



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Název:** Systém pro sledování výpůjček pomocí RFID  
**Student:** Peter Guľa  
**Vedoucí:** Ing. Jiří Budek  
**Studijní program:** Informatika  
**Studijní obor:** Poítačové inženýrství (bakalářský)  
**Katedra:** Katedra číslicového návrhu  
**Platnost zadání:** do konce letního semestru 2014/15

### Pokyny pro vypracování

Prostudujte standard ISO/IEC 14443 pro komunikaci s bezkontaktními čipovými kartami s vazbou na blízko. Prostudujte knihovnu libnfc pro komunikaci s čipovými kartami a NFC (Near Field Communication) zařízeními. Navrhněte prototyp systému pro evidenci přítomnosti sledovaných předmetů. Systém bude umět detekovat vyjmutí předmetu z předepsaného místa a vyzve uživatele k identifikaci kartou. Následně zaznamená výpůjčku předmetu a obdobným způsobem i jeho vrácení. Systém realizujte formou prototypu v omezeném rozsahu (jednotky sledovaných předmetů). Předpokládejte uplatnění jako systém pro půjčování hardwarových přípravků studentům.

### Seznam odborné literatury

Dodá vedoucí práce.

L.S.

doc.Ing. Hana Kubátová, CSc.  
vedoucí katedry

prof.Ing. Pavel Tvrdík, CSc.  
děkan

V Praze dne 13. března 2014



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA POČÍTAČOVÝCH SYSTÉMŮ



Bakalářska práce

## **System pro sledování výpůjček pomocí RFID**

*Peter Guľa*

Vedúci práce: Jiří Buček

27. júna 2014



---

## Pod'akovanie

Rád by som touto cestou podakoval pánovi Ing. Jirímu Bučkovi za odborné vedenie, cenné rady a venovaný čas.



---

## Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 14. mája 2015

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2015 Peter Guľa. Všetky práva vyhradené.

*Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. K jej využitiu, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.*

### **Odkaz na túto prácu**

Guľa, Peter. *Systém pro sledování výpůjček pomocí RFID*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.



---

# Abstract

This bachelor thesis deals with an analysis of RFID NFC chips and its use for borrowing items in scope of development boards within the faculty. The theoretical part deals in evaluating the current state of borrowing and analysing chosen technologies. In the practical part I am occupied with the proposal of a prototype and its implementation. The result of the thesis is a prototype system for borrowing items capable of detecting removal and addition.

**Keywords** RFID, ISO/IEC 14443, libnfc, prototype, borrowing, Raspbian

---

# Abstrakt

Táto bakalárska práca sa zaoberá analýzou možností RFID NFC čipov a ich využitie pre požičiavanie predmetov, s uplatnením na hardvérové prípravky vrámci fakulty. Teoretická časť sa zaoberá hodnotením aktuálneho stavu s požičiavaním a analýzou vybraných technológií. V praktickej časti sa zaoberám návrhom prototypu a jeho implementáciou. Výsledkom práce je prototyp systému pre požičiavanie predmetov s detekciou ich odobrania a pridania.

**Kľúčová slova** RFID, ISO/IEC 14443, libnfc, prototyp, požičiavanie, Raspbian



---

# Obsah

Úvod	1
<b>1 Cieľ práce</b>	<b>3</b>
<b>2 Popis problému</b>	<b>5</b>
2.1 Súčasný stav riešenia problematiky . . . . .	5
2.2 Požiadavky na implementovaný systém . . . . .	6
2.3 Spracovanie existujúcich implementácií . . . . .	7
2.4 Popis štruktúry BP . . . . .	7
<b>3 Analýza</b>	<b>9</b>
3.1 Norma ISO/IEC 14443 [9] . . . . .	9
3.2 Libnfc . . . . .	15
3.3 Analýza možných riešení bez využitia NFC . . . . .	16
3.4 Analýza možných riešení s využitím NFC . . . . .	17
3.5 Riadiaca platforma . . . . .	20
3.6 Vstupné a výstupné periférie . . . . .	21
3.7 Jednodoskový počítač . . . . .	22
3.8 Periférie pre Raspberry Pi . . . . .	24
3.9 Programovací jazyk . . . . .	26
3.10 Operačný systém . . . . .	26
3.11 Databázový stroj . . . . .	26
3.12 SQLite . . . . .	28
3.13 Grafická knižnica QT . . . . .	32
<b>4 Návrh</b>	<b>33</b>
4.1 Návrh počítača . . . . .	33
4.2 Návrh zapojenia hardvéru . . . . .	33
4.3 Databázový návrh . . . . .	34
4.4 Procedurálny návrh . . . . .	35

4.5 Grafický návrh . . . . .	37
<b>5 Realizácia a testovanie</b>	<b>39</b>
5.1 Priebežné testovanie . . . . .	39
5.2 Merania . . . . .	40
5.3 Raspberry Pi . . . . .	40
5.4 Testovanie . . . . .	41
<b>6 Výhľad do budúcnosti</b>	<b>43</b>
<b>Záver</b>	<b>45</b>
<b>Literatúra</b>	<b>47</b>
<b>A Zoznam použitých skratiek</b>	<b>53</b>
<b>B Diagramy</b>	<b>55</b>
<b>C Obsah priloženého CD</b>	<b>57</b>

---

## Zoznam obrázkov

3.1	Komunikácia . . . . .	10
3.2	Schéma zapojenia s využitím USB hub . . . . .	18
3.3	Schéma zapojenia s využitím multiplexoru . . . . .	18
3.4	Schéma zapojenia s oddelenou anténou . . . . .	19
3.5	Znázornenie volania funkcií pri použití funkcie exec . . . . .	30
3.6	Znázornenie možných poradí volania funkcií . . . . .	31
3.7	Znázornenie volania funkcií pri použití funkcie prepare . . . . .	32
4.1	Schéma systému . . . . .	34
4.2	Databázový model . . . . .	35
4.3	Legenda k databázovému modelu . . . . .	35
4.4	Grafický návrh . . . . .	38
B.1	Algoritmus vrátenia . . . . .	55
B.2	Algoritmus vypožičania . . . . .	56



---

# Zoznam tabuliek





---

# Úvod

V posledných rokoch sa bezkontaktné čipové karty začali masovo využívať.

Prvý výraznejší náznak používania bezdrôtových čipových kariet sa objavil už 21. 11. 1994 [1], v oblasti civilného letectva, v patente s názvom „Electronic ticketing and reservation system and method“ [1], kde sa píše o umelohmotných identifikačných kartách, ktoré v rukách cestujúcich budú pristupovať k počítačom za účelom overovania [1]. O dekádu neskôr bolo len 20% leteniek elektronických [2]. Z dôvodu zvýšenia pohodlia cestujúcich si združenie IATA dalo za úlohu do 1. júna 2008 používať výhradne elektronické lístky, čo sa im podarilo napriek skepticizmu ohľadom návratnosti investície [2].

Táto technológia sa priblížila bežným ľuďom až zavádzaním v mestskej hromadnej doprave. Priekopníkmi v tomto smere boli mestá Soul (1996) [3] a Hongkong (1997) [4]. Pre porovnanie Londýn zaviedol karty Oyster až v roku 2003 [5] a Praha zaviedla OpenCard v roku 2006 [6]. Podrobnejšie sú tieto prípady popísané v tabuľke 0.1.

Zavedenie bezkontaktných čipových kariet malo niekoľko pozitívnych dopadov a ponúka niektoré výhody. Spomeniem tie, ktoré považujem za zaujímavejšie:

- V systémoch, v ktorých sa pred zavedením bezkontaktných kariet platilo za jazdu u vodiča sa urýchlil nástup cestujúcich a dopravný podnik sa odbremenil od veľkého množstva mincí.
- V systémoch, kde sa po zavedení kariet, je potrebné pri každom nástupe tzv. „pípnuť“, či už za účelom platby alebo preukázania časového lístku, sa zjednodušilo monitorovanie obsadenosti liniek.
- Ponúka možnosť rozličného spoplatnenia jázdy v špičke a mimo nej.

Bezkontaktné čipové karty nie sú čarovný prútik, ktorý rieši problémy. No ponúka nástroje, ktoré môžu k riešeniu pomôcť.

Tabuľka 0.1: Tabuľka histórie zavádzania bezkontaktných čipových kariet v mestskej hromadnej doprave

mesto	názov karty	rok zavedenia karty	technológia	rok zavedenia technológie
Soul	Upass	1996 [3]	MIFARE [3]	1994 [3]
Hongkong	Octopus	1997 [4]	FeliCa [7]	1994 [8]
Londýn	Oyster	2003 [5]	ISO/IEC 14443 A [5]	2001 [9]
Praha	Opencard	2006 [6]	ISO/IEC 14443 A	2001 [9]

Myšlienka lokalizovať predmety pomocou rádiových vln sú skoro také staré ako samotné využívanie rádiových vln. Prvým praktickým využitím sa javí byť vynález pána Christian Hülsmeier, ktorý v roku 1904 verejne demonštroval svoj systém slúžiaci na detekciu lodí v okolí detektoru. Vychádzal z teoretických poznatkov pánov Heinrich Hertz a James Clerk Maxwell. Systém však nezaujal. Dopyt po tejto technológii nastal až začiatkom tridsiatych rokov minulého storočia za účelom detekovania lietadiel vo vzdušnom priestore, čo viedlo k vytvoreniu radaru, ako ho poznáme dnes. [10]

Radar nebol jedinou technológiou, založenou na rádiových vlnách, ktorá uzrela svetlo sveta v prvej polovici minulého storočia. Začiatkom druhej svetovej vojny malo Kráľovské letectvo Veľkej Británie problémy s presnosťou nočného bombardovania. Na druhej strane denné bombardovanie prinášalo vysoké straty. V júny roku 1940 Dr. Robert J. Dippy predviedol plán na lokalizovanie cieľu pomocou rádiových signálov. Tento systém dostal meno GEE. Poprvý krát bol tento systém použitý pri nočnom bombardovaní z 11. na 12. augusta 1941. Jeho výsledky boli prekvapivo dobré. [11]

Tento systém bol vhodný pre lokalizáciu na európske vzdialenosti, ale bol nepoužiteľný v prípade veľkých vzdialeností. Z tohto dôvodu američania na základe systému GEE vyvinuli systém LORAN pre potreby vojny v Pacifiku. Hlavný rozdiel bol v použití väčších vlnových dĺžok. Systém GEE sa zo začiatku volal len *G* ako *grid* - mriežka. Z bezpečnostných dôvodov bol premenovaný na GEE. [11]

Dôvod pre implementáciu systému pre vypožičiavanie predmetov, s predpokladanou aplikáciou na FIT ČVUT, je zrýchlenie vypožičiavania, hlavne v prípade prezenčného požičiavania predmetov.

---

## Ciel' práce

V tejto práci mám za úlohu navrhnúť systém pre evidenciu prítomnosti sledovaných predmetov. Okrem toho bude systém pri vybratí predmetu schopný vyzvať užívateľa k identifikácii kartou a túto identifikáciu vykonať. K tomu je potrebné si naštudovať štandard ISO/IEC 14443, ktorý pojednáva o komunikácii zariadenia s kartou/kartami[9]. Z dôvodu využívania ISO/IEC 14443 A ISIC kartami, bol tento štandard zvolený pre potreby navrhovaného systému. Štandard je zameraný na komunikáciu na krátke vzdialenosti, tzv. NFC.

Komunikáciu zo softvérového hľadiska bude zabezpečovať externá knižnica libnfc, ktorá podporuje niekoľko ďalších podobných NFC štandardov. Úlohou je sa s touto knižnicou oboznámiť a správne ju zakomponovať do výsledného prototypu funkčného systému.

V rámci evidencie sledovaných predmetov je nutné vytvoriť databázu obsahujúcu aktuálne a spoľahlivé informácie o výpožičkách. Spolu s týmto sa naskytá možnosť výberu podporných prostriedkov na ukladanie záznamov do databáze s prihliadnutím na dostupné zdroje.

Výsledný systém bude prototypom, ktorý bude možné ďalej rozširovať pre väčšie množstvo predmetov. Predpokladá sa, že takto modifikovaný systém bude zavedený a využitý pre evidenciu výpožičiek a vrátení hardvérových prípravkov študentom na FIT ČVUT. Predpokladom pre správne využitie systému je označenie sledovaných predmetov RFID tagmi, ktoré komunikujú podľa štandardu ISO/IEC 14443 A. RFID je skratka pre rádiový frekvenčnú identifikáciu.



---

## Popis problému

V tejto kapitole popisujem problém a požiadavky na jeho riešenie.

### 2.1 Súčasný stav riešenia problematiky

Na niektorých predmetoch, napríklad BI-SAP, BI-SRC, BI-VES, BI-PNO a BI-JPO, ktoré vrámci výuky využívajú hardverové prípravky. Nezriedka sa stáva, že študent nestihne vypracovať zadané úlohy v rámci vyučovacej hodiny. V tomto prípade má študent možnosť zapožičať si prípravok, dokončiť úlohu a odovzdať ju na nasledujúcej hodine. Požičanie potrebného prípravku študentovi môže byť realizované niekoľkými rôznymi spôsobmi.

V súčasnosti sa vypožičiavanie delí na tieto dve skupiny: prezenčné a absenčné.

#### 2.1.1 Prezenčné výpožičky

Pri prezenčnom študent príde za učiteľom do kancelárie, kde si vypožičia prípravok a používa ho v respíriu bez dohľadu. Za účelom evidencie je možné viesť knihu výpožičiek, no pri momentálnom, dočastnom, riešení sa spoliehame na česťnosť študenta, že prípravok vráti a nahlási prípadné poškodenie. Tento môže byť v blízkej budúcnosti nahradený možnosťou vypožičania len v rámci priestoru učebne. V učebni by bola, v stanovené časy, osoba zodpovedná za požičiavanie prípravkov. Tieto prípravky nebude možné odnieť mimo miestnosť.

Mojou úlohou je vylepšiť stávajúci problém praktickejším riešením, ako je papierová evidencia prípravkov a zamedziť riziku pochybenia ľudského faktoru pri zápise výpožičky. Po zavedení navrhovaného systému ostanú prípadnej poverenej osobe niektoré povinnosti:

- skontrolovať totožnosť študenta podľa preukazu ISIC alebo karty ČVUT a to porovnaním fotky na karte s tvárou študenta.

## 2. POPIS PROBLÉMU

---

- dozerať na vrátenie prípravkov
- dohliadať na správne používanie systému

### 2.1.2 Absenčné výpožičky

Absenčné vypožičanie sa od prezenčného líši v možnosti odnieť si prípravok mimo budovu a vrátenia v iný deň. Tento spôsob je podstatne menej využívaný bežnými študentami.

Zahrňuje niekoľko problémov:

- malé množstvo prípravkov konkrétneho druhu, ktoré vlastní fakulta
- riziko poškodenia a straty

Z tohto dôvodu systém nie je určený na tento druh požičiavania, ale môže ho podporovať.

## 2.2 Požiadavky na implementovaný systém

Na systém nasadený do používania budú kladené nasledujúce požiadavky. Náplňou tejto práce je vývoj prototypu. Preto niektoré zo spomenutých bodov nebudú zahrnuté v prvotnej implementácii.

### 2.2.1 Rozšíriteľnosť a jednoduchosť implementácie

Systém bude pravdepodobne neskôr modifikovaný a rozšírený, aby ho bolo možné používať za stanoveným účelom. Z dôvodu rýchleho porozumeniu kódu je potrebné implementovať systém tak, aby bol kód zrozumiteľný. Navyše jednoduchý kód má za následok menší počet chýb, čo umožňuje beh bez vonkajšieho zásahu.

### 2.2.2 Jednoduché ovládanie

Po nasadení, systém zodpovedný za výpožičky, bude využívaný študentmi. Preto je nutné zabezpečiť čo najjednoduchšiu obsluhu tohto systému, aby sa zamedzilo častým procesným chybám.

### 2.2.3 Výpočetná nenáročnosť

Z dôvodu nasadenia systému na vlastný hardvér, je vhodné nemrhať výpočetnými prostriedkami a vyhnúť sa potrebe operačného systému. Následkom toho je možnosť kúpy finančne nenáročného hardvéru. Požiadavok na výpočetnú nenáročnosť bude splnený len čiastočne. Dôvody budú uvedené v sekcii analýzy.

### 2.2.4 Zabezpečenie

Systém bude identifikovať užívateľov na základe ISIC kariet. Je potrebné, aby karty nebolo možné jednoducho okopírovať a vydávať sa za niekoho iného. Z tohto dôvodu nie je bezpečné využívať UID pre potreby identifikácie, pretože karta prezrádza svoje UID každej čítačke a teda sa dá okopírovať. Riešenie zabezpečenia nie je súčasťou bakalárskej práce. Bude však spomenuté ako možné rozšírenie.

### 2.2.5 Rýchlosť

Pre pohodlie užívateľov je potrebné minimalizovať odozvu systému.

## 2.3 Spracovanie existujúcich implementácií

Obvykle sa pre vypožičiavanie predmetov používa identifikácia pomocou čiarových kódov, ktorá je lacnejšia, no neumožňuje nám detekciu vybratia predmetu z miesta, kde je uložený.

Naopak technológia RFID a podobné, sa často používa v kartách pre identifikáciu ich vlastníkov alebo ako kontrola proti odcudzdeniu v obchodoch.

Nie je mi známa implementácia, kde by sa rádiová identifikácia s väzbou na krátku vzdialenosť používala pre monitorovanie prítomnosti predmetu pravidelnou komunikáciou s predmetom.

Ako najbližší podobný problém sa mi javí, dobre známy, systém GPS. Ten lokalizuje predmety na základe ich vzdialenosti od satelitov, ktorá sa meria dĺžkou odozvy. Z toho vyplýva, že kritickou časťou systému GPS sú presné hodiny na satelitoch. Túto alternatívu riešenia priblížim v časti Analýza možných riešení bez využitia NFC 3.3.

## 2.4 Popis štruktúry BP

V nasledujúcich častiach bakalárskej práce bude spracovaná analýza a návrh prototypu. Jednotlivé časti sú rozdelené na tieto sekcie:

1. Analýza a návrh – V tejto sekcii sa budem zaoberať rozborom rôznych možností riešenia. Vyhodnotím potreby alternatív a zvolím najvhodnejšiu možnosť vzhľadom k dostupným zdrojom. V návrhu zkonštruujem modely výslednej implementácie systému.
2. Realizácia a testovanie – V tejto kapitole opíšem funkčnosť riešenia, metódy a výsledky testovania.
3. Možnosti rozšírenia – Rozoberiem doposiaľ uvažované konkrétne vylepšenia pre stávajúci prototyp.

## 2. POPIS PROBLÉMU

---

4. Záver – Zhrniem všetky doposiaľ uvažované a realizované návrhy.



---

# Analýza

V tejto časti budú analyzované literárne zdroje, možnosti výberu komponentov a spôsob ich zostavenia.

## 3.1 Norma ISO/IEC 14443 [9]

Štandard ISO/IEC 14443 bol vydaný postupne v rokoch 2000 a 2001. Bol rozdelený na štyri časti. Česká verzia štandardu ČSN ISO/IEC 14443 vychádzala v rokoch 2001 a 2002. Názvy týchto častí sú nasledujúce:

1. Physical characteristics – Fyzikální charakteristiky
2. Radio frequency power and signal interface –  
Rádiofrekvenční výkonové a signálové rozhraní
3. Initialization and anticollision – Inicializace a antikoze
4. Transmission protocol – Protokol přenosu

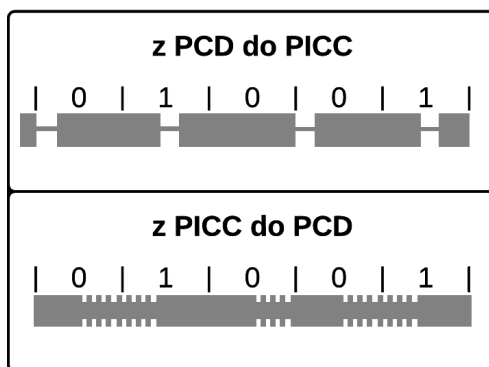
### 3.1.1 Časť 1 – Fyzikálne charakteristiky

Prvá časť obsahuje iba osem strán. Obsah môžem zhrnúť do niekoľkých bodov:

- Definícia pojmov – definícia PICC a PCD a podporných pojmov.
- Rozmery karty – odvoláva sa na štandard ISO/IEC 7810.
- Odolnosť voči vonkajším vplyvom – od namáhania ohybom po intenzitu röntgenového žiarenia.

Z tejto časti stojí za pozornosť rozdelenie pojmov PICC a PCD:

- PICC – karta s väzbou na blízko – Karta obsahujúca integrovaný obvod, s ktorým je možné komunikovať pomocou indukčnej väzby na blízko.



Obr. 3.1: Príklad komunikačných signálov pre rozhranie typu A [9]

- PCD – Zariadenie, ktoré používa indukčnú väzbu pre napájanie karty a komunikáciu s ňou.

### 3.1.2 Časť 2 – Rádiofrekvenčné výkonové a signálové rozhranie

Druhá časť sa zaoberá elektromagnetickým poľom potrebným pre napájanie PICC a komunikáciu medzi PCD a PICC. Štandard sa delí na verziu A a B. Nebudem sa zvlášť zaoberať verziou B, pretože, ako spomínam v Ciele práce, 1 verziu A používajú ISIC karty.

Vlastnosti spoločné pre obe verzie štandardu (A aj B):

- Nosná frekvencia, ktorú vysiela PCD, je  $13,56\text{MHz}$  ( $\pm 7\text{ kHz}$ )
- Pracovné pole vysielať PCD, bez modulácie, má efektívnu hodnotu od  $1,5\text{A/m}$  do  $7,5\text{A/m}$

Vlastnosti špecifické pre štandard A:

- Prenosová rýchlosť =  $\text{nosná frekvencia}/128 = (13,56\text{MHz})/128 \cong 106\text{kbit/s}$

Pre komunikáciu sa využíva amplitúdová modulácia. Vzhľadom na to, že karta je napájaná elektromagnetickým poľom skrz ktoré súčasne komunikuje, nie je možné využívať pre komunikáciu oboma smermi rovnaké kódovanie viz. 3.1. Komunikácia je obojsmerná, ale v jeden okamih môže prebiehať len v jednom smere – takzvaný half-duplex.

Kódovanie komunikácie z PCD do PICC:

- logická 1 sa vysiela tak, že po polovici trvania bitu nastane pauza.
- logická 0 sa vysiela dvoma spôsobmi:

- ak sa jedná o prvý bit od začiatku rámca alebo sa jedná o bit, ktorému predchádza logická nula, tak sa pauza vyskytne na začiatku doby trvania bitu.
- ak nespadá do podmienky v predchádzajúcom bode, tak sa po celú dobu trvania bitu nevyskytne modulácia.

Pauza je definovaná ako stopercentná amplitúdová modulácia signálu. Presnejšie ide o pokles amplitúdy poľa pod 5% jeho nemodulovanej hodnoty. Táto pauza môže trvať od  $0,5\mu s$  do  $3\mu s$  v závislosti od doby poklesu amplitúdy zo 100% pod 5%.

Kódovanie komunikácie z PICC do PCD tak isto využíva amplitúdovú moduláciu:

- logická 1 – modulácia pomocnou nosnou frekvenciou po dobu prvej polovice trvania bitu
- logická 0 – modulácia pomocnou nosnou frekvenciou po dobu druhej polovice trvania bitu

Pomocná nosná frekvencia je  $847kHz/s$  ( $nosná\ frekvencia/16 = 13,56MHz/16 \cong 847kHz/s$ ), čo znamená, že doba trvania jedného bitu ( $nosná\ frekvencia/128$ ) je rovná osemnásobku pomocnej nosnej frekvencie ( $128/16 = 8$ ). Z čoho vyplýva, že behom jednej polovice doby trvania bitu vzniknú na obálke elektromagnetického poľa 4 "zuby". A to vždy v jednej polovici trvania bitu. V prípade, že budú naraz vysielat dve karty rôznu hodnotu bitu, nastane modulácia po celú dobu trvania bitu. Táto metóda modulácie umožňuje rozpoznanie viacerých PICC komunikujúcich s PCD. O rozpoznaní viacerých PICC sa píše v časti 3.

### 3.1.3 Časť 3 – Inicializácia a antikolízia

V tejto časti sa píše o dátach posielaných za účelom inicializácie komunikácie medzi PCD a PICC. Presnejšie sa píše o:

- bitový formát rámca
- časové návaznosti, ktoré sú kritické pre rozpoznanie viacerých kariet
- metódy vybratia jednej PICC z niekoľkých v elektromagnetickom poli PCD

PICC pri prvotnej inicializácii prechádza troma stavmi. Tieto stavy sú:

- IDLE – PICC je napájaná elektromagnetickým poľom a čaká len na príkazy REQA alebo WUPA. Obdržaním jedného z týchto príkazov sa dostane do stavu READY

### 3. ANALÝZA

---

- READY – Ak PICC nie je vybratá príkazom SELECT, presúva sa do stavu IDLE. Inak sa presúva do stavu ACTIVE.
- ACTIVE – PICC, v tomto stave môže buďto komunikovať podľa časti 4 tohto štandardu alebo prijať správu HLTA. V tomto prípade sa PICC presunie do stavu HALT, ktorý je ppísaný nižšie.

Okrem týchto stavov existujú ďalšie tri stavy. Slúžia na 'odloženie' inicializovaných kariet a komunikáciu s inými.

- HALT – Tento stav sa chová analogicky k stavu IDLE, ale karta sa z neho prebudí len na príkaz WUPA a presunie sa do stavu READY\*. To znamená, že narozdiel od stavu IDLE, ignoruje príkaz REQA.
- READY\* – Tento stav je analogický k stavu READY. Na rovnaký príkaz ako sa PICC dostane zo stavu READY do stavu ACTIVE sa dostane aj zo stavu READY\* do ACTIVE\*. Konkrétne sa jedná o príkaz SELECT.
- ACTIVE\* – Správa sa rovnako ako stav ACTIVE.

Toto zdvojenie (IDLE – HALT, READY – READY\* a ACTIVE – ACTIVE\*) má za následok zjednodušenie inicializácie a identifikácie všetkých kariet v poli. PCD si najprv môže inicializovať všetky PICC v dosahu a presunúť ich do stavu HALT. A to bez rizika, že by sa inicializované karty opakovali (za predpokladu, že neboli vyňaté z elektromagnetického poľa). Potom si môže vybrať s ktorou kartou bude komunikovať postupnosťou príkazov WUPA a SELECT.

Správy sa posielajú v troch rôznych rámcoch:

- Krátky rámec – Vysiela start bit, následne 7 bitov v poradí od LSB a ukončovací bit. Paritný bit sa neposiela.
- Štandardný rámec – Prenáša najprv start bit, potom 8 bitov správy, jeden paritný bit, 8 bitov správy, jeden paritný bit a tak ďalej až za odošle celá správa. Po jej konci následuje ukončovací bit.
- Ostatné – Správy CRC\_A a SAK majú vlastný (spoločný) typ rámca.

Tieto rámce používajú správy nasledovne:

- REQA a WUPA majú ako jediné formu krátkeho rámca.
- ANTICOLISION pozostáva z dvoch bajtov a prípadne časti prijatej správy ako je popísané v bode 6.
- SELECT je správa ANTICOLISION, ktorá obsahuje celú prijatú správu obsahujúcu UID.
- CRC\_A je veľkosti 2 bajty a má vlastný typ rámca.

- ATQA má dĺžku 2 bajty a obsahuje informáciu o dĺžke UID. UID môže mať 3 rôzne dĺžky a to: 4, 7 a 10 bajtov. Teda je možné adresovať  $2^{32} + 2^{56} + 2^{80} \cong 1.2 \times 10^{24}$  kariet.
- SAK má veľkosť 1 bajt a jeho rámec má rovnaký typ ako CRC\_A.

Základná procedúra pri inicializácii novej karty, v prípade viacerých kariet v poli, prebieha nasledovne:

1. Všetky karty sú v stave IDLE
2. PCD vyšle správu REQA alebo WUPA
3. Každá PICC vyšle správu ATQA a presunie sa do stavu READY
4. PCD pošle správu ANTICOLISION obsahujúcu parameter kaskádnej úrovne. V tomto prípade 1.
5. Každá PICC vyšle 5 bajtov, ktoré obsahujú:
  - Ak sa jedná o PICC s UID veľkosti 4 bajty, pošle celé svoje UID a kontrolný súčet.
  - Inak pošle 1 bajt s hodnotou CT, 3 najvyššie bajty svojej adresy a kontrolný súčet.
6. Ak nastane kolízia:
  - a) PCD pošle správu ANTICOLISION obsahujúcu parameter kaskádnej úrovne, rovnakú ako v bode 4, a časť prijatej správy po 1. kolíziu a bit 0 alebo 1 za účelom vylúčenia aspoň jednej PICC. Táto správa nemusí byť zarovnaná na bajty.
  - b) Každé PICC, ktorého správa začínala postupnosťou bitov, ktorú PCD vyslalo v predchádzajúcom bode, pokračuje od tohto bitu posielat správu posielanú v bode 5.
7. Ak nastane ďalšia kolízia vrátíme sa na bod 6.
8. Inak PCD pošle správu SELECT s celou správou poslanou v bode 5 a pripojí správu CRC\_A.
9. Zvolené PICC odpovedá správou SAK, ktorá obsahuje informáciu, či je UID kompletné a zopakuje správu CRC\_A.
10. Ak je UID kompletné, PICC sa presunie do stavu ACTIVE a týmto končí výber.
11. V opačnom prípade sa vrátíme na bod 4 s tým rozdielom, že kaskádna úroveň bude o jedno vyššia a výber prebieha analogicky. Úroveň výberu sú 3 a zodpovedajú dĺžkam správ.

#### 3.1.4 Časť 4 – Protokol prenosu

Obsahom tejto časti je prenos datových blokov po nadviazaní komunikácie podľa predchádzajúcej časti. Pripúšťa sa tu aj komunikácia s niekoľkými PICC naraz tzv. multicasting, pričom každá PICC musí byť presunutá do stavu active separátne.

Komunikácia podľa tohto štandardu, po nadviazaní komunikácie podľa minulého bodu, prebieha nasledovne:

1. PCD pošle bajt RATS, ktorý obsahuje informácie o maximálnej dĺžke rámca vysielaného PCD a definuje CID pre PICC, ktorá je v stave active a nemá pridelené CID. CID je identifikátor PICC, ktorý je nemenný po dobu kým je PICC aktívna. V prípade komunikácie s viacerými PICC naraz sa definuje CID pre všetky PICC rovnako. V opačnom prípade sa definuje unikátne pre každú PICC.
2. PICC odpovedá správou ATS, ktorá pozostáva z niekoľkých častí. Z nich štyri sú povinné:
  - a) Bajt TL definuje dĺžku správy.
  - b) Bajt T0, bajt formátu, definuje maximálnu veľkosť rámca vysielaného PICC a či budú nasledovať nepovinné bajty TA(1), TB(1) a TC(1).
  - c) Nasledujú nepovinné bajty TA(1), TB(1) a TC(1) a historické bajty T1 až Tk.

Pre samotnú komunikáciu sa používajú rámce, ktoré sa delia:

1. Pole PROLOG, ktoré sa delí na 3 bajty:
  - a) PCB nesie informáciu o tom, či budú nasledovať bajty CID a NAD. Okrem toho hlavne definuje typ rámca. Rozoznáva 3 hlavné typy rámca:
    - i. I-typ prenáša informácie pre aplikačnú vrstvu.
    - ii. R-typ sa používa pre pozitívne alebo negatívne potvrdenie.
    - iii. S-typ nesie riadiacu informáciu ako napríklad predĺženie doby čakania.
  - b) Nepovinný CID spomínaný vyššie.
  - c) Nepovinný NAD, ktorý je ďalším identifikátorom.
2. Nepovinná datová časť, ktorá nie je ďalej špecifikovaná.
3. Pole EPILOG, ktoré obsahuje len kód detekcie chyby.

## 3.2 Libnfc

Na stránke nfc-tools, ktorá zastrešuje projekt knižnice libnfc sa píše, prečo tento projekt vznikol: „Since the RFID/NFC market is spoiled by proprietary hard and software we want to contribute constructively by distributing a free library and free tools which can be used for various RFID and NFC applications.“ [12] Čo môžem preložiť následovne: Keďže trh RFID/NFC je skazený licencovaným hardvér a softvér, chceme konštruktívne prispieť distribuovaním voľne dostupných knižníc a nástrojov, ktoré sú použiteľné pre rôzne RFID a NFC aplikácie. Projekt nfc-tools zahŕňa nasledujúce štyri projekty, z ktorých posledné dva sú v beta verzii.

- libnfc – prvé slobodné nízko úrovňové NFC SDK a programátorské API, ktoré je nezávislé na platforme.
- libfreefare – API pre MIFARE karty postavené na libnfc
- ifdnfc – IFDHaldler pre čítačky kompatibilné s libnfc
- nfc-eventd – démon monitorujúci NFC

Knižnica sa skladá z deviatich modulov:

- Library initialization/deinitialization
- NFC Device/Hardware manipulation
- NFC initiator
- NFC target
- Error reporting
- Special data accessors
- Properties accessors
- Miscellaneous
- To-string converters

Na základnú komunikáciu sú potrebné prvé tri moduly. Prvý na inicializáciu libnfc, druhý na inicializáciu PCD a tretí na inicializáciu PICC podľa ISO/IEC 14443-3. Konkrétne tieto funkcie:

- Z prvého modulu – Library initialization/deinitialization:
  - `nfc_init` – iniciátor knižnice
  - `nfc_exit` – deiniciátor knižnice
- Z druhého modulu – NFC Device/Hardware manipulation:

- `nfc_list_devices` – vyhľadá všetky prítomné kompatibilné PCD.
- `nfc_open` – otvorí PCD.
- `nfc_close` – zavrie PCD.
- Z tretieho modulu – NFC iniciator:
  - `nfc_initiator_init` – nastaví PCD ako čítačku.
  - `nfc_initiator_list_passive_targets` – vyhľadá všetky PICC v dosahu PCD.

Štandard aj knižnica podporujú komunikáciu s niekoľkými PCD funkciou `nfc_list_devices` a pre každé PCD podporuje komunikáciu s niekoľkými PICC funkciou `nfc_initiator_list_passive_targets`.

### 3.3 Analýza možných riešení bez využitia NFC

V tejto sekcii je navrhnutých niekoľko možností ako realizovať systém bez využitia technológie NFC.

#### 3.3.1 Riešenie pomocou dotykových senzorov

V tejto možnosti by ku každému vypožičiavanému predmetu prislúchalo jedno tlačidlo, ktoré by predmet stláčal vlastnou váhou. Pri vrátení by človek musel identifikovať seba aj predmet ktorý vracia, napríklad pomocou čiarového kódu a položiť ho na tlačidlo. Pri vypožičaní by musel identifikovať len seba pretože systém vie aký predmet má na konkrétnom tlačidle. Táto alternatíva je najlacnejšia na zavedenie, ale chýba tu priebežná kontrola predmetu, ktorý je odložený. Systém by vedel či je predmet na mieste alebo nie, ale nevie overiť jeho identitu.

#### 3.3.2 Riešenie pomocou jednej antény

Tento systém je veľmi podobný systému GPS. Pri tejto alternatíve je potrebná aspoň jedna anténa s väzbou na väčšiu vzdialenosť ako je bežná NFC. Každý predmet by mal prilepenú anténu, napríklad RFID tag. Väzba by reagovala na vzdialenosť aspoň do dvoch metrov a pri vyňatí predmetu z poličky by systém registroval jeho oddialenie od antény a vyžiadal autentifikáciu užívateľa. Pri vrátení predmetu do predpísanej vzdialenosti by systém nadviazal komunikáciu s predmetom a následne vyžiadal autentifikáciu užívateľa. Počet antén by bolo možné rozšíriť až na štyri a tým identifikovať polohu predmetu v 3D priestore so znížením pravdepodobnosti výskytu chyby. Najväčšou nevýhodou tejto možnosti je potreba veľmi presného merania času, aby bolo možné zistiť



vzialenosť predmetu od antény. Ďalšou nevýhodou je rušenie elektromagnetického signálu.

## 3.4 Analýza možných riešení s využitím NFC

Ako protiklad k predchádzajúcej sekcii, je v tejto sekcii krátka obhajoba riešenia s využitím NFC a diskusia niekoľkých variant ako je možné navrhnúť systém s využitím NFC.

Vzhľadom na to, že RFID tagy nie sú cenovo náročné a je to rovnaká technológia, ako sa bude využívať pre identifikáciu študentov, som sa rozhodol pre RFID tagy podľa štandardu ISO/IEC 14443 A.

Na základe tohto výberu je možná stála kontrola prítomnosti predmetu na jeho mieste v elektromagnetickom poli antény RFID čítačky. Navyše je vďaka RFID tagu možné identifikovať evidovaný predmet.

Naskytá sa niekoľko hardvérových topológií. Každá z nich je ďalej popísaná aj s diagramom:

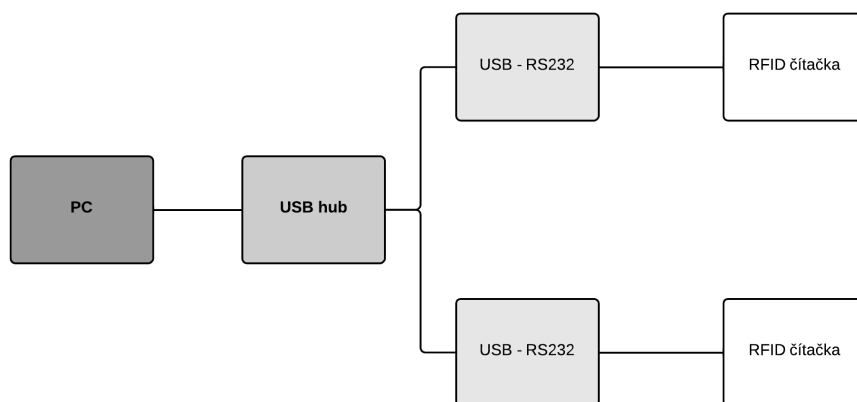
1. Každý prípravok bude mať vlastnú RFID čítačku a tie budú prepojené cez USB hub.
2. Každý prípravok bude mať vlastnú RFID čítačku a tie budú prepojené cez vlastný multiplexor.
3. Každý prípravok bude mať vlastnú RFID anténu a tieto antény budú prepojené cez vlastný interface.

### 3.4.1 Zapojenie s využitím USB hub

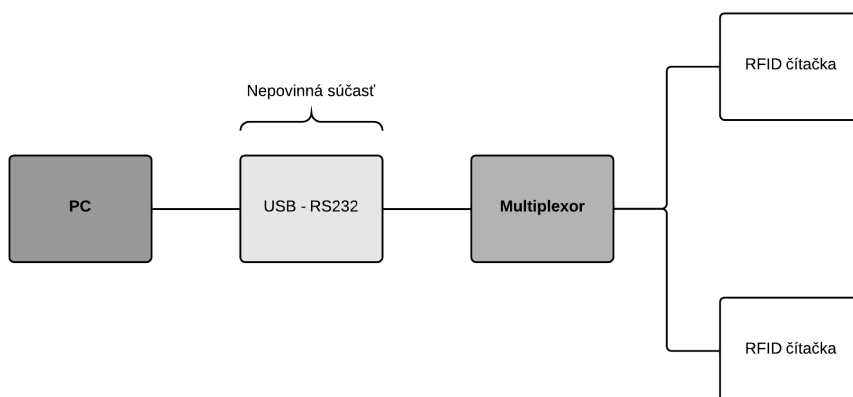
Každý prípravok bude mať vlastnú RFID čítačku a tie budú prepojené cez USB hub. Najväčšou výhodou tejto alternatívy je jednoduchá implementácia. Knižnica libnfc navyše podporuje komunikáciu s niekoľkými PCD. Tie môžu byť zapojené buďto priamo k počítaču alebo cez USB hub a USB-serial konvertory. Týmto odpadá práca s prepínaním spojenia medzi jednotlivými PCD. Nevýhodou tohto riešenia je vysoká cena z dôvodu potreby nakúpenia hardvéru:

- 1 počítač, na ktorom bude spustená aplikácia
- 0-1 USB hub – v závislosti od počtu USB portov na počítači a počtu čítačiek
- $n$  USB – RS-232 konvertorov
- $n$  RFID čítačiek

Pričom  $n$  je maximálny počet predmetov, ktoré budú k dispozícii na požívanie. Toto číslo sa dá dvakrát alebo trikrát znížiť tým, že do poľa jednej



Obr. 3.2: Schéma zapojenia s využitím USB hub

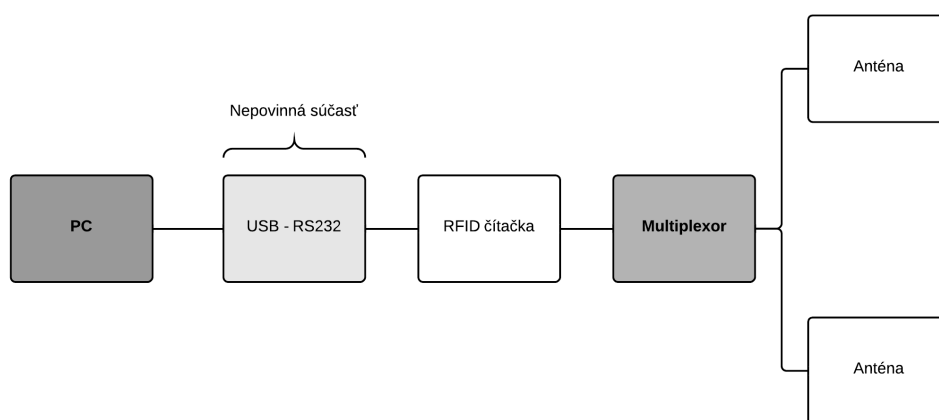


Obr. 3.3: Schéma zapojenia s využitím multiplexoru

čítačky vložím viac požičiavaných predmetov. Keďže jedna RFID čítačka stojí \$39.95 [13] a cena ostatného hardvéru je, v prípade väčšieho množstva čítačiek, zanedbateľná, tak závislosť počtu požičiavateľných predmetov k cene bude lineárny.

#### 3.4.2 Zapojenie s využitím multiplexoru

Táto možnosť riešenia je podobná tej predchádzajúcej. Hlavným rozdielom je nahradenie USB hubu vlastným multiplexorom, ktorý sa bude starať o prepojenie RFID čítačiek s počítačom. Výčet hardvéru sa teda zmení takto:



Obr. 3.4: Schéma zapojenia s oddelenou anténou

- 1 počítač, na ktorom bude spustená aplikácia
- 1 vlastný multiplexor
- 0-1 USB – RS-232 konvertor – v závislosti od portov, ktoré sa nachádzajú na počítači
- $n$  RFID čítačiek

Táto alternatíva nám ceny hardvéru zníži len minimálne pretože najväčší podiel ceny pripadá na RFID čítačky. Výhodou tejto alternatívy je možnosť nepoužiť USB konektor a nahradiť ho sériovou linkou. To umožňuje použiť lacnejšie mikrokontroléry, ktoré nepodporujú USB. Nevýhodou tejto možnosti je zabezpečenie ovládania multiplexoru, ktorý narozdiel od USB hub(u), nie je možné kúpiť ako hotový produkt a používať bez zmeny.

#### 3.4.3 Zapojenie s oddelenou anténou

V tejto možnosti, narozdiel od predchádzajúcej, sa multiplexor vloží medzi RFID čítačku a anténu. Antény sa nachádzajú na jednom plošnom spoji, ktorý by bolo potrebné rozrezať a na časť bez antény napájkovať spojenie s novou anténou. Táto anténa bude musieť mať podobné parametre ako odrezaná anténa, aby bola schopná komunikovať. V tejto alternatíve navyše vznikajú problémy s prenosom signálu na vzdialenosť minimálne niekoľko decimetrov medzi čítačkou a anténou. Jedná sa o vysokofrekvenčný signál, ktorý je určený pre anténu v bezprostrednej vzdialenosti od RFID čítačky.

Pri tejto alternatíve sa množstvo potrebného hardvéru zníži nasledovne:

- 1 počítač, na ktorom bude spustená aplikácia

### 3. ANALÝZA

---

- 1 vlastný analógový multiplexor
- 0-1 USB – RS-232 konvertor
- 1 RFID čítačka – časť, ktorá ostane po odrezaní antény
- $n$  RFID antén

Pri tejto možnosti nám výrazne klesne cena, keďže potrebujeme len jednu RFID čítačku. No v tomto prípade je potrebné zaobstarať vhodný multiplexor, vyriešiť prenos signálu k anténam a zostrojiť antény. Náročnosť tohoto riešenia je vyššia ako je rámec bakalárskej práce.

### 3.5 Riadiaca platforma

V závislosti na jednoduchosti modifikácie a prístupu sa ponúka niekoľko možností:

- mikrokontrolér
- jednodoskový počítač (single-board computer)
- plnohodnotný počítač

V prípade potreby využitia veľkého množstva prípravkov by bolo výhodné zvoliť prvú možnosť a to z dôvodu nízkej nákupnej ceny. Na druhej strane mikrokontroléry majú slabú podporu grafiky a k zmene softvéru je potrebná externá výpočetná jednotka.

Ďalšou možnosťou, ktorá je z uvažovaných možností protipólom k prvej, je plnohodnotný počítač prípadne laptop. Hlavnou nevýhodou je vysoká nákupná cena. Okrem ceny prekážajú aj jeho veľké rozmery a spotreba energie. Tieto nevýhody sú vyvážené vysokým výpočtovým výkonom, veľkým množstvom vstupných a výstupných portov, možnosťou prispôsobenia hardvéru podľa potrieb a komfortným prístupom cez plnohodnotný operačný systém. Vďaka tomu je práca aj obsluha takého zariadenia jednoduchšia. Pomáha tomu aj grafické rozhranie, ktoré je obvykle súčasťou operačného systému.

Poslednou možnosťou je jednodoskový počítač. Táto platforma spája výhody oboch predchádzajúcich variánt. Dôraz je kladený na pomer ceny a využiteľnosti platformy. Je možné s nimi pracovať aj bez potreby externej výpočetnej jednotky, len za potreby vstupných a výstupných periférií. Výpočtový výkon je výrazne nižší ako u plnohodnotných počítačov, no postačuje pre beh prispôbovaných operačných systémov s podporou grafického rozhrania.

## 3.6 Vstupné a výstupné periférie

Pre obsluhu systému je potrebné, aby mal užívateľ možnosť komunikovať so systémom. Na to slúžia vstupné a výstupné periférie. Tieto môžu byť spojené, teda vstupno-výstupné, alebo oddelené. Vstupno-výstupné periférie sa vyznačujú vyššou užívateľskou prívetivosťou a vyššou modularitou, ale aj vyššou cenou.

### 3.6.1 Oddelené vstupné periférie

Jednou z možností pre zaistenie vstupu je niekoľko samostatných tlačidiel. Každé z nich samostatne zasiela dotaz o akcii, ktorá sa spracuje v riadiacej jednotke. Výhodou tohto riešenia je nízka cena tlačidiel. Nevýhodou je potreba panelu, kde budú tlačidlá umiestnené.

Druhou možnosťou je klávesnica, ktorá rieši problém s umiestnením tlačidiel do panelu. Nevýhodou je znížený komfort užívateľa v prípade, že bude potrebovať len niekoľko tlačidiel.

### 3.6.2 Oddelené výstupné periférie

Pre komunikáciu s užívateľom je potrebné zobraziť výstup v zodpovedajúcej forme. Naskytá sa možnosť použitia rôznych druhov displejov.

- jednofarebný LED displej so svietivými bodmi usporiadanými do štvorcovej mriežky – výhodou je nízka cena no postačuje len pre zobrazenie jedoduchých oznamov.
- farebný LCD displej – užívateľsky prívetivejšia alternatíva, ktorá vyžaduje podporu grafického výstupu od hardvéru a operačného systému.

### 3.6.3 Spojené vstupné a výstupné periférie

V tejto alternatíve je možné využiť len dotykový LCD displej. Ten však môže byť dvoch typov:

- rezistívny dotykový displej
  - dotyk sa detekuje pri ohybe vrchnej vodivej vrstvy, čo dlhodobým používaním znižuje schopnosť prijímať vstupy.
  - pre dotyk je potrebné použiť necht alebo stylus
  - táto technológia je lacnejšia a takéto displeje sa bežne predávajú samostatne
- kapacitný dotykový displej

### 3. ANALÝZA

---

- dlhšia životnosť, pretože nedochádza k fyzickej deformácii materiálu
- vyžaduje dotyk bruškom prsta
- technologicky náročná výroba má za následok malú dostupnosť samostatne predajných displejov
- bežne používaná technológia na mobilných telefónoch a tabletoch

Po prehodnotení možností a zdrojov som sa rozhodol pre zapojenie s využitím USB hub 3.4.1 z niekoľkých dôvodov:

- vytváram prototyp v obmedzenom rozsahu (jednotky sledovaných predmetov) => Kritérium ceny v závislosti od počtu predmetov nie je dôležitým merítkom.
- jednoduchá implementácia minimalizuje chybovosť
- v prípade poruchy jednoduchá výmena komponent
- túto alternatívu bude možné zmeniť na alternatívu 2 alebo 3 bez zásahu do väčšiny kódu

Rovnako ako pri ostatných alternatívach je potrebné zvážiť nasledujúce súčasti systému.

### 3.7 Jednodoskový počítač

Z možností, o ktorých pojednávam v sekcii Riadiaca platforma 3.5 som si vybral jednodoskový počítač, pretože sa najviac zhoduje s požiadavkami na systém. Konkrétne podpora prispôbeného operačného systému s grafickým rozhraním, ktoré spňujú požiadavky na Rozšíriteľnosť jednoduchosť implementácie 2.2.1 a jednoduché ovládanie 2.2.2. Odhadované softvérové nároky sú dostatočne nízke pre výber tejto platformy. Navyše jednodoskové počítače sú orientované, okrem iného, na jednoduché prototypovanie.

V tejto oblasti sa ponúka niekoľko značiek. Z tých známejších som vybral:

- Raspberry Pi
- Beagleboard
- Intel Galileo
- Cubieboard
- UDOO

Od týchto výrobcov som vybral najlacnejšie dostupné počítače:

Tabuľka 3.1: Tabuľka porovnania jednodoskových počítačov číslo 1 [14]

	Beaglebone Black	Intel Galileo	Raspberry Pi
SoC	Texas Instruments AM3358	Intel Quark X1000	Broadcom BCM283
CPU Arch.	ARM Cortex-A8	Intel X1000 i586	ARM1176
Clock	1GHz	400MHz	700MHz
Mem.	512MB	256MB	model A 256MB model B 512MB
GPU	PowerVR SGX530	None	Broadcom VideoCore IV
Cena	55\$	80\$	model A 25\$ model B 35\$

- Raspberry Pi model A
- Raspberry Pi model B
- Beaglebone Black
- Intel Galileo
- Cubieboard 2
- UDOO Dual Basic

Skrátená charakteristika týchto počítačov je uvedená v tabuľkách 3.1 a 3.2. Nedostatky jednotlivých počítačov vzhľadom k ich zamýšľanej aplikácii sú nasledovné:

- Intel Galileo neobsahuje grafický procesor. Táto vlastnosť by sa stala výrazným nedostatkom v prípade grafického rozhrania s užívateľom.
- UDOO Dual Basic a Cubieboard 2 sú výkonné jednodoskové počítače s dvojjadrovým procesorom, vhodným pre zložitejšie výpočty. Vzhľadom na zamýšľané využitie vybraného počítača je tento výpočtový výkon zbytočne veľký. Táto ich prednosť sa odráža aj v ich cene a kôli nej tieto počítače vylučujem z výberu.
- Beaglebone Black by bolo tiež možné použiť, no v porovnaní s Raspberry Pi, nemá skoro žiadnu podporu zariadení, ktoré sú optimalizované pre počítače od Beagleboard. To síce neznemožňuje použitie Beaglebone Black, no zťažuje ho.

Mojim výberom je Raspberry Pi, konkrétne model B revízia 2. Jedná sa o verziu s 512MB RAM. Revízia 1 má 256MB RAM a už nie je v predaji. Tento počítač bol zvolený z nasledujúcich dôvodov:

Tabuľka 3.2: Tabuľka porovnania jednodoskových počítačov číslo 2

	UDOO Dual Basic [15]	Cubieboard 2 [16]
CPU	ARM Cortex-A9 Dual core	ARM Cortex-A7 Dual core
Arch.	ARMv7-A	ARMv7-A
Clock	1GHz	1GHz
Mem.	1GB DDR3	1GB DDR3
GPU	GPU Vivante GC 880 + Vivante GC 320	OpenGL ES GPU
Cena	73€	45.95£

- dva milióny predaných kusov Raspberry Pi sú príslubom veľkej užívateľskej základne [17]
- počítače Raspberry Pi podporujú niekoľko operačných systémov s grafickým rozhraním [18]
- nízka cena modelu \$39.95 [19]
- model B sa stal všeobecne obľúbený a to sa odrazilo aj na dostupnom hardvéri prispôbenom pre model B

### 3.8 Periférie pre Raspberry Pi

Vzhľadom na analýzu vstupných a výstupných periférií 3.6 a výber riadiacej platformy 3.7 som sa rozhodol zamerať na periférie určené pre Raspberry Pi. V obchode Adafruit [20] je dostupných niekoľko periférií. Rozdelím ich na rovnaké druhy ako v analýze periférií. Parametre týchto periférií sú popísané v tabuľkách 3.3 a 3.4.

- Oddelené periférie
  - Výstupné periférie – farebné displeje Niektoré z týchto produktov majú tlačidlá, no tie slúžia na nastavenie farebnosti obrazu. Jedná sa o produkty číslo 910 [21], 911 [22], 912 [23], 913 [24], 946 [25], 947 [26].
  - Oddelené periférie na jednom plošnom spoji. Jedná sa o displeje, ktoré slúžia na zobrazovanie krátkeho textu. Niektoré z nich ponúkajú rôznovarebné podsvietenie. Sú to produkty číslo 1109 [27], 1110 [28] a 1115 [29].
- Spojené periférie
  - Rezistívny dotykový displej je produkt číslo 1601 [30].
  - Kapacitný dotykový displej je produkt číslo 1983 [31].



Tabuľka 3.3: Tabuľka porovnania displejov číslo 1

iden- tifi- kátor	cena	konektivita	zapúz- drenie	počet znakov	Počet nastavi- teľných tlačdiel
1109	24.95\$	I <sup>2</sup> C	nie	16x2	5
1110	24.95\$	I <sup>2</sup> C	nie	16x2	5
1115	19.95\$	I <sup>2</sup> C	nie	16x2	5

Tabuľka 3.4: Tabuľka porovnania displejov číslo 2

iden- tifi- kátor	uhlo- priečka	cena	konektivita	zapúz- drenie	rozlí- šenie	Počet nastavi- teľných tlačdiel
910	1.5"	39.95\$	Composite video	nie	160x240	0
911	2.0"	39.95\$	Composite video	nie	320x240	0
912	2.5"	44.95\$	Composite video	nie	320x240	0
913	3.5"	44.95\$	Composite video	nie	320x240	0
946	4.3"	49.95\$	Composite video	áno	480x272	0
947	7.0"	74.95\$	Composite video	áno	480x234	0
1601	2.8"	34.95\$	SPI, GPIO, I2C	nie	320x240	4
1983	2.8"	44.95\$	SPI, GPIO, I2C	nie	320x240	4

Z dôvodov uvedených v analýze periférií 3.6 som sa rozhodol pre kapacitný dotykový displej. Celý názov je *PiTFT 2.8" TFT 320x240 + Capacitive Touchscreen - Raspberry Pi Model B*. Naviac jeho cena v porovnaní s ostatnými nie je výrazne vyššia vzhľadom na parametre. Aj keď sa displej číslo 1983 nepredáva v plastovom ráme, je možné dokúpiť zapúzdrenie zdieľané s počítačom. To je možné vďaka tomu, že rovnako ako produkty 1109, 1110, 1115 a 1601 sa dá pripojiť priamo na vrch Raspberry Pi model B.

Ďalšie technické parametre produktu číslo 1983:

- single-touch tzn. obrazovka podporuje len jeden dotyk naraz
- 16-bitová farebnosť
- pri komunikácii s Raspberry Pi používa len niekoľko pinov, ostatné piny nie sú využívané, ale sú fyzicky napojené na plošný spoj a dajú sa z neho vyviesť. Používané piny sú:
  - I<sup>2</sup>C piny (SDA & SCL)
  - SPI piny (SCK, MOSI, MISO, CE0)
  - GPIO piny #25 & #24

- tlačidlá sú napojené cez ďalšie GPIO porty

## 3.9 Programovací jazyk

Jazyk C som zvolil z toho dôvodu, lebo je rýchly a nízkoúrovňový. Systém je dostatočne jednoduchý na to, aby nebolo potrebné používať objekty k zjednodušeniu chápania systému. Navyše aj knižnica libnfc je napísaná v jazyku C.

## 3.10 Operačný systém

Zoznam distribúcií oficiálne kompatibilných s Raspberry Pi, prevzatý z oficiálnej stránky Raspberry Pi [18]:

- NOOBS – Jednoduchý operačný systém odporúčaný pre začiatočníkov. Obsahuje Raspbian a ponúka ďalšie operačné systémy na stiahnutie. [18]
- NOOBS Lite – Verzia OS NOOBS bez predinštalovaného Raspbianu. [18]
- Raspbian – Neoficiálny port Linuxového operačného systému Debian. [32]
- Snappy Ubuntu Core – jadro Linuxovej distribúcie Ubuntu pre vývojárov. [18]
- OpenELEC – Open Embedded Linux Entertainment Centre – Operačný systém postavený tak, aby na ňom bolo možné spustiť softvér Kodi. [33] Kodi je open source multimediálne centrum. [34]
- OSMC – Open Source Media Centre – Open source multimediálny prehrávač založený na projekte Kodi. [35]
- Pidora – Linuxová distribúcia vychádzajúca z Linuxu Fedora. [36]
- RISC OS – Nelinuxový operačný systém dizajnovaný pre procesory ARM. Jeho vývoj začal už v roku 1989. [37]

### 3.10.1 Raspbian

Moja voľba operačného systému padla na Raspbian. Vybral som si ju kôli dobrým ohlasom, jednoduchosti používania [38], a veľkej komunite používateľov. Navyše je Debian operačný systém, s ktorým mám skúsenosti. V základnej konfigurácii Raspbian používa LXDE grafické rozhranie.

## 3.11 Databázový stroj

Databázový stroj je potrebný pre uchovávanie záznamu o predmetoch výpožičky, oprávnených užívateľoch a výpožičkách. Hlavne z dôvodu uchovania záznamu aj pri výpadku napájania som uprednostnil databázový stroj pred vlastnou implementáciou záznamu do súboru. Samozrejme, použitím bežne používanej knižnice sa zamedzí zbytočným chybám pri vlastnom návrhu a jeho implementácii.

V prvom rade je potrebné vybrať vhodný databázový stroj pre použitie na vstavanom systéme. Ďalej potrebujem zohľadniť špecifické požiadavky vzhľadom na operačný systém.

Existuje niekoľko typov databázových systémov. Uvažoval som tieto alternatívy o ktorých si myslím, že stoja za zmienku vzhľadom k používanosti a použiteľnosti.

- NoSQL – Systém nevyužíva štandardný koncept databáz, tzn. tabuľky pre ukladanie dát nie sú realizované poľom štruktúr ale odlišnými štruktúrami, ktoré sa dajú voliť v závislosti od povahy požiadavkov a dát. Pozitíva tohto prístupu sa najviac prejavujú pri spracovaní veľkých dát, čo nie je môj prípad.
- Objektová databáza – Hlavnou výhodou týchto systémov je uchovanie objektov z OO jazykov. Z tohto dôvodu som túto možnosť zamietol.
- Štandardné relačné databázy – Zvyčajne vyžadujú veľa pamäťového priestoru a požadujú pripojenie na internet. Na druhej strane ponúkajú veľa možností ktoré nie sú k dispozícii na menších systémoch.
- Databázový stroj pre embedded systémy / mobile database – Obvykle sú to štandardné relačné databázy prispôbené pre použitie v vstavaných systémoch. To znamená že zaberajú menej pamäťového miesta a nepodporujú niektoré optimalizácie, ktoré nie sú kritické pre vstavané systémy. Niektoré optimalizácie ako napríklad multivláknový prístup je obvykle možné deaktivovať a tým ušetriť ďalšie pamäťové miesto.
- Databázový stroj pre embedded systémy / real time – sú to databázy, ktoré garantujú odozvu do určitého času za daných podmienok. Navrhovaný systém neposudzujem ako RT systém a teda sa aspektom garancie časovej odozvy nebudem zaoberať kým to nebude užívateľsky nepríjemne dlhá odozva.

### 3.11.1 Databáza pre embedded systémy / mobile database

Tento typ databáz je prispôbený pre zariadenia s malou kapacitou pamäte a sú vhodné pre menšie databázy. Z týchto dôvodov som si vybral tento typ databáz.

Z ďalších potrieb som si ujasnil, že potrebujem:

- Interface pre jazyk C, ktorý som si vybral v časti 3.9.
- Databázový stroj bez potreby pripojenia na internet, pretože bude jednoduchšie a bezpečnejšie vytvoriť systém bez potreby pravidelnej sieťovej komunikácie.
- Databázový stroj, ktorý je určený pre operačný systém Linux. Dôvody pre tento výber sú uvedené v časti 3.10.

Po malom prieskume na internete som našiel bezpočet odporúčaní SQLite a nenašiel som žiadny iný databázový stroj, ktorý by bol na internete porovnateľne odporúčaný. Preto som sa rozhodol oboznámiť sa s týmto databázovým strojom a zistil som, že vyhovuje všetkým mojim požiadavkám.

## 3.12 SQLite

SQLite je knižnica, v ktorej je implementovaný databázový stroj.[39] Tento stroj je:

- Self-Contained[40] – potrebuje len minimálnu podporu knižníc alebo operačného systému.
- Serverless[41] – nepotrebuje vzdialený server k behu. Vystačí si so súborom umiestneným na disku. Tento súbor je navyše prenositeľný.
- Zero-Configuration[42] – nie je nutné nič nastavovať.
- Transactional[43] – všetky operácie z vonku vyzerajú ako:
  - atomické – V prípade prerušenia behu aplikácie sa transakcia vykoná celá, alebo vôbec.
  - konzistentné – Dáta v databáze budú stále spĺňať pravidlá, ktoré sú definované, napr. `not null`.
  - izolované – Pri paralelnom spustení transakcií bude výsledok rovnaký, ako keby boli spustené za sebou.
  - trvalé – Výsledok transakcie je uložený permanentne na non-volatilej pamäti.

Navyše, knižnica obsahujúca databázu má menej ako 500KiB a je použiteľná zadarmo za akýmkoľvek účelom, no na druhej strane ponúka aj platenú profesionálnu podporu.

Knižnica SQLite podporuje len štyri datové typy uložitelné do knižnice.

- **INTEGER** – Celé číslo zaberajúce maximálne osem bajtov.

- **REAL** – Desatinné číslo uložené v reprezentácii s pohyblivou datovou čiarkou. Ukladá sa do ôsmich bajtov.
- **NULL** – Datový typ pre hodnotu NULL.
- **TEXT** – Reťazec uložený v kódovaní UTF-8 alebo UTF-16.
- **BLOB** – Dáta uložené bezo zmeny tak, ako boli prebrané.

### 3.12.1 C/C++ API

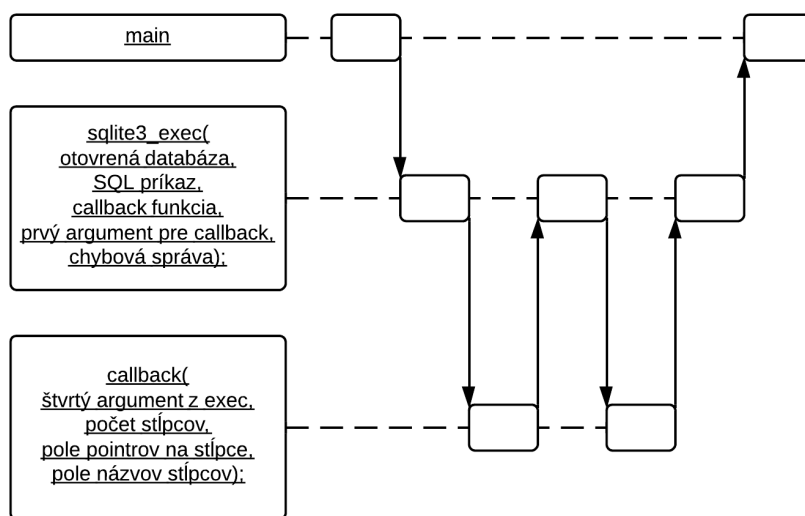
Rozhranie ponúka 207 funkcií[45]. Z nich je pre pochopenie základnej funkčnosti potrebných len zopár:

- **sqlite3\_open\_v2** – Otvorí alebo vytvorí súbor obsahujúci databázu a inicializuje sa.
- **sqlite3\_exec** – Vykoná príkaz a jeho výstup, každý riadok zvlášť, pošle do funkcie určenej tretím parametrom, nazývaný **callback**. Takže funkcia **callback** je volaná *počet\_riadkov\_výstupu* ráz. Táto funkcia je volaná len v prípade, keď sa do funkcie **sqlite3\_exec** vloží databázový dotaz. Nevolá sa ak je vložený príkaz **CREATE** ani **PRAGMA**. Ak sa táto funkcia preberie zo vzorových príkladov, tak vypíše na štandardný výstup všetky parametre v riadku. Poradie parametrov na výstupe je dané databázovým dotazom alebo ak dotaz neurčuje parametre v časti **SELECT** tak v poradí danom uložením v databázi. Funkciu **callback** je možné upraviť tak, aby namiesto výpisu dát ich rovno spracovávala. Lenže tento prístup je nepraktický viz obr. 3.5.
- **sqlite3\_close** – Zavrie súbor a uvoľní prostriedky.

Obr. 3.5 znázorňuje volanie funkcií pomocou funkcie **sqlite3\_exec** s druhým parametrom obsahujúcim databázový dotaz, ktorému zodpovedajú dva riadky v databáze, ktorá je určená prvým parametrom. Funkcia **callback** môže mať iný názov a môže volať ďalšie funkcie. Pomocou štvrtého argumentu funkcie **sqlite3\_exec** je možné z funkcie **main** do funkcie **callback** dostať štruktúru premenných potrebnú pre spracovanie spolu s výsledkom databázového dotazu.

Pre spríjemnenie práce s databázou a zrýchlenie behu programu je vhodné nahradiť funkciu **sqlite3\_exec** nasledujúcimi funkciami:

- **sqlite3\_prepare\_v2** – kompiluje SQL príkaz do bajtkódu uloženého vo výstupnom parametri s datovým typom **sqlite3\_stmt**. Okrem textového príkazu je vyžadovaný aj ukazateľ na otvorenú databázu. Textový reťazec je možné vložiť v kódovaní UTF-8 touto funkciou alebo UTF-16 použitím funkcie **sqlite3\_prepare16\_v2**. Alternatívy týchto funkcií bez **\_v2** je možné použiť, ale neodporúča sa to pri nových programoch.

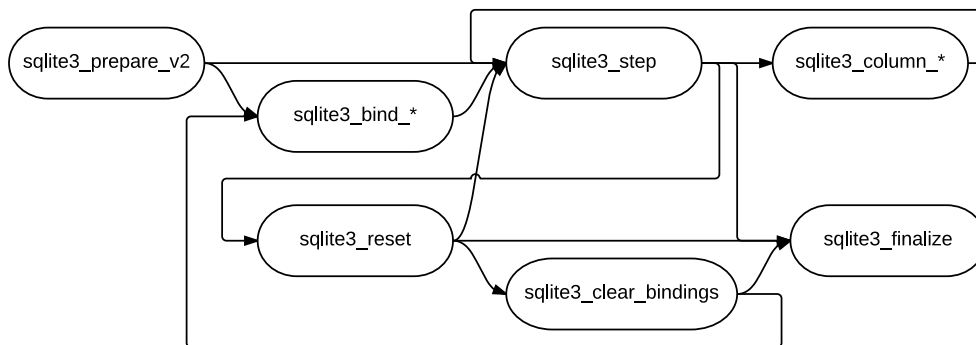


Obr. 3.5: Znáznorenie volania funkcií pri použití funkcie exec

V knižnici zostávajú kôli spätnej kompatibilitate. Do kompilovaného reťazca je možné vložiť aj špeciálne reťazce. Ich šablóny sú:

- ?
- ?NNN
- :VVV
- @VVV
- \$VVV

- `sqlite3_bind_datový_typ` – vkladá do bajtkódu pripraveného funkciou `sqlite3_prepare_v2` premenné za špeciálne reťazce spomínané vyššie. Pričom NNN znamená číselnú hodnotu a VVV reťazec. V prípade, že vkladaná premenná je datového typu `TEXT` alebo `BLOB`, funkcia umožňuje vytvárať hlbokú alebo plytkú kópiu. Funkcia podporuje vkladanie všetkých datových typov vrátane textu v kódovaní UTF-16.
- `sqlite3_step` – Vykoná príkaz pre jeden riadok a vráti informáciu o výsledku. Validná návratová hodnota môže byť `SQLITE_ROW` ak je možné získať ďalší riadok a `SQLITE_DONE` ak už boli odkrokované všetky riadky.
- `sqlite3_column_datový_typ` – Vracia hodnotu prečítanú z databázy zo stĺpca zadaného druhým argumentom. V prípade, že sa jedná o datový typ `TEXT` alebo `BLOB`, vracia ukazateľ na dáta. Tieto dáta budú



Obr. 3.6: Znáznorenie možných poradí volania funkcií

uveľnené ktoroukoľvek z funkcií `sqlite3_step`, `sqlite3_reset` alebo `sqlite3_finalize`. Funkcia nepožaduje aby datový typ dát uložených v databáze sa striktnie zhodoval s datovým typom funkcie `sqlite3_column_datový_typ`. V prípade, že sa datové typy líšia, dáta sa skonvertujú do takej miery ako to je možné.

- `sqlite3_reset` – Zresetuje SQL príkaz v premennej datového typu `sqlite3_stmt` pre opätovné použitie funkciami `sqlite3_step` atď. Táto funkcia neresetuje to, čo nastavila funkcia `sqlite3_bind_datový_typ`. O to sa stará nasledujúca funkcia.
- `sqlite3_clear_bindings` – Odstraňuje nabindované premenné pridelené funkciou `sqlite3_bind_datový_typ`.
- `sqlite3_finalize` – Uvoľňuje dáta datového typu `sqlite3_stmt`, ktoré alokovala funkcia `sqlite3_prepare_v2`. Funkciu `sqlite3_finalize` je možné volať nezávisle na tom, v ktorom bode životného cyklu sa príkaz nachádza.

Pre zjednodušenie prikladám obr. 3.6, kde je vyobrazené možné poradie volania funkcií. Funkciu `sqlite3_finalize` je možné volať kedykoľvek, no v diagrame uvádzam len prechody ktoré dávajú dobrý zmysel. Po funkcii `sqlite3_finalize` sa na SQL príkaz nesmie volať žiadna funkcia.

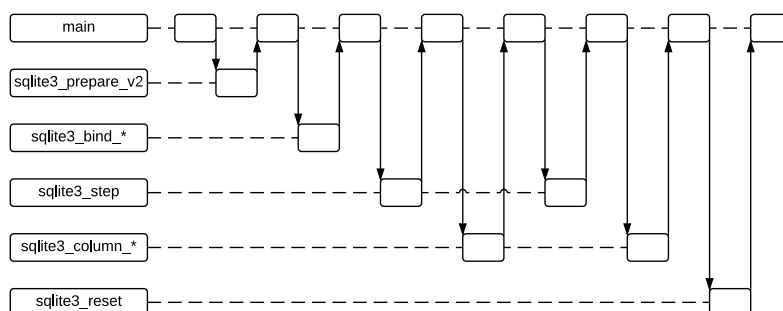
Obr. 3.7 odráža začiatok rovnakého scenáru ako obr. 3.5. Hlavný rozdiel medzi týmito dvoma prístupmi je ten, že v prvom má programátor lepšiu kontrolu nad spúšťaným dotazom a nemusí využívať funkciu `callback` a posielanie ďalších dát cez štvrtý argument funkcie `sqlite3_exec`.

Použitie tohoto spôsobu má aj tú výhodu, že pri opätovnom volaní rovnakého dotazu, s rovnako alebo rôzne nastavenými premennými, nie je potrebné opätovne volať funkciu `sqlite3_prepare_v2`. Čo zrychľuje beh programu.

Nevýhodou je zložitejšia postupnosť príkazov, ktorú je potrebné si osvojiť a potreba uchovávaní si predpripravených SQL príkazov.

### 3. ANALÝZA

---



Obr. 3.7: Znáznorenie volania funkcií pri použití funkcie prepare

### 3.13 Grafická knižnica QT

Pre vytvorenie grafického prostredia som zvolil knižnicu QT. Vzhľadom na to, že sa jedná o prototyp, neprikladám výberu grafickej knižnice veľký význam. Dôvodmi pre výber knižnice sú:

- knižnica QT je predinštalovaná v OS Raspbian
- je to jednoduchý nástroj pre vytvorenie grafického prostredia

Súčasťou knižnice QT je aj databázový modul (Qt SQL) a modul pre komunikáciu technológiou NFC (Qt NFC). Rozhodol som sa ich nepoužiť z dôvodu jednoduchšej zmeny grafickej knižnice v prípade potreby.



---

# Návrh

V tejto kapitole navrhнем ako zostaviť systém.

## 4.1 Návrh počítača

### 4.1.1 Hardvér

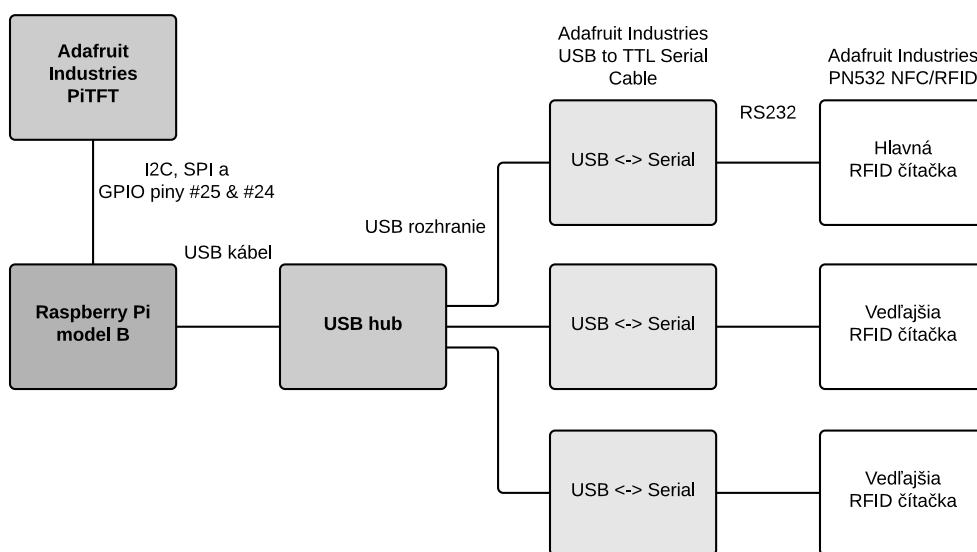
*Raspberry Pi model B* [19] bude spojené so vstupnovýstupnou perifériou *PiTFT 2.8" TFT 320x240 + Capacitive Touchscreen* [31] a zapúzdené v obale *PiTFT Enclosure for Raspberry Pi Model B* [46]. Tieto rozhodnutia sú odvodené v analýze 3.5, 3.6, 3.7, 3.8.

### 4.1.2 Softvér

Na počítači pobeží operačný systém Raspbian Wheezy. Na ňom bude nainštalovaný driver pre použitie vybranej periférie. V OS budú skompilované a nainštalované knižnice Libnfc a SQLite. Vzhľadom na to, že Operačný systém už obsahuje knižnicu QT, nebude ju potrebné inštalovať. Výber týchto softvérových komponentov je odôvodnený 3.10, 3.11, 3.12, 3.13.

## 4.2 Návrh zapojenia hardvéru

Výsledný prototyp systému s tromi RFID čítačkami bude vyzeráť podľa schémy na Obr. 4.1. Dôvod pre výber tejto možnosti som popísal v Analýze 3.4. Hlavná RFID čítačka slúži na komunikáciu s ISIC/ČVUT kartami užívateľov. Týmto spôsobom sa užívateľ autentifikuje. Vedľajšie RFID čítačky slúžia na identifikáciu predmetov pomocou RFID tagov. Medzi USB <-> Serial konvertorom pobeží komunikácia (TX/RX) na úrovni 3.3V. Napájanie je na úrovni 5V a maximálny prúd je 500mA.



Obr. 4.1: Schéma systému

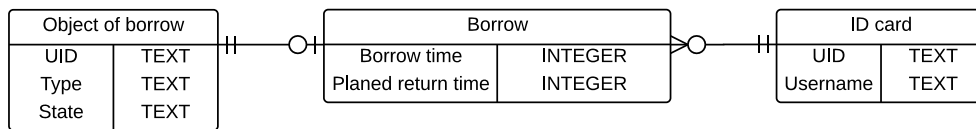
### 4.3 Databázový návrh

Model na obr. 4.2 znázorňuje uloženie dát v databáze. Tento model sa líši od štandardu, ktorý používa firma Oracle a ktorý sa používa na fakulte, preto som priložil legendu na obr. 4.3 vysvetľujúcu použité relácie. Pre detailnejší popis som nižšie uviedol zoznam vzťahov medzi tabuľkami. Model popisujem z ľava do prava:

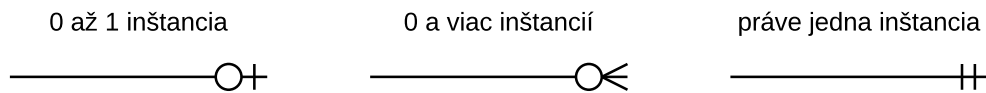
- Jeden riadok tabuľky `Passive_target` nemusí mať žiadnu, ale môže mať maximálne jednu asociáciu s riadkom tabuľky `Borrow`.
- Jeden riadok tabuľky `Borrow` má práve jednu asociáciu s riadkom tabuľky `Passive_target` a súčasne aj s jedným riadkom tabuľky `ID card`.
- Jeden riadok tabuľky `ID card` nemusí mať žiadnu, ale môže mať niekoľko asociácií s riadkami tabuľky `Borrow`.

Vránci tabuliek, ktoré tvoria databázu, môžu nastať nejasnosti vo voľbe dátových typov. Z toho dôvodu zdôvodním voľbu dátových typov atribútov a popíšem ich význam:

- `Passive_target` – UID a `ID card` – UID: Jednoznačný identifikátor RFID PICC podľa normy ISO/IEC 14443. Zvolil som dátový typ `TEXT`, pretože podľa ISO/IEC 14443-3, ako spomínam v časti 3.1.3, môže mať identifikátor karty až 10 bajtov, no dátový typ `SQLite` – `INTEGER` podporuje len osem. V databáze bude uložený ako 20-znakový reťazec.



Obr. 4.2: Databázový model



Obr. 4.3: Legenda k databázovému modelu

- **Passive\_target – Type:** Krátky popis predmetu. Zvolaný datový typ **TEXT**
- **Passive\_target – State:** Obsahuje informáciu, v akom stave sa predmet nachádza. Môže byť prítomný, požičaný alebo stratený. Pre jednoduchšiu implementáciu som zvolil datový typ **TEXT**, kde je napísané slovo v angličtine, ktoré stav vystihuje.
- **Borrow – Borrow time a Planed return time:** Čas výpôžičky a plánovaný návrat predmetu. Z dôvodu absencie dátového typu **TIMESTAMP**, spomínanej v kapitole 3.12, som zvolil datový typ **INTEGER**. Čas je ukladaný ako počet sekúnd od 1.1.1970.
- **ID card – Username:** Osem znakové užívateľské meno, ktoré je osobe pridelené fakultou. Zvolený datový typ **TEXT**.

Za kľúče v tabuľkách **Passive\_target** a **ID card** som zvolil **UID**, pretože je to unikátny identifikátor **PICC**, teda pre karty aj tagy.

Vzhľadom na to, že pôžička je existenčne závislá na požičiavanom predmete a užívateľovej identifikačnej karte, rozhodol som sa, že identifikátorom tejto tabuľky budú cudzie kľúče **UID** z oboch tabuliek. Z toho vyplýva obmedzenie, že v tabuľke môže byť len jeden záznam o vypožičaní konkrétneho predmetu konkrétnou osobou. Vzhľadom k tomu, ale hlavne vzhľadom k veľkosti databázy a rýchlosti k jej prístupu, som sa rozhodol históriu výpožičiek zaznamenávať len do logu. Neprehľadnosť sa môže javiť ako jedna z mála nevýhod tohto riešenia, avšak spätná rekonštrukcia logu na databázu je v prípade potreby možná.

## 4.4 Procedurálny návrh

Tu rozoberiem návrh algoritmov použitých v programe.

### 4.4.1 Návrh vlákien

Aplikácia bude používať dve vlákna. Jedno zabezpečí cyklické čítanie tagov a kariet v dosahu čítačiek a druhé odozvy užívateľského prostredia. Pre tieto potreby využijem vlákna z knižnice QT pretože aj vzolené užívateľské prostredie využíva knižnicu QT. Komunikácia medzi vláknami bude prebiehať pomocou správ. Nasleduje zoznam správ posielaných vláknom obsluhujúcim PCD a ich krátky popis:

- `sgiStopMainWindow` – vyšle signál hlavnému oknu, aby sa ukončilo.
- `sigSetUser` – nastaví meno užívateľa
- `sigUnsetUser` – zruší meno užívateľa
- `sigSetInfo` – nastaví informáciu na zobrazenie
- `sigUnsetInfo` – zruší informáciu na zobrazenie

Správa posielaná hlavným oknom:

- `sgiStop` – vyšle signál vláknou, ktoré obsluhuje PCD, aby sa ukončilo.

### 4.4.2 Návrh vlákna bežiaceho v pozadí

Vlákno má za účel zareagovať na pôžičku alebo vrátenie, zaznamenať to a vygenerovať nejaký výstup.

Program cyklicky prechádza všetky čítačky RFID kariet. Alternatívu s programom čakajúcim na signál od čítačky pri odobratí alebo pridaní PICC do dosahu som vylúčil z nasledujúcich troch dôvodov:

- Potreba napájania všetkých RFID čítačiek paralelne, čo môže byť problém pri vyššom počte RFID čítačiek.
- Riziko zahltenia pri odobratí viacerých prípravkoch naraz.
- V prípade straty signálu by nastala strata synchronizácie.

RFID čítačky rozdeľujem na hlavnú a vedľajšie. Hlavná RFID čítačka je zodpovedná za komunikáciu s identifikačnými kartami užívateľov a vedľajšie detekujú pridanie alebo odobratie predmetu z ich dosahu. Ak sa od posledného stavu zmenil počet PICC v dosahu práve dotazovanej čítačky, spustí sa jeden z nasledujúcich scenárov:

- V prípade odobratia predmetu z dosahu vedľajšej RFID čítačky sa spustí scenár na obr. B.2.
- V prípade vloženia predmetu do dosahu vedľajšej RFID čítačky sa spustí scenár na obr. B.1.

- V prípade priloženia identifikačnej karty na hlavnú RFID čítačku, mimo ostatné scenáre, sa zobrazí užívateľské meno a zoznam výpožičiek.

V oboch prípadoch sa preruší beh hlavného cyklu, kontrolujúceho zmeny na čítačkách.

Z priloženého diagramu na obr. B.2 vyplýva, že scenár výpožičky môže dopadnúť tromi spôsobmi. Každý podscenár predpokladá, že nebol naplnený žiadny z predchádzajúcich:

1. Ak systém nemá v databáze tag odobraného predmetu, systém na jeho odobratie nereaguje.
2. Inak, ak nie je užívateľova identifikačná karta v databáze alebo sa vôbec nepreukáže, zaznamená sa strata predmetu
3. Inak, ak je v databáze, výpožička sa zaznamená do databázy.

Podobne z diagramu na obr. B.1 vyplývajú tieto 3 podscenáre. Opäť, každý z nich predpokladá, že nebol naplnený žiadny z predchádzajúcich:

1. Ak sa tag pridaného predmetu nenachádza v databáze, systém požiada o jeho odobratie.
2. Inak, ak je v databáze tag pridaného predmetu vyskytuje ako stratený, tak sa označí, že nie je stratený.
3. V opačnom prípade sa výpožička odstráni a predmet sa označí ako možný na požičanie.

Scenár, ktorý sa spustí priložením identifikačnej karty na hlavnú RFID čítačku považujem za triviálny a nebudem ho ďalej rozoberať.

## 4.5 Grafický návrh

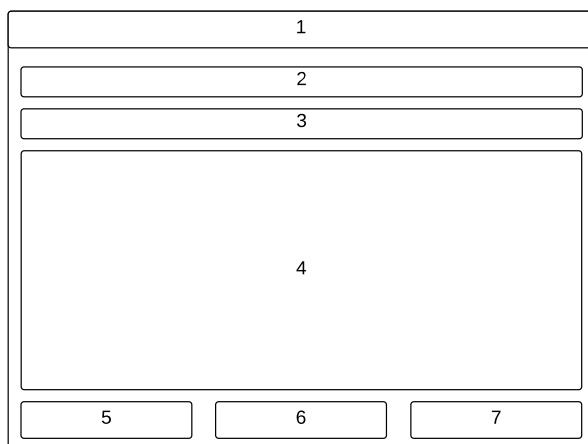
V tejto sekcii navrhнем grafické rozhranie ktoré zodpovedá potrebám a možnostiam knižnice QT. Grafické prostredie bude umožňovať zobrazíť meno užívateľa, ktorý sa identifikoval kartou a jeho výpožičky v zozname. Ďalej bude umožňovať vypnúť aplikáciu a dotázať sa na zoznam výpožičiek.

Grafické prostredie navrhnuté na obr. 4.4 splňuje vyššie spomenuté potreby. V očíslovaných častiach sa budú zobrazovať:

1. názov okna
2. meno identifikovaného užívateľa
3. oznam pre užívateľa
4. zoznam podľa požiadavky

#### 4. NÁVRH

---



Obr. 4.4: Grafický návrh

- 5. tlačidlo na zobrazenie všetkých výpožičiek
- 6. tlačidlo na zobrazenie výpožičiek užívateľa
- 7. tlačidlo pre ukončenie aplikácie

---

## Realizácia a testovanie

V tejto časti popíšem ako je systém realizovaný, výsledky testovania a dopady testovania na úpravu systému.

Pred nasadením na Raspberry Pi bol celý softvér vyvíjaný a priebežne testovaný na laptoppe DELL E6400 s vybranými parametrami:

- Intel®Core™2 Duo CPU T9600 @ 2.80GHz
- Operačný systém Debian GNU/Linux 7.4 (Wheezy)

Vzhľadom na to, že Raspbian je neoficiálnou vetvou Debianu [], je tento operačný systém pravdepodobne najvhodnejším systémom na vývoj a testovanie.

### 5.1 Priebežné testovanie

Dopady priebežného testovania:

- V pôvodnom návrhu bolo vypínanie a znovuzapínanie každej PCD funkciami `nfc_open` a `nfc_initiator_init` pri každom dotaze na PCD. Tento postup sa ukázal ako neefektívny a bol nahradený jedným zapnutím a vypnutím každej čítačky za celú dobu behu programu. To viedlo aj k problému popísanom v nasledujúcom bode. V záujme zníženia spotreby som bložil funkciu `nfc_idle(nfc_device *)`, ktorá presunie PCD do neaktívneho módu.
- Vloženie funkcie `nfc_device_set_property_bool(nfc_device *, NP_ACTIVATE_FIELD, false)` za účelom vypnutia elektromagnetického poľa PND. Jeho vypnutím sa docieli vypnutie všetkých PICC v dosahu a pri následovnom zapnutí budú všetky v stave IDLE. Odtiaľ si ich funkcia `nfc_initiator_list_passive_targets` prelistuje a presunie do stavu HALT. Dôvodom pre tento postup bola vlastnosť funkcie `nfc_initiator_list_passive_targets`. Tá prechádza len tie PICC,

Tabuľka 5.1: Meranie elektrického prúdu medzi počítačom a USB hub(om)

	elektrický prúd [ <i>mA</i> ]		
počet PCD	1	2	3
kludový stav	105	164.7	225.6
komunikácia bez karty	157.6	226.4	278.1
komunikácia s kartou	205.3	252	308.6

ktoré sú v stave IDLE. Pri pokuse pridať k tejto funkcii funkcii `nfc_initiator_target_is_present`, som nenedosiahol očakávaný výsledok. Mojim úmyslom bolo zistiť, či PICC so známym UID, ktoré boli presunuté do stavu HALT, sa ešte nachádzajú v elektromagnetickom poli. Naviac vypínaním elektromagnetického poľa sa zníži spotreba neaktívnych PCD.

## 5.2 Merania

Cieľom merania bolo zistiť, koľko prúdu každé zariadenie spotrebuje a koľko zariadení je možné napojiť na jeden USB hub. V prípade separátneho napájania USB hub z rozvodnej siete toto obmedzenie odpadá.

Merania prebehli na USB kábli medzi počítačom a USB hub, presnejšie na drôte prenášajúcom napájanie  $V_{cc}$ . Meral som digitálnym multimetrom. Táto metóda merania odberu nie je presná no je dostačujúca pre splnenie vytýčeného cieľa.

V tabuľke 5.1 sú hodnoty nameraného elektromagnetického prúdu, medzi laptopom popísanom vyššie a USB hubom. V tabuľke 5.2 bol laptop nahradený počítačom Raspberry Pi. Tabuľka 5.1 narozdiel od tab. 5.2 ukazuje spotrebu prúdu aj pri viacerých čítačkách, kde už Raspberry Pi nemohlo poskytnúť viac prúdu. Vo všetkých meraniach nemá USB hub ani čítačky pripojené externé zdroje napájania. Po pripojení externých zdrojov sa znížila spotreba na zanedbateľnú úroveň. Políčka tabuliek označené ako n/a neboli merateľné. V prípade riadku 0 to vyplýva z jeho povahy. Na riadku 1 z obmedzenia na maximálne jednu PICC na PCD. Na riadku číslo 3 bol prúd požadovaný od Raspberry Pi príliš veľký, než aby bolo možné nadviazať komunikáciu.

## 5.3 Raspberry Pi

Na počítači Raspberry Pi je nainštalovaný operačný systém Raspbian GNU/Linux 7 (wheezy). Následne bol doinštalovaný softvér potrebný pre sfunkčnenie obrazovky, dodávaný výrobcom. Potom boli skompilované a nainštalované knižnice Libnfc verzia 1.7.1 a SQLite 3080803.



Tabuľka 5.2: Meranie elektrického prúdu medzi Raspberry Pi a USB hub(om)

počet PCD	elektrický prúd [ <i>mA</i> ]				
	v pasívnom stave	za behu programu	s jedným priloženým tagom	s dvomi priloženými tagmi	po dobehnutí programu
0	42.1	n/a	n/a	n/a	n/a
1	97.4	153	192	n/a	118
2	149.0	240	250	260	188
3	193.9	n/a	n/a	n/a	n/a

## 5.4 Testovanie

V tejto sekcii sú popísané parametre behu výsledného softvéru.

Softvér v aktívnej slučke bez vlozenej funkcie `sleep` alebo `usleep` spotrebuje 11% až 12% výpočetnej kapacity procesoru. Pritom spotrebuje v špičke 54114kB pamäti RAM. Odozva na priloženie a odobratie karty je voľným okom nemerateľná. Odozva na priloženia alebo odobratia predmetu sa pohybuje okolo pól sekundy. Takáto odozva nie je užívateľsky príjemná, no odozva predmetu nie je tak užívateľsky kritická ako odozva karty.



---

## Výhľad do budúcnosti

Tento prototyp systému zďaleka nie je dokonalý a dá vylepšiť niekoľkými spôsobmi. Za všetky spomeniem dve ktoré považujem za najdôležitejšie. Týkajú sa cieľov, ktoré podľa môjho názoru, neboli naplnené v dostatočnej miere. Konkrétne bezpečnosti autentifikácie a ceny pre bežné použitie.

1. Zvýšenie bezpečnosti autentifikácie na ktorú sa v tomto systéme používa iba UID. UID je určené na rozpoznanie kariet v prípade prítomnosti viacerých kariet v dosahu jednej čítačky. UID nie je vhodné na identifikáciu z toho dôvodu, že karta ho vyzerá pri prvom kontakte s akoukoľvek čítačkou. Lepším riešením je použitie symetrického šifrovania pre autentifikáciu. Nevýhodou tejto možnosti je okrem technologickej náročnosti aj problém získať kľúče od výrobcu čipových kariet.
2. Zmena topológie na inú alternatívu tak, aby bolo cenovo dostupné požívať výrazne väčší počet predmetov.



---

## Záver

V tejto práci som navrhol prototyp systému pre evidenciu prítomnosti sledovaných predmetov v obmedzenom rozsahu. Systém navyše dokáže detekovať odobranie požičiavaného predmetu z dosahu RFID čítačiek. Predmet je nutné predom označiť RFID tagom a zaniest do databázy systému. Na odobratie systém reaguje dotazom o autentifikáciu užívateľa pomocou RFID karty štandardu ISO/IEC 14443 A. Do tohto štandardu spadajú aj preukaz študenta ČVUT a preukaz ISIC. V prípade úspešného zaznamenania výpožičky túto skutočnosť systém zaznamená do databázy. Vrátenie predmetu do dosahu RFID čítačky systém detekuje a ukončí výpožičku. Pre komunikáciu počítača s RFID kartou, pomocou RFID čítačky, používam knižnicu libnfc.

Systém z dôvodov spomínaných v časti 4.3 neukladá históriu výpožičiek do databázy. História je možné rekonštruovať z generovaného logu.

Systém bude možné v budúcnosti rozšíriť o bezpečnú autentifikáciu užívateľov. Čo sa týka hardvérovej stránky systému, tú bude, pred zavedením systému do praxe, potrebné zmeniť tak, aby sa znížila závislosť počtu výpožičiavateľných predmetov na cene systému. Diskusii takýchto možností sa venujem 3.4 a 3.3.

Verím, že výsledky mojej práce uľahčia ďalší vývoj tohto systému alebo systémov jemu podobných.



---

## Literatúra

- [37] 3QD Developments Ltd. *RISC OS FAQs: The answers to commonly asked questions*. [online]. 29 Alma Road, Oversetts, South Derbyshire, Derbyshire, East Midlands, England, United Kingdom: [prístup 23. jún 2014] Dostupné z:  
[www.riscos.com/the\\_faqs/index.htm](http://www.riscos.com/the_faqs/index.htm)
- [29] Adafruit Industries. *Adafruit Blue&White 16x2 LCD+Keypad Kit for Raspberry Pi: PRODUCT ID: 1115*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/1115](http://www.adafruit.com/products/1115)
- [28] Adafruit Industries. *Adafruit RGB Negative 16x2 LCD+Keypad Kit for Raspberry Pi: PRODUCT ID: 1110*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/1110](http://www.adafruit.com/products/1110)
- [27] Adafruit Industries. *Adafruit RGB Positive 16x2 LCD+Keypad Kit for Raspberry Pi: PRODUCT ID: 1109*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/1109](http://www.adafruit.com/products/1109)
- [21] Adafruit Industries. *NTSC/PAL (Television) TFT Display - 1.5"Diagonal: PRODUCT ID: 910*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/910](http://www.adafruit.com/products/910)
- [22] Adafruit Industries. *NTSC/PAL (Television) TFT Display - 2.0"Diagonal: PRODUCT ID: 911*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/911](http://www.adafruit.com/products/911)
- [23] Adafruit Industries. *NTSC/PAL (Television) TFT Display - 2.5"Diagonal: PRODUCT ID: 912*. [online]. New York City, New York, United

- States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/912](http://www.adafruit.com/products/912)
- [24] Adafruit Industries. *NTSC/PAL (Television) TFT Display - 3.5"Diagonal: PRODUCT ID: 913*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/913](http://www.adafruit.com/products/913)
- [25] Adafruit Industries. *NTSC/PAL (Television) TFT Display - 4.3"Diagonal: PRODUCT ID: 946*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/946](http://www.adafruit.com/products/946)
- [26] Adafruit Industries. *NTSC/PAL (Television) TFT Display - 7"Diagonal: PRODUCT ID: 947*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/947](http://www.adafruit.com/products/947)
- [31] Adafruit Industries. *PiTFT 2.8"TFT 320x240 + Capacitive Touchscreen - Raspberry Pi Model B: PRODUCT ID: 1983*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/1983](http://www.adafruit.com/products/1983)
- [46] Adafruit Industries. *PiTFT Enclosure for Raspberry Pi Model B: PRODUCT ID: 1892*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/1892](http://www.adafruit.com/products/1892)
- [30] Adafruit Industries. *PiTFT - Assembled 320x240 2.8"TFT+Touchscreen for Raspberry Pi: PRODUCT ID: 1601*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/1601](http://www.adafruit.com/products/1601)
- [13] Adafruit Industries. *PN532 NFC/RFID controller breakout board - v1.3: PRODUCT ID: 364*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/products/364](http://www.adafruit.com/products/364)
- [19] Adafruit Industries. *Raspberry Pi Model B 512MB RAM: PRODUCT ID: 998*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014] Dostupné z:  
[www.adafruit.com/products/998](http://www.adafruit.com/products/998)
- [20] Adafruit Industries. *Raspberry Pi / Mini Displays*. [online]. New York City, New York, United States of America: [prístup 23. jún 2014]. Dostupné z:  
[www.adafruit.com/categories/160](http://www.adafruit.com/categories/160)



- 
- [1] Anthony V. Pugliese. *Electronic ticketing and reservation system and method*. [online]. Vynálezca: Joel R. Goheen. Prihl. 21.11.1994. MPT: G06Q50/00; G07B15/00; G06Q10/00. Čís. patentu: US5724520 (A). The United States Patents and Trademark Office [přístup 20. apríl 2014] Dostupné z:  
[www.google.com/patents/US5724520](http://www.google.com/patents/US5724520)
- [11] CAMPBELL W.P. *GEE and LORAN: RADAR NAVIGATIONAL SYSTEMS*. WORLD WAR II [online]. [přístup 23. jún 2014] Dostupné z:  
[www.rquirk.com/cdnradar/cor/chapter16.pdf](http://www.rquirk.com/cdnradar/cor/chapter16.pdf)
- [12] CONTY Romuald. *Main Page: Introduction*. [online]. 2013-06-13 [přístup 23. jún 2014] Dostupné z:  
[nfc-tools.org](http://nfc-tools.org)
- [9] ČSN ISO/IEC 14443. *Identifikační karty – Bezkontaktní karty s integrovanými obvody – Karty s vazbou na blízko*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2002. Třídící znak 369760.
- [14] DICOLA Tony. Embedded Linux Board Comparison: Overview. In: Adafruit Industries. Adafruit Learning System. [online]. New York City, New York, United States of America: 2014-05-06 [přístup 23. jún 2014] Dostupné z:  
[learn.adafruit.com/embedded-linux-board-comparison](http://learn.adafruit.com/embedded-linux-board-comparison)
- [10] Encyclopædia Britannica, Inc. *radar / electronics :: Ground-probing radar / Encyclopedia Britannica*. [online]. [přístup 23. jún 2014] Dostupné z:  
[www.britannica.com/EBchecked/topic/488278/radar/237374/Ground-probing-radar](http://www.britannica.com/EBchecked/topic/488278/radar/237374/Ground-probing-radar)
- [38] HAYWARD David. Raspberry Pi operating systems: 5 reviewed and rated. In: Future Publishing Limited. *TechRadar*. [online]. 2013-May-09 [přístup 23. jún 2014] Dostupné z:  
[www.techradar.com/news/software/operating-systems/raspberry-pi-operating-systems-5-reviewed-and-rated-1147941](http://www.techradar.com/news/software/operating-systems/raspberry-pi-operating-systems-5-reviewed-and-rated-1147941)
- [39] Hipp, Wyrick & Company, Inc. *About SQLite*. [online]. [přístup 23. jún 2014] Dostupné z:  
[www.sqlite.org/about.html](http://www.sqlite.org/about.html)
- [45] Hipp, Wyrick & Company, Inc. *List Of SQLite Functions*. [online]. [přístup 23. jún 2014] Dostupné z:  
[www.sqlite.org/c3ref/funclist.html](http://www.sqlite.org/c3ref/funclist.html)
- [40] Hipp, Wyrick & Company, Inc. *SQLite Is Self-Contained*. [online]. [přístup 23. jún 2014] Dostupné z:  
[www.sqlite.org/selfcontained.html](http://www.sqlite.org/selfcontained.html)

- [41] Hipp, Wyrick & Company, Inc. *SQLite Is Serverless*. [online]. [prístup 23. jún 2014] Dostupné z:  
[www.sqlite.org/serverless.html](http://www.sqlite.org/serverless.html)
- [43] Hipp, Wyrick & Company, Inc. *SQLite is Transactional*. [online]. [prístup 23. jún 2014] Dostupné z:  
[www.sqlite.org/transactional.html](http://www.sqlite.org/transactional.html)
- [42] Hipp, Wyrick & Company, Inc. *Zero-Configuration*. [online]. [prístup 23. jún 2014] Dostupné z:  
[www.sqlite.org/zeroconf.html](http://www.sqlite.org/zeroconf.html)
- [2] International Air Transport Association. *E-ticketing*. [online]. [prístup 20. apríl 2014] Dostupné z:  
[www.iata.org/whatwedo/stb/Pages/e-ticketing.aspx](http://www.iata.org/whatwedo/stb/Pages/e-ticketing.aspx)
- [6] Magistrát hl. m. Prahy. *Audit Opencard prokázal, že vynaložené náklady neodpovídají současnému stavu projektu*. [online]. [prístup 20. apríl 2014] Dostupné z:  
[opencard.praha.eu/jnp/cz/aktuality/archiv\\_zprav/rok\\_2009/audit\\_opencard\\_prokazal\\_ze\\_vynalozene.html](http://opencard.praha.eu/jnp/cz/aktuality/archiv_zprav/rok_2009/audit_opencard_prokazal_ze_vynalozene.html)
- [16] New IT Limited. *Cubieboard 2*. [online]. [prístup 23. jún 2014] Dostupné z:  
[www.newit.co.uk/shop/Cubieboard/Cubieboard2](http://www.newit.co.uk/shop/Cubieboard/Cubieboard2)
- [3] NXP Semiconductors. *NXP celebrates 20th anniversary of MIFARE products*. [online]. 2014-0128. [prístup 20. apríl 2014] Dostupné z:  
[www.mifare.net/en/aboutmifare/news/nxp-celebrates-20th-anniversary-mifare-products/](http://www.mifare.net/en/aboutmifare/news/nxp-celebrates-20th-anniversary-mifare-products/)
- [5] NXP Semiconductors. *Showcase - London: Easing travel in London's congested public transport network*. [online]. [prístup 20. apríl 2014] Dostupné z:  
[www.mifare.net/en/showcases/showcase-london/](http://www.mifare.net/en/showcases/showcase-london/)
- [4] Octopus Cards Limited. *Corporate Profile: Our History*. [online]. [prístup 20. apríl 2014] Dostupné z:  
[www.octopus.com.hk/about-us/corporate-profile/our-history/en/index.html](http://www.octopus.com.hk/about-us/corporate-profile/our-history/en/index.html)
- [33] OpenELEC. *What Is OpenELEC?* [online]. [prístup 23. jún 2014] Dostupné z:  
[openelec.tv/home/what-is-openelec](http://openelec.tv/home/what-is-openelec)
- [35] OSMC. *About*. [online]. [prístup 23. jún 2014] Dostupné z:  
[osmc.tv/about/](http://osmc.tv/about/)

- 
- [18] Raspberry Pi Foundation UK registered charity 1129409. *Downloads*. [online]. [prístup 23. jún 2014] Dostupné z: [www.raspberrypi.org/downloads/](http://www.raspberrypi.org/downloads/)
- [15] SECO USA Inc. *UDOO Dual Basic*. [online]. [prístup 23. jún 2014] Dostupné z: [shop.udoo.org/eu/product/udoo-dual-basic.html](http://shop.udoo.org/eu/product/udoo-dual-basic.html)
- [36] Seneca Centre for Development of Open Technology : *Pidora*. [online]. 2014-03-16 [prístup 23. jún 2014] Dostupné z: [zenit.senecac.on.ca/wiki/index.php/Pidora](http://zenit.senecac.on.ca/wiki/index.php/Pidora)
- [7] Smart Card Alliance. *Hong Kong Octopus Card*. [online]. [prístup 20. apríl 2014] Dostupné z: [www.smartcardalliance.org/resources/lib/Hong\\_Kong\\_Octopus\\_Card.pdf](http://www.smartcardalliance.org/resources/lib/Hong_Kong_Octopus_Card.pdf)
- [8] Sony Corporation. *ANNUAL REPORT 1994 - 1994-E*. [online]. [prístup 20. apríl 2014] Dostupné z: [www.sony.net/SonyInfo/IR/financial/ar/8ido18000005flqz-att/1994-E.pdf](http://www.sony.net/SonyInfo/IR/financial/ar/8ido18000005flqz-att/1994-E.pdf)
- [32] THOMPSON Michael. *RaspbianAbout - Raspbian*. [online]. [prístup 23. jún 2014] Dostupné z: [www.raspbian.org/RaspbianAbout](http://www.raspbian.org/RaspbianAbout)
- [17] UPTON Liz. TWO MILLION! In: Raspberry Pi Foundation UK registered charity 1129409. *Raspberry Pi*. [online]. 2013-11-17 [prístup 23. jún 2014] Dostupné z: [www.raspberrypi.org/two-million/](http://www.raspberrypi.org/two-million/)
- [34] XBMC Foundation. *About*. [online]. [prístup 23. jún 2014] Dostupné z: [kodi.tv/about/](http://kodi.tv/about/)



## Zoznam použitých skratiek

- 3D** Three-dimensional
- API** Application programming interface
- ARM** Acorn RISC Machine architecture
- ČVUT** České vysoké učení technické
- FIT** Fakulta informačných technológií
- GPIO** General-purpose input/output
- GPS** Global Positioning System
- I<sup>2</sup>C** Inter-Integrated Circuit
- IATA** Medzinárodné združenie leteckých dopravcov
- IEC** International Electrotechnical Commission
- IFDHaldler** Interface Driver Handler
- ISO** International Organization for Standardization
- LCD** Liquid-crystal display
- LED** Light-emitting diode
- LORAN** LOng RAnge Navigation
- MIFARE** MIkron FARE Collection System
- NFC** Near Field Communication

## A. ZOZNAM POUŽITÝCH SKRATIEK

---

**OO** Object Oriented

**OS** Operating System

**PCD** Proximity Coupling Device

**PICC** Proximity Integrated Circuit Card

**RFID** Radio-frequency identification

**SDK** Software development kit

**SPI** Serial Peripheral Interface Bus

**SQL** Structured Query Language

**UCS** Universal Character Set

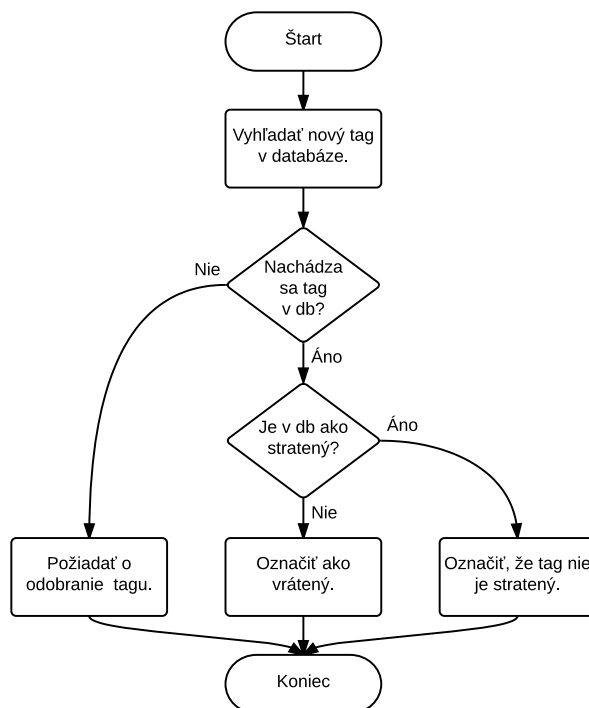
**UID** Unique Identifier

**USB** Universal Serial Bus

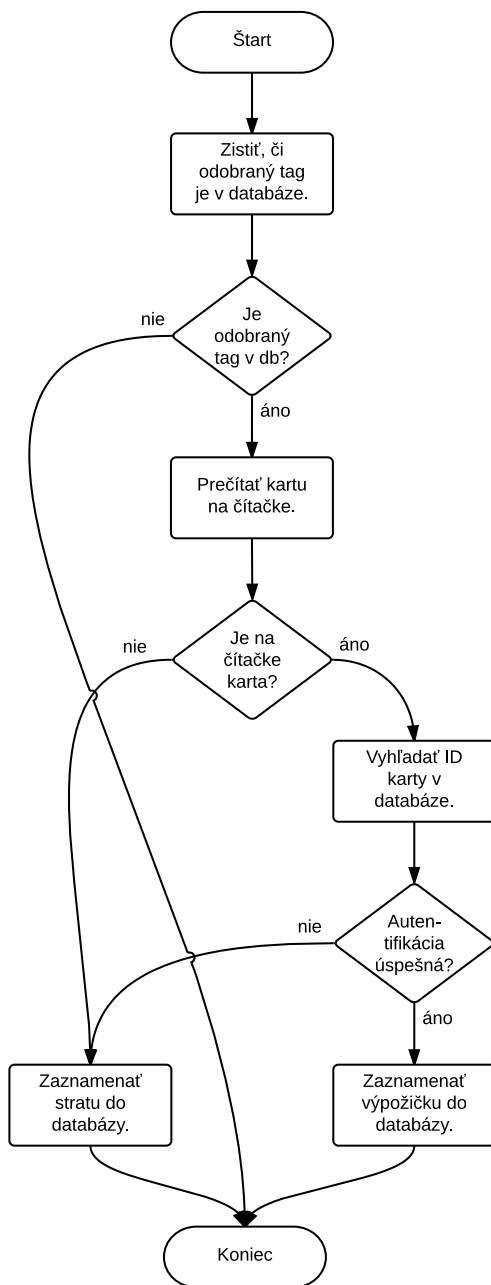
**UTF-16** UCS Transformation Format—16-bit

**UTF-8** UCS Transformation Format—8-bit

# Diagramy



Obr. B.1: Algoritmus vrátenia



Obr. B.2: Algoritmus vypožičania



## **Obsah priloženého CD**

## C. OBSAH PRILOŽENÉHO CD

---

readme.txt	.....	stručný popis obsahu CD a krátka užívateľská príručka
src	.....	zdrojové kódy implementácie
_	MainWindow.h	.....
_	MainWindow.cpp	.....
_	BackendThread.h	.....
_	BackendThread.cpp	.....
_	database.h	.....
_	database.cpp	.....
_	nearFiendCommunication.h	.....
_	nearFiendCommunication.cpp	.....
_	main.cpp	.....
_	html	..... dokumentácia
text	.....	text práce
_	source	..... zdrojové súbory k $\text{\LaTeX}$
_	pic	..... obrázky pre kompiláciu
_	algoritmus_vratenia.pdf	.....
_	algoritmus_vypozicania.pdf	.....
_	alt1.png	.....
_	alt2.png	.....
_	alt3.png	.....
_	communication_PICC_and_PCD_example.png	.....
_	DBmodelERD.pdf	.....
_	exec.png	.....
_	graf_navrh.pdf	.....
_	legenda-crop.pdf	.....
_	prepare.pdf	.....
_	prepare2.pdf	.....
_	schema_sys.pdf	.....
_	analyza.tex	.....
_	BP_Gula_Peter_2014.tex	..... hlavný $\text{\LaTeX}$ pre prácu
_	ciel.tex	.....
_	diagramy.tex	.....
_	FITthesis.cls	..... šablóna pre $\text{\LaTeX}$
_	navrh.tex	.....
_	realizacia.tex	.....
_	uvod.tex	.....
_	vyhlad.tex	.....
_	cvut-logo-bw.pdf	..... logo ČVUT
_	BP_Gula_Peter_2014.pdf	..... text práce vo formáte PDF