

Česká zemědělská univerzita v Praze

Technická fakulta

Laboratoř výpočetních aplikací



Bakalářská práce

Uživatelský interface pro domácí SMART zařízení

Tomáš Krlička

© 2021 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Technická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Tomáš Krlička

Zemědělské inženýrství

Informační a řídicí technika v agropotravinářském
komplexu

Název práce

Uživatelský interface pro domácí SMART zařízení

Název anglicky

User interface for home SMART devices

Cíle práce

Cílem práce je navrhnout strukturu, logickou koncepci a navrhnout konkrétní řešení pro univerzální uživatelský interface domácích SMART zařízení (lednice, mrazák, TV,..). Výstupem práce bude i praktické otestování navržené koncepce na vybraném typu zařízení včetně mobilní aplikace.

Metodika

1. Cíl práce
2. Metodika
3. Smart zařízení
4. Rozbor možných technologií
5. Návrh struktury interface
6. Praktické ověření
7. Zhodnocení a závěr
8. Doporučení pro další vývoj

Doporučený rozsah práce

30 až 40 stran textu včetně obrázků, grafů a tabulek

Klíčová slova

smart zařízení, IoT, mobilní aplikace

Doporučené zdroje informací

COOK, S. *Domain-specific development with Visual Studio DSL tools.*

Upper Saddle River, NJ: Addison-Wesley, 2007. ISBN 9780321398208.

EDDELBUETTEL, D. *Seamless R and C++ integration with Rcpp.* New York:

Springer, 2013. ISBN 978-1-4614-6867-7.

GARDNER, J. W. – VARADAN, V. K. – AWADELKARIM, O. O. *Microsensors, MEMS, and smart devices.*

Chichester: Wiley, 2001. ISBN 0-471-86109-.

GOODWIN, S. *Smart home automation with Linux and raspberry Pi.* New York:

Apress, 2013. ISBN 978-1-4302-5887-2.

HAMMOND, J. S. – KEENEY, R. L. – RAIFFA, H. *Smart choices: a practical guide to making better decisions.*

Boston, Mass.: Harvard Business School Press, 1999. ISBN 0875848575.

PRATA, S. – SOKOL, B. *Mistrovství v C++.* Brno: Computer Press, 2013. ISBN 978-80-251-3828-1.

Předběžný termín obhajoby

2020/2021 LS – TF

Vedoucí práce

Ing. Zdeněk Votruba, Ph.D.

Garantující pracoviště

Katedra technologických zařízení staveb

Konzultant

Ing. Jan Lešetický, Ph.D.

Elektronicky schváleno dne

3. 3. 2020

doc. Ing. Jan Malaťák,
Ph.D.

Vedoucí katedry

Elektronicky schváleno dne

10. 2. 2021

doc. Ing. Jiří Mašek, Ph.D.

Děkan

V Praze dne 12. 05. 2021

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci „Uživatelský interface pro domácí SMART zařízení“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne _____

Tomáš Krlička

Poděkování

Rád bych touto cestou poděkoval svému vedoucímu práce Ing. Zdeňkovi Votrubovi, Ph.D. za skvělé vedení mé bakalářské práce. Dále bych chtěl poděkovat své přítelkyni Kateřině, která při mně stálá po celou dobu mé bakalářské práce a pečlivě kontrolovala gramatiku každého odstavce a Vojtovy za věcné připomínky k mé práci. V neposlední řadě, bych chtěl poděkovat rodině za podporu při celém studiu a mým přátelům, bez kterých bych se tak daleko nedostal.

Uživatelský interface pro domácí SMART zařízení

Abstrakt

Práce se zabývá IoT technologií a její implementací v řešení „Smart“. Na začátku je probrána technologie čárových kódů a automatizovanou identifikací. V další je popsána IoT technologie pro uvažované nasazení, včetně návrhu struktury příslušné databázi. V praktické části, je realizace systému zaměřena na „chytrou“ domácnost a automatizovanou identifikací čárového kódu s popisem možného budoucího vývoje a možností převedení na mobilní aplikaci.

Klíčová slova: smart zařízení, IoT, mobilní aplikace

User interface for home SMART devices

Abstract

The work deals with IoT technology and its implementation in the "Smart" solution. At the beginning, bar code technology and automated identification are discussed. The following describes the IoT technology for the intended deployment, including the design of the structure of the relevant database. In the practical part, the implementation of the system is focused on a "smart" household and automated identification of a barcode with a description of possible future developments and the possibility of transfer to a mobile application.

Keywords: smart device, IoT, mobile application

Obsah

1	ÚVOD	1
2	CÍLE PRÁCE	2
2.1	Metodika	2
3	NÁVRH ŘEŠENÍ	3
4	ČÁROVÉ KÓDY	4
4.1	Počátky čárových kódů	5
4.2	Konstrukce čárových kódů	5
4.3	Technologie snímání a přenos kódů	7
4.3.1	Vstupní modul.....	7
4.3.2	Modul pro elektrické zpracování signálu.....	9
4.3.3	Modul pro logické vyhodnocení signálu	10
4.3.4	Výstupní modul.....	11
4.3.5	Dekodéry.....	12
4.4	GS-1	12
4.5	EAN	13
4.5.1	EAN 8	13
4.5.2	EAN 13	14
4.6	Další druhy.....	15
4.6.1	QR kódy.....	15
4.6.2	RGB QR Code	16
4.6.3	RFID	17
5	DATABÁZE	19
5.1	Relace.....	19
5.1.1	1:1	19
5.1.2	1:N	19
5.1.3	M:N.....	19
5.2	Databázové objekty.....	20
5.3	Databázové modely.....	20
5.3.1	Hierarchický model.....	20
5.3.2	Síťový model	21
5.3.3	Relační model	22
5.3.4	Objektová databáze.....	22
5.3.5	Objektově-relační databáze.....	23
6	INTERNET VĚCÍ	24
6.1	SMART řešení	26
7	VOLBA PLATFORMY	27

7.1	Arduino UNO	27
7.2	Raspberry Pi 3 Model B.....	27
8	PRAKTICKÁ ČÁST	29
8.1	Volba databáze.....	29
8.2	Identifikace čárového kódu.....	30
8.3	Popis databáze.....	30
8.4	Program.....	31
8.5	Budoucí vývoj.....	33
9	MOBILNÍ APLIKACE.....	34
10	ZÁVĚR	35
11	POUŽITÁ LITERATURA.....	36
	TABULKY.....	40

Seznam obrázků

Obr. 1 Návrh realizace řešení „chytré“ lednice [vlastní].....	3
Obr. 2 První čárový kód, „bull's eye“ [7].....	5
Obr. 3 Popis čárového kódu [1].....	6
Obr. 4 Vztah kontrastu [1].....	6
Obr. 5 Ukázka a důsledek mech. poškození kódu [1].....	7
Obr. 6 Schéma sériového modulu [1].....	8
Obr. 7 Polygonomické zrcátko [1]	8
Obr. 8 Schéma paralelního modulu [1]	8
Obr. 9 Ideální změna napětí příchozího signálu [1]	9
Obr. 10 Reální změna napětí příchozího signálu [1].....	9
Obr. 11 Ukázka zachování poměru šířky [1]	10
Obr. 12 Dekódovací algoritmus [1].....	10
Obr. 13 Schéma EAN8 [2]	14
Obr. 14 Schéma EAN13 [2]	14
Obr. 15 Schéma QR kódu [34].....	16
Obr. 16 RGB QR Code [18].....	16
Obr. 17 Hierarchický model [24]	21
Obr. 18 Síťový model [24]	21
Obr. 19 Relační model [24].....	22
Obr. 20 Zařízení připojené k internetu [35]	24
Obr. 21 Schéma IoT systému od sběru dat po podniknutí kroků [34]	25
Obr. 22 Arguino UNO [40]	27
Obr. 23 RaspberryPi 3 Modul B [42].....	28
Obr. 24 Popis databáze [vlastní].....	30
Obr. 25 Ukázka záznamu z databáze [vlastní]	31
Obr. 26 Vývojový diagram programu [vlastní].....	32
Obr. 27 Náhled seznamu potravin [vlastní].....	32

Seznam tabulek

Tab. 1 Ceník podle rozsahu kódu. [9]	13
Tab. 2 Ceník podle tržby firmy. [9].....	13
Tab. 3 Ceník podle zaměstnanců. [9]	13
Tab. 4 Typy a dosah vysílaných frekvencí pasivních RFID tagů [vlastní]	17
Tab. 5 Porovnání databází [vlastní].....	29
Tab. 6 Dekódovací tabulka A liché parity [1]	40
Tab. 7 Dekódovací tabulka B sudé parity [1].....	40
Tab. 8 Dekódovací tabulka C sudé parity [1].....	41
Tab. 9 Určení hodnoty na 13. pozici [1].....	41

Seznam zkratek

LED – Light-Emitting Diode

TTL – Tranzistorovo-tranzistorová logika

EAN – European Article Number

RFID – Radio Frequency Identification

UPC – Universal Product Code

GTIN – Global Trade Item Number

QR – Quick Response

RGB – Red-Green-Blue

USD – United States Dollar

EPC – Electronic Product Code

TID – Tag Identifier

DBMS – Database Management System

ID – Identifier

IoT – Internet of Things

WAN – Wide Area Network

IDE – Integrated Development Environment

USB – Universal Serial Bus

HDMI – High-Definition Multimedia Interface

GPIO – General-purpose input/output

OS – Operating systém

1 Úvod

Od počátku lidstva se lidé snažili kompenzovat své slabé stránky nástroji, které jim usnadňovaly práci. Pokud se lidem nechtělo vykonávat namáhavou fyzickou činnost, nechali vykonávat tyto úkony zvířata či stroje. Jelikož lenost je matka pokroku, lidé neustále vyhledávají způsoby, jak si všemožné procesy usnadnit.

Postupem času, když lidé začali mezi sebou stále více a více obchodovat, bylo nutné i v tomto ohledu přijít se snadným a efektivním způsobem identifikace produktů. Z tohoto důvodu byly vynalezeny první identifikátory a následně i čárové kódy, které slouží k automatické identifikaci v mnoha různých odvětvích. Za pomoci automatické identifikace je možno snadno a rychle určit, kde se daný produkt právě nachází. To v kombinaci s technologií Internetu věci dalo následně za vznik například prvním automatizovaným skladům.

Právě technologie Internetu věci se používá pro účely automatizace, jako příklad může být Průmysl 4.0, který automatizuje jednoduché a repetitivní procesy, načež je tedy nemusí vykonávat člověk. Dále se tato technologie uplatňuje i pro „chytrá“ řešení v domácnosti, kde se repetitivní úkony vykonávají sami a běh celé domácnosti provádí samotný systém.

2 Cíle práce

Cílem práce je analyzovat a ověřit možnosti pro automatickou identifikaci v konkrétních podmínkách chytré domácnosti spolu s napojením na Internet věcí. Tímto způsobem by mohl uživatel získat větší přehled, například o svých potravinách, a tím zamezit zbytečnému plýtvání nebo nadměrnému nakupování.

2.1 Metodika

Práce je rozdělena na dvě části. První část se věnuje teoretickému návrhu „chytré“ lednice, popisem čárových kódů a dalších způsobů automatizované identifikace. Následně jsou probrány databáze a je nastíněna technologie Internetu věcí. Druhá část se věnuje realizaci „chytré“ lednice, návrhu a tvorbě databáze, vytvoření programu na identifikaci produktů a vytvoření webové aplikace. Na závěr je probrán možný budoucí vývoj a možnost převést tento koncept na mobilní aplikaci.

3 Návrh řešení

Při návrhu řešení pro „chytrou“ lednici, která má za úkol uživateli zlepšit přehled o potravinách v ní, si musíme uvědomit, že nechceme uživateli nikterak znepříjemnit užívání daného systému. Proto je potřeba zanalyzovat děje, které provádí při vkládání, nebo vykládání potravin. Jelikož jsou tyto úkony velice primitivní, nemůžeme po uživateli chtít, aby je zbytečně komplikoval. Z toho vyplývá, že musíme naopak využít stávajících pohybů, zvolit optimální umístění scanneru a použít je tak pro náš systém.



Obr. 1 Návrh realizace řešení „chytré“ lednice [vlastní]

Protože se většina ledniček skládá z dvou samostatných chladících komor, je potřeba nainstalovat čtečku do každé komory zvlášť, aby uživatel nemusel zbytečně otevírat tu komoru, do které právě nekládá produkt, a tím zbytečně nenarušoval teplotu v nich. Díky tomuto uložení čteček, uživateli postačí pouze mírně pozměnit trajektorii pohybu při vkládání nebo odebírání produktu. Čtečka by následně byla propojena se zařízením, které bude uložené, jak výrobce uzná za vhodné a bude produkt identifikovat a následně ho zapisovat do databáze.

4 Čárové kódy

V dnešní době jsme v obklopení čárových kódů, které nalezneme téměř na každém rohu. Tyto kódy nám slouží k jasnému a rychlému identifikování, které je v přírodě naprosto kritické, na rozdíl od říše zvířat, lidé tyto kódy nevyužíváme k identifikaci nebezpečí, ale k identifikaci produktů, za účelem urychlení procesů a bezchybnému určení produktu. Jelikož je jejich výroba a případná implementace velice snadná, nic nebránilo tomu, aby se rozšířily takovým způsobem, jakým je tomu dnes.

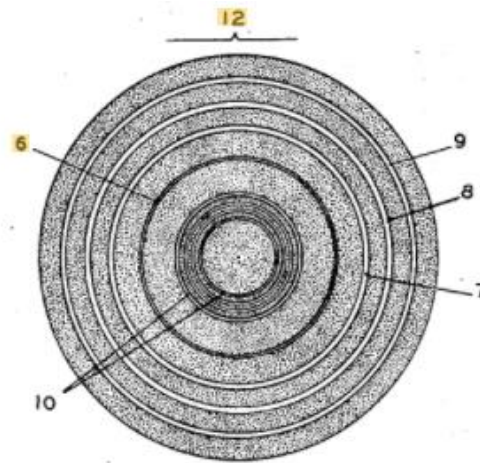
S čárovými kódy se v největším rozsahu setkáme v supermarketech, kde zastávají klíčovou roli. Jsou totiž využívány k rychlé identifikaci produktů, které zákazník kupuje a tím je supermarketu i zákazníkovi ušetřen čas při nákupu u pokladny, kde nemusí obsluha zdlouhavě identifikovat jednotlivé položky a urychlí se tak prodej. Dále jsou tyto informace, které jsou poskytovány při těchto úkonech, využívány pro vedení podrobných záznamů odbytu zboží a obchody tak mohou rychle reagovat na doplňování zásob jak v regálech samotného obchodu, tak i na skladu. Pokročilé systémy pak mohou na těchto shromážděných datech stavět různé strategie prodeje, nebo analyzovat trh.

Díky své schopnosti precizní identifikace, si své uplatnění našli i ve sféře farmacie a medicíny, kde jsou používány pro podání správných léků pacientům, nebo jejich samotné identifikaci. Tím se zamezí podání nesprávného léku a předejde se tak ublížení na zdraví pacientům, v případě i jejich smrti z důvodu zaměnění léku.

Dále tuto technologii využívají i přepravní společnosti, které tak zvyšují svoji efektivitu tím, že ztrácí minimální čas identifikací zboží a tím se proces přepravy urychlí. Jelikož se údaje z čárových kódů ukládají do systému, jde pak velice snadno zjistit, kde se konkrétní zboží momentálně nachází, a tak je lze velice snadno a precizně sledovat. [2]

4.1 Počátky čárových kódů

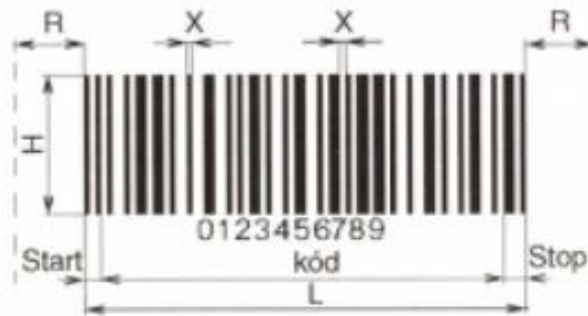
Čárové kódy jsou tu s námi již od druhé poloviny 20. století, za jejich vynálezem stojí Norman Joseph Woodland a jeho přítel Bernard Silver, kteří na žádost vedoucího supermarketu měli vytvořit systém pro kódování produktů. První kódy využívali fluorescentní inkoust a čtení bylo prováděno za pomoci ultrafialového světla. Komerčně se však neuchytili, jelikož byli velmi drahé a nespolehlivé. V pozdějším vývoji, byl vytvořen kruhový model kódu, který využíval rozdílné tloušťky čar a mezer jenž byl inspirován Morseovou abecedou. I přes to, že se kód dal číst, jakkoliv otočený, ani tentokrát se komerčně neuchytil, protože scanner pro čtení kódu využíval drahou 500wattovou žárovku. [4;5;6;7]



Obr. 2 První čárový kód, „bullseye“ [7]

4.2 Konstrukce čárových kódů

Čárové kódy se skládají ze sekvence mezer a čar, které jsou nositelem informací. Šířka mezer a čar, jejich seřazení, tak i celková délka kódu je specifická pro jednotlivé druhy kódu. Každý kód může mít jinou sekvenci Stop a Start, která označuje začátek a konec kódu. Některé mají i dělicí znak, který však nijak nenarušuje celistvost kódu. Nicméně stále se vyznačují charakteristickými znaky.



Obr. 3 Popis čárového kódu [1]

X – Šířka modulu. Jedná se o nejužší element kódu.

R – Světlé pásmo. Jedná se o prázdnou část kódu, aby při čtení nedocházelo k nepřesnostem nebo chybám při čtení Start/Stop sekvence. Na tuto část se nedává žádná grafika ani text. Pásmo by mělo být desetkrát širší než **X**, minimálně však 2,5mm.

L – Celková délka kódu.

Kód – Kódový řetězec

Start – Startovací sekvence

Stop – Ukončovací sekvence

Další parametr je hustota neboli density, která určuje hodnotu **X**. Čím je hustota větší, tím jsou vyšší požadavky na čtecí zařízení. Čím je hustota menší, tím bude kód zabírat větší plochu. Podle hustoty rozdělujeme kódy na tři následující skupiny.

- **High Density** (vysoká hustota)
- **Medium Density** (střední hustota)
- **Low Density** (nízká hustota)

Dalším klíčovým prvkem je poměr odrazu. Aby byl kód čitelný, musí být odraz pozadí rozdílný od odrazu čar. Tento kontrast je dán vztahem.

$$C = \frac{\text{Odras pozadí} - \text{Odras čárky}}{\text{Odras pozadí}} \geq 0,7$$

Obr. 4 Vztah kontrastu [1]

Pokud jsou dodrženy všechny podmínky, kód je velice spolehlivý, chyby při čtení se takřka nevyskytují. Nicméně může dojít k narušení, nebo poškození samotného kódu a v důsledku toho, může dojít k nesprávnému přečtení sekvence, jak je vidět na obrázku.



Obr. 5 Ukázka a důsledek mech. poškození kódu [1]

Za tímto účelem se do kódu přidává kontrolní znak, který nese informaci o předcházejících znacích a jeho hodnota může být vyjádřením operace typu XOR nebo MODULO. Porovnáním hodnot se odhalí případné chybné přečtení kódu. [1]

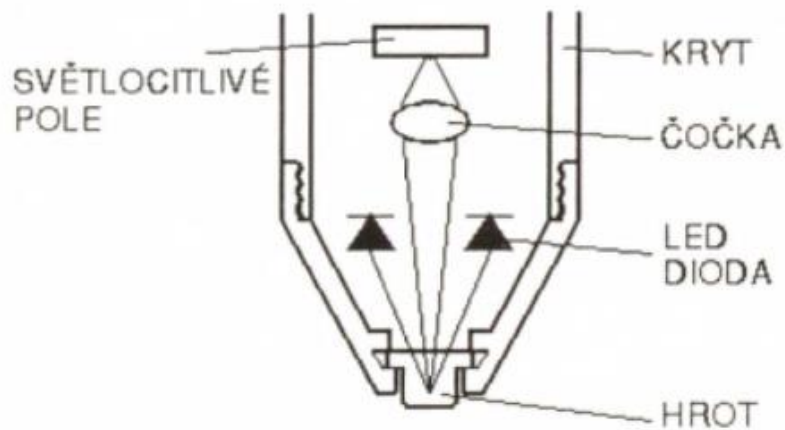
4.3 Technologie snímání a přenos kódů

Čtecí zařízení lze rozdělit na čtyři základní části, jsou nimi vstupní a výstupní moduly, dále pak modul pro elektrické zpracování signálu a modul pro logické vyhodnocení signálu. Zmíněné moduly jsou v nadcházejících kapitolách popsány.

4.3.1 Vstupní modul

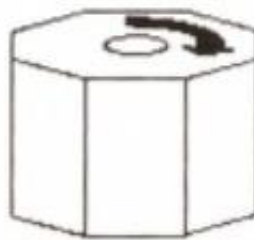
Jedná se o část, která jako první přijde do kontaktu s kódem. Kontakt je zprostředkován za pomoci elektromagnetického záření a nedochází tak k přímému fyzickému kontaktu. Elektromagnetické záření bývá vytvářeno v těchto modulech buď LED diodami nebo Helium-Neon laserem, které emitují světlo okolo 600nm vlnové délky. Dále se využívá i spektrum 900nm vlnové délky, a to ze dvou důvodů. Jelikož toto spektrum člověk nevidí, používá se pro utajení kódu a v místech, kde není možné snímat ve viditelném spektru, jako jsou například válcovny a slévárny. Modul má tedy za úkol emitovat elektromagnetické záření, za pomoci optických čoček, dojde ke sjednocení paprsku, následně je usměrněn zrcátkem a vyslán na cílené místo, kde nastane odraz, který zpátky přijme a nasměruje na další modul. Dle provedení, se moduly dále dělí na sériové a paralelní.

Sériové moduly, jsou využívány převážně u čtecích per. Modul je konstruován tak, že při čtení kódu, zaznamenává jednotlivé body a jaký je v těchto bodech odraz, tímto způsobem je možné číst libovolně dlouhý kód, nicméně zde dochází k dotyku mezi čtecím perem a kódem, proto je potřeba chránit čočku.

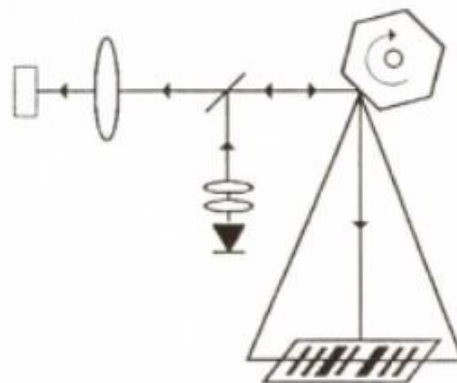


Obr. 6 Schéma sériového modulu [1]

Paralelní moduly se využívají například u ručních skenerů. Modul je tu konstruován bez paměti, jelikož využívá celkový odraz celé délky kódu najednou, a proto je schopen přečíst kódy pouze určité délky. Čtení celého kódu je dosaženo za pomoci polygonomického zrcátka, na která emituje laser. Jelikož zrcátko rotuje, je laser odrážen v různých úhlech a dopadá tak na celou šíři kódu.[1]



Obr. 7 Polygonomické zrcátko [1]



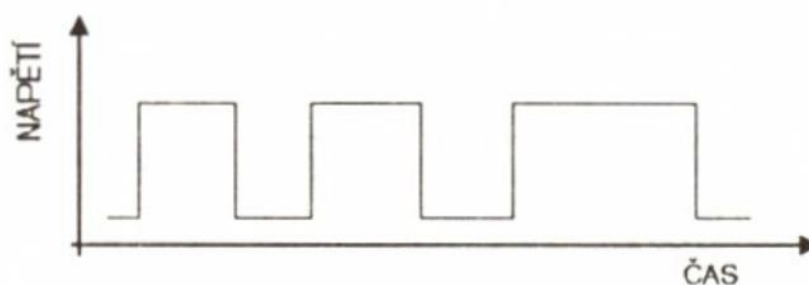
Obr. 8 Schéma paralelního modulu [1]

4.3.2 Modul pro elektrické zpracování signálu

Hlavní funkcí tohoto modulu, je převádět data ze vstupního modulu, nebo jejich hodnoty, z jednoho média, v tomto případě je to elektromagnetické záření, do média druhého, nejčastěji v podobě elektrického napětí nebo odporu. Toho bývá dosaženo, za pomoci fotodiody nebo fotorezistoru, které reagují na množství dopadajícího světla.

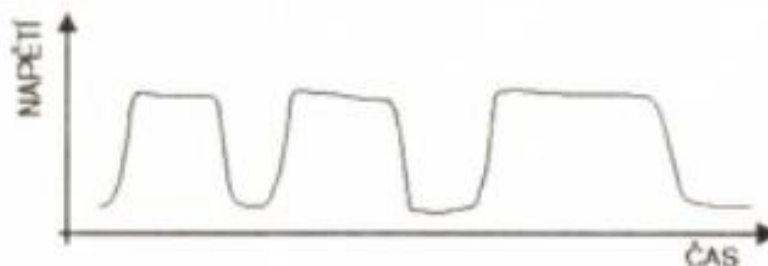
Jak již bylo zmíněno v předešlé kapitole, sériový způsob čtení kódu zaznamenává odraz v daném bodě, proto je nutné, aby tyto moduly měly paměťové buňky, do kterých se bude ukládat, zda došlo k odrazu v jednotlivých bodech. Jakmile čtení čárového kódu skončí, máme v paměťových buňkách zaznamenaný celý obraz kódu.

Při paralelním čtení kódu, máme informace o celém kódu najednou, který je také ukládán do paměti ve formě sekvencí 0 a 1. Vstupní signál však nemá ideální průběh, musí se před uložením hodnot tedy zpracovat.



Obr. 9 Ideální změna napětí příchozího signálu [1]

V běžném provozu dostáváme signál ze čtení podobný, jak je uvedeno níže na obrázku. V tomto případě je obtížnější přesně určit, kdy se jedná o plnohodnotnou změnu. Z tohoto důvodu jsou v modulu zabudovány ještě dočišťovací obvody, které signál „vyhladí“ pro lepší čtení.

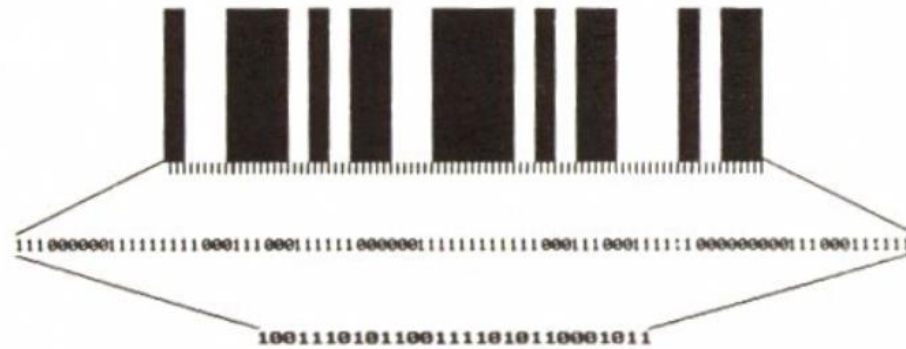


Obr. 10 Reálná změna napětí příchozího signálu [1]

Filtrační obvody fungují na vyhodnocování hodnoty signálu v určitém intervalu, pokud bude signál dosahovat hodnot v daném intervalu, považujeme signál za logickou 1, pokud hodnot nenabyla, považujeme to za logickou 0. Počet 0 a 1 je také brán s ohledem na čas, v jakém časovém úseku byli zaznamenány, a proto se jedna 1 může finálně nahradit větším počtem 1 a to stejné platí i s 0. Tímto způsobem uhladíme průběh signálu, hodnoty můžeme uložit do paměti a poslat na další zpracování dalšímu modulu.

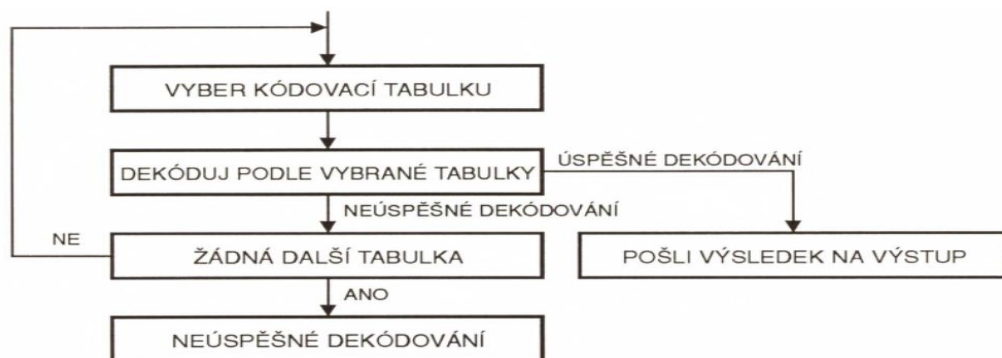
4.3.3 Modul pro logické vyhodnocení signálu

Tento modul vyhodnocuje data, která obdrží od předcházejícího modulu. Data jsou předávána nejnázne za pomoci paměti, jelikož se tento modul zajímá o jejich logickou hodnotu. Nejprve se musí dodržet poměr šířky sloupců k počtu uložených 1 a 0, to je patrné na následujícím obrázku.



Obr. 11 Ukázka zachování poměru šířky [1]

Dále se modul snaží vyhledat odpovídající kód z dekodovací tabulky, kde jednotlivá sekvence 0 a 1 odpovídá danému produktu. Pro dekodování různých kódů z různých tabulek, se využívá metoda pokus-omyl. Modul projíždí jednotlivé tabulky, dokud nenarazí na shodu, poté výsledek odešle. Pokud nebyla nalezena shoda, modul ukončí proces.



Obr. 12 Dekodovací algoritmus [1]

4.3.4 Výstupní modul

Jedná se o koncový modul celého zařízení, v podstatě se nejedná jen o výstupní modul, ale o vstupně-výstupní modul, jelikož může komunikovat i se zařízením, bývá označován i za komunikační zařízení. Hlavní funkcí tohoto modulu je převzít data od modulu pro logické vyhodnocení, které probíhá opět za pomoci paměti, kde je již uložena dekódovaná hodnota, která se předá na export pro uživatele. Pro komunikaci může modul využívat celou řadu sériových rozhraní. Ty nejdůležitější jsou níže popsány. [1]

RS232 je protokol, ale také i rozhraní, pracující na fyzické vrstvě a jeho přenos je asynchronní. Pro vyhodnocení logických hodnot využívá rozsah napětí $\pm 5 \text{ V}$, $\pm 10 \text{ V}$, $\pm 12 \text{ V}$ a $\pm 15 \text{ V}$. Nejčastěji se využívá rozsah $\pm 12 \text{ V}$, kde napětí $+12 \text{ V}$ odpovídá logické 0 a napětí -12 V zase logické 1. Pro komunikaci se využívá jeden kabel a komunikace probíhá pouze jedním směrem, proto se jedná o bezkolizní přenos a je za potřeby tří kabelů pro komunikaci mezi dvěma zařízeními. Jeden kabel, označený jako Tx, slouží pro odesílání signálu, druhý kabel, označený jako Rx, slouží pro přijímání signálu a poslední kabel je Ground, který slouží jako referenční napětí. Maximální vzdálenost propojených zařízení je 5 m se zachováním optimální rychlosti. [1;10;11]

RS422 je velice podobný RS232, hlavním rozdílem je rozdělení Tx a Rx na dva páry kroucené dvoulinky, a podle signálu se pak dodatečně značí jako páry A a B, což se využívá pro odfiltrování rušení signálu, jelikož jsou vzájemně posunuté a rušení se tedy neprojeví, Pro vyhodnocení logické 1 se využívá vztah $A-B < -200\text{mV}$, pro vyhodnocení logické 0 se využívá vztah $A-B > +200\text{mV}$. Další výhodou je možnost vysílat a zároveň přijímat data a oproti RS232 je možnost připojení více zařízení mezi sebou. Možná vzdálenost mezi propojenými zařízeními může dosahovat až 1500 m při rychlosti přenosu 100 kbit/s. [1;12]

RS485 je podobný již zmíněnému RS422, na rozdíl od něj, může propojit až 32 vysílačů a 32 přijímačů, to je možné, jelikož při komunikaci všechny neaktivní vysílače nezasahují do komunikace. Zařízení je možno propojit do vzdálenosti 1200 m s přenosovou rychlostí až 10Mbit/s. Vyhodnocování logických hodnot probíhá stejně jako u RS422. [1;12]

TTL neboli tranzistorovo-tranzistorová logika patří do bipolární technologie, která využívá bipolární tranzistory. Pracují v rozsahu 5 V, kde je logická 0 reprezentována vstupním napětím v rozsahu 0 V až 0,8 V a logická 1 v rozsahu 2 V až 5 V. Napětí 0,8 V

až 2 V je tzv. zakázané pásmo, které nemá definovanou hodnotu. Využívá se na střední a nízké hustoty integrace. [1;13]

4.3.5 Dekodéry

Jedná se o spojení dvou předešlých modulů, a to modulu pro logické vyhodnocení signálu a výstupního modulu. Podle konstrukce je můžeme rozdělit na tři typy.

Externí dekodér se nenachází ve vlastním šasi a jedná se o samostatnou jednotku. Má možnost být napojený na více různých vstupních modulů najednou s možností přiřadit jednotlivým modulům prioritu při zpracování. Dále disponují i větším počtem výstupů pro další zařízení.

Interní dekodéry jsou přímo zabudované na základní desce počítače a disponují menším počtem možností připojení vstupních zařízení než externí. Disponují klávesnicovým, nebo sériovým výstupem.

Integrované dekodéry jsou současně se vstupním modulem v jednom šasi. Disponují jedním vstupem i výstupem. Bývá implementována převážně u mobilních skenerů. [1]

4.4 GS-1

Jedná se o nadnárodní neziskovou organizaci, která se stará o udržování, provoz a vývoj ve sféře identifikace produktů po celém světě, dále také zavádí velké množství standardů v téže sféře. Společnost vznikla roku 1990 sjednocením Electronic Code Council of Canada, EAN International a Uniform Code Council. Roku 1995 vytváří první standard zaměřený na zdravotnictví. Roku 2004 je vytvořen standard pro stále více používanou technologii RFID (Radio Frequency Identification). [8]

Česká republika je také členem GS-1 a to od roku 1983, kdy vstoupila do celku EAN ještě jako Československo. Stalo se tak 19. členem společnosti a vůbec prvním členem z bývalého socialistického bloku a dostalo předvolbu 859.

V dnešní době má každý možnost si pořídit svůj vlastní bar kód, který bude označovat jeho produkt, GS-1 následně bude požadovat poplatek za registraci podle kritérií, které můžete vidět níže. [2]

Formát GCP	Kapacita rozsahu	Vstupní poplatek (v Kč)	Roční provozní poplatek (v Kč)
GCP-9	999	5 000	dle kategorie tržeb
GCP-8	9999	7 000	dle kategorie tržeb
GCP-7	99999	9 000	dle kategorie tržeb
GCP-11	9	0	1 500

Tab. 1 Ceník podle rozsahu kódu. [9]

Kategorie	Roční tržby (v milionech Kč)	Roční provozní poplatek (v Kč)
A	0 do 1	1040
B	nad 1 do 5	2080
C	nad 5 do 10	3110
...
O	nad 35 000	165 750

Tab. 2 Ceník podle tržby firmy. [9]

Kategorie	Počet zaměstnanců	Roční provozní poplatek (v Kč)
X	0 až 3	3 000
Y	4 až 10	7 000
Z	11 a více	12 000

Tab. 3 Ceník podle zaměstnanců. [9]

4.5 EAN

Jedná se o standard, který vznikl roku 1977, za účelem sjednotit kódovací systémy. Tento standard používá více než 60 zemí a je evropskou implementací UPC standardu. EAN je lineární, číselný, spojitý kód s pevně danou délkou. V podstatě se jedná o médium, na kterém se pomocí 0 a 1 zapisuje GTIN (Global Trade Item Number), neboli globální číslo obchodní položky, který je jedinečný pro každý produkt. Kód se skládá ze START a STOP znaku, který se skládá ze sekvence 1-0-1, dělicího znaku, který se skládá ze sekvence 0-1-0-1-0 a 8 nebo 13 čísel, záleží podle zvoleného standardu, kde je každé číslo reprezentováno sedmimístnou sekvencí nul a jedniček, pozice jednotlivých čísel jde zprava do leva. [1;14]

4.5.1 EAN 8

Jak již název napovídá, tento standard používá 8 číslic, jednotlivá čísla jsou rozdělena na půl dělicí sekvencí. Levá část čísel se dekóduje tabulkou A a jejich parita je vždy lichá, pravá část se dekóduje za pomoci tabulky C a jejich parita je vždy sudá. Dále čísla levé strany vždy začínají 0 a končí 1, u pravé části čísel je tomu přesně naopak. Pro

zkontrolování správného dekódování se využívá následující postup. Všechna čísla na sudých pozicích sečteme a vynásobíme třemi. Všechna čísla na lichých pozicích, od 3. pozice sečteme a sečteme se sudými. Výsledek přičteme k číslu na 1. pozici a výsledné číslo vydělíme modulem 10. Pokud se výsledek rovná 0, kód jsme úspěšně dešifrovali.

Pokud by některá firma žádala EAN8 pro značení svého produktu, je zapotřebí požádat GS-1 písemnou formou, která následně daný kód přidělí přímo danému produktu. [1]



Obr. 13 Schéma EAN8 [2]

4.5.2 EAN 13

U tohoto standardu používáme 13 čísel. Čísla na 1. až 6. pozici, se dekódují podle tabulky C a čísla na pozicích 7. až 12. se dekódují, podle parity, jejich znaků. Pokud je parita znaků lichá, používáme tabulku A, pokud je sudá, používáme tabulku B. Podle kombinace dekódování, mezi 12. až 7. pozicí, se následně určí hodnota na pozici 13., jelikož toto číslo se nenachází na čárovém kódu, pouze za START znakem se vypisuje její číselná hodnota. K ověření správného dekódování, se pak využívá stejný postup, jako u EAN8. [1;14]



Obr. 14 Schéma EAN13 [2]

4.6 Další druhy

Existuje celá řada lineárních čárových kódů, avšak, pokud by byla potřeba uložit větší množství dat do lineárního čárového kódu, byl by kód velice dlouhý, čtení takového kódu, by bylo časově náročné, stejně tak pořizovací náklady na tisk. Proto bylo nutné navrhnout nové formáty automatické identifikace. Některé z dalších řešení jsou popsány v následujících kapitolách. [15;16;17]

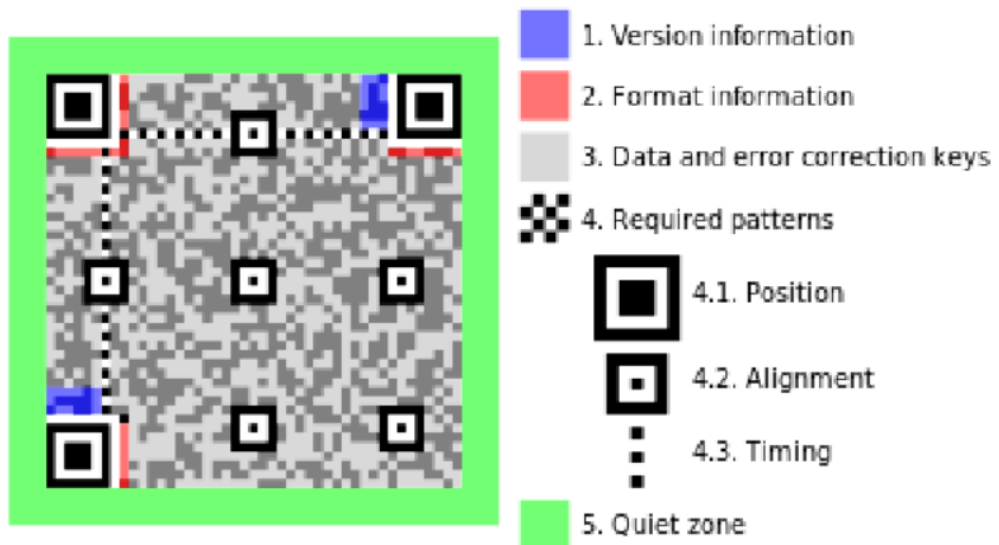
4.6.1 QR kódy

Zkratka QR je ze slova „quick response“, což v doslovném překladu znamená rychlá odpověď, jelikož zařízení dokážou velmi rychle rozkódovat a provést zadané úkony v kódu, jako je zobrazit danou webovou stránku, nebo provést platbu online. Jedná se o dvou dimenzionální kód, který je schopen přenášet větší množství dat na menší ploše než jiné jednodimenzionální kódy. Další výhodou je lepší čtení kódu za pomoci kamery místo specializovaného skeneru, díky tomu není potřeba dalšího hardwaru pro čtení kódu, ale bohatě postačí kamera například u mobilního telefonu. Velkou výhodou je také samotná struktura kódu, která je konstruována tak, aby byl kód čitelný i přes to, že dojde k většímu poškození kódu a velká část se tak dá odvodit od zbytku čitelného kódu. [15;16;17]

QR kód se skládá z několika základních prvků, které jsou níže popsány.

- **Position** – Nachází se v rozích a slouží k přesnému určení QR kódu a jeho rozměrů. Dále se po jejich okrajích ukládají bílá místa pro oddělení od informací.
- **Quiet zone** – Slouží pro oddělení celého kódu od okolí a pro vytvoření dostatečného kontrastu pro spolehlivé a korektní čtení.
- **Format information** – V těchto místech se ukládají potřebné informace pro určení masking vzoru, který slouží pro zakódování dat v kódu a úroveň korekce, která může být od 7 % do 30 %, to představuje, kolik plochy může být poškozené, aby byl kód stále čitelný.
- **Version information** – Představují označení verze QR kódu. Jednotlivé verze se od sebe liší především ve velikosti, kde verze 1 má rozměry 21x21 bodů, zatímco verze 40 má rozměry 1777x1777 a jedná se momentálně o největší QR kód.

- **Timing** – Jedná se o pruhy střídajících se černých a bílých bodů, které nám určují jak rozměry jednotlivých bodů, tak uložení sloupců a řádků v kódu.
- **Alignment** – Jsou čtverce rozmístěné po diagonálně po celém kódu a využívají se ke korekci čtení. Po vytištění na fyzické médium, může nastat vyblednutí kódu, což může vést ke špatnému čtení. Z tohoto důvodu jsou rozmístěny tyto prvky, aby došlo k normalizaci zkreslení barvy. [34]



Obr. 15 Schéma QR kódu [34]

4.6.2 RGB QR Code

Jedná se o třidimenzionální kód, se stejným vzhledem jako QR code. Tento kód se skládá ze 3 QR kódů, ale každý z nich je reprezentován jednou z barev RGB (red, green, blue). Pokud se jednotlivé body překryjí, výsledná barva bude kombinací již zmíněných barev. Takto překryté kódy nám tím pádem dovolí ukládat 3x více dat než běžný QR kód. [15;16;17]



Obr. 16 RGB QR Code [18]

4.6.3 RFID

Jedná se o technologii automatické identifikace za pomoci rádiových frekvencí. Identifikace je zprostředkována za pomoci RFID čtečky a RFID tagu. RFID tagy se dělí na aktivní a pasivní.

Pasivní tag je zařízení skládající se z antény, kapacitatoru, čipu a paměti. Při čtení tagu, RFID čtečka začne vysílat rádiové vlny do okolí a pokud je v dostatečné vzdálenosti k tagu, tak energii z rádiových vln přijme přes anténu a nabije kapacitatory. Ty následně dodají energii čipu, který vyšle signál zpět čtečce. Díky absenci baterie a snadné konstrukci mají dlouhou životnost a jejich cena se pohybuje řádech korun. Další výhodou oproti čárovým kódům je paměť, která se dělí na 4 třídy – rezervovaná paměť, EPC paměť, TID paměť a uživatelská paměť.

Rezervovaná paměť slouží k uložení kill a přístupového hesla. Kill heslo slouží pro deaktivaci tagu a přístupové heslo se využívá pro omezení přístupu k citlivým datům v tagu. Paměti jsou konstruovány o velikostí 64bitů, to znamená 32bitů pro každé heslo.

EPC paměť neboli Electronic Product Code paměť slouží k definování daného produktu. Paměti jsou realizovány o velikostech 64, 96 a 128 bitů. Pro snížení nákladů je však nejvíce využívána velikost 64 bitů.

TID paměť neboli Tag Identifier, je označení tagu od výrobce o libovolné délce, které si volí sám výrobce a nelze ho měnit.

Uživatelská paměť slouží k dodatečným informacím o produktu. Velikost paměti se pohybuje od 512bitů až do 8 tisíc bytů.

Pasivní RFID tagy vysílají na frekvencích, které jsou uvedeny v následující tabulce.

Frekvence	Název	Dosah
125KHz / 134KHz	Low Frequency	1–10 cm
13,56MHz	High Frequency	1 cm – 1 m
865 - 960MHz	Ultra High Frequency	1–30+ m

Tab. 4 Typy a dosah vysílaných frekvencí pasivních RFID tagů [vlastní]

Aktivní tag je podobný pasivnímu, na rozdíl od něj však disponuje ještě baterií, která vydrží provozovat tag 3-5 let. Dále se tyto tagy dělí na „Beacon“, který nečeká na aktivaci od čtečky v pravidelných intervalech, z pravidla 3-5 sekund, vysílá svoje údaje (například polohu) a „Transponders“, který komunikuje až po iniciaci od čtečky, ale pro komunikaci využívá energii z baterie. Tento druh tagů se využívá ke komunikaci na větší vzdálenost (okolo 100 m) a v odvětvích jako je například průmysl. [19;20;21;22]

5 Databáze

Databáze slouží k ukládání dat ve strukturované formě, které se ukládají v elektronické podobě. Když mluvíme o databázi, máme na mysli celek, který se však skládá ze dvou částí, ze samotných dat, což jsou údaje, které popisují danou skutečnost a pouze v kontextu nám dají nějaké informace a DBMS (Database Management System), v českém znění jako Systém Řízení Báze Dat (SŘBD), který pracuje se samotnými daty jako například, jak budou uložena, jakého budou typu a jaké operace budeme nad nimi provádět, například vkládání nového záznamu, odstranění daného záznamu nebo aktualizování současného záznamu. Mezi nepoužívanější SŘBD patří například Oracle, MS SQL Server, nebo MySQL. [23;24]

5.1 Relace

Jedná se o vazbu mezi tabulkami nebo záznamy, které rozdělujeme na několik základních druhů. Relace jsou důležité, pokud chceme pracovat s více než jednou tabulkou nebo záznamem, v takovém případě využíváme tzv. „primární klíč“, který slouží k jednoznačnému odkazu na daný záznam v dané tabulce. Jedná se o číslo, které je pro každý záznam jedinečný a v tabulkách se označuje jako „ID“. Druhy relací jsou níže popsány.

5.1.1 1:1

Jedná se o vztah, kde právě jeden záznam v jedné tabulce, odpovídá právě jednomu záznamu v druhé tabulce a naopak. Tento vztah se využívá pro rozdělení velkých tabulek, popřípadě z bezpečnostního hlediska, pokud chceme mít záznam oddělen od hlavní tabulky.

5.1.2 1:N

Vztah jeden-ku-mnoha představuje vztah jednoho záznamu v jedné tabulce, který odpovídá více záznamům ve více tabulkách. Tento vztah je jeden z nejčastějších.

5.1.3 M:N

Vztah mnoho-ku-mnoha je vztah více záznamů, který odpovídá více záznamům ve více tabulkách a pro jeho realizaci využíváme dva vztahy 1:N, které jsou propojené přes spojovací tabulku. [25;26]

5.2 Databázové objekty

U pokročilejších databází se můžeme setkat s různými druhy objektů, které jsou níže popsány.

- **Trigger** – Jedná se o spouštěč, který provede úkony nad daty, pokud se splní určitá podmínka, jako je např. změna dat nebo vymazání údajů.
- **Event** – Procedura, která se provádí opakovaně v určitý datum nebo po uplynutí daného časového intervalu. Slouží především k údržbě databáze, promazání nepotřebných a dočasných dat, popřípadě pro kontrolu integrity databáze.
- **Procedura** – Je kus kódu, který provede danou sekvenci příkazů, zadanou tvůrcem kódu.
- **Sestavy** – Umožňuje vytvářet graficky přívětivou formou výstupní data, ideální pro tisk nebo prezentaci.
- **Formuláře** – Umožňuje vytvářet grafické rozhraní pro uživatele za účelem snadného zadávání dat.
- **Partitioning** – Slouží k rozdělení velkých a obsáhlých tabulek na více uložišť.
- **Pohled** – Poskytuje data uživateli, které jsou přepsány z tabulky. [24;32]

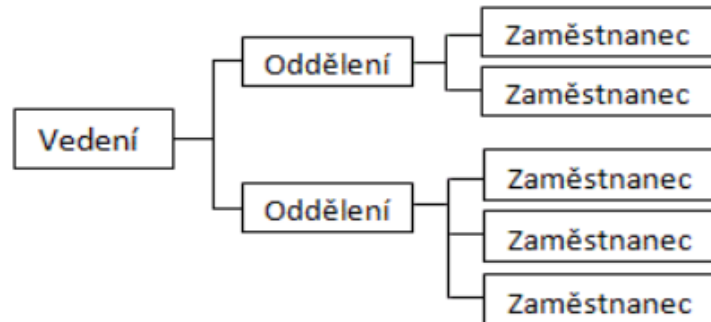
5.3 Databázové modely

Pod databázovým modelem se stará, jak se budou data ukládat, v jaké struktuře a jaké budou mezi nimi vazby. Dále jsou popsány některé z nich.

5.3.1 Hierarchický model

Jedná se o jeden z prvních komerčně úspěšných modelů, který se začal využívat. Graficky se zobrazuje jako obrácený strom. V modelu jsou data ve vztahu 1:n a jsou reprezentována termíny rodič a potomek. Rodič tedy vždy odkazuje na jednoho nebo více potomků, naopak potomek vždy odkazuje pouze na jednoho rodiče. V případě kořenu se jedná o počáteční záznam, který neodkazuje na žádného rodiče, jelikož je v nejvyšší vrstvě.

Výhody tohoto modelu jsou – snadné pochopení celkové struktury, vysoká rychlost navigace přes záznamy, které jsou prováděny přes ukazatele, nikoli přes klíče. Nevýhody jsou – malá flexibilita modelu a jakákoliv změna ve struktuře je velice obtížná, není zde možnost vytvářet vztah m:n a kvůli vztahu rodiče a potomka se data musí často ukládat duplicitně. [24;27;29]

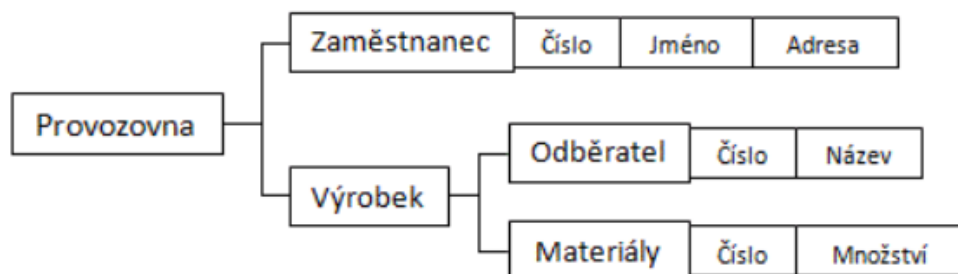


Obr. 17 Hierarchický model [24]

5.3.2 Síťový model

Tento model vychází z hierarchického modelu a doplňuje jej o možnost vytváření relací m:n, což znamená že potomek může mít nyní vztah s více rodiči, na rozdíl od hierarchického modelu a nemusíme tedy data ukládat redundantně.

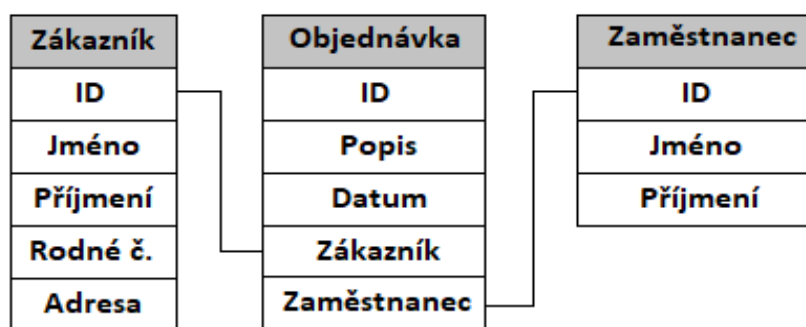
Výhody a nevýhody tohoto modelu jsou podobné jako u modelu hierarchického, nicméně oproti němu je zde možnost vytvářet více druhů relací a snazší přístup k datům. Na druhou stranu se jedná o komplexnější a složitější strukturu, která je flexibilnější, ale oproti hierarchickému modelu je zde více ukazatelů, které znesnadňují práci.[24;29]



Obr. 18 Síťový model [24]

5.3.3 Relační model

Jedná se o nejpoužívanější model v současnosti. Základním prvkem tohoto modelu jsou tabulky, do kterých se ukládají data v pevně dané formě. Tato forma je určována v jednotlivých sloupcích a nazývá se atribut, každý řádek v tabulce se pak nazývá záznam. Jednotlivé tabulky jsou provázány relacemi a je zde možnost provést i relaci m:n za pomoci relační tabulky. Samotné relace se neprovádějí za pomoci ukazatelů, ale za pomoci primárních a cizích klíčů, které jsou jedinečné pro danou tabulku a odkazují vždy na jeden daný záznam v tabulce. Se vznikem relačních tabulek je svázán i vznik SQL (Structured Query Language), který se stal standardem pro práci s daty v tomto modelu. Oproti již zmíněným modelům jsou data ukládána v přehledné formě, práce se záznamy je velmi jednoduchá a změna struktury jednotlivých tabulek není nikterak obtížná. [24;28;29]



Obr. 19 Relační model [24]

5.3.4 Objektová databáze

S příchodem objektového programování byla snaha přenést tuto myšlenku i na databáze, což dalo za vznik objektovým databázím, které kombinují prvky objektového programování a vlastnosti databází. Základem jsou objekty, které obsahují data i metody pro práci s nimi, odpadá tak nutnost používat mezistupeň k přístupu a práci s daty ve formě SQL dotazů. Pro vytváření objektů a metod postačí jakýkoliv objektově orientovaný programovací jazyk jako je C#, Java, Smalltalk nebo Python, tím je zároveň dosažena konzistence v rámci jednoho prostředí. K datům je za pomoci objektových identifikátorů, které odkazují na daný objekt v paměti, snadný a rychlý přístup, odpadá tak nutnost vyhledávat data přes ID (identifikátory), které jsou například u relačního modelu. [30;31]

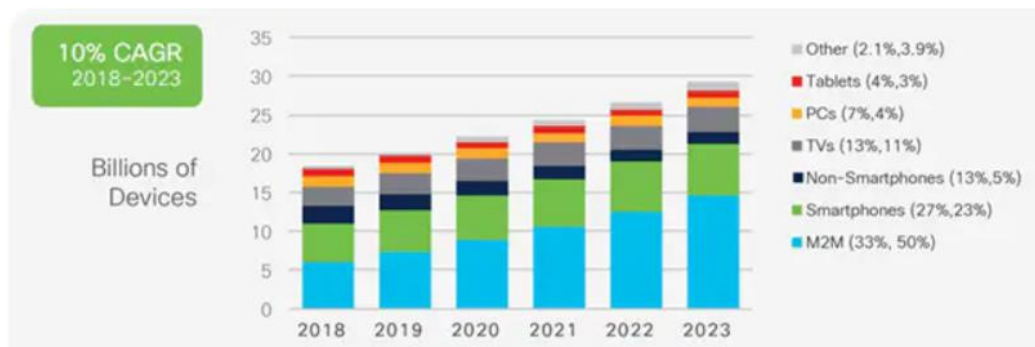
5.3.5 Objektově-relační databáze

Objektově-relační model vznikl jako kompromis mezi relační a objektovou databází. Je založen na relační technologii s možností vytvářet a využívat objekty. Objekt je však v databázi rozdělen na více částí, ty jsou následně uloženy do tabulek a propojené relacemi. Jelikož se jedná z části o relační databázi, standardem pro práci s daty se opět stává SQL s rozšířením o objektové prvky. [33]

Nejlépe vystihuje tento model následující citát: *„Použit tabulky pro uložení objektu je podobné, jako když řídíte auto a pak ho před uložením do garáže celé rozmontujete. Auto může být ráno zase smontováno, je však otázkou, je-li toto ten nejefektivnější způsob parkování.“* (E. Dysonová)

6 Internet věcí

Technologie Internetu věcí, označované také zkratkou IoT (Internet of Things), lze definovat jako elektronické zařízení s výpočetním výkonem, senzory nebo čidly s možností připojení drátově či bezdrátově k jiným zařízením v rámci sítě za účelem výměny dat, dalšího zpracování a reagovat na ně, bez nutnosti lidské interakce. Každé z těchto zařízení je jasně identifikovatelné a je schopné pracovat i samostatně. S příchodem 5G sítě, se tato technologie stále více využívá napříč všemi odvětvími, to prokázal i průzkum firmou Cisco, který byl proveden roku 2018 a prokazuje rapidní nárůst komunikace v rámci globální sítě. Zjistili, že 33 % veškeré komunikace je na bázi machine-to-machine neboli komunikace stroj se strojem a komunikace tohoto typu má do roku 2023 narůst přibližně na 50 %, kde největší zástupcem tohoto druhu komunikace zastupuje právě zařízení IoT. [34;35]

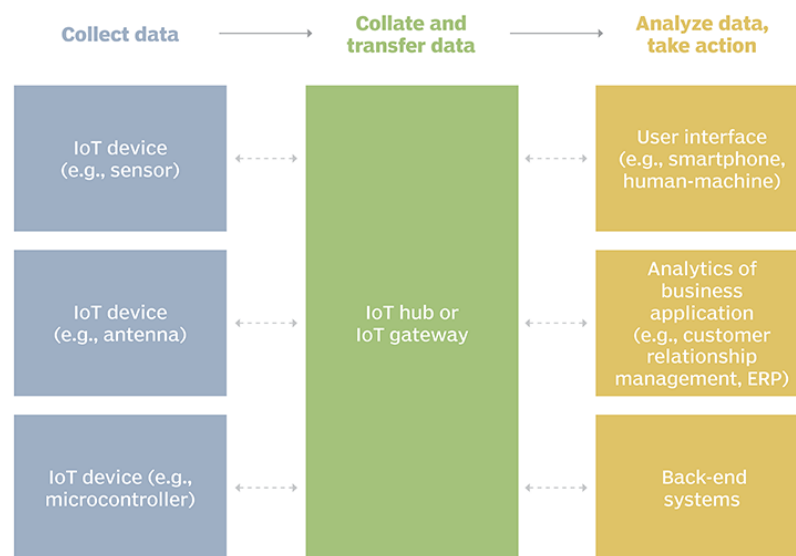


Obr. 20 Zařízení připojené k internetu [35]

Jelikož zařízení IoT pracuje s daty v reálném čase a komunikuje s centrálním cloudem, tak s přibývajícím počtem těchto zařízení roste i latence. Aby se latence snížila, využívá se na sběr dat, kontrolní funkce a ukládání edge computing, který rozkládá výpočetní výkon na koncová zařízení a zároveň probíhá pouze ta nejnutnější komunikace se serverem. Jedná se o zařízení, která považujeme za koncová v síti a zároveň jsou součástí cloudové infrastruktury. Takové zařízení se také označuje jako IoT gateway a je využíváno i jako bezpečnostní prvek, usnadňuje komunikaci se staršími zařízeními, a dokonce i na vizualizaci dat. [36;37]

Výhodou této technologie je snadný a rychlý přístup k datům z jakéhokoliv zařízení, lepší propojení a monitoring jednotlivých zařízení mezi sebou a automatizaci úkonů, které zlepšují kvalitu a snižují náklady. Nevýhodou je, v případě velkého množství zařízení, sdílené velké množství dat, která zvyšují potenciál útok na síť a odcizení těchto mnohdy citlivých dat. V případě více druhů zařízení od jiných výrobců, může být i problematická komptabilita mezi nimi.

Example of an IoT system



Obr. 21 Schéma IoT systému od sběru dat po podniknutí kroků [34]

Nicméně i přes absenci pevně daných standardů, je zde pár začínajících, které jsou níže popsány.

- **6LoWPAN** – Umožňuje zařízením s nízkou spotřebou energie komunikovat v rámci internetu.
- **ZigBee** – Jedná se o bezdrátovou, nízkoenergetickou komunikační technologii, která se převážně používá v průmyslu.
- **LoRaWAN** – Je technologie, která podporuje velké množství připojených zařízení o malém výkonu a pracuje s LoRa rádiovou modulací.

Jelikož IoT propojuje obrovské množství zařízení mezi sebou, je velmi důležité dbát na bezpečnost, jelikož jednotlivá zařízení mnohdy odesílají citlivá data, bývají častým terčem útoků hackerů, nebo malwarů, kteří se jich snaží zmocnit. Pro získání dat postačí útočníkovi, zneužití jedné chyby v zabezpečení, popřípadě obejít bezpečnostní opatření jednoho zařízení, čím si usnadní přístup k celé síti, protože zařízení jsou úzce propojena. Aby byla data v bezpečí, musí se zachovat zabezpečené připojení k internetu a k síti, případně vybavit jednotlivá zařízení firewallem nebo antivirem, implementace však bývá problematická, jelikož všechna zařízení nemají dostatečný výpočetní výkon pro tak sofistikovanou ochranu. Dále je nutné vzít v potaz, že čím více způsobů připojení k internetu, tím více způsobů napadení, před kterým je potřeba je chránit. [34]

6.1 SMART řešení

Implementace technologie IoT do celkového systému, který je zaměřen na určité odvětví, popřípadě na koncového uživatele je možné nazvat „Smart“ řešením, i když se spíše jedná o marketingové označení nežli o technický pojem. Podle pole působení, se pak dělí například na Smart cities, v tomto případě je snaha zlepšit kvalitu života ve městě, pokud by byl podobný systém implementován v zemědělství, získáme zemědělství 4.0.

Dalším druhem implementace jsou například chytré domy, které se snaží zlepšit komfort v domácnosti. V současné době se však již nejedná pouze o nějaký hlasový komunikátor, popřípadě „chytrou“ televizi, ale celkový systém, který zastřešuje jak zabezpečení objektu, tak i ekologický a ekonomický aspekt domu s možností ovládat vše vzdáleně, například pomocí mobilní aplikace. [38]

7 Volba platformy

K praktické části bylo nezbytné zvolit zařízení, na kterém se bude systém realizovat. Nakonec byl výběr zúžen na následující dvě platformy, se kterými je již zkušenost.

7.1 Arduino UNO

Jedná se o malý jednodiskový počítač, který byl vytvořen za účelem výukové a open source platformy, která měla pomoci s řízením různých zařízení pomocí počítače. Samotné Arduino se skládá typicky z programovatelného mikrokontroleru, digitálních a analogových pinů a paměti s možností přidání dalších modulů, jako je například Bluetooth modul. Arduino je celkově energeticky nenáročný a pro napájení postačí ideálně 9V baterie. Pro programování mikrokontroleru se využívá vývojové prostředí Arduino IDE, které využívá pro psaní kódu jazyk Wiring, který je založen na jazycích C a C++. Program se vytváří zvlášť a je následně nahrán na mikrokontroler, typicky pak obsahuje dvě části, a to při spuštění, kde se například nastaví, které piny se budou využívat a jestli mají být nastaveny na input či output a smyčka, která běží po celou dobu provozu v předem nastavené frekvenci. [39]



Obr. 22 Arguino UNO [40]

7.2 Raspberry Pi 3 Model B

Jedná se o rozměrově malý, jednodeskový počítač a podobně jako Arduino, je i Raspberry Pi open source výuková platforma, která je zaměřena jak na řízení, i na celkovou podporu výuku informatiky. Zařízení obsahuje USB porty pro připojení periférií, jako je myš nebo klávesnice, HDMI pro připojení monitoru, GPIO piny, které slouží pro ovládání elektronických zařízení a je na uživateli, jak je využije, a RJ45

konektor pro připojení k síti, včetně Wifi adaptéru. Primárním operačním systémem je Raspbian, který je založen na Linuxu včetně podpory programovacích jazyků, jako je Scratch a Python. [41]



Obr. 23 RaspberryPi 3 Modul B [42]

Nakonec byla vybrána platforma Raspberry Pi, jelikož na rozdíl od Arduina, je zde již zabudované síťové připojení, dále je Raspberry Pi výkonnější, je zde možnost zprovoznit webovou aplikaci a také podporuje programovací jazyk Python, který bude využíván pro psaní programu.

8 Praktická část

Praktická část se zaměřuje na vytvoření systému, který bude schopen identifikovat čárový kód, zjistit požadované údaje a následně je zapíše do databáze, které pak budou vypsané na webové stránce. Tento systém bude zaměřený jako doplněk lednice a bude se starat o přehlednost potravin v ní, tak jak bylo popsáno v předešlé kapitole *3 Návrh řešení*.

8.1 Volba databáze

Při výběru databáze, byly zohledněny především parametry jako je uživatelská základna, její model, náročnost, jestli je uživatelsky přívětivá a jestli je zdarma.

	FirebirdSQL	MariaDB
Rychlost	4.	1.
Databázový model	Relační model	Relační model
Licence	Open source	Open source
Podpora OS	Linux, OS X, Unix, Windows	Linux, Windows
SQL	Ano	Ano
Podporované jazyky	C, C#, C++, Java, PHP, Python	C, C#, C++, Java, PHP, Python
Triggers	Ano	Ano
Oblíbenost	4.	3.

	MySQL	PostgreSQL
Rychlost	2.	3.
Databázový model	Relační model	Objektově-relační model
Licence	Open source	Open source
Podpora OS	Linux, OS X, Windows	Linux, OS X, Unix, Windows
SQL	Ano	Ano
Podporované jazyky	C, C#, C++, Java, PHP, Python	.NET, C, C++, Java, PHP, Python
Triggers	Ano	Ano
Oblíbenost	1.	2.

Tab. 5 Porovnání databází [vlastní]

Nakonec byla zvolena MariaDB, jelikož jsem se rozhodl že, budu využívat databázi s relačním modelem, která je postavena na základu MySQL, která je aktuálně nejvíce používaná a díky tomu mohu dohledat více informací a řešení na případné problémy, které by mohli nastat. Další výhodou je, že oproti MySQL podporuje více připojení najednou a je rychlejší.

8.2 Identifikace čárového kódu

Při tvorbě tohoto modulu se objevil závažným problémem, a to je absence centrální databáze pro všechny čárové kódy. Tento problém pramení z toho, že výrobci dostanou k dispozici velký interval čárových kódů pro označování produktů, podle jejich vlastního uvážení, a tak si označení produktů nechávají pouze ve vlastních databázích pro vlastní účely. Tento problém se již snaží GS-1 řešit a vytváří centrální databázi pro všechny čárové kódy, nicméně stále není úplná a není k dispozici pro širokou veřejnost.

Z tohoto důvodu byla použita komunitní databáze, nicméně ani ty nemají uchované informace o všech čárových kódech a mnohdy je nutné za každou identifikaci zaplatit poplatek nebo platit měsíční předplatné, a to vše samozřejmě bez záruky úspěšné identifikace. Nakonec bylo rozhodnuto, že budu využívat komunitní databázi Open food facts, která je zaměřena na identifikaci potravin, je zdarma a neustále se rozrůstá o další produkty. Další výhodou je, že poskytuje údaje plně v české lokalizaci, a to i včetně alergenů, rozměrů, nutričních hodnot atd.

8.3 Popis databáze

Vytvořená tabulka obsahuje pět sloupců. Typicky „ID“ je v tabulce primární klíč, který se automaticky vyplňuje. Do sloupce „Nazev“ se ukládá název produktu do sloupce „Pod_Nazev“ se ukládá značka produktu, oba sloupce jsou textového datového typu. Ve sloupci „Pocet“ se ukládá aktuální počet produktů a jedná se o číselný datový typ. Aby byl proces identifikace již známých produktů rychlejší, nachází se zde i sloupec „Barcode“, který ukládá samotný čárový kód produktu, jedná se o decimální datový typ o délce 14 čísel bez desetinného místa, díky němu je zde možnost identifikovat produkt pouhým porovnáním v databázi a není nutné komunikovat s databází třetí strany.

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Nazev	varchar(255)	YES		NULL	
Pod_Nazev	varchar(255)	YES		NULL	
Pocet	int(11)	YES		NULL	
Barcode	decimal(14,0)	YES		NULL	

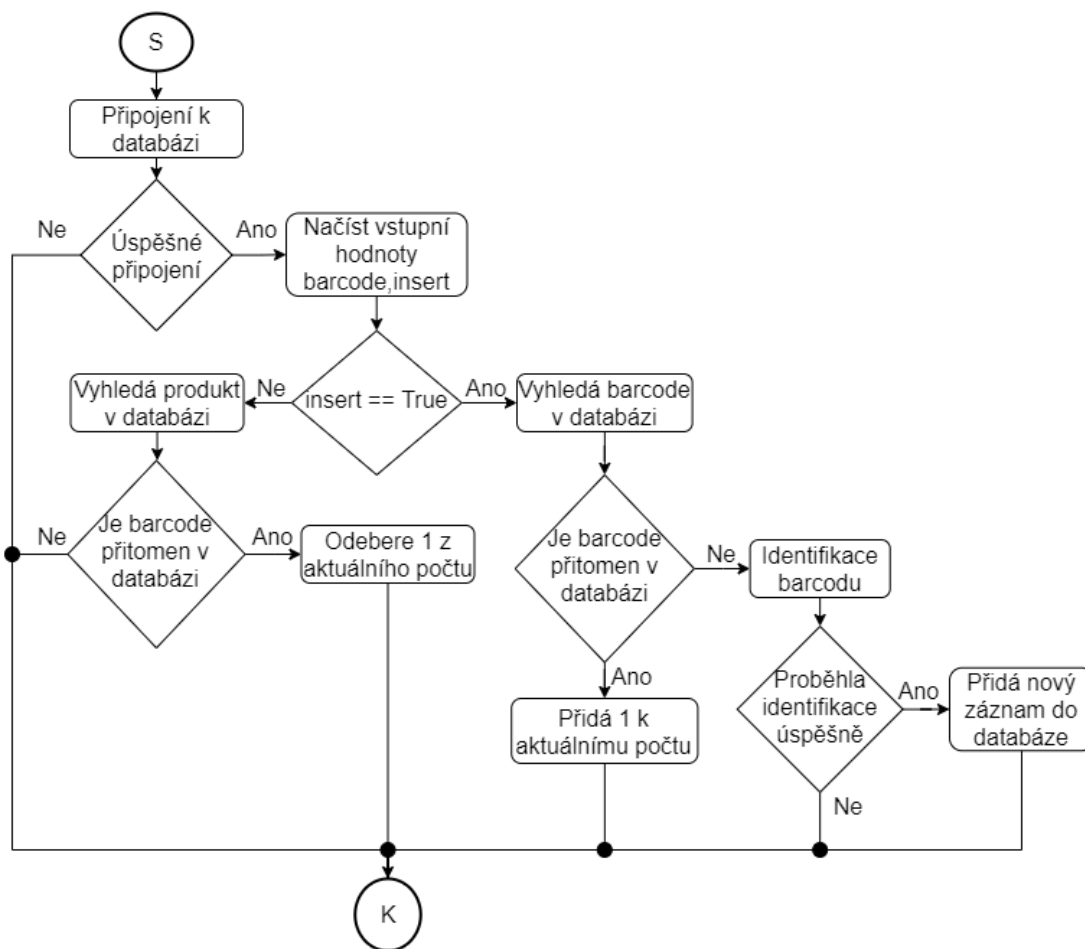
Obr. 24 Popis databáze [vlastní]

ID	Nazev	Pod_Nazev	Pocet	Barcode
1	Eiskaffe	Hochwald	2	4003490049661

Obr. 25 Ukázka záznamu z databáze [vlastní]

8.4 Program

Program, jak je již zmíněno v předešlé kapitole, je psán v jazyce Python, má hlavní funkci *main()*, která má dva vstupní parametry a to *barcode*, který je celočíselného typu a reprezentuje čárový kód, který chceme identifikovat, další je *insert*, jenž je typu bool a uvádí zda chce uživatel daný produkt vložit nebo odebrat ze seznamu, což se realizuje podmínkou, tedy pokud je *insert == True*, znamená to, že produkt chceme vkládat, v opačném případě chceme produkt odebírat. Na začátku funkce se provádí připojení k databázi a následuje rozhodnutí, jestli produkt vkládáme nebo odebíráme. Pokud produkt vkládáme, nejprve dotazem na databázi zjišťujeme, jestli se již produkt v aktuálním seznamu nachází. To provádíme dotazem „*SELECT COUNT(*) FROM Bc.Seznam WHERE Barcode = '{barcode}'*“, který nám vrátí počet záznamů, jenž mají v kolonce *Barcode* odpovídající barcode. Jelikož je barcode pro každý produkt jedinečný, očekávaný výsledek je 0, nebo 1, které si převedeme jako *False* nebo *True*. Pokud se tedy produkt, již nachází v databázi, není nutné provádět identifikaci, ale pouze navýšíme kolonku „*Pocet*“ v databázi o jedna. Pokud produkt není v databázi, musí se nejprve podle čárového kódu provést identifikace produktu, kterou provádíme za pomoci *requestu* s daným čárovým kódem, který směřujeme na databázi Open food facts, o která již byla zmíněna v předešlé kapitole. Pokud identifikace proběhne úspěšně, uloží se název produktu do proměnných *product_name* a název výrobce do *brands*. Tyto proměnné následně se vstupní proměnnou *barcode* použijeme v dotazu *INSERT*, kde vkládáme tyto informace do databáze a kolonku „*Pocet*“, dáváme vždy na hodnotu jedna. Pokud však produkt odebíráme, zjišťujeme pouze aktuální hodnotu *Pocet* od které odečítáme jedna za pomoci dotazu *UPDATE*. Po provedení všech akcí se na konci funkce ukončí navázání s databází a funkce se ukončí.



Obr. 26 Vývojový diagram programu [vlastní]

Další částí je webová aplikace, která je vytvořena ve frameworku Django. V tomto frameworku bylo vytvořen *view*, který naváže komunikaci s databází a následně zobrazí všechny záznamy, které mají v kolonce „*Pocet*“ více než nula, v přehledné formě na webové stránce.

Seznam

Produkt	Výrobce	Počet
Eiskaffe	Hochwald	2
Tatarská omáčka	Hellmanns, unilever ČR	3

Obr. 27 Náhled seznamu potravin [vlastní]

8.5 Budoucí vývoj

V případě dalšího vývoje by proběhlo zlepšení stávajícího ovládání, aby bylo více uživatelsky přívětivé a byla zde možnost rychlého přidávání a odebrání položek. Toho by bylo docíleno instalací přepínače připojeného na GPIO piny, který by automaticky vyhodnocoval, zda chce uživatel provádět vkládání nebo odebrání. Případně instalací snímačů po stranách scanneru, které by snímaly pohyb a na základě jeho vyhodnocení by bylo možné určit, jestli uživatel daný produkt vkládá nebo odebírá.

Další funkcí, kterou by bylo možné přidat, je analýza kupovaných produktů, podle které by se dalo určit, kdy a co uživatel pravidelně nakupuje. Podle odhadu by se pak vygeneroval očekávaný nákup, který by uživateli navrhnul položky, které pravidelně nakupuje, s možností přidat nebo odebrat další položky podle jeho uvážení. Následně by zjistil, kde se vybrané položky dají nakoupit u donáškových službách, jako je Košík nebo Rohlík, za celkově nejnižší cenu. Poté by bylo nabídnuto provést nákup těchto potravin od vybrané donáškové služby. Tímto způsobem by uživatel nemusel trávit zbytečně čas při nakupování potravin, navíc ještě nakoupí za nejnižší cenu.

9 Mobilní aplikace

Se stále zvětšující se oblibou mobilních aplikací a výkonnostním pokrokem, které mobilní zařízení zažila, lze podobný koncept aplikovat i na mobilní zařízení v podobě aplikace. Muselo by dojít k menším změnám, co se týče čtení čárových kódů, nicméně většina telefonů již disponuje dosti kvalitním fotoaparátem, který by byl schopen identifikovat čárový kód. Výhodou by pak byla možnost implementace čtení QR kódů, které právě využívají tuto techniku pro čtení, případně by zde byla i možnost zabudovat identifikaci RFID tagů.

Momentálně se na trhu vyskytují dvě hlavní platformy pro mobilní aplikace – Android od Google a iOS od Apple. Android je na rozdíl od iOS open source platforma, více značek využívá právě tento operační systém pro své mobilní zařízení, z čehož plyne i větší uživatelská základna pro tuto platformu. Dále je Android díky své platformě Google play více otevřen vydávání nových aplikací, na rozdíl od AppStoru, kde je potřeba zaplatit vstupní poplatek, který je mnohonásobně vyšší než na konkurenční platformě, na druhou stranu si strhává menší provizi za prodej aplikací.

Programovacím jazykem pro vývoj aplikací na iOS je Swift, který je určen také pro vývoj na platformách WatchOS a macOS, jedná se o open source, kompilovaný jazyk, vytvořený jako alternativa k Objective-C jazyku. Android využívá pro tvorbu aplikací jazyk Kotlin, který je založen na jazyku Java, se kterým je schopen efektivně spolupracovat, dále byl ovlivněn i jazyky C# a JavaScript.

Pro vývoj aplikace by byla zvolena platforma Android a programovací jazyk Kotlin, jelikož se jedná o používanější platformu. Protože by byla aplikace zcela zdarma, vstupní podmínky platformy Google play by byly na rozdíl od konkurence výhodnější.

10 Závěr

Cílem práce bylo navrhnout řešení pro „chytrou“ lednici, která bude využívat automatickou identifikaci produktů, bude provázána s databází a webovou aplikací. Způsoby automatické identifikace byly analyzovány v teoretické části práce spolu s popisem databází a přiblížením technologie Internetu věcí. Dále byla zvolena platforma, na které se dané řešení provozuje.

V praktické části byla zvolena a vytvořena databáze. V této části se ukázala jako nejlepší volba databáze MariaDB, která splňuje všechny požadavky a její vlastnosti umožňují další bezproblémový vývoj. Dále byl vytvořený modul pro automatickou identifikaci čárových kódů, kde se objevil problém s absencí kvalitní databáze pro identifikaci produktů. Následně byla vytvořena webová aplikace, jenž má za úkol uživatelům zlepšit přehled o stávajících produktech, a tím zamezit zbytečnému nakupování a plýtvání potravin. Na závěr byly probrány další možnosti budoucího vývoje a možnost převést tento koncept na mobilní aplikaci.

Výsledek této práce může sloužit jako základ všem systémům, jenž používají pro identifikaci produktů čárové kódy. Může být využit například i v systému skladování vakcín, což je v dnešní době velmi aktuální téma. Dále se mohou části práce publikovat na platformě Odo, která podporuje širokou škálu modulů psaných v Pythonu a může sloužit jako identifikační modul většímu systému.

Tento koncept bych rád rozvinul v případném navazujícím studiu v mé diplomové práci a vytvořil tak celkový systém pro správu potravin.

11 Použitá literatura

- [1] BENADIKOVÁ, Adriana. Čárové kódy: Automatická identifikace. Praha: Grada, 1994. ISBN 80-85623-66-8.
- [2] GS1 CZECH REPUBLIC – Váš partner ve světě automatické identifikace. GS1 CZECH REPUBLIC – Váš partner ve světě automatické identifikace [online]. Copyright © GS1 Czech Republic [cit. 21.11.2020]. Dostupné z: <https://www.gs1cz.org/>
- [3] N. Joseph Woodland, Inventor of the Bar Code, Dies at 91 - The New York Times. The New York Times - Breaking News, US News, World News and Videos [online]. Copyright © [cit. 16.11.2020]. Dostupné z: <https://www.nytimes.com/2012/12/13/business/n-joseph-woodland-inventor-of-the-bar-code-dies-at-91.html>
- [4] N. Joseph Woodland, Inventor of the Bar Code, Dies at 91 - The New York Times. The New York Times - Breaking News, US News, World News and Videos [online]. Copyright © [cit. 17.11.2020]. Dostupné z: <https://www.nytimes.com/2012/12/13/business/n-joseph-woodland-inventor-of-the-bar-code-dies-at-91.html>
- [5] NIHF Inductee N. Joseph Woodland Created Technology for Reading Bar Codes. National Inventors Hall of Fame® [online]. Copyright © [cit. 17.11.2020]. Dostupné z: <https://www.invent.org/inductees/n-joseph-woodland>
- [6] NIHF Inductee Bernard Silver Invented the Bar Code Reader. National Inventors Hall of Fame® [online]. Copyright © [cit. 17.11.2020]. Dostupné z: <https://www.invent.org/inductees/bernard-silver>
- [7] Google Patents [online]. Dostupné z: <https://patents.google.com/patent/US2612994>
- [8] How we got here | GS1. GS1 | The Global Language of Business [online]. Dostupné z: <https://www.gs1.org/about/how-we-got-here>
- [9] GS1 CZECH REPUBLIC – Váš partner ve světě automatické identifikace [online]. Copyright ©E [cit. 21.11.2020]. Dostupné z: https://www.gs1cz.org/media/nezarazene/cenik_2019_v7.1.pdf
- [10] How RS232 Works: The Easy Guide to RS232.. The best Microcontroller projects and resources. [online]. Copyright © 2005 [cit. 28.11.2020]. Dostupné z: <https://www.best-microcontroller-projects.com/how-rs232-works.html>

- [11] Sériový port RS232 | Papouch.com. Papouch.com [online]. Dostupné z: <https://papouch.com/seriovy-port-rs232-p3740/>
- [12] Základní rozdíly mezi RS-232, RS-422 a RS-485. Průmyslové počítače pro jakýkoli úkol [online]. Copyright © 2020, ipc2u.cz [cit. 28.11.2020]. Dostupné z: <https://ipc2u.cz/articles/simple-decisions/zakladni-rozdily-mezi-rs-232-rs-422-a-rs-485/>
- [13] 301 Moved Permanently [online]. Copyright © [cit. 28.11.2020]. Dostupné z: <http://www.cs.vsb.cz/licev/technologieco.pdf>
- [14] Tajomstvo ČIAROVÝCH KÓDOV - YouTube. YouTube [online]. Copyright © 2020 Google LLC [cit. 29.11.2020]. Dostupné z: <https://youtu.be/2ZO5L8tMtRA>
- [15] Rakesh Patel: Difference Between 1D, 2D & 3D Barcode. Rakesh Patel [online]. Dostupné z: <https://rakeshnpatel.blogspot.com/2015/01/difference-between-1d-2d-3d-barcode.html>
- [16] Čím a jak QR kódy číst | qikni.cz. QR kódy | qikni.cz [online]. Copyright © webdesign by Kobra design 2013 [cit. 29.11.2020]. Dostupné z: <https://www.qikni.cz/qr-kod/chci-vedet-cim-a-jak-qr-kody-cist.html>
- [17] QR kod, co je QR kód? | QR-KODY.cz. QR-KODY.cz - qr kódy a qr code technologie [online]. Copyright © QR [cit. 29.11.2020]. Dostupné z: <http://www.qr-kody.cz/qr-kod>
- [18] Generated RGB colored QR Code | Download Scientific Diagram. ResearchGate | Find and share research [online]. Copyright © 2008 [cit. 29.11.2020]. Dostupné z: https://www.researchgate.net/figure/9-Generated-RGB-colored-QR-Code_fig19_301788314
- [19] Active RFID vs. Passive RFID: What's the Difference? - atlasRFIDstore. atlasRFIDstore | Buy RFID Technology | The RFID Hardware Experts [online]. Copyright © 2021 atlasRFIDstore [cit. 25.01.2021]. Dostupné z: <https://www.atlasrfidstore.com/rfid-insider/active-rfid-vs-passive-rfid>
- [20] Types of Memory in RFID Tags - atlasRFIDstore. atlasRFIDstore | Buy RFID Technology | The RFID Hardware Experts [online]. Copyright © 2021 atlasRFIDstore [cit. 25.01.2021]. Dostupné z: <https://www.atlasrfidstore.com/rfid-insider/types-of-memory-in-gen-2-uhf-rfid-tags>
- [21] RFID and the Differences in Passive, Semi-Passive, and Active Tags. Custom Barcode & Labeling Solutions | Computype [online]. Copyright © Computype

- 2021 All rights reserved. [cit. 25.01.2021]. Dostupné z:
<https://www.computype.com/blog/rfid-and-the-difference-in-passive-semi-passive-and-active-tags>
- [22] SMILEY, Suzanne, 2020. The Insider's Guide to Working with RFID. Birmingham, AL 35203. ISBN 9780578778778
- [23] Úvod do tabulek - Access. Microsoft Support [online]. Dostupné z:
<https://support.microsoft.com/cs-cz/office/%C3%BAvod-do-tabulek-78ff21ea-2f76-4fb0-8af6-c318d1ee0ea7>
- [24] Databáze. Databáze [online]. Copyright © 2010 Misha [cit. 15.02.2021]. Dostupné z: <http://www.databaze.chytrak.cz/objekty.htm>
- [25] The 3 Types of Relationships in Database Design | Database.Guide. Database.Guide | [online]. Dostupné z: <https://database.guide/the-3-types-of-relationships-in-database-design/>
- [26] Příručka k relacím mezi tabulkami - Access. Microsoft Support [online]. Dostupné z: <https://support.microsoft.com/cs-cz/office/p%C5%99%C3%ADru%C4%8Dka-k-relac%C3%ADm-mezi-tabulkami-30446197-4fbe-457b-b992-2f6fb812b58f>
- [27] 05 Hierarchical Databases - YouTube. YouTube [online]. Copyright © 2021 Google LLC [cit. 16.02.2021]. Dostupné z:
https://www.youtube.com/watch?v=Rk5S3VWhuZw&ab_channel=Informatica
- [28] Co je relační databáze | Oracle Česká Republika. [online]. Dostupné z:
<https://www.oracle.com/cz/database/what-is-a-relational-database/>
- [29] What is a Database Model | Lucidchart. Online Diagram Software & Visual Solution | Lucidchart [online]. Copyright © [cit. 16.02.2021]. Dostupné z:
https://www.lucidchart.com/pages/database-diagram/database-models#section_1
- [30] Objektově orientované systémy databází – Wikisofia. [online]. Copyright © 2013 ISSN [cit. 01.03.2021]. Dostupné z:
https://wikisofia.cz/wiki/Objektov%C4%9B_orientovan%C3%A9_syst%C3%A9my_datab%C3%A1z%C3%AD
- [31] Objektové databáze - Linux E X P R E S. Linux E X P R E S [online]. Copyright © 2021 CCB, spol. s r. o., všechna práva vyhrazena. [cit. 01.03.2021]. Dostupné z: <https://www.linuxexpres.cz/business/objektove-databaze>
- [32] ŠTEFANÍK, Štěpán. Nástroj pro údržbu databáze. Brno, 2010. Diplomová práce. Masarykova Univerzita. Vedoucí práce RNDr. Karel Kiszovi.Computerworld.cz

- [cit. 03.03.2021]. Deník pro IT profesionály [online]. Dostupné z:
<https://computerworld.cz/archiv/objektive-relacni-databaze-9932>
- [33] How to Decode a QR Code by Hand | A Step by Step Guide - YouTube.
YouTube [online]. Copyright © 2021 Google LLC [cit. 12.03.2021]. Dostupné z:
https://www.youtube.com/watch?v=kHTWTyV7VJQ&ab_channel=CodeQuriouZ
- [34] What is IoT (Internet of Things) and How Does it Work?. Internet of Things (IoT) news, blogs and analysis - IoTAgenda.com [online]. Dostupné z:
<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
- [35] Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper - Cisco. Cisco - Networking, Cloud, and Cybersecurity Solutions [online]. Dostupné z: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- [36] Co je to edge computing? Aneb cesta k nižší latenci | MasterDC. MasterDC – Specialisté na firemní IT infrastrukturu [online]. Copyright © 2021 Master Internet, s.r.o. [cit. 19.04.2021]. Dostupné z: <https://www.master.cz/blog/edge-computing-slibuje-nizsi-latenci/>
- [37] What is an IoT Gateway? *Open Automation Software* [online]. Lakewood [cit. 2021-04-19]. Dostupné z: <https://openautomationsoftware.com/open-automation-systems-blog/what-is-an-iot-gateway/>
- [38] Chytré byty, domy a budovy - Pravidla, nápady, řešení.... Odborné časopisy [online]. Dostupné z:
<http://www.odbornecasopisy.cz/flipviewer/Specialy/21/KnowHow//index.html#p=3>
- [39] Arduino - Introduction . Arduino - Home [online]. Dostupné z:
<https://www.arduino.cc/en/guide/introduction>
- [40] Arduino Uno Rev3 | Arduino Official Store. Arduino Official Store | Boards Shields Kits Accessories [online]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>
- [41] What is a Raspberry Pi?. Teach, Learn, and Make with Raspberry Pi [online]. Dostupné z: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
- [42] Buy a Raspberry Pi 3 Model B – Raspberry Pi. Teach, Learn, and Make with Raspberry Pi [online]. Dostupné z:
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Tabulky

0	0	0	0	1	1	0	1
1	0	0	1	1	0	0	1
2	0	0	1	0	0	1	1
3	0	1	1	1	1	0	1
4	0	1	0	0	0	1	1
5	0	1	1	0	0	0	1
6	0	1	0	1	1	1	1
7	0	1	1	1	0	1	1
8	0	1	1	0	1	1	1
9	0	0	0	1	0	1	1

Tab. 6 Dekódovací tabulka A liché parity [1]

0	0	1	0	0	1	1	1
1	0	1	1	0	0	1	1
2	0	0	1	1	0	1	1
3	0	1	0	0	0	0	1
4	0	0	1	1	1	0	1
5	0	1	1	1	0	0	1
6	0	0	0	0	1	0	1
7	0	0	1	0	0	0	1
8	0	0	0	1	0	0	1
9	0	0	1	0	1	1	1

Tab. 7 Dekódovací tabulka B sudé parity [1]

0	1	1	1	0	0	1	0
1	1	1	0	0	1	1	0
2	1	1	0	1	1	0	0
3	1	0	0	0	0	1	0
4	1	0	1	1	1	0	0
5	1	0	0	1	1	1	0
6	1	0	1	0	0	0	0
7	1	0	0	0	1	0	0
8	1	0	0	1	0	0	0
9	1	1	1	0	1	0	0

Tab. 8 Dekódovací tabulka C sudé parity [1]

13	12	11	10	9	8	7	6 až 1
0	A	A	A	A	A	A	C
1	A	A	B	A	B	B	C
2	A	A	B	B	A	B	C
3	A	A	B	B	B	A	C
4	A	B	A	A	B	B	C
5	A	B	B	A	A	B	C
6	A	B	B	B	A	A	C
7	A	B	A	B	A	B	C
8	A	B	A	B	B	A	C
9	A	B	B	A	B	A	C

Tab. 9 Určení hodnoty na 13. pozici [1]