kjer se nahajajo viri aplikacije, tj. razne slike in ikone;
- datoteke `Makefile`, `Makefile.release` in `RFID Lib.pro`, ki vsebujejo navodila za program `make`, da zna prevesti aplikacijo.

V imeniku `RFID Lab` zaženemo ukaz `make`, brez parametrov, ki samodejno prebere navodila za prevajanje aplikacije in jo prevede. Kot rezultat dobimo izvršljivo datoteko z imenom `RFID Lab`, ki jo lahko zaženemo.

6. Zaključek

V sklopu diplomske naloge je bila razvita aplikacija za sistem RFID, ki omogoča sledenje knjigam in opremi v laboratoriju. Omogoča hranjenje podatkov o knjigah in opremi v podatkovni bazi in njihovo izposojanje s strani oseb v laboratoriju. Vsa funkcionalnost je dostopna preko preprostega uporabniškega vmesnika. Zaradi tehničnih omejitev sistem ne omogoča dejanskega beleženja dostopov oseb do laboratorija pač pa vsebuje simulacijo takega sistema.

Sistem bi lahko v nadaljevanju razširili na:

- Beleženje vstopov in izstopov

Z uporabo večjih anten ali z uporabo večjega števila anten ob izhodu iz laboratorija, bi lahko izvedli beleženje dostopov, ki bi delovalo v realnih pogojih. Z uporabo še enega čitalca, ki bi bil namenjen samo za preverjanje značk ob izhodu, bi še dodatno povečali zanesljivost takega sistema. Omogočili bi lahko tudi samodejno izposojanje knjig in opreme. Osebi v tem primeru ne bi bilo potrebno položiti knjige ali opreme v območje antene, ki je namenjena za izposojanje, ampak bi antene ob izhodu zaznale, katera oseba je odnesla knjigo ali opremo iz laboratorija in bi se izposoja izvedla samodejno.

- Delovanje na več operacijskih sistemih

Ogrodje Qt omogoča, da aplikacijo napišemo enkrat, prevedemo pa, brez da bi morali kodo popravljati, na več različnih operacijskih sistemov. Zaradi težav, ki sem jih imel pri uporabi knjižnic, ki jih Qt ponuja za mrežno komunikacijo, je del programa, ki izvaja mrežno komunikacijo s čitalcem napisan z uporabo knjižnic, ki jih ponuja jedro operacijskega sistema Linux. Aplikacijo je tako možno prevesti le na operacijskem sistemu Linux. Z razrešitvijo problemov s knjižnicami Qt, bi aplikacijo enostavno lahko prevedli tudi za druge operacijske sisteme na trgu.

- Podpora različnim podatkovnim bazam

Ogrodje Qt omogoča tudi delo z različnimi podatkovnimi bazami, ne samo s PostgreSQL, vendar moramo v kodi v določenih primerih vsako bazo obravnavati na

drugačen način. Aplikacijo bi lahko nadgradili do te mere, da bi znala na potrebnih koncih v programu posebej obravnavati različne baze in tako zagotovili podporo aplikacije različnim podatkovnim bazam.

- Podpora več jezikom v grafičnem vmesniku
Ogrodje Qt omogoča podporo in širok nabor orodij, za prevajanje aplikacije v druge jezike. S pripravo ustreznih datotek s prevodi, bi lahko aplikacija podpirala več jezikov, uporabnik pa bi lahko poljubno izbiral med njimi.

7. Literatura

- [1] D. E. Brown, *RFID Implementation*, New York: McGraw-Hill, 2007, pogl. 1
- [2] S. Garfinkel, B. Rosenberg, *RFID Applications, Security and Privacy*, Upper Saddle River, NJ: Addison-Wesley, 2005, pogl. 1
- [3] (2009) Linux. Dostopno na:
<http://www.comptechdoc.org/os/linux/articles/ardistro.html>
- [4] (2009) Linux. Dostopno na:
<http://www.linux.org/info/index.html>
- [5] (2009) PostgreSQL. Dostopno na:
<http://www.postgresql.org/about>
- [6] (2009) Qt. Dostopno na:
<http://qt.nokia.com/products>
- [7] (2009) Qt (toolkit). Dostopno na:
[http://en.wikipedia.org/wiki/Qt_\(toolkit\)](http://en.wikipedia.org/wiki/Qt_(toolkit))