



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra ekonomiky, manažerství a humanitních věd

BAKALÁŘSKÁ PRÁCE

## **Dálkový monitoring zaplnění nádob na tříděný odpad**

Autor: Kryštof Novák

Studijní program: Elektrotechnika, Energetika a management

Studijní obor: Elektrotechnika a management

Vedoucí práce: Ing. Vladimír Janíček, Ph.D.

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Novák** Jméno: **Kryštof** Osobní číslo: **434642**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra ekonomiky, manažerství a humanitních věd**  
Studijní program: **Elektrotechnika, energetika a management**  
Studijní obor: **Elektrotechnika a management**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Dálkový monitoring zaplnění nádob na tříděný odpad**

Název bakalářské práce anglicky:

**Remote Monitoring of Filling of Containers for Sorted Waste**

Pokyny pro vypracování:

- 1) Proveďte rozbor současných systémů pro dálkový sběr dat a jejich vhodnost použití pro monitoring nádob s tříděným odpadem.
- 2) Navrhněte koncept zařízení pro monitoring.
- 3) Vytvořte obslužný software (databázový systém, GUI pro ovládání a interpretaci dat).
- 4) Proveďte ekonomickou rozvahu přínosů projektu.

Seznam doporučené literatury:

- 1) Raspberry Pi IoT Projects: Prototyping Experiments for Makers, Shovic J. C., ISBN-13: 978-1484213780, 2016
- 2) Remote Sensor Monitoring by Radio with Arduino: Detecting Intruders, Fires, Flammable and Toxic Gases, and other Hazards at a Distance, Leithauser D., ISBN-13: 978-1534640559, 2016
- 3) [www.lora-alliance.org](http://www.lora-alliance.org)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Vladimír Janíček, Ph.D., katedra mikroelektroniky FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **07.02.2018** Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Vladimír Janíček, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

# Čestné prohlášení

---

Prohlašuji, že jsem předloženou bakalářskou práci vypracoval zcela samostatně pod vedením vedoucího práce a že jsem uvedl veškeré informační zdroje v souladu s metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne .....

.....

Podpis autora práce

# Poděkování

---

Rád bych poděkoval vedoucímu práce Ing. Vladimíru Janíčkoví, Ph.D. za pomoc a odborné konzultace při tvorbě bakalářské práce. Dále bych chtěl poděkovat firmě Ing. Pavel Novák s.r.o. za poskytnuté informace z oblasti odpadového hospodářství a finance na vývoj prototypu systému na dálkový monitoring zaplnění odpadových nádob. V neposlední řadě také děkuji rodině a přítelkyni Siqi za neutichající morální podporu.



# Abstrakt

---

Tato bakalářská práce se zabývá návrhem, realizací a ekonomickým vyhodnocením přínosů systému pro sledování zaplněnosti odpadových nádob. Navrhuje a ověřuje koncept měření ultrazvukem, přenos dat GSM sítí a interpretaci dat internetovou aplikací. Koncept vyhodnocuje jako správně zvolený a funkční. Z analýzy ekonomických přínosů vyplývá, že sledováním zaplněnosti odpadových nádob s následnou optimalizací svozových tras lze ušetřit až 40 % svozových nákladů.

# Abstract

---

This bachelor thesis is dealing with a design, realization and economic contributions of a system for waste level measuring in waste containers. It proves that the designed concept which is composed of ultrasonic measuring, GSM data transfer and information interpretation in an internet application is functional. The economic contributions show that using waste level measuring and collection route optimization can save up to 40 % costs of waste collection.

# Klíčová slova

---

Monitoring odpadových nádob, internet věcí, smart city, tříděný odpad, ultrazvukové měření

# Keywords

---

Waste bins monitoring, Internet of things, Smart city, sorted waste, ultrasonic measuring

# Obsah

<b>1</b>	<b>Úvod .....</b>	<b>9</b>
1.1	Cíl práce .....	9
1.2	Motivace.....	9
1.3	Koncept Smart city a Internet věcí.....	10
<b>2</b>	<b>Současné systémy pro dálkový monitoring odpadových nádob .....</b>	<b>11</b>
2.1	Automatické systémy .....	11
2.1.1	Měření ultrazvukem .....	11
2.1.2	Měření kamerou s následným rozpoznáním obrazu .....	12
2.1.3	Automatické systémy v České republice.....	13
2.2	Manuální systémy .....	13
2.2.1	Manuální systémy v České republice.....	14
2.3	Topologie a požadované parametry měřící jednotky.....	14
2.3.1	Požadované parametry měřící jednotky .....	15
<b>3</b>	<b>Současné technologie pro bezdrátový přenos dat pro IoT.....</b>	<b>15</b>
3.1	GSM.....	15
3.2	LTE .....	16
3.3	Bluetooth.....	16
3.4	WiFi .....	16
3.5	ZigBee .....	17
3.6	SigFox .....	17
3.7	LoRa.....	17
3.7.1	LoRaWAN.....	17
3.8	Narrowband IoT.....	18
3.9	Výběr vhodné komunikační technologie.....	18
<b>4</b>	<b>Části systému .....</b>	<b>18</b>
4.1	Koncept systému .....	18
4.2	Měřící jednotka .....	19
4.2.1	Mikrokontrolér.....	19
4.2.2	Zdroj napětí.....	19
4.2.3	Výpočet výdrže měřící jednotky .....	20
4.2.4	Ultrazvukový senzor vzdálenosti .....	21
4.2.5	GSM modul .....	22
4.2.6	RTC.....	23
4.3	Server .....	25
4.3.1	Hardware .....	25
4.3.2	Software – LAMP .....	26
<b>5</b>	<b>Realizace měřící jednotky.....</b>	<b>28</b>
5.1	Hardware.....	28
5.1.1	Elektrické schéma.....	28
5.1.2	Deska plošného spoje .....	31

5.1.3	Osazení komponentů na DPS .....	31
5.1.4	Ochranný obal měřící jednotky .....	32
5.1.5	Kompletace hardware části.....	33
<b>5.2</b>	<b>Software .....</b>	<b>33</b>
5.2.1	Inicializace.....	33
5.2.2	Funkce Setup().....	34
5.2.3	Nastavení RTC – funkce nastaveniRTC().....	35
5.2.4	Nastavení spánku - funkce nastaveniSpanku() .....	36
5.2.5	Smyčka loop().....	37
5.2.6	Ultrazvukové měření - funkce zmerit() .....	39
5.2.7	Uspání mikrokontroléru – funkce usni() .....	40
5.2.8	Řetězec funkcí k odeslání dat .....	40
<b>5.3</b>	<b>Testování měřící jednotky .....</b>	<b>45</b>
5.3.1	Krátkodobý test.....	45
5.3.2	Dlouhodobý test.....	45
<b>6</b>	<b>Realizace internetové aplikace.....</b>	<b>47</b>
<b>6.1</b>	<b>MySQL databáze.....</b>	<b>47</b>
<b>6.2</b>	<b>Grafické uživatelské prostředí .....</b>	<b>48</b>
6.2.1	Stránka přihlášení.....	48
6.2.2	Domovská stránka (aktuální data).....	49
6.2.3	Stránka detailu odpadové nádoby.....	50
6.2.4	Stránka Mapa .....	50
<b>7</b>	<b>Ekonomická analýza projektu .....</b>	<b>51</b>
<b>7.1</b>	<b>Náklady.....</b>	<b>51</b>
7.1.2	Provozní náklady .....	52
7.1.3	Výrobní náklady.....	52
<b>7.2</b>	<b>Cena služby.....</b>	<b>54</b>
<b>7.3</b>	<b>Cash flow .....</b>	<b>55</b>
<b>8</b>	<b>Přínosy projektu.....</b>	<b>55</b>
<b>8.1</b>	<b>Neekonomické přínosy projektu.....</b>	<b>55</b>
<b>8.2</b>	<b>Ekonomické přínosy projektu .....</b>	<b>55</b>
8.2.1	Běžný svoz.....	56
8.2.2	Chytrý svoz .....	57
8.2.3	Závěr ekonomických přínosů projektu.....	59
<b>9</b>	<b>Závěr.....</b>	<b>59</b>
	<b>Literatura .....</b>	<b>61</b>
	<b>Seznam tabulek a obrázků .....</b>	<b>63</b>
	<b>Seznam příloh.....</b>	<b>65</b>

# Přehled použitých zkratk

IoT	Internet of Things
NFC	Near Field Communication
SMS	Short message service
GSM	Groupe Spécial Mobile
GPRS	General Packet Radio Service
EDGE	Enhanced Data Rates for GSM Evolution
LTE	Long Term Evolution
IEEE	Institute of Electrical and Electronics Engineers
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
MAC	Media Access Control
I/O	Input/output
IDE	Integrated Development Environment
UART	Universal Asynchronous Receiver and Transmitter
RTC	Real Time Clock
SCL	Serial clock
SDA	Serial data
NAS	Network Attached Storage
DPS	Deska plošného spoje
ICSP	In-Circuit Serial Programming
SPI	Serial Peripheral Interface
MOSI	Master Out/Slave In
MISO	Master In/Slave Out
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
SMT/SMD	Surface mount technology/surface mount device
CAD	Computer-aid design
PLA	Poly lactid acid
LED	Light emitting diode
BOD	Brown out detector
SMCR	Sleep mode control register
APN	Access point name
HTTP	Hyper text transfer protocol
DC	direct current
CSS	Cascading Style Sheets
GUI	Graphical User interface
XML	Extensible markup language
DPPO	Daň z příjmů právnických osob
NPV	Net present value – čistá současná hodnota peněz
CF	Cash flow
DCF	Diskontované cash flow
M2M	Machine to machine
NPV	Net present value
ISR	Interrupt service routine

# 1 Úvod

---

## 1.1 Cíl práce

Cílem této bakalářské práce je realizace funkčního systému pro měření zaplněnosti odpadových nádob a vyhodnocení ekonomických přínosů tohoto projektu. Nejdříve v teoretické části budou přiblíženy a zváženy jednotlivé běžně používané technologie pro sledování zaplněnosti odpadových nádob a přenos dat zařízeních internetu věcí, následně v praktické části budou popsány jednotlivé prvky celého systému a průběh samotné realizace. Na závěr bude provedena rozvaha přínosů projektu.

## 1.2 Motivace

Otcova poradenská firma na odpadové hospodářství, Ing. Pavel Novák s.r.o., má klienta, který by měl zájem o systém sledování zaplněnosti odpadových nádob. Na trhu se neprodává hotový použitelný hardware, který by umožnil provozování vlastní služby, ale vždy pouze monitoring jako kompletní služba. Z tohoto důvodu jsem se rozhodl systém sledování zaplněnosti odpadových nádob realizovat sám.

V nynější době svozové firmy většinou nemají zpětnou vazbu o zaplněnosti jejich odpadových nádob, tudíž vyvázejí odpad paušálně jednou za určité časové období. To funguje v celku spolehlivě pro odpady, jejichž produkce je konstantní a dá se tedy předpokládat zaplnění za určitý čas. Pro odpady, které se produkují nepravidelně a nárazovitě, paušální vývoz funguje špatně. Například sběrné nádoby na textil. Při stěhování nebo velké obměně šatníku se nádoba zaplní okamžitě. To je jev, který je zcela náhodný a nedá se dost dobře předpokládat. Bez zpětné vazby zaplnění nádoby, se vývoz uskuteční až při následujícím vývozovém termínu, což je v mnoha případech pozdě. Mezitím lidé nechávají odpad vedle nádob (obr. 1), čímž vzniká nelegální skládka a majitel odpadových nádob může dostat od města pokutu nebo v extrémním případě i vypovězení smlouvy na umístění kontejnerů. Pro eliminaci těchto případů, firmy se snaží vyvázet nádoby v co nejkratších časových úsecích, což ale vede k vysypávání nenaplňených nádob a tím i tvorbě zbytečných nákladů.

Majitel nádob pomocí monitoringu zaplnění získá zajímavé informace, které mu můžou pomoci k zefektivnění samotného vývozového procesu. Pomocí plánovače tras a informací o zaplněnosti může optimalizovat svozovou trasu a tím ušetřit náklady na provoz. Při dostatečném množství dat lze také zjistit optimální potřebnou velikost vozového parku, počet odpadových nádob a zaměstnanců, což může vést k dalšímu šetření nákladů.



Obrázek 1 - Nelegální skládka kolem sběrných nádob na textil. 5.5. 2017, Olbrachtova, Praha 4

### 1.3 Koncept Smart city a Internet věcí

Monitoring odpadových nádob s optimalizovaným systémem svozu je část chytrého odpadového hospodářství, což je neodmyslitelnou součástí konceptu „Smart City“. Koncept Smart City využívá informační a komunikační technologie pro zlepšení kvality života ve městech. Tento koncept vznikl kvůli přibývajícím počtu obyvatel ve městech a tím i rostoucím nárokům na infrastrukturu, která by v budoucnu měla zajistit udržitelnost kvality života. V konkrétním případě svozu odpadů, zvýšení počtu obyvatel znamená zvětšení produkce odpadů a zvětšení hustoty dopravy ve městech. Systém měření zaplněnosti odpadových nádob reaguje na oba podmínky – pomůže včasnému vývozu rychle plnicích se nádob a zároveň dokáže omezit svozy na nezbytné minimum. Lze tedy v následujících letech v tomto odvětví na území České republiky očekávat velký růst a investice a tím i slibnou obchodní příležitost. Hlavní město Praha sledování zaplněnosti odpadových nádob a optimalizaci svozových tras již zakomponovala do svého konceptu „Smart Prague do roku 2030“ [1].

Měřicí jednotky také spadají do konceptu internetu věcí. IoT je síť zařízení vybavených síťovou konektivitou, které pomocí různých senzorů, čidel a modulů sbírají a vyměňují si data pro určitý úkol.

## 2 Současné systémy pro dálkový monitoring odpadových nádob

---

Nynější technologie pro sledování zaplněnosti se dají rozdělit do dvou hlavních kategorií – automatické a manuální. Automatické systémy zajišťují veškerý proces – od měření po předání informace – zcela automaticky bez nutného zásahu člověka. Manuální technologie pro sledování odpadových nádob vyžadují v procesu měření a předání informace spolupráci člověka.

### 2.1 Automatické systémy

Od automatické měřicí jednotky umístěné v odpadové nádobě se požaduje, aby spolehlivě změřila množství odpadu a tuto informaci předala požadovanému subjektu. Tyto požadavky se dají splnit mnoha technologiemi, nicméně kromě proveditelnosti, je také důležité, aby daná technologie byla praktická, spolehlivá a finančně nenáročná. Z rozhovoru se zástupcem italské firmy Baron.it, která vyrábí měřicí jednotky do odpadových nádob, a provedeném průzkumu trhu vyplynulo, že se na trhu usadily dvě technologie používané k sledování odpadových nádob – měření ultrazvukem a snímání kamerou s následným rozpoznáváním obrazu.

Tato podkapitola bude věnována pouze procesu měření, přenosu informace se bude zabývat kapitola následující.

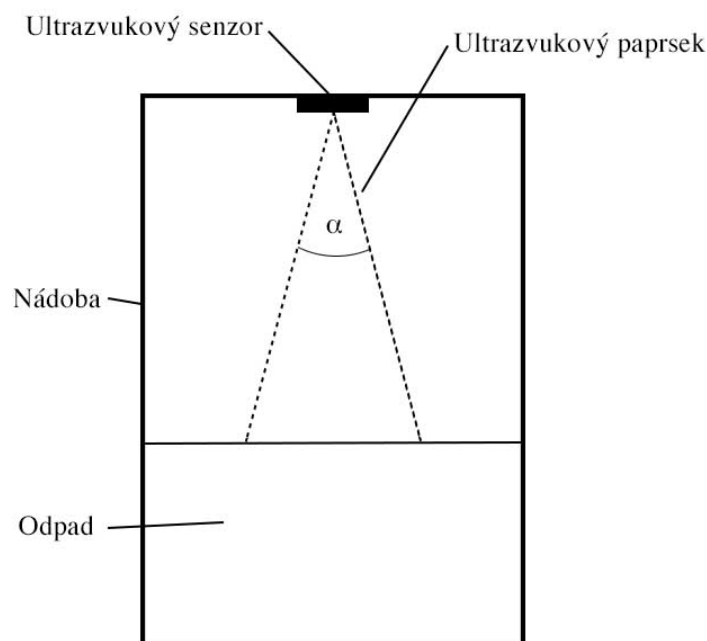
#### 2.1.1 Měření ultrazvukem

Ultrazvukové senzory měří vzdálenost mezi vysílačem a povrchem (odpadem) pomocí zvuku na vysoké frekvenci. Senzor z vysílače vyšle signál, který se odrazí od odpadu a vrátí zpět do přijímače (obr. 2). Pomocí známé rychlosti zvuku (343 m/s pro 20 °C) a doby mezi odesláním a přijmutím signálu, se dá pomocí vzorce 3 vypočítat vzdálenost odpadu od senzoru. Vyslaný paprsek ultrazvuku má kuželovitý tvar s úhlem  $\alpha$  vypočítaným ze vzorce 1, kde  $v$  je rychlost vlny,  $d$  průměr vysílače a  $f$  frekvence signálu. Většina běžných ultrazvukových senzorů vysílá zvuk o frekvenci 40 kHz a mají průměr vysílače od 1,5 do 2 cm. Úhel vysílaného signálu  $\alpha$  se tedy pohybuje kolem 30°.

$$\alpha = \sin^{-1}\left(1.2 \frac{v}{d * f}\right) \quad (1)$$

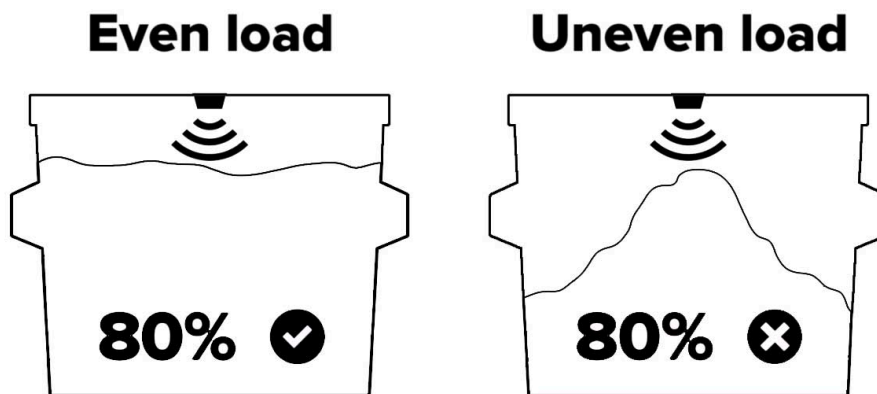
*Výpočet velikosti úhlu vyslaného ultrazvukového paprsku [3]*

Tato metoda měření zaplněnosti odpadových nádob je nejrozšířenější na trhu. Z mého průzkumu trhu vyplynulo, že víceméně všechny firmy nabízející řešení měření zaplněnosti běžných komerčních odpadových nádob používají tuto technologii.



Obrázek 2 - Ultrazvukové měření odpadové nádoby

Největší nevýhoda této technologie se považuje za falešnou signalizaci zaplnění při nerovnoměrné distribuci odpadu v kontejneru (obr. 3). Když je odpad nerovnoměrně rozmístěn nebo po vložení do kontejneru změnil svůj tvar (např. papírová krabice – rozloží se), tak měřicí jednotka může nahlásit milný údaj o zaplněnosti. Dále se tímto způsobem nedají monitorovat otevřené velkoobjemové kontejnery z důvodu jejich velikosti a nemožnosti umístění měřicí jednotky na víko kontejneru.



Obrázek 3 - Nerovnoměrné rozprostření odpadu v kontejneru [4]

### 2.1.2 Měření kamerou s následným rozpoznáním obrazu

Velkoobjemové kontejnery na suš či stavební odpad se ultrazvukem měřit nedají z důvodu nemožnosti umístění senzoru na víko kontejneru a velikým rozměrům. Jediná možnost, jak je automaticky měřit, je pomocí kamery a softwaru na rozpoznání obrazu (obr. 4). Tato technologie by se dala použít i pro běžné odpadní nádoby, ale pořizovací a provozní náklady jsou několikanásobně vyšší než ultrazvukové měření.



Jelikož tato bakalářská práce se věnuje monitoringu běžných odpadových nádob, dále se této technologii nebude věnováno.



Obrázek 4 - Velkoobjemový kontejner s obrazovým senzorem [4]

### 2.1.3 Automatické systémy v České republice

V České republice, podle zástupce poradenské odpadové firmy Ing. Pavel Novák s.r.o., se automatické ultrazvukové systémy využívají zejména v podzemních kontejnerech, kde kontrola hladiny odpadu je velmi problematická.

První velký pokus o plošné nasazení sledování odpadových nádob se uskutečnil v roce 2016 ve spolupráci společnosti O2 IT services, svozové firmy AVE a města Kolín. Projekt spočíval v umístění 330 ultrazvukových měřicích zařízení do nádob na tříděný odpad [5]. Starosta města Kolín řekl, že tento projekt byl pouze z poloviny úspěšný, protože informace o zaplnění nádob městu sice pomohla optimalizovat rozmístění odpadových nádob a tím zabránit přeplnování a znečištění města, ale dramatické snížení nákladů nepřinesla [6]. Kolín se po uplynutí pilotního projektu, kdy měl službu zadarmo, rozhodl zredukovat počty umístění měřicích jednotek na 21 na pouze podzemní kontejnery a v projektu pokračovat v normálním placeném režimu. Hlavní důvod pro zredukování počtu měřicích jednotek byla cena jednoho zařízení. Při ceně 3000 Kč za jedno zařízení a vysokým měsíčním provozním poplatkem si město nemohlo dovolit vybavit měřicím zařízením všech 400 odpadových nádob [6].

## 2.2 Manuální systémy

Mnoho subjektů spoléhá na vlastní uživatele, u kterých předpokládá, že zaplněnou nádobu sami ohlásí. Ohlášení je řešeno pomocí umístění QR kódu, NFC nálepky, případně telefonního čísla. V případě QR kódu si uživatel stáhne příslušnou aplikaci do chytrého telefonu a při přeplnění či závadě nádoby QR kód naskenuje a tím odešle informaci svozové firmě. Podobné je to s NFC nálepkou – oskenováním NFC čipu telefonem se odešle informace o zaplnění. V případě telefonního čísla, informace je sdělena pomocí SMS nebo telefonního hovoru. Výhoda tohoto systému je přesná informace od člověka, tudíž chybovost měření by měla být podstatně menší než u automatických systémů. Dále je možnost sdělení

různorodé informace – automatické systémy sdělují v drtivé většině případů jenom informaci o zaplnění, sdělení od člověka může kromě zaplněnosti obsahovat informaci o poškození či jiném problému.

Velkým nedostatkem tohoto způsobu sledování odpadových nádob je malá uživatelská iniciativa. Pro uživatele je mnohem snazší, než někam volat nebo skenovat QR kód či NFC nálepku, odpad nechat vedle nádoby s tím, že se svozová firma s tím sama vypořádá. Málokdo si také při vynášení odpadu bere s sebou mobilní telefon.



Obrázek 5 - Odpadové nádoby s QR kódem v Kolině [6]

### 2.2.1 Manuální systémy v České republice

Systém telefonních čísel a QR kódů využívá více měst a městských částí v České republice. Úspěšnost tohoto systému je přímo úměrná schopnosti města iniciovat obyvatele k poskytování zpětné vazby. Dle zástupce odpadové poradenské firmy Ing. Pavel Novák s.r.o., se to českým městům moc nedaří a z tohoto důvodu manuální systémy jsou neefektivní a téměř nepřinášejí žádný přínos pro zlepšení svozu odpadů.

Město Příbram v roce 2017 nainstalovalo QR kódy na všechny své kontejnery. Dle zástupce města odboru pro odpadové hospodářství, za poslední půlrok přišly pouze jednotky oznámení.

## 2.3 Topologie a požadované parametry měřící jednotky

Žadatel řešení monitoringu sběrných nádob – klient firmy Ing. Pavel Novák s.r.o. - vyžaduje automatický sběr dat bez zásahu člověka. Na trhu je více firem (Smartbin, eCube, SmartUp Cities, O2,...) poskytující kompletní řešení monitoringu jako službu, ale pořizovací cena a provozní náklady měřící jednotky jsou pro české subjekty velmi vysoké. Jediná firma na trhu nabízející samostatnou měřící jednotku k provozování vlastní služby je italská firma BARON.it. Společnost Ing. Pavel Novák s.r.o. tři měřící jednotky koupila a použila je na testovací provoz. Měřící jednotky se neosvědčily. Byly velmi rozměrné, čímž znesnadňovaly manipulaci se sběrnou nádobou a s cenou 150 euro za kus ekonomicky neúnosné. Z tohoto důvodu bylo rozhodnuto vyvinout měřící jednotku vlastní.

### 2.3.1 Požadované parametry měřící jednotky

- Automatický sběr dat
- Ultrazvuková technologie měření
- Napájení z baterie s výdrží alespoň dva roky
- Pořizovací cena za kus nižší než 1000 Kč

## 3 Současné technologie pro bezdrátový přenos dat pro IoT

---

Pro správnou funkčnost zařízení internetu věcí, a tedy i měřících jednotek pro monitoring odpadových nádob, je potřeba zajistit vhodnou komunikaci pro přenos dat. Na trhu je spousta technologií. Navzájem se liší frekvenčním pásmem, dosahem, energetickou náročností či použitými protokoly. Z těchto důvodů nelze říci, že jedna technologie je lepší než druhá. Vždy záleží na konkrétních požadavcích systému. V této kapitole se bude věnováno rozboru jednotlivých technologií a na závěr bude provedeno srovnání a výběr nejvhodnější pro měřící jednotky.

### 3.1 GSM

Globální systém pro mobilní komunikaci je technologie, které se také říká mobilní technologie druhé generace – 2G. Je používána pro přenos hlasu a SMS na frekvenčním pásmu 1800MHz (Band 3) a 900MHz (Band 8) [7]. Rozšíření 2G sítě je GPRS, někdy také nazýváno jako síť 2.5 generace. Dalším vývojovým krokem GPRS je EDGE. Tato technologie pouze přináší především novou modulaci, které umožňuje efektivnějšího přenosu dat a tím i vyšších rychlostí [8]. Veřejnou GSM síť v České republice provozují firmy Vodafone ČR a.s., O2 Telefónica ČR a.s., T-Mobile ČR a.s. na frekvencích viz tabulka 1.

Tabulka 1 - Operátoři v ČR a jejich frekvenční pásma pro GSM [7]

Operátor v České republice	Frekvenční pásma pro GSM 2G	Frekvenční pásma pro GSM 3G
T-Mobile ČR a.s.	900/1800MHz	2100MHz
O2 Telefónica ČR a.s.	900/1800MHz	2100MHz
Vodafone ČR a.s.	900/1800MHz	2100MHz

**Výhody:** pokrytí celé České republiky, velké množství operátorů, spolehlivost, roaming free v Evropské Unii

**Nevýhody:** velká spotřeba energie při přenosu dat, drahý měsíční tarif za SIM kartu

## 3.2 LTE

O LTE se někdy hovoří jako o síti 4. generace (4G), což ale není pravda. Požadavky pro plnohodnotné 4G dle Mezinárodní telekomunikační unie jsou přenosové rychlosti nad 1 Gb/s pro statické zařízení a 100 Mb/s pro rychle se pohybující se zařízení [9]. V České republice LTE bylo přiděleno pět pásem – Band 1 (2100MHz), Band 3 (1800MHz), Band 7 (2600MHz), Band 8 (900MHz) a Band 20 (800MHz) [7]. Dnes čeští operátoři na drtivé většině území provozují LTE na 800MHz, ale stále jsou i oblasti s 1800MHz a 2100MHz (tabulka 2). Další vývojový mezikrok mezi LTE a 4G je LTE-Advanced. Tato technologie se označuje jako síť 3.9 generace a umožňuje ještě vyšší rychlosti.

Tabulka 2 - Operátoři v ČR a jejich frekvenční pásma pro LTE [7]

Operátor v České republice	Frekvenční pásma pro LTE
T-Mobile ČR a.s.	800/1800/2100 MHz
O2 Telefónica ČR a.s.	800/1800/2100 MHz
Vodafone ČR a.s.	800/900/1800/2100 MHz

**Výhody:** vysoká přenosová rychlost, spolehlivost, roaming free v Evropské Unii

**Nevýhody:** velká spotřeba energie při přenosu dat, vysoká cena LTE modulů, drahý měsíční tarif

## 3.3 Bluetooth

Technologie Bluetooth je navržena pro bezdrátovou komunikaci na malé vzdálenosti. Používá frekvence od 2,400GHz až 2,483GHz [11].

**Výhody:** malá spotřeba energie

**Nevýhody:** krátký dosah, placená licence, nekompatibilita mezi jednotlivými verzemi, přehlcené pásmo 2.4 GHz, vysoká tarifní cena za zařízení, neveřejná síť v ČR

## 3.4 Wi-Fi

Wi-Fi je technologie pro bezdrátovou komunikaci v počítačových sítích. Funguje v tzv. bezlicenčních pásmech 2.4 GHz a 5 GHz – tj. pásma, které uživatel může volně používat aniž by si musel žádat o speciální povolení (licenci) [12]. Využívá se pro levné budování bezdrátových sítí a dnes je masivně rozšířená. První WiFi standard IEEE 802.11 přišel v roce 1997, pracoval v pásmu 2,4 GHz a umožňoval maximální teoretické přenosové rychlosti až 2 Mbit/s [13]. Nejnovější dnes běžně používaný standard je IEEE 802.11ac. Pracuje v pásmu 2.4GHz a 5GHz a umožňuje maximální rychlost až 1000 Mbit/s [13]. Dosah je střední vzdálenosti – v budovách řádově v jednotkách až desítkách metrů, v otevřeném prostranství až stovky metrů. Dosah je velmi ovlivněn. V České republice neexistuje veřejná síť.

**Výhody:** vysoké přenosové rychlosti, cena

**Nevýhody:** vysoká spotřeba energie, neexistuje veřejná síť v ČR

### 3.5 ZigBee

ZigBee standard IEEE 802.15.4 je bezdrátová technologie používaná zejména v průmyslových aplikacích. ZigBee je podobné Wi-fi – také používá 2,4GHz pásmo, ale má mnohem menší spotřebu energie a je navrženo pro rychlosti kolem 250 kb/s [14]. Dosah je až 30 m pro jediné zařízení (záleží na překážkách), ale dá se prodloužit pomocí „přeposílačů“ signálu až na teoreticky neomezenou potřebnou vzdálenost. V České republice neexistuje veřejná síť.

**Výhody:** vysoká bezpečnost, malá spotřeba energie

**Nevýhody:** poměrně malý dosah, zahlcené pásmo 2,4 GHz, malé přenosové rychlosti, neexistuje veřejná síť v ČR

### 3.6 SigFox

SigFox je technologie speciálně vyvinutá pro IoT. Její specifikum je přenos malých objemů dat na relativně velké vzdálenosti (až 10 km v městské zástavbě a 30 km v otevřeném prostranství) při malé spotřebě energie. SigFox je velmi odolný vůči překážkám. Pro přenos dat využívá 868 MHz frekvenční pásmo a přenosové rychlosti se pohybují kolem 100 bit/s [15]. Tato technologie je také velmi bezpečná a spolehlivá. V Česku veřejnou síť SigFox provozuje firma SimpleCell Networks a.s.

**Výhody:** malá spotřeba energie, poměrně velký dosah, roaming free, veřejná síť v ČR provozovaná operátorem

**Nevýhody:** malá rychlost, jednostranná komunikace

### 3.7 LoRa

Lora je bezdrátová technologie také vyvinutá hlavně pro účel IoT. V Evropě pracuje v bezlicenčním frekvenčním pásmu 868 MHz, v USA 915 MHz. Její hlavní výhoda je relativně velký dosah (více než 10 km v městské zástavbě) a malá spotřeba energie jak při odesílání, tak přijímání dat. Rychlosti dosahují až 250 kb/s [16].

#### 3.7.1 LoRaWAN

LoRaWAN definuje samotný komunikační protokol a síťovou architekturu pro vzájemnou jednotnou komunikaci zařízení v síti [16].

Ve zkratce, LoRa je samotná technologie pro přenos dat, LoRaWAN je protokol, který zajišťuje správnou komunikaci mezi veřejnou LoRa sítí a zařízeními.

**Výhody:** malá spotřeba energie, relativně dlouhý dosah, veřejná síť v ČR provozovaná operátorem

**Nevýhody:** malá rychlost, jednostranná komunikace

### 3.8 Narrowband IoT

Nb-IoT je speciální úzkopásmová síť speciálně vyvinutá pro IoT. Příznačné pro tuto technologii je malá spotřeba energie při komunikaci, dobrá prostupnost signálu (v městské zástavbě, budovách, hustě zalesněným terénem atd.) a oboustranná komunikace. Pracuje v licencovaném pásmu 700 až 900 MHz a přenosová rychlost je 50 kb/s [17]. V České republice veřejnou síť provozuje Vodafone.

**Výhody:** malá spotřeba energie, dobrá propustnost signálu, oboustranná komunikace, veřejná síť v ČR provozovaná operátorem

**Nevýhody:** malá přenosová rychlost

### 3.9 Výběr vhodné komunikační technologie

Od přenosové technologie pro měřicí jednotku je požadována pouze veřejná síť s pokrytím minimálně celé České republiky. Energetická náročnost přenosu se může zanedbat, protože měřicí jednotka bude odesílat data pouze jednou denně po dobu několika vteřin. Z toho vychází pět použitelných technologií – GSM, LTE, SigFox, LoRa a Nb-IoT. V době, kdy se navrhovalo zařízení, specializované technologie pro IoT – LoRa, SigFox a Nb-IoT – neměly plné pokrytí České republiky, což by se mělo změnit během roku 2018. LTE moduly jsou mnohem dražší než GSM moduly a jediná výhoda, která tato technologie má, je vyšší rychlost, což tomto případě není klíčové. Z výše uvedených důvodů byla vybrána technologie GSM. V budoucnu by měla být technologie přenosu dat změněna na vhodnější hlavně z důvodu poměrně vysokých měsíčních tarifů za SIM kartu pro GSM komunikaci. Operátoři provozující sítě LoRa, SigFox a Nb-IoT mají několikanásobně nižší měsíční tarifní ceny.

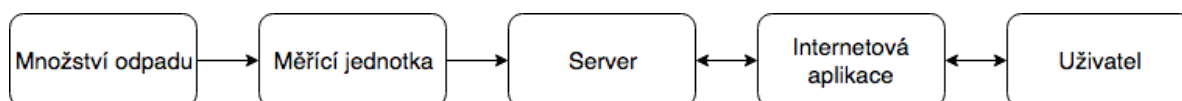
## 4 Části systému

---

### 4.1 Koncept systému

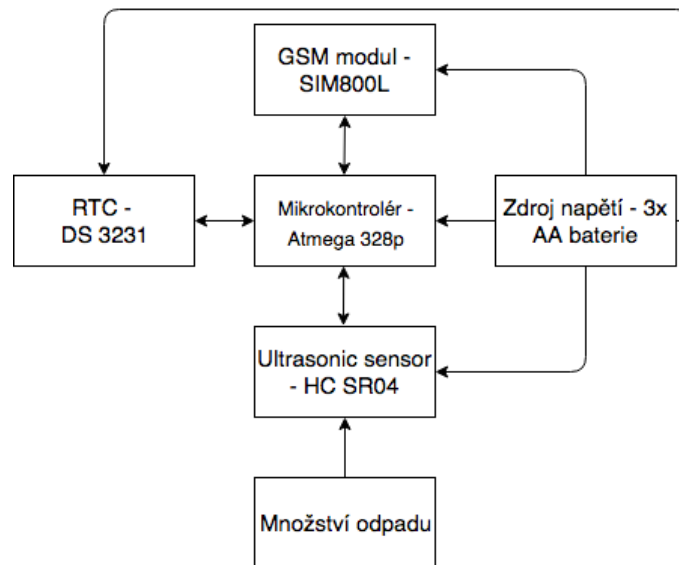
Celý systém se skládá z dvou hlavních částí – z měřicí jednotky v odpadové nádobě a serveru, na kterém běží aplikace a který zajišťuje příjem, uložení a interpretaci informací (obr. 6).

Množství odpadu bude měřeno pomocí ultrazvukového senzoru, který předá informaci mikrokontroléru a ten využívajíc GSM síť, určenou na základě výsledku srovnání bezdrátových technologií pro přenos dat v kapitole 3, odešle data na server, kde se uloží do databáze. Uživatelsky přívětivá interakce mezi uživatelem a databází je realizována pomocí internetové aplikace.



Obrázek 6 - Blokový diagram konceptu systému

## 4.2 Měřicí jednotka



Obrázek 7 - Blokový diagram měřicí jednotky

Měřicí jednotka má za úkol jednou denně změřit zaplněnost odpadové nádoby a předat informaci serveru, který ji následně zpracuje. Jelikož servis je časově a technologicky náročný, jednotka by měla vydržet fungovat alespoň dva roky bez jakéhokoliv zásahu. Blokový diagram měřicí jednotky je na obrázku 7.

### 4.2.1 Mikrokontrolér

Mikrokontrolér řídí komunikaci mezi jednotlivými moduly a kontroluje celý proces měření odpadu. Od mikrokontroléru je v této aplikaci požadována funkce uložení do hlubokého spánku, pracovní napětí od 3 V do 5 V, vybavenost alespoň 15 I/O piny a vnějším přerušením. Dále s tímto čipem bude použita nadstavba Arduino, která pomocí velkého počtu knihoven urychlí programování. Programování bude provedeno v prostředí Arduino IDE, v němž se programuje v jazyce Arduino programming language (programovací jazyk Arduino), který je odvozený od jazyka C.

Z důvodu použití nadstavby Arduino, musí být vybrán mikrokontrolér, pro který jsou napsané Arduino knihovny. Byl vybrán Microchip ATmega 328p (obr. 8), protože to je nejběžněji používaný mikrokontrolér pro Arduino programování, splňuje veškeré požadavky a je velmi finančně dostupný.



Obrázek 8 - Microchip ATmega328P [18]

### 4.2.2 Zdroj napětí

Požadavky na zdroj napětí jsou následující – pracovní napětí 3 – 5 voltů, pracovní teplota od



-20°C do 40°C a dostatečná energie pro funkci měřící jednotky alespoň po dobu dvou let. Je technicky velmi nepohodlné a obtížné vést ke každému senzoru napájecí kabel z rozvodné sítě, tudíž senzory budou operovat na baterii. Aby baterie vydržela dlouhou (více než dva roky) dobu bez jakéhokoliv zásahu, je potřeba malé úrovně samovybití. Samovybití je přímo ovlivněno teplotou a typem baterií [19]. Nejnižší úrovně samovybití dosahují baterie alkalické a lithiové [20]. Jelikož cena lithiových baterií obvykle dosahuje několikanásobku ceny alkalických baterií, vybrány byly baterie alkalické.

Běžné AA alkalické baterie na trhu mají kapacitu kolem 2,7 Ah a pracovní napětí od 1,5 V (plně nabitě) do 1 V (plně vybité). Spojením tří baterií do série v jeden battery pack, maximální výstupní napětí bude 4,5 V a minimální napětí 3 V při úplném vybití. V tomto napěťovém rozhraní jsou schopny veškeré použité komponenty pracovat, tudíž není potřeba použití žádného napěťového měniče. Mikrokontrolér bude při každém měření odpadu také měřit napětí baterií a při kriticky nízkých hodnotách pošle upozorňovací zprávu.

#### 4.2.3 Výpočet výdrže měřící jednotky

Senzor se nachází vždy ve dvou módech – pracovní mód a spící mód. Při spícím módu odhadovaná spotřeba celého systému je 10  $\mu$ A a při pracovním módu 150 mA. Pracovní mód trvá 30 vteřin jednou za 24 hodin, zbytek dne senzor spí. Z těchto informací lze vypočítat spotřebu energie za jeden den.

$$S_s = \text{Spotřeba}_{\text{spící mód}} = 23.9917 \text{ hodin} * 10\mu\text{A} \approx 2.39917 * 10^{-4} \text{ Ah}$$

$$S_p = \text{Spotřeba}_{\text{pracovní mód}} = 8.333 * 10^{-4} \text{ hodin} * 150\text{mA} \approx 1.25 * 10^{-3} \text{ Ah}$$

$$\text{Celková denní spotřeba energie} = S_p + S_s \approx 1.49 * 10^{-3} \text{ Ah}$$

$$\text{Celková roční spotřeba energie} \approx 1.49 * 10^{-3} \text{ Ah} * 365 \approx 0.54 \text{ Ah}$$

Není potřeba přepočítávat na watt hodiny, protože zdroj i spotřebič mají vždy stejné napětí.

Baterie budou zapojeny v sérii, tudíž jejich počáteční napětí je 4,5 voltu a kapacita 2.7 Ah. Při vzatí do úvahy roční hodnotu samovybití 3 % [20] a samotné roční spotřeby energie 0.54 Ah, výdrž na jeden battery pack je více než čtyři roky (tabulka 3), což plně splňuje požadavek, který byl stanoven na dva roky.

$$\text{Kapacita rok}_k = \text{Kapacita rok}_{k-1} * 0.97 - \text{roční spotřeba energie [Ah]} \quad (2)$$

Tabulka 3 - Kapacita baterií v jednotlivých letech

Začátek roku	Kapacita (Ah)
0	2.7
1	2.08
2	1.47
3	0.88
4	0.31
5	0.00



#### 4.2.4 Ultrazvukový senzor vzdálenosti

V kapitole č. 2 bylo rozhodnuto o použití ultrazvukové technologie měření. Základní princip ultrazvukového senzoru spočívá ve vyslání signálu v podobě zvuku o vysoké frekvenci, který putuje prostředím, odrazí se od překážky a vrací se zpátky k senzoru, kde je zaznamenán přijímačem. Při známe rychlosti zvuku (zhruba 340 m/s při 20 °C) a času od vyslání signálu po přijetí, lze vypočítat vzdálenost překážky od senzoru ze vzorce 3.

$$Vzdálenost = \frac{\text{čas od vyslání po přijetí signálu} * v_{zvuk}}{2} [m] \quad (3)$$

*Výpočet vzdálenosti od ultrazvukového senzoru*

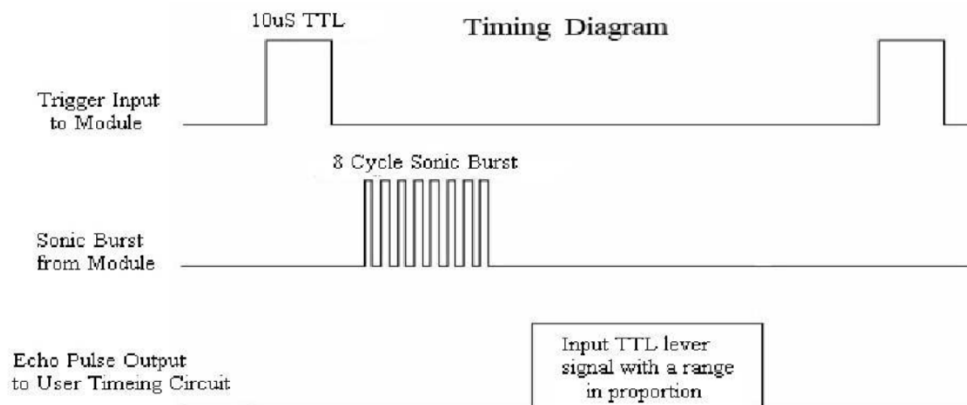
##### 4.2.4.1 Porovnání ultrazvukových senzorů

V tabulce 4 je srovnání vybraných řešení ultrazvukových senzorů

*Tabulka 4 - Porovnání ultrazvukových senzorů*

Značka a typ	HC – SR04	LV-MaxSonar MB1000	RB-Mab-31
Rozsah měření	2 – 400 cm	15 – 645 cm	3 – 400 cm
Napájecí napětí	3 – 5.5 V	2,5 – 5,5 V	5 V
Komunikace	I <sup>2</sup> C	Sériová	I <sup>2</sup> C
Měřicí úhel	30°	32°	30°
Cena	3,95\$	25,95\$	19,99\$

Byl vybrán ultrasonický senzor HC – SR04 (obr. 10) pro jeho cenu, dostupnost a vyhovující parametry. Senzor má čtyři piny – Vcc, GND, Echo a Trig. Vcc a GND slouží pro napájení. Echo a Trig pro samotnou komunikaci mezi mikrokontrolérem a senzorem. Na obrázku 9 je vidět, že pro úspěšné změření vzdálenosti je potřeba poskytnout 10 μs dlouhý puls na pin Trig. Senzor následně vyšle zvuk o frekvenci 40 kHz, který se odrazí od překážky a vrátí se zpět. Návrat je senzorem zaznamenán a ten dává okamžitě vědět mikrokontroléru na pinu Echo. Mikrokontrolér změří délku Echo signálu, který je proporcionální času mezi odesláním a přijmutím ultrazvukového signálu a pomocí vzorce 3 vypočítá vzdálenost překážky od senzoru.



Obrázek 9 - Časový diagram komunikace ultrazvukového senzoru [21]



Obrázek 10 - Ultrazvukový senzor HC – SR04 [21]

#### 4.2.5 GSM modul

V kapitole č. 3 bylo rozhodnuto pro přenos informací mezi senzorem a serverem použít GSM datových přenosů. Pro připojení a využití GSM sítě je potřeba modulu, který se sítí umí komunikovat. Byl zvolen model SIM800L od firmy SIMCom (obr. 11). Je to jediný GSM modul na trhu, který je od renomovaného výrobce a má cenu pod 10\$. Tento modul plně podporuje frekvence 2G sítě českých operátorů (900/1800MHz), umožňuje přenášet hlas, SMS a data. Výměna informací mezi GSM modulem a mikrokontrolérem probíhá pomocí AT příkazů přes sériovou asynchronní komunikaci – UART.



Obrázek 11 - GSM modul SIM8001 [22]

### 4.2.5.1 AT příkazy

AT příkazy jsou odvozeny od slova „Attention“ („pozor“) a jsou to příkazy na ovládání různých síťových prvků – nejčastěji modemů, případně GSM/LTE modulů jako v tomto případě. Příkazy nejsou standardizované, každý přístroj má pro sebe specifické. Všechny příkazy začínají předponou „AT“. Po předponě následuje samotný příkaz. Příkazy se dělí na dvě části – základní a rozšířené. Základní příkazy jsou takové, které začínají bez „+“. Např. „ATD“ (Dial – vytočit číslo) nebo „ATA“ (Answer – odpovědět na vyzvánění). Rozšířené příkazy začínají znakem „+“. Např. „AT+CPIN“ (zadej pin).

### 4.2.5.2 Sériová komunikace (UART)

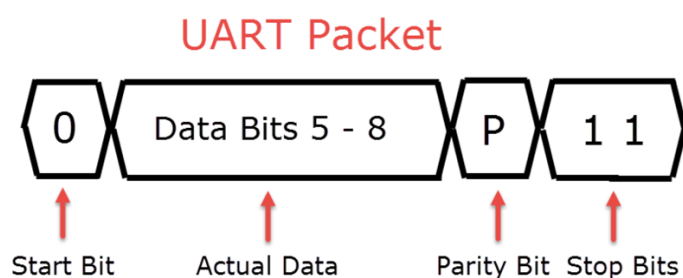
UART je fyzický kus mikrokontroléru, který pomocí dvou pinů Rx (Receive) a Tx (Transmit) zajišťuje přijímání a odesílání dat. Jelikož se jedná o asynchronní komunikaci, vysílač a přijímač nemají společný generátor hodinového signálu, ale každý má vlastní. Pro správnou funkci komunikace je tedy potřeba, aby vysílač i přijímač měly naprosto stejné nastavení. Nastavuje se:

**Baud rate** – Nastavuje, jak rychle data budou přenášena. Jednotky jsou v bitech za vteřinu. Rychlost může být jakákoliv, jediná podmínka je, že přijímač i vysílač musí mít nastavenou stejnou hodnotu. Standardní rychlosti jsou 1200, 2400, 4800, 9600, 19200, 38400, 57600, a 115200 bit/s. Při vyšších rychlostech se mohou začínat objevovat chyby v přenosu.

**Parity bit** – Parity bit je jednoduchý způsob ověření chyb. Pokud je parity zapnutý, může být nastavený na lichý či sudý. Znamená to, že každý kus přenesených dat (paket) musí mít součet buď sudý nebo lichý. Např. Předpokládáme, že parity je nastaveno na lichý a je přidáno k bajtu 0b01011101, což má lichý součet. V tomto případě přidaný parity bit by byl 0. Pokud by parity byl nastavený na sudý, v předchozím případě by měl hodnotu 1.

**Stop bit** – Stop bit určuje konec přenášeného paketu. Může být nastavený jako jednobitový nebo dvoubitový.

Příklad struktury UART paketu je na obrázku 12.



Obrázek 12 - Grafické znázornění UART datového paketu [23]

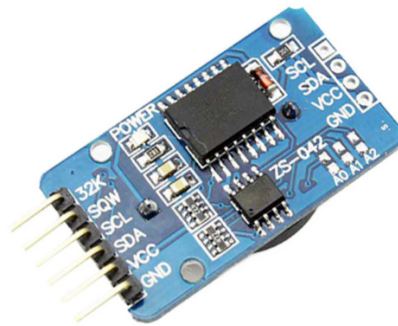
### 4.2.6 RTC

Zapnutý mikrokontrolér v pracovním módu se všemi periferiemi má spotřebu proudu kolem 150 mA. Kdyby byl v pracovním módu neustále, vydržel by na baterii jenom několik málo dní. Z tohoto důvodu se veškeré periférie vypínají a mikrokontrolér ukládá do hlubokého spánku, kde spotřeba elektrické energie je v řádech několika mikroampérů. Z hlubokého spánku je nutné mikrokontrolér jednou za čas

probudit, aby provedl měření a odeslal data na server. Probuzení je provedeno pomocí RTC, které jednou v určitý nastavený čas vyšle signál do mikrokontroléru a ten se vzbudí.

Probuzení probíhá následujícím způsobem: během spánku má mikrokontrolér nastavený interrupt (přerušení) na určitém pinu, kde při změně logického levelu je vzbuzen. Logický level RTC změní, když se nastavený čas alarmu v RTC shoduje s aktuálním časem.

Byl zvolen model DS3231 (obr. 13). Tento konkrétní model byl vybrán z důvodu nízké ceny, vybaveností dvěma alarmy, velmi nízké spotřeby energie a dá se zakoupit už jako hotové řešení na desce plošného spoje. Komunikace – nastavení času a alarmů - mezi mikrokontrolérem a RTC probíhá pomocí I<sup>2</sup>C sběrnice [24].



Obrázek 13 - RTC DS3231 modul [25]

#### 4.2.6.1 I<sup>2</sup>C

I<sup>2</sup>C je komunikační sběrnice sloužící pro přenos dat mezi jednotlivými zařízeními. Obousměrný provoz probíhá po dvou drátech – hodiny SCL a data SDA. Jedná se o synchronní komunikaci – tj. komunikaci, kde je společný hodinový signál generovaný tzv. masterem. Na jednu linku může být připojeno více zařízení (slaves). Každý slave má unikátní adresu pro jednoznačnou identifikaci. Master, který chce odeslat či přijmout data, musí nejdříve definovat adresu cílového zařízení a zda půjde o čtení či zápis. Přenos dat má několik fází:

**Klidový stav** – Na SDA i SCL jsou logické jedničky a neprobíhá žádný přenos.

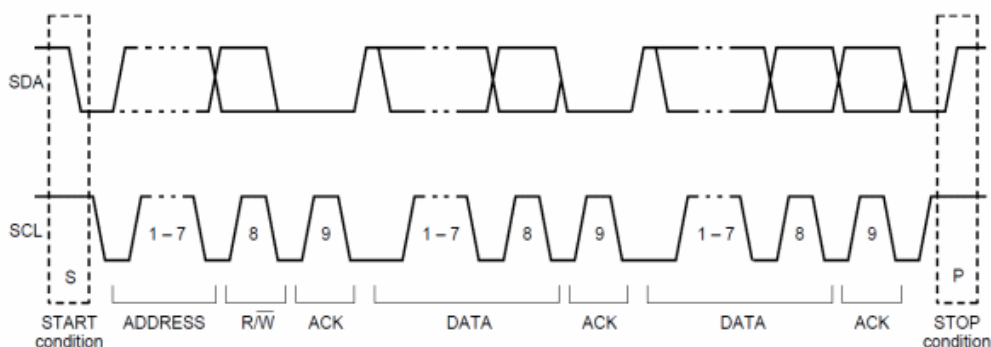
**Start bit** – Zahájení přenosu. SDA se změní z 1 na 0, SCL zůstává 1.

**Přenos dat** – Samotný přenos informací. SDA se může měnit pouze když je SCL v logické 0. Data se přenášejí po 8 bitech.

**Acknowledge bit** – Příjímačem generovaný bit sloužící k potvrzení správného příjmu dat. Pokud příjem dat byl proveden bezchybně, příjímač odešle 0. Logická 0 také znamená, že zařízení je připraveno na další příjem dat. Pokud byl příjem chybný, příjímač odešle 1.

**Stop bit** – Ukončení přenosu. SCL je v logické 1 a SDA se změní z 0 na 1.

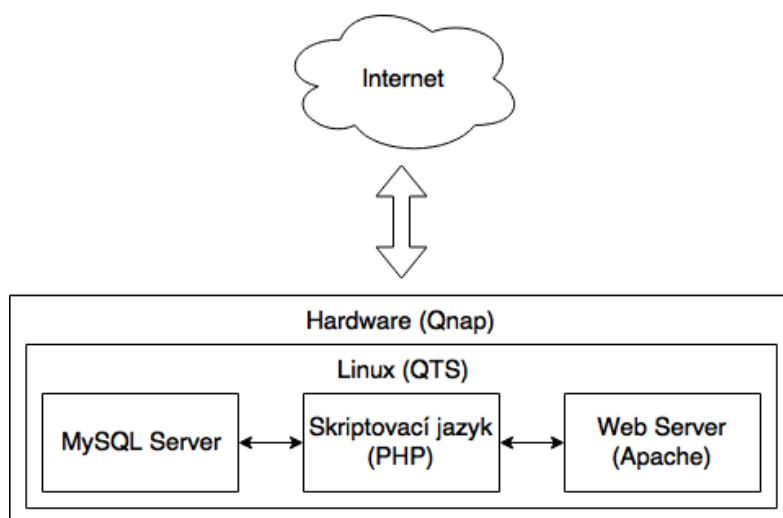
Průběh I<sup>2</sup>C komunikace je na obrázku 14.



Obrázek 14 - Grafické znázornění průběhu I<sup>2</sup>C komunikace [26].

## 4.3 Server

V předchozím textu bylo několikrát odkázáno na slovo „server“. Tímto výrazem byl myšlen hardware a sada softwaru, které jsou zapotřebí pro příjem, zpracování a uložení dat a běh webové aplikace pro pohodlnou uživatelskou obsluhu. Diagram serveru je na obrázku 15.



Obrázek 15 - Blokové schéma serveru

### 4.3.1 Hardware

Celý systém běží na NASu TS-231P od firmy Qnap. Tento síťový disk disponuje dvoujádrovým procesorem ARM Cortex A9 o frekvenci 1800 Mhz a 1GB RAM operační paměti, což mu dává dostatečný výkon pro běh operačního systému Linux na kterém běží všechny aplikace, které zajišťují správnou funkci celého webového systému.

### 4.3.2 Software – LAMP



Obrázek 16 - Lamp logo [27]

LAMP je balíček svobodného softwaru používající se pro implementaci dynamických webových stránek a aplikací. Zkratka **LAMP** je odvozená od názvů jeho jednotlivých komponent – Linux operační systém, Apache HTTP server, **MySQL** databáze a **PHP** programovací jazyk.

LAMP se stal velmi populární zejména kvůli svobodně šiřitelné licenci (open-source) všech jeho jednotlivých komponent, což umožňuje instalaci a provoz bez jakýchkoliv licenčních poplatků.

#### 4.3.2.1 Linux

Linux je unixový operační systém pod licencí open source. Operační systém je nejnižší level softwaru, který běží přímo na hardwaru a zajišťuje správnou komunikaci mezi ním a ostatními aplikacemi. Na Qnap TS-231P je nainstalovaná linuxová distribuce QTS 4.0 vyvinutá přímo pro tento NAS.

#### 4.3.2.2 Apache server

Apache je webový server. Webový server se stará o komunikaci mezi klientem a webovým obsahem. Na základě konkrétního požadavku od klienta mu předá přímo požadovaný obsah (webovou stránku), případně spustí potřebné moduly a skripty pro generování požadovaného obsahu.

Apache server je vyvíjený a udržovaný společností Apache Software Foundation. Je velmi populární kvůli jeho rychlosti, zabezpečení, spolehlivosti a možnosti uzpůsobení přesně pro potřebné požadavky a úkoly. Je multiplatformní podporující operační systémy Linux, Windows a Mac OS.

#### 4.3.2.3 PHP skriptovací jazyk

Pro generování obsahu a komunikaci mezi webovým serverem a MySQL databází je potřeba skriptovací jazyk. Byl vybrán jazyk PHP, protože je poměrně snadný na pochopení a je nejrozšířenějším serverovým jazykem.

PHP je navržený zejména pro webové stránky a aplikace, nicméně dá se použít i pro obecné účely. PHP kód je interpretován webovým serverem pomocí PHP modulu, který generuje samotnou webovou stránku. PHP nemusí být jako samostatný soubor, ale může být implementován přímo do HTML kódu. PHP je svobodný software vydaný pod PHP licenci. V softwarovém balíku LAMP může být PHP nahrazeno jazykem Perl či Python.

#### 4.3.2.4 MySQL databáze

Data ze senzoru se musejí strukturovaně ukládat do nějakého strukturovaného celku pro budoucí možnou interpretaci. K tomu jsou využívány serverové databáze. Pro systém monitoringu zaplněnosti odpadových nádob byla vybrána MySQL databáze pro její volně šiřitelnou licenci a rozšířenost.

MySQL je open-source databázový systém využívající relační model. Tento databázový systém je multiplatformní a komunikace s ním probíhá pomocí jazyka SQL.

##### 4.3.2.4.1 Relační model

V tomto modelu jsou data organizována do tabulek (entity), které jsou navzájem určitým způsobem závislé – je mezi nimi logická vazba (relace). Tabulky se skládají ze sloupců a řádků. Sloupce reprezentují vlastnosti entity a musí mít jedinečný název a určený datový typ. Tyto vlastnosti se také nazývají atributy (např. obr. 17 - id, ulice, latitude, longitude, majitel). Řádky představují jednotlivé záznamy – jeden řádek reprezentuje jeden záznam. Každý řádek by měl obsahovat unikátní identifikátor, podle kterého jde určit konkrétní záznam – primární klíč.

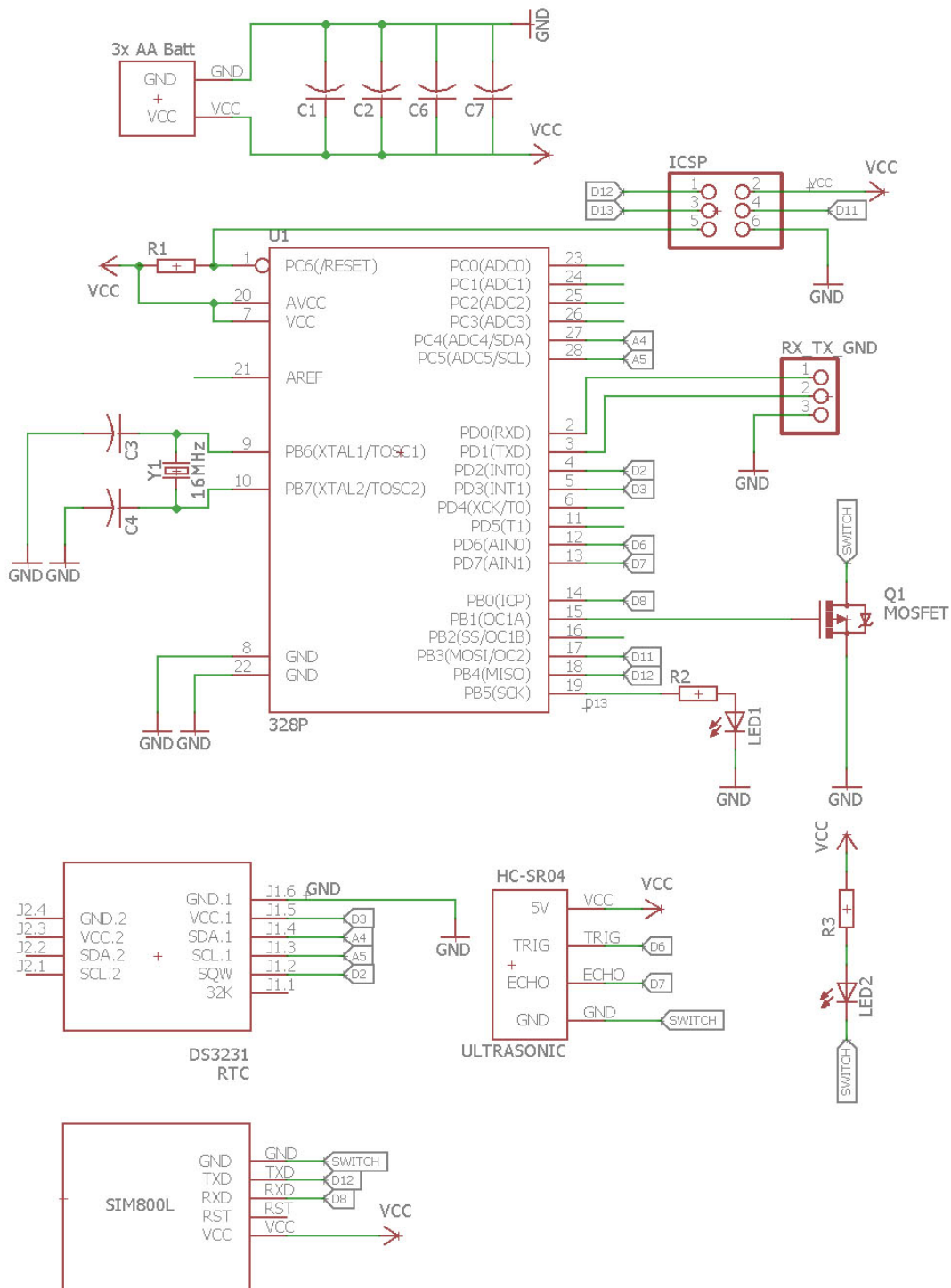
id	ulice	latitude	longitude	majitel
JER1	Jeremenkova ul.	50.045329	14.420782	potex
KVV1	Kvetnoveho v.	50.033570	14.505073	potex
BOL1	Boletova ul.	50.008831	14.417105	potex

Obrázek 17 Relační model databáze

# 5 Realizace měřicí jednotky

## 5.1 Hardware

### 5.1.1 Elektrické schéma



Obrázek 18 - Elektrické schéma měřicí jednotky



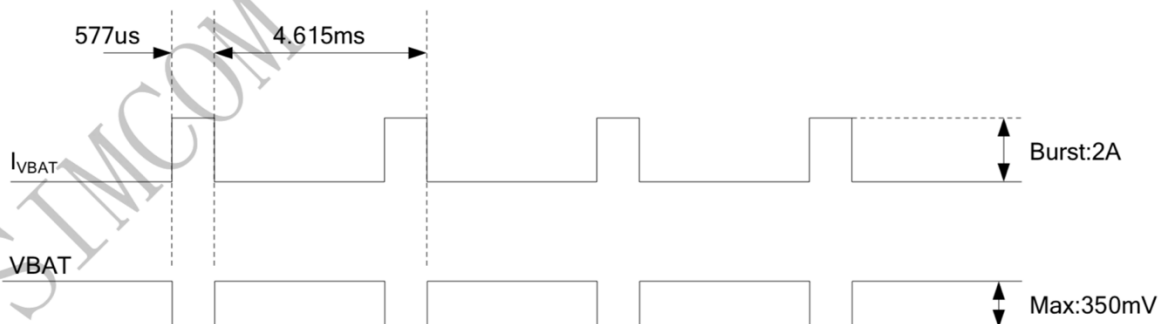
### 5.1.1.1 Blok 3x AA BATT

Blok „3x AA BATT“ jsou dva piny, na které je připojen zdroj napětí složený z tří AA baterií. Mezi pozitivní větví (VCC) a zemí (GND) jsou připojeny čtyři blokovací kondenzátory. C1, C2 a C6 jsou filtrační blokovací kondenzátory s hodnotami 0.1  $\mu\text{F}$ , 0.1  $\mu\text{F}$  a 10  $\mu\text{F}$ . C1 a C2 mají za úkol odfiltrovat vysokofrekvenční složky v obvodu tak, že zmíněné složky svedou na zem. C6 slouží jako zdroj energie při náhlém proudovém odběru, aby nekleslo napětí v obvodu.

GSM modul při odesílání dat má proudové špičky až 2 A po dobu 577  $\mu\text{s}$  (obr. 19). Při takto náhlých velkých proudových impulsích zdroj energie z baterií nedokáže dost rychle reagovat a klesá napětí v systému. Pokud napětí klesne o více než  $\Delta U = 350 \text{ mV}$ , GSM modul se resetuje. Aby se tomuto zabránilo, je přidán lokální blokovací kondenzátor C7, který vykryvá nedostatky energie. Kapacita lokálního blokovacího kondenzátoru se vypočítá z vzorce 4, kde  $I_{\text{imp}}$  je amplituda impulsu,  $t_{\text{imp}}$  je čas impulsu a  $\Delta U_{\text{nap}}$  je maximální dovolený pokles napětí. C7 tedy musí být větší 330  $\mu\text{F}$ . Pro kondenzátor C7 jsem zvolil hodnotu 470  $\mu\text{F}$ .

$$C_L > \frac{I_{\text{imp}} * t_{\text{imp}}}{\Delta U_{\text{nap}}} [F] \quad (4)$$

*Výpočet velikosti blokovacího kondenzátoru [38]*



*Obrázek 19 - Proudové špičky GSM modulu [28]*

### 5.1.1.2 Blok U1

Blok U1 reprezentuje mikrokontrolér Atmega 328p. Pin 1 (PC6/RESET) je reset pin, který se chová jako „active low“ tj. pokud je na pinu napětí blízké 0 V, mikrokontrolér se resetuje. Z tohoto důvodu mezi tento pin a VCC je umístěn pull up odpor R1, který drží hodnotu napětí na VCC.

Na Pinu 9 (PB6) a 10 (PB7) je připojen krystalový oscilátor Y1, který určuje, na jaké frekvenci bude mikrokontrolér pracovat. Pro Atmega 328p je maximální možná pracovní frekvence 20 MHz [29]. Byl vybrán oscilátor o frekvenci 16 MHz, protože veškeré časové funkce Arduino knihovny jsou napsané pro tuto frekvenci. Kdyby byla zvolena jiná frekvence, časové funkce jako delay(), Serial.xxx() by nefungovaly správně a musely by se přepočítávat na konkrétní zvolenou frekvenci. Kondenzátory C3 a C4 mají hodnotu 22 pF a slouží ke správné oscilaci oscilátoru. Jejich hodnota je definována v datasheetu krystalu [30].

LED1 je připojena na pin 19 (PB5) a signalizuje zapnutí mikrokontroléru. Pokud by byla dioda připojena přímo do země, vznikl by zkrat a mikrokontrolér by se mohl vysokým proudem poškodit. Proto je v obvodu odpor R2, který slouží k omezení proudu. Hodnota R2 se vypočítá z vzorce 5.

$$R2 > \frac{VCC - V_F}{I_{max}} \quad (5)$$

*Výpočet odporu R2*

Maximální proud z I/O pinů Atmega 328p je 40 mA [31]. Pokles napětí  $V_f$  na diodě je 2,2 V a max napětí VCC v obvodu je 4,5 V [32]. Odpor R2 tedy musí být větší než 57 Ohmů. Byla vybrána hodnota odporu 220 Ohmů.

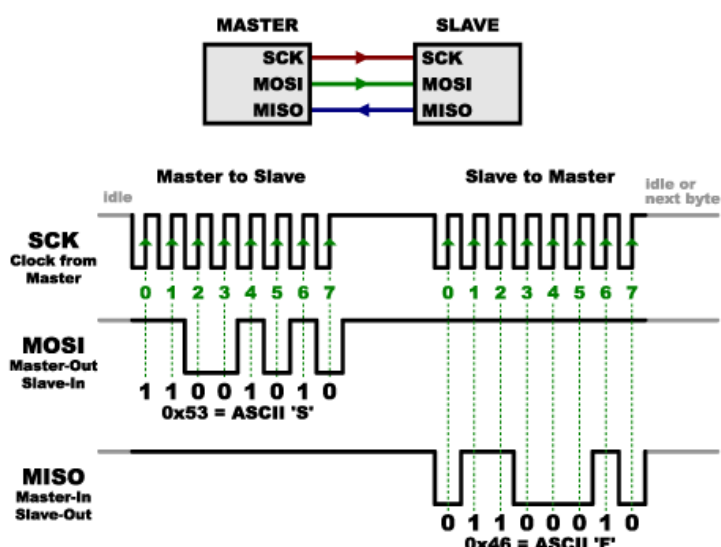
### 5.1.1.3 Blok RX\_TX\_GND

Tento blok slouží k sériové komunikaci při debuggování mezi mikrokontrolérem a počítačem. Skládá se z pinu RX (Receive), TX (Transfer) a GND.

### 5.1.1.4 Blok ICSP

Poté, co je mikrokontrolér vložen či napájen na desku plošného spoje, je nemožné ho programovat, pokud nemá funkci In-System Programming (ISP). Tato funkce mu dovoluje být programován externím programátorem pouze připojením pinů MISO, SCK, RESET, VCC, MOSI a GND. Firma Microchip, která vyrábí mimo jiné zde použitý mikrokontrolér Atmega 328p, má vlastní ISP protokol, který nazývá ICSP.

Komunikace je v podstatě stejná jako u SPI rozhraní. Master (externí programátor) generuje hodinový signál (SCK) a slave (mikrokontrolér) se podle tohoto signálu řídí. Data z masteru se posílají na pinu MOSI a data ze slave se posílají z pinu MISO viz obrázek 20.



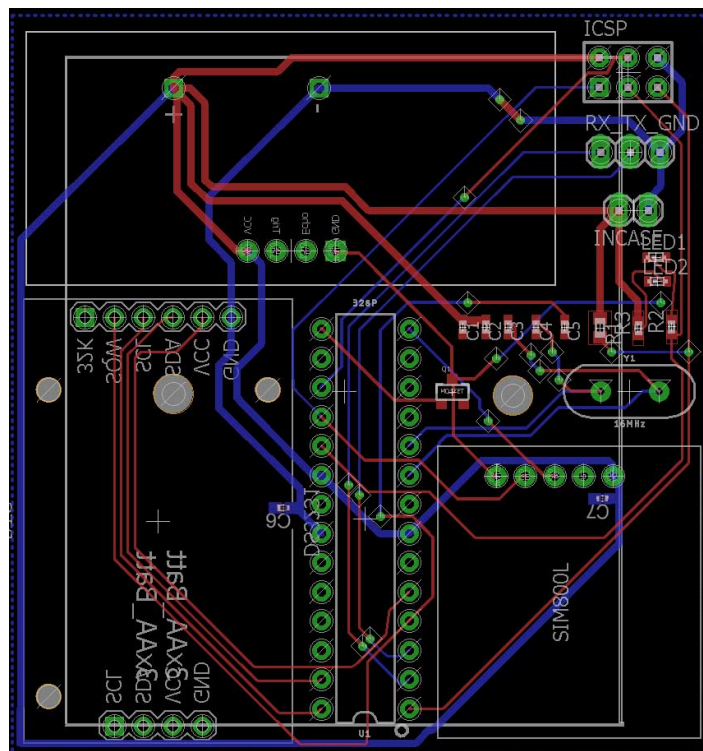
Obrázek 20 - SPI komunikace [33]

### 5.1.1.5 Vypínač GSM modulu a ultrazvukového senzoru

GSM modul a ultrazvukový senzor v nečinném stavu spotřebovávají proud v řádech jednotek miliampér. Jelikož měřicí jednotka má vydržet v odpadové nádobě několik let a zdroj energie jsou tři alkalické baterie, takový proud je nepřijatelný. Z tohoto důvodu se tyto dvě části zcela vypínají po dobu spánku mikrokontroléru. Vypnutí je zajištěno MOSFETem Q1, který je zařazen mezi zemními piny GSM modulu, ultrazvukového senzoru a zemí. Při sepnutí má MOSFET typický odpor 16 mΩ, při rozepnutí se dostává do vysokoimpedančního stavu s maximálním únikovým proudem 1 μA při 25°C a 16 V. Pro signalizaci zapnutí modulu a senzoru je přidána LED dioda LED2, která je také „uzemněna“ do MOSFETu Q1.

### 5.1.2 Deska plošného spoje

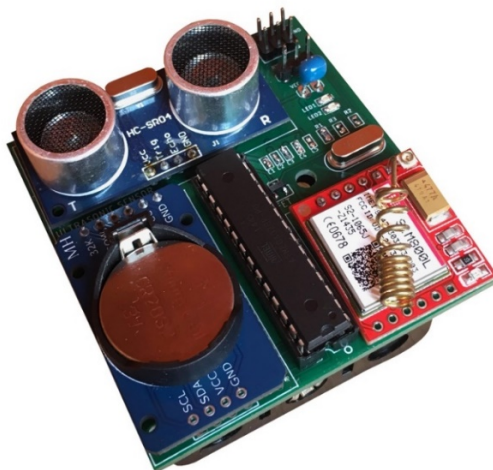
Deska plošného spoje je navržena v softwaru Autodesk Eagle a vychází z elektrického schématu. Deska je dvouvrstvá a bude osazena jak SMD, tak i dírovými komponenty. Výroba byla zadána čínské firmě PCBWay. Layout DPS je na obrázku 21.



Obrázek 21 - Návrh desky plošného spoje

### 5.1.3 Osazení komponentů na DPS

Osazení komponentů na DPS probíhalo pomocí ručního měkkého pájení hrotovou páječkou. Pro pájení byla použita trubičková cínová olovnatá pájka (Sn63Pb37) s teplotou tavení kolem 185 °C. Vzhledem k malým rozměrům SMD komponentů by bylo vhodnější použití pájení přetavením, ale bohužel tato technika nebyla k dispozici. Hotová měřicí jednotka je na obrázku 22.



Obrázek 22 - Osazená deska plošného spoje

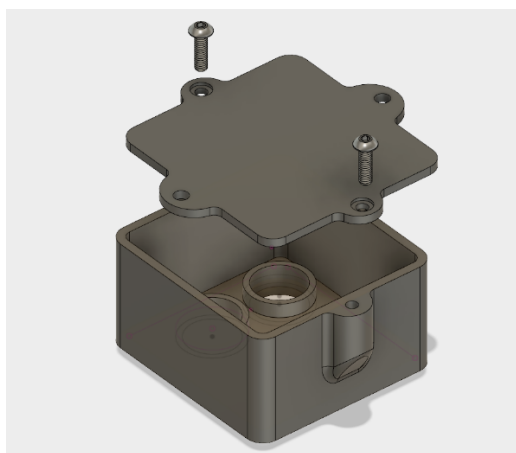
#### 5.1.4 Ochranný obal měřicí jednotky

Elektronika měřicí jednotky z důvodu vystavení povětrnostním a mechanickým vlivům musí být opatřena ochranným obalem. Tento obal bude mít také za úkol mechanické uchycení na víko odpadové nádoby.

Na obrázku 23 je vidět obal, který byl vymodelován v CAD softwaru Autodesk Fusion 360 a vytisknut na 3D tiskárně Prusa i3 MK2S. Jelikož se očekávají náročnější mechanické podmínky – otřesy při výsypu, kontakt s odpadem a jiné – byly vybrány robustnější parametry tisku (tab. 5). Jako materiál byl použit plast PLA.

Tabulka 5 - Základní parametry 3D tisku

Výška vrstvy	0,2 mm
Počet vrchních vrstev	7
Výplň	50%
Počet vrstev stěn	7
Počet spodních vrstev	7



Obrázek 23 - 3D model ochranného obalu

### 5.1.5 Kompletace hardware části

Po vytisknutí ochranného obalu, bylo nutno elektroniku do něj vložit a řádně sešroubovat samořeznými šrouby. Finální podoba měřicí jednotky je na obrázku 24.



Obrázek 24 - Finální podoba měřicí jednotky

## 5.2 Software

Ovládací program běžící na mikrokontroléru Atmega 328p má za úkol řízení celého procesu sledování hladiny odpadu v odpadové nádobě. Program je napsán v programovacím jazyce Arduino, který je odvozený od jazyka C. Vývojový diagram programu je v příloze 1. Čas od probuzení, změření vzdálenosti odpadu a poslání dat na server a znovu usnutí trvá zhruba 30 vteřin. Čas se může nepatrně měnit na základě síly GSM signálu.

### 5.2.1 Inicializace

V inicializaci programu (obr. 25) se importují knihovny (`#import`), dále se přiřadí hodnoty ke jménům (`#define`) a inicializují proměnné. Řádek č. 26 určuje ID senzoru, kterým se jednotka prezentuje při odesílání a interpretaci dat na serveru. Proměnná „url“ obsahuje IP adresu serveru, kam se data budou odesílat. Funkce „`SoftwareSerial()`“ nastavuje piny RX a TX pro sériovou komunikaci mezi mikrokontrolérem a GSM modulem.

```

12 //importovani knihoven
13 #include <SoftwareSerial.h>
14 #include <DS3232RTC.h>
15 #include <Streaming.h>
16
17 //jmenovani pinu
18 #define LED_PIN 13 //pouze na signalizaci spanku
19 #define RTC_POWER_PIN 3
20
21 //piny pro mereni ultrazvukem
22 #define trigPin 6
23 #define echoPin 7
24
25 //*****ID senzoru*****
26 #define ID "1"
27 //*****
28
29 //inicializace promennych
30 String vzdalenost;
31 String fill, batt, httppara;
32 String id = ID;
33 //prikaz + adresa na poslani informaci
34 String url = "AT+HTTPPARA=URL,http://31.30.94.90/bakalarka.php/?id=";
35 String Arsp, Grsp;
36 SoftwareSerial gsm(12, 8);
37 unsigned long timecheck;

```

Obrázek 25 – Inicializace

## 5.2.2 Funkce Setup()

Funkce Setup() (obr. 26) proběhne právě jednou při startu nebo resetu mikrokontroléru. Jsou zde nastaveny piny na vstup či výstup. LED\_PIN, RTC\_POWER\_PIN a pin č. 9 jsou zde nastaveny na výstup, protože ovládají určitý prvek. LED\_PIN ovládá LED diodu, která signalizuje zapnutí mikrokontroléru, RTC\_POWER\_PIN dodává energii RTC a pin č. 9 spíná MOSFET, který vypíná a zapíná ultrazvukový senzor a GSM modul pro šetření energie. Funkce „nastaveniRTC()“ konfiguruje RTC a funkce „nastaveniSpanku()“ volí typ a režim spánku.

```

40 void setup() {
41   //nastaveni LED pinu na output
42   pinMode(LED_PIN, OUTPUT);
43
44   //napajeni RTC kdyz arduino je vzhuru
45   pinMode(RTC_POWER_PIN, OUTPUT);
46   digitalWrite(RTC_POWER_PIN,HIGH);
47
48   //pin pro vypnuti/zapnuti gsm modulu
49   pinMode(9, OUTPUT);
50
51   //nastaveni RTC a nastaveni spanku
52   nastaveniRTC();
53   nastaveniSpanku();
54 }

```

Obrázek 26 - Funkce setup()

### 5.2.3 Nastavení RTC – funkce nastaveniRTC()

Funkce nastaveniRTC() (obr. 27) využívá knihovnu „DS3232RTC.h“, která zjednodušuje obsluhu RTC modulu. Na řádce 119 začíná nastavování času. Pokud čas v RTC již nastavený je a modul nebyl mezitím odpojený od napájení nebo mu nebyla vyňata baterie, tak znovu nastavení času není potřeba. Jelikož hodnoty v RTC jsou neznámé, musí se inicializovat na známé hodnoty – to se provádí od řádku 130 do 136. Dále je zvolení alarmu. Možnosti alarmu jsou v tabulce 6.

Tabulka 6 - Typy alarmů [34]

ALM2_EVERY_MINUTE	Alarm jednou za minutu
ALM2_MATCH_MINUTES	Alarm, když se minuty shodují (tj. jednou za hodinu)
ALM2_MATCH_HOURS	Alarm, když se hodiny a minuty shodují (tj. jednou za 24h)
ALM2_MATCH_DATE	Alarm, když se den v měsíci, hodiny a minuty shodují
ALM2_MATCH_DAY	Alarm, když se den v týdnu, hodiny a minuty shodují

Alarm má být spuštěn jednou za den, je tedy vybrán „ALM2\_MATCH\_HOURS“. Při spuštění alarmu, alarmový indikační bit změni svojí logickou hodnotu z 0 na 1. Pro znovuobnovení alarmu je potřeba, aby se indikační bit resetoval zpět na hodnotu 0 – to se dělá na řádce 142 a 143. Dále na řádce 146 se vypíná generování obdélníkové vlny z multifunkčního pinu a na řádcích 148 a 149 se aktivuje/deaktivuje alarm.

```

117 void nastaveniRTC(){
118     // Nastaveni casu - netreba, pokud je cas je uz nastaveny
119     tmElements_t tm;
120     tm.Hour = 18;           //set the RTC time to 06:29:50
121     tm.Minute = 19;
122     tm.Second = 50;
123     tm.Day = 25;
124     tm.Month = 3;
125     tm.Year = 2018 - 1970; //rozdíl od roku 1970
126     RTC.write(tm);        //zapiš čas do RTC
127
128
129     //inicializace RTC na zname hodnoty
130     RTC.setAlarm(ALM1_MATCH_DATE, 0, 0, 0, 1);
131     RTC.setAlarm(ALM2_MATCH_DATE, 0, 0, 0, 1);
132     RTC.alarm(ALARM_1);
133     RTC.alarm(ALARM_2);
134     RTC.alarmInterrupt(ALARM_1, false);
135     RTC.alarmInterrupt(ALARM_2, false);
136     RTC.squareWave(SQWAVE_NONE);
137
138     //nastaveni casu alarmu 2; "0, minuty, hodiny, 0"
139     RTC.setAlarm(ALM2_MATCH_HOURS, 0, 0, 16, 0);
140
141     //resetovani alarmovejch flagu
142     RTC.alarm(ALARM_1);
143     RTC.alarm(ALARM_2);
144
145     //vypnuti square wave na MFP RTC pinu
146     RTC.squareWave(SQWAVE_NONE);
147     //zapnout alarm 2, alarm 1 neni pouzit
148     RTC.alarmInterrupt(ALARM_1, false);
149     RTC.alarmInterrupt(ALARM_2, true);
150 }

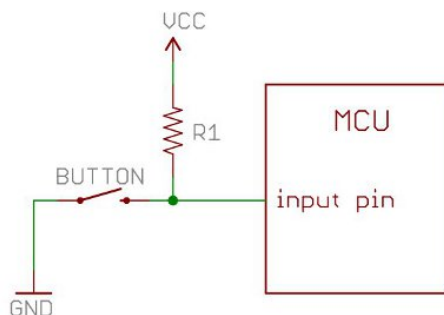
```

Obrázek 27 - Funkce nastaveniRTC()

#### 5.2.4 Nastavení spánku - funkce nastaveniSpanku()

Pro správnou funkci probuzení mikrokontroléru je potřeba nastavit pin přerušení na pull-up resistor (obr. 29). Pull-up resistor je odpor, který určuje logickou hodnotu pinu (logickou jedničku), když jí žádné zařízení nedefinuje. Pull-up resistor je velmi často využíváný v elektronice, a proto mnoho mikrokontrolérů a procesorů, včetně Atmega328p, má zabudovaný tento prvek interně. Při nastavení pinu na „INPUT\_PULLUP“ (obr. 28, řádek 155), mikrokontrolér vnitřně použije vlastní odpor a připojí pin na svůj zdroj napětí. Opak pull-up resistoru je pull-down resistor.





Obrázek 28 - Schéma pull-up resistoru [35]

Ve funkci nastaveniSpanku() (obr. 28) na řádce 157 je nastavení přerušení při sestupné hraně na pinu 2. RTC při spuštění alarmu změní hodnotu na pinu 2 z logické jedničky na logickou nulu, tím se mikrokontrolér probudí a spustí se ISR digitalInterrupt. ISR je funkce, která se provede při aktivaci přerušení. Jelikož záměr je mikrokontrolér pouze probudit ze spánku, tato funkce může být ponechána prázdná.

Řádek 160 vypíná A/D převodník z důvodu šetření energie při spánku. Napětí baterií je měřeno GSM modulem, tudíž není potřeba převodník znovu zapínat. V SMCR registru se nastavuje spánek. Atmega328p má čtyři úrovně spánku – „extended standby“, „standby“, „power-save“ a „power-down“. Úrovně se od sebe liší počtem funkcí, které mikrokontrolér má během spánku aktivní. Jelikož čip bude probuzen vnějším signálem, je možno použít spánek power-down, téměř veškeré funkce vypnout a tím snížit spotřebu energie na nezbytné minimum. V řádce 164 se zapisuje do registru SMCR logická jednička pro aktivaci spánku.

```

153 void nastaveniSpanku(){
154     //nastaveni interrupt pinu 2 na PULL UP resistor
155     pinMode(2, INPUT_PULLUP);
156     //aktivace interruptu 0 na padajici hranu - interrupt 0 je na D2
157     attachInterrupt(0, digitalInterrupt, FALLING);
158
159     //vypnutí ADC - setri energii
160     ADCSRA &= ~(1 << 7);
161
162     //povoleni a zvoleni hlubokeho spanku (nejmensi spotreba)
163     SMCR |= (1 << 2); //power down mod
164     SMCR |= 1;      //povolit spanek
165
166 }

```

Obrázek 29 - Funkce nastaveniSpanku

### 5.2.5 Smyčka loop()

Loop() (obr. 30) je hlavní nekonečná smyčka programu, která po inicializaci a provedení funkce setup() běží stále dokola. Nejdříve se zapne signalizační dioda LED\_PIN, která dává vědět, že se mikrokontrolér probudil z hlubokého spánku. Poté se změní logický level pinu 9 z 0 na 1, čímž se sepne MOSFET a přivede se napětí na GSM modul a ultrazvukový senzor. Následně se do proměnné vzdalenost pomocí funkce zmerit() uloží informace o vzdálenosti odpadu od měřící jednotky.

Na řádce 70 a 72 se inicializuje UART sériová komunikace. `Serial.begin(9600)` inicializuje komunikaci o baud rate 9600 mezi počítačem a mikrokontrolérem, která je používána pro výpis informací při debugování. Defaultní nastavení UART komunikace v knihovně `Serial` je 8 bitů, jeden stop bit a žádný parity bit. `Gsm.begin(4800)` inicializuje komunikaci mezi mikrokontrolérem a GSM modulem o rychlosti 4800 bit/s, žádným parity bitem a jedním stop bitem [36]. Byla zvolena pomalejší rychlost přenosu kvůli minimalizace chyb během přenosu. Modul SIM800L umí sám rozpoznat baud rate a tedy není potřeba ho nastavovat v samotném modulu.

Řádek 76 do proměnné `url`, která obsahuje AT příkaz, adresu serveru a informaci o ID měřící jednotky, přidá údaj o vzdálenosti odpadu. Následuje odeslání informace pomocí řetězce funkcí začínajícím funkcí `connection()`. Po odeslání informace se měřící jednotka připravuje na spánek. Nejdříve se vypne pomocí pinu 9 GSM modul a ultrazvukový senzor, poté LED dioda a napájení RTC. Ještě před vypnutím RTC je resetována signalizace alarmu v RTC změnou alarm flag bitu z 1 na 0 pro znovuspuštění alarmu. Dále se piny A4 a A5, které jsou připojeny na pull-up odpory přepnou do vstupního módu, kdy mají impedanci několik mega ohmů a tím šetří energii. Na řádce 100, funkce `usni()` ukládá mikrokontrolér do hlubokého spánku. Na stejném řádku se také mikrokontrolér po spuštění alarmu probudí. Po probuzení se piny A4 a A5 znovu připojí na pull-up odpor pro správnou funkci I<sup>2</sup>C komunikace, zapne se RTC a smyčka se opakuje od začátku.

```
56 void loop() {
57
58     //LED na signalizaci
59     digitalWrite(LED_PIN, HIGH);
60     delay(200);
61     //zapnutí gsm modulu
62     digitalWrite(9, HIGH);
63
64     //zmereni vzdalenosti odpadu
65     delay(500);
66     vzdalenost = zmerit();
67     delay(500);
68
69     //Inicializace seriovych komunikaci - jedna pro info, jedna pro gsm modul
70     Serial.begin(9600);
71     delay(50);
72     gsm.begin(4800);
73     delay(50);
74
75     //pridani informace vzdalenosti do url
76     httppara = url + id + "&fill=" + vzdalenost + "&batt=";
77
78     //funkce connection pro odeslani informaci
79     connection();
80
81     //vypnutí gsm modulu a LED diody
82     digitalWrite(9, LOW);
83     delay(100);
84     digitalWrite(LED_PIN, LOW);
```

Obrázek 30a - První část smyčky `loop()`

```

86 //zde zacina priprava na spanek
87 //resetovani flagu alarmu 2
88 RTC.alarm(ALARM_2);
89 delay(10);
90 //vypnuti RTC
91 digitalWrite(RTC_POWER_PIN,LOW);
92
93 //nastaveni pinu na vysokou inpedanci - setri energii
94 pinMode(A5, INPUT);
95 pinMode(A4, INPUT);
96
97 //ulozeni do spanku
98 Serial.println("Jdu spat!");
99 delay(10);
100 usni(); //zde arduino usne a take se probudi
101 delay(10);
102
103 //nastaveni pinu na INPUT_PULLUP pro I2C komunikaci
104 digitalWrite(A5, INPUT_PULLUP);
105 digitalWrite(A4, INPUT_PULLUP);
106
107 //zapnuti RTC
108 digitalWrite(RTC_POWER_PIN,HIGH);
109 delay(100);
110 }

```

Obrázek 30b - Druhá část smyčky loop()

### 5.2.6 Ultrazukové měření - funkce zmerit()

Ve funkci zmerit() (obr. 31) se provádí změřeni vzdálenosti odpadu od měřicí jednotky. Nejdříve se inicializují proměnné a piny. Pin trigPin dává signál senzoru k vyslání ultrazvuku a pin echoPin zaznamenává, jak dlouho trvalo ultrazvuku se dostat k odpadu a zpět. Po inicializaci je proveden 10 mikrosekundový puls na trigPin pinu, kterým se spustí měření senzorem. Ten změní vzdálenost a vrátí signál na echoPin, jehož délka je přímo úměrná délce času letu signálu. Délka impulzu se měří funkcí pulseIn(). Následně se pomocí rychlosti zvuku (v cm/mikrosekund) vypočítá vzdálenost odpadu, která se převede na datový typ String a vloží do návratové hodnoty výsledek.

```

347 String zmerit (){
348   String vysledek;
349   int vzdalenost;
350   long duration;
351   //nastaveni pinu
352   pinMode(trigPin, OUTPUT);
353   pinMode(echoPin, INPUT);
354
355   digitalWrite(trigPin, LOW);
356   delay(100);
357
358   //10 mikrosekundovy puls, kterym spoustime signal z ultrazvukoveho senzoru
359   digitalWrite(trigPin, HIGH);
360   delayMicroseconds(10);
361   digitalWrite(trigPin, LOW);
362
363   duration = pulseIn(echoPin, HIGH);
364   //vypocitani vzdalenosti pomoci znamy rychlosti zvuku
365   vzdalenost = duration*0.034/2;
366   //prevedeni vysledku na string
367   vysledek = String(vzdalenost);
368   //piny na input - setri energii
369   pinMode(echoPin, INPUT);
370   pinMode(trigPin, INPUT);
371   return vysledek;
372 }

```

Obrázek 31 - Funkce zmerit()

### 5.2.7 Uspání mikrokontroléru – funkce usni()

Uložení mikrokontroléru do spánku probíhá ve funkci usni() (obr. 32). Nejdříve se na řádce 172 a 173 vypne BOD pro šetření energie během spánku a poté na řádce 176 pomocí assemblerového příkazu se mikrokontrolér uloží do samotného spánku.

BOD slouží k neustálé kontrole napájecího napětí mikrokontroléru. Pokud napětí klesne pod určitou hranici, mikrokontrolér se resetuje. Tato ochrana je z důvodu ochránění mikrokontroléru před trvalým „uzamčením“ z důvodu nečekaného poklesu napětí a tím přerušení prováděné operace.

```

170 void usni(){
171   //vypnutí BOD - setri energii - postup z datasheetu
172   MCUCR |= (3 << 5);
173   MCUCR = (MCUCR & ~(1 << 5)) | (1 << 6);
174
175   //(assembler) mikrokontroler jde zde do hlubokeho spanku
176   __asm__ __volatile__ ("sleep");
177 }

```

Obrázek 32 - Funkce usni()

## 5.2.8 Řetězec funkcí k odeslání dat

### 5.2.8.1 Ověření připojení k síti - funkce connection()

V každé funkci, kde se očekávají konkrétní návratová data od GSM modulu, je systém ověření, který zabraňuje zacyklení mikrokontroléru, když GSM modul nevrací očekávaná data. Tento ověřovací systém je realizován na řádce 181 a 186 (obr. 36). Na řádce 181 se do proměnné timecheck uloží pomocí

funkce `millis()` čas od zapnutí mikrokontroléru a na řádku 186 se ověří, zda mikrokontrolér nestrávil ve funkcích k odesílání dat více než minutu. Pokud ano, program příkazem `break` vyskočí ze všech cyklů a pokračuje v hlavní smyčce, kde uloží mikrokontrolér do spánku. Návrat nestandardní velikosti dat může být zejména kvůli výpadku GSM sítě a tedy nemožnosti se k ní připojit.

AT příkaz „AT+COPS?“ se dotazuje GSM modulu, zda se připojil a registroval do GSM sítě. Pokud ano, modul pošle mikrokontroléru data o velikosti 37 bitů, které se uloží do proměnné *Grsp* a program vyskočí z cyklu a spustí funkci `BATT()`. Pokud ne, program počká 3 vteřiny a dotáže se znovu. Dotazuje se do té doby, dokud se modul nepřipojí či nevyprší časový limit 60 vteřin.

Veškeré příkazy sériové komunikace s `PC Serial.xxx()` slouží pouze k usnadnění debuggování a na běh programu nemají žádný vliv.

```
180 void connection(){
181     timecheck = millis();
182     int test = 0;
183     //testovani ocekavaneho vstupu od GSM modulu
184     while (test != 37){
185         //pokud se GSM modul pripojuje vice nez 60s, vyskoc z funkce
186         if((millis() - timecheck) > 60000) break;
187         //poslani AT prikazu na zjisteni pripojeni do site
188         gsm.println("AT+COPS?");
189         Serial.println("AT+COPS?");
190         Grsp = gsm.readString();
191         Serial.println(Grsp);
192         test = Grsp.length();
193         delay(3000);
194     }
195     Serial.println("BATT");
196     BATT();
197 }
```

Obrázek 33 - Funkce `connection()`

### 5.2.8.2 Měření napětí baterie - funkce `BATT()`

Ve funkci `BATT()` (obr. 34) se GSM modulem měří napětí baterie. Po odeslání příkazu „AT+CBC“, GSM vrátí informaci o napětí a ta se uloží do proměnné *baterie*. Jelikož informace o napětí je obalená nepotřebnými znaky, je potřeba hodnotu napětí vyseparovat pro dostání čistě číselné reprezentace. To se provádí na řádcích 207 až 209. Dále se vyseparované znaky spojí do stringu *batt* a přidají se k proměnné *httppara*, která v sobě uchovává informace k odeslání na server. V této funkci není možné žádné zacyklení, nemusí se tedy použít ověřovací systém.

```

200 void BATT(){
201     delay(300);
202     //poslani AT prikazu
203     gsm.println("at+cbc");
204     String baterie = gsm.readString();
205     Serial.println(baterie);
206     //vyseparovani informace ze stringu
207     char singleCharacter = baterie.charAt(21);
208     char singleCharacter2 = baterie.charAt(22);
209     char singleCharacter3 = baterie.charAt(23);
210     delay(50);
211     batt = String(singleCharacter) + "." + String(singleCharacter2)
212     + String(singleCharacter3);
213     //pridani informace do URL
214     httppara = httppara + batt;
215     Serial.println(httppara);
216     Serial.println("CGATT");
217     CGATT();
218 }

```

Obrázek 34 - Funkce BATT()

### 5.2.8.3 Povolení GPRS - funkce CGATT()

Funkce CGATT() (obr. 35) pomocí AT příkazu „AT+CGATT?“ povolí GPRS přenos. Má stejnou strukturu jako funkce connection(). Po úspěšném povolení GPRS přenosu, program vyskočí z cyklu a spustí funkci SAPBR().

```

220 void CGATT(){
221     delay(300);
222     int test = 0;
223
224     do {
225         //pokud se GSM modul pripojuje vice nez 60s, vyskoc z funkce
226         if((millis() - timecheck) > 60000) break;
227         //poslani AT prikazu
228         gsm.println("AT+CGATT?");
229         Grsp = gsm.readString();
230         Serial.println(Grsp);
231         test = Grsp.length();
232         if (test == 29) break;
233         delay(3000);
234         //otestovani ocekavaneho vstupu od GSM modulu
235     } while (test != 29);
236
237     Serial.println("SAPBR");
238     SAPBR();
239 }

```

Obrázek 35 - Funkce CGATT()

### 5.2.8.4 Typ připojení – funkce SAPBR()

Pro odeslání dat je nutné nastavit v GSM modulu typ připojení. To se provádí funkcí SAPBR() (obr. 36). Jelikož se přenášejí data, je zvoleno GPRS. Po zvolení typu připojení, program skočí do funkce SAPBR2().

```

242 void SAPBR(){
243     delay(300);
244     int test = 0;
245     do {
246         //pokud se GSM modul pripojuje vice nez 60s, vyskoc z funkce
247         if((millis() - timecheck) > 60000) break;
248         //poslani AT prikazu
249         gsm.println("AT+SAPBR=3,1,CONTYPE,GPRS");
250         Grsp = gsm.readString();
251         Serial.println(Grsp);
252         test = Grsp.length();
253         if (test == 32) break;
254         delay(3000);
255         //otestovani ocekavaneho vstupu od GSM modulu
256     } while (test != 32);
257
258     Serial.println("SAPBR2");
259     SAPBR2();
260 }

```

Obrázek 36 - Funkce SAPBR()

### 5.2.8.5 Parametry připojení – funkce SAPBR2()

Ve funkci SAPBR2() (obr. 37) se nastavuje jméno přístupového bodu (APN) pro GPRS připojení. APN je jméno nastavení, které si zařízení stáhne, aby mohlo uskutečnit připojení mezi mobilní sítí a internetem. Každý operátor má jiné jméno přístupového bodu. V této měřicí jednotce jsou využívány služby operátora Vodafone, který má APN „internet“.

```

262 //nastaveni parametru prenosove sluzby
263 void SAPBR2(){
264     delay(300);
265     int test = 0;
266
267     do {
268         //pokud se GSM modul pripojuje vice nez 60s, vyskoc z funkce
269         if((millis() - timecheck) > 60000) break;
270         //poslani AT prikazu
271         gsm.println("AT+SAPBR=3,1,APN,internet");
272         Grsp = gsm.readString();
273         Serial.println(Grsp);
274         test = Grsp.length();
275         if (test == 32) break;
276         delay(3000);
277         //otestovani ocekavaneho vstupu od GSM modulu
278     } while (test != 32);
279
280     Serial.println("SAPBR3");
281     SAPBR3();
282 }

```

Obrázek 37 - Funkce SAPBR2()

### 5.2.8.6 Aktivace připojení – funkce SAPBR3()

Ve funkci SAPBR3() (obr. 38) se aktivuje připojení GPRS. Po aktivaci připojení, program přejde do funkce HTTPINIT().

```

285 void SAPBR3(){
286     delay(300);
287     int test = 0;
288
289     do {
290         //pokud se GSM modul pripojuje vice nez 60s, vyskoc z funkce
291         if((millis() - timecheck) > 60000) break;
292         //poslani AT prikazu
293         gsm.println("AT+SAPBR=1,1");
294         Grsp = gsm.readString();
295         Serial.println(Grsp);
296         test = Grsp.length();
297         if (test == 19) break;
298         delay(3000);
299         //otestovani ocekavaneho vstupu od GSM modulu
300     } while (test != 19);
301
302     Serial.println("HTTPINIT");
303     HTTPINIT();
304 }

```

Obrázek 38 - Funkce SAPBR3()

### 5.2.8.7 Inicializace HTTP – funkce HTTPINIT()

Před odesláním HTTP požadavku, je potřeba inicializovat HTTP službu a poté nastavit parametry požadavku. Inicializace je provedena příkazem „AT+HTTPINIT“ ve funkci HTTPINIT() (obr. 39). Po úspěšné inicializaci, program skočí do funkce HTTPPARA() pro nastavení parametrů.

```

307 void HTTPINIT(){
308     delay(300);
309     int test = 0;
310
311     do {
312         //pokud se GSM modul pripojuje vice nez 60s, vyskoc z funkce
313         if((millis() - timecheck) > 60000) break;
314         //poslani AT prikazu
315         gsm.println("AT+HTTPINIT");
316         Grsp = gsm.readString();
317         Serial.println(Grsp);
318         test = Grsp.length();
319         if (test == 18) break;
320         delay(3000);
321         //otestovani ocekavaneho vstupu od GSM modulu
322     } while (test != 18);
323
324     Serial.println("HTTPPARA");
325     HTTPPARA();
326 }

```

Obrázek 39 - Funkce HTTPINIT()



### 5.2.8.8 Nastavení parametrů HTTP – funkce HTTPPARA()

Ve funkci HTTPPARA() (obr. 40) se nastavují HTTP parametry požadavku. Příkaz pro nastavení HTTP požadavku obsahuje IP adresu serveru, jméno webové stránky a informace, které bude předávat – v tomto případě ID měřicí jednotky, vzdálenost odpadu a napětí baterie. Všechny tyto informace jsou uloženy v proměnné *httppara*, která byla naposledy upravena ve funkci BATT().

```
328 //nastaveni parametru HTTP
329 void HTTPPARA(){
330     delay(300);
331     //poslani AT prikazu
332     gsm.println(httppara);
333     Serial.println("HTTPACTION");
334     HTTPACTION();
335 }
```

Obrázek 40 - Funkce HTTPPARA()

### 5.2.8.9 Provedení HTTP požadavku – funkce HTTPACTION()

AT příkaz „AT+HTTPACTION=0“ ve funkci HTTPACTION() (obr. 41) zvolí HTTP GET požadavek a provede ho. GET požadavek si vyžádá konkrétní zvolenou stránku a předá serveru informace o *ID*, *fill* a *batt*, které jsou v proměnné *httppara*.

```
337 //odeslani dat na server
338 void HTTPACTION(){
339     delay(300);
340     //poslani AT prikazu
341     gsm.println("AT+HTTPACTION=0");
342     delay(4000);
343 }
```

Obrázek 41 - Funkce HTTPACTION()

## 5.3 Testování měřicí jednotky

Po provedené kompletaci hardwaru a nahrání softwaru do mikrokontroléru, byla jednotka otestována ve dvou fázích – krátkodobý a dlouhodobý test.

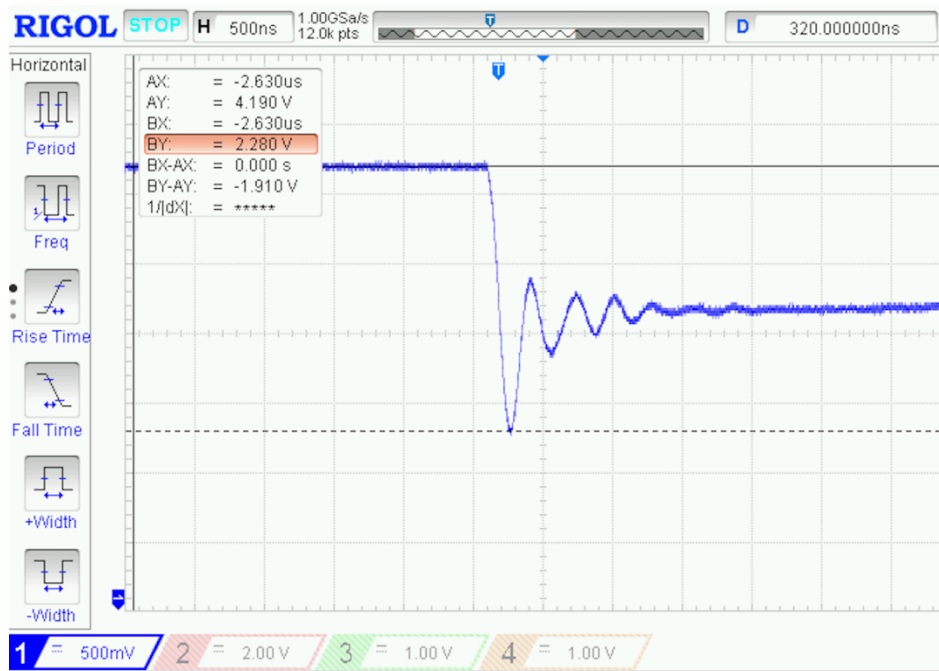
### 5.3.1 Krátkodobý test

Krátkodobý test spočíval v změření vzdálenosti a odeslání informace jednou za minutu po dobu tří hodin. V tomto testu se nevyskytl žádný problém a vše fungovalo tak, jak bylo navrženo.

### 5.3.2 Dlouhodobý test

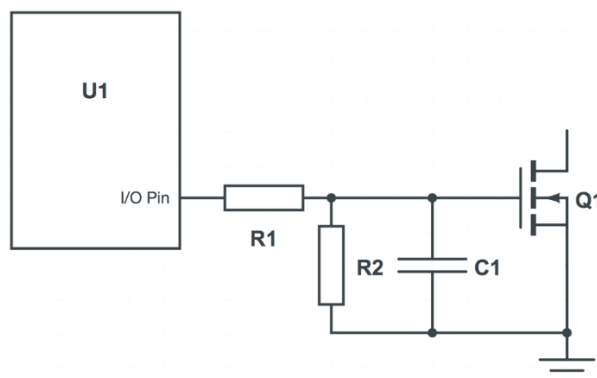
V dlouhodobém testu měřicí jednotka byla naprogramována, aby provedla jedno měření a odeslání informace denně po dobu jednoho týdne. Při tomto testu se vyskytla značná nestabilita. Po třetím dnu, jednotka přestala komunikovat a začala se periodicky po několika vteřinách vypínat a zapínat. Po opakování testu se tento problém objevil už při druhém dnu.

Následná analýza na osciloskopu prozradila, že se jedná o hardwarový problém. Při skokovém sepnutí tranzistoru Q1 pro zapnutí ultrazvukového senzoru a GSM modulu, docházelo k velkému špičkovému proudu z důvodu nabití kapacit modulů a tím pokles napájecího napětí VCC v celém systému na hranici, pod kterou se mikrokontrolér restartoval (obr. 42). Tento jev se při krátkodobém testu neobjevil, protože měřící jednotka byla napájena z laboratorního DC zdroje, což je mnohem tvrdší napěťový zdroj než baterie použité při dlouhodobém testu.



Obrázek 42 - Pokles napětí v systému

Poklesu napětí jde zabránit postupným sepnutím tranzistoru Q1 místo skokového sepnutí. Tranzistor Q1 je typu MOSFET, který je ovládán napětím aplikovaným na hradlo gate. Postupné sepnutí tedy znamená pozvolný růst tohoto řídicího napětí. To může být realizováno RC obvodem viz obrázek 43.



Obrázek 43 - Obvod pro postupné sepnutí tranzistoru MOSFET

Odpor R1 zpomaluje nabíjení kondenzátoru C1 a tím způsobuje postupné sepnutí tranzistoru Q1. Odpor R2 slouží jako pull-down rezistor, který zajišťuje následné vybití kondenzátoru.

Protože v době nalezení problému poklesu napájecího napětí byla už DPS vyrobena, nebylo možno její elektrické schéma upravit. Problém byl tedy vyřešen externím 1000 $\mu$ F kondenzátorem, který byl napájen mezi řídicí pin tranzistoru a zem. Maximální proud z mikrokontroléru je 40 mA, což v kombinaci s velkou kapacitou kondenzátoru dosáhne podobného efektu jako schéma na obrázku 43. Tento způsob je nicméně nevhodný, protože při nabíjení kondenzátoru je pin zkratován do země, což by v dlouhodobém hledisku mohlo trvale poškodit mikrokontrolér.

## 6 Realizace internetové aplikace

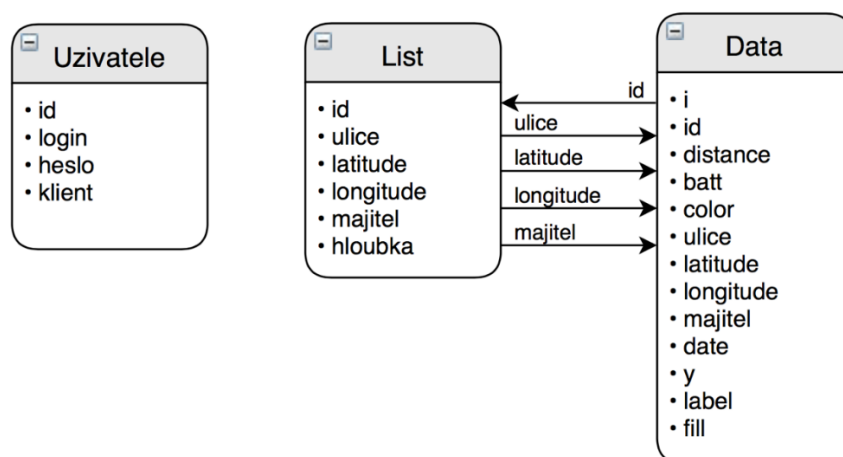
Data z měřicí jednotky se ukládají do MySQL databáze. Jelikož měřících jednotek může mít klient až stovky a posílají data alespoň jednou denně, čtení výsledků přímo z databáze se po čase stane velmi nepřehledné a nekomfortní. Z tohoto důvodu je potřeba vytvořit internetovou aplikaci s uživatelským prostředím, která bude schopna přehledně a strukturovaně interpretovat chtěná data.

Internetová aplikace je napsána v jazycích PHP, HTML, CSS a Javascript a využívá Apache a MySQL server. PHP skripty v kombinaci s SQL příkazy se starají o veškerou komunikaci s MySQL databází, HTML a CSS zajišťují vzhled, strukturu a obsah internetové aplikace. Javascript je použit pro generování funkčních prvků – grafů a map s umístěnými kontejnery.

### 6.1 MySQL databáze

V operačním systému QTS síťového disku Qnap je MySQL server defaultně nainstalován a stačí ho tedy jenom ve webovém rozhraní síťového disku aktivovat. Po aktivaci byl stáhnut nástroj pro správu databází – phpMyAdmin, který se umí sám nakonfigurovat pro lokální MySQL server. PhpMyAdmin je nástroj napsaný v jazyce PHP, který pomocí přehledného grafického uživatelského prostředí umožňuje jednoduchou správu MySQL databázového systému.

V MySQL byla vytvořena databáze „bakalarka“, která obsahuje tři tabulky – „data“, „list“ a „uzivatele“ (obr. 44).



Obrázek 44 - Struktura a vzájemné vztahy tabulek databáze „bakalarka“

V tabulce „uzivatele“ jsou informace pro přihlášení. *Id* je unikátní identifikační číslo uživatele, *login* přihlašovací jméno, *heslo* přihlašovací heslo a *klient* je název uživatele, který se objevuje v hlavičce aplikace po přihlášení.

Tabulka „List“ obsahuje informace o jednotlivých měřících jednotkách. Každá měřící jednotka má své unikátní identifikační číslo, které souhlasí právě s jedním *id* z tabulky „List“ a na základě této shody jsou danému záznamu v tabulce *Data* přiřazeny informace. Parametr *ulice* je název ulice s odpadovou nádobou, v níž je měřící jednotka umístěna. *Latitude* je zeměpisná šířka odpadové nádoby, *longitude* je zeměpisná délka měřící nádoby. Tyto dva parametry slouží k přesnému zobrazení na mapě. *Majitel* označuje majitele nádoby a souhlasí právě s jedním parametrem *login* v tabulce „uzivatele“. *Hloubka* je hloubka dané odpadové nádoby, pomocí které se vypočítává relativní zaplnění v procentech.

V tabulce „data“ jsou uložena jednotlivá data, která byla obdržena od měřících jednotek a tabulky „list“. *I* je unikátní číslo záznamu. Parametry *id*, *distance* a *batt* byly obdrženy od měřící jednotky a reprezentují identifikační číslo, vzdálenost odpadu a napětí baterie. *Color* je barva, která se přiřadí záznamu na základě parametru relativního zaplnění - *fill* – a slouží pro barevné znázornění odpadové nádoby na mapě. *Ulice*, *latitude*, *longitude* a *majitel* jsou parametry získané z tabulky *list*. *Date* je datum a čas připsání záznamu. *Y* slouží ke grafické interpretaci dat a obsahuje hodnotu osy Y – tj. relativní zaplnění *fill*. *Label* je popis každé hodnoty Y. *Fill* je vypočítáno z parametrů *distance* a *hloubka* (tabulka „list“).

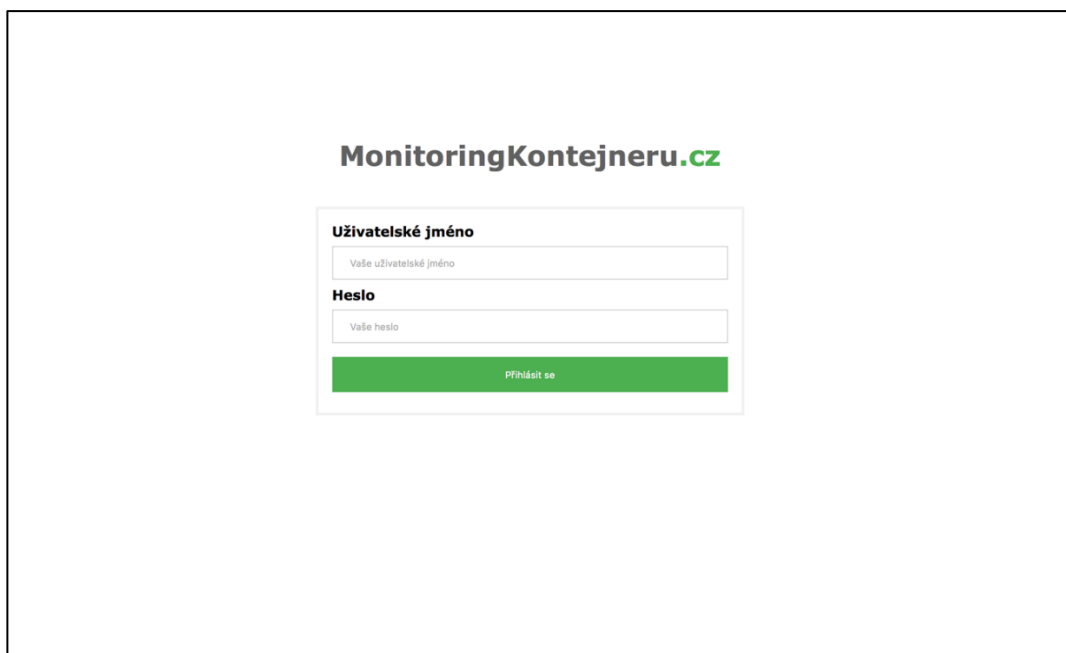
## 6.2 Grafické uživatelské prostředí

Grafické uživatelské prostředí internetové aplikace pro sledování zaplnění odpadových nádob je realizováno v jazyce HTML, CSS a funkční doplňky v Javascriptu. HTML slouží k vytvoření struktury a obsahu webové stránky. Kaskádové styly (CSS) zajišťují správné zobrazení, vizuální efekty a design všech elementů.

Aplikace by měla být funkční se zařízeními s různými rozlišeními obrazovky a internetovými prohlížeči. Vytváření vlastních kaskádových stylů, které splňují tyto požadavky vyžaduje velké množství zkušeností a testování. Jelikož není možno splnit ani jedno, byl využit CSS Framework W3.CSS, který určuje pravidla zobrazení a velmi usnadňuje vytváření vzhledu webové stránky. Důvod k vybrání tohoto frameworku bylo zejména to, že využívá pouze čisté CSS bez jakýchkoliv JQuery nebo Javascript knihoven, podporuje moderní responzivní web design, umožňuje správné zobrazení na PC, notebooku, tabletu a mobilním telefonu a je kompatibilní se všemi předními internetovými prohlížeči.

### 6.2.1 Stránka přihlášení

Přihlašovací stránka (obr. 41) zajišťuje přihlašování do internetové aplikace. Při úspěšném přihlášení, klient je přesměrován na domovskou stránku. Při neúspěšném přihlášení – uživatelské jméno nebo heslo bylo zadáno chybně – aplikace odmítne přihlášení s požadavkem opětovného zadání přihlašovacích údajů.



Obrázek 45 - Stránka přihlášení

### 6.2.2 Domovská stránka (aktuální data)

Domovská stránka (obr. 46) obsahuje tabulkový přehled aktuálních dat všech odpadových nádob, které daný klient má vybavené měřicí jednotkou. V přehledu je ID odpadové nádoby, relativní zaplnění, ulice, napětí baterie a datum poslední aktualizace údajů. Ikona .CSV exportuje tabulku do CSV souboru a umožní stáhnutí do počítače.

ID	Zaplnění	Ulice	Baterie	Datum obnovy
2	56%	Vymyslens	4.5 V	2018-04-16 16:08:52
1	35%	Technicka	4.32 V	2018-04-25 16:00:01

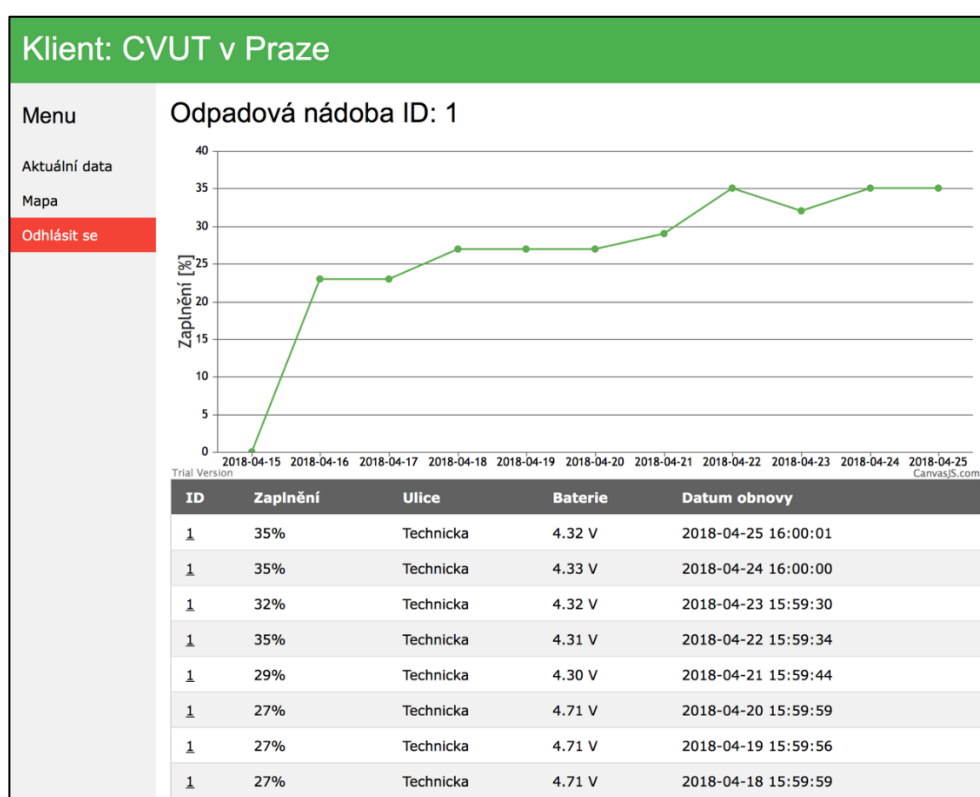
.CSV

Obrázek 46 - Domovská stránka (aktuální data)

### 6.2.3 Stránka detailu odpadové nádoby

Po kliknutí na ID měřicí jednotky na domovské stránce, aplikace přesměruje uživatele na detail a historii dané měřicí nádoby (obr. 47). V detailu odpadové nádoby je historie zaplnění nádoby v tabulkové a grafické podobě za posledních 30 dní. Na patě stránky je opět možnost exportovat výsledky do CSV dokumentu.

Graf je realizován Javascript knihovnou CanvasJS.com. Pro správné zobrazení, je nejdříve potřeba pomocí PHP vygenerovat z MySQL databáze potřebná data do dvourozměrného pole, kde osa X je datum připsání výsledku (tabulka „data“, parametr *label*) a osa Y relativní zaplnění (tabulka „data“, parametr *y*) a poté předat toto pole Javascript funkci, která vyvolá graf. CanvasJS.com disponuje přehledným a jednoduchým API, v kterém je možné upravit styl grafu. V tomto grafu byl oproti defaultním vzhledu doplněn popis osy Y, změněny barvy a mřížka pozadí.



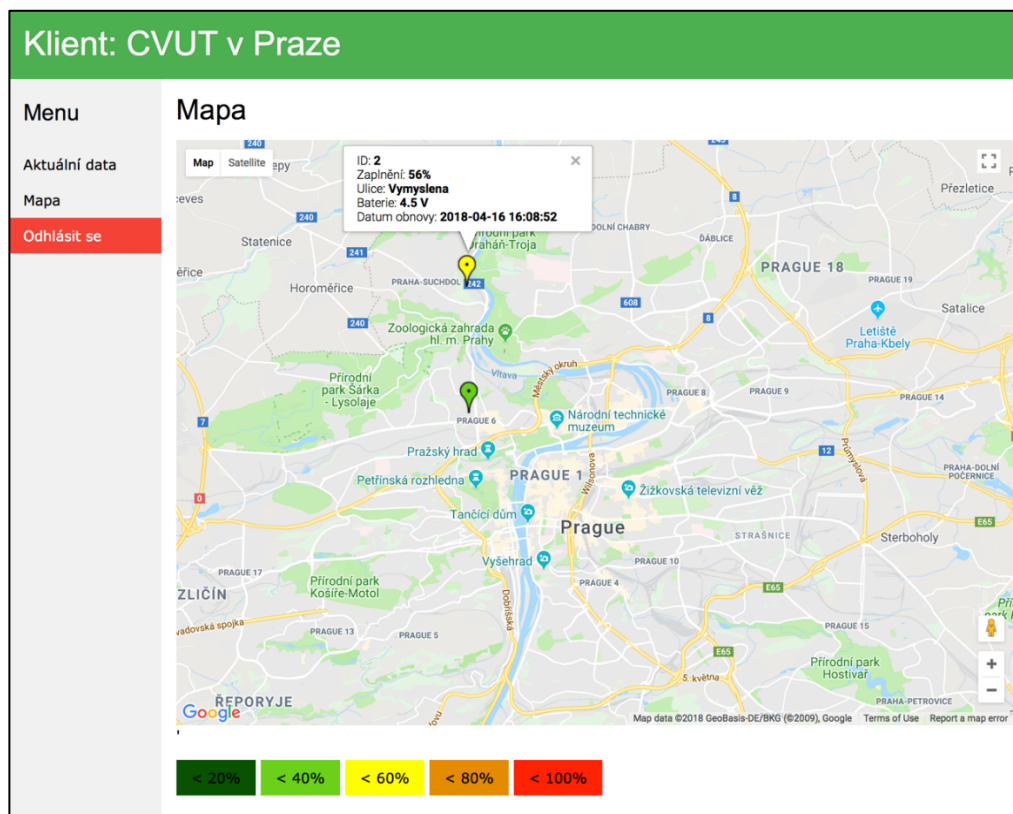
Obrázek 47 - Detail odpadové nádoby

### 6.2.4 Stránka Mapa

Stránka mapa (obr. 48) zobrazuje odpadové nádoby s měřicí jednotkou dle jejich skutečné lokace. Měřicí jednotky jsou zde reprezentovány špendlíky, které mění barvu dle zaplnění. Kliknutím na špendlík se zobrazí základní informace o zaplnění a stavu měřicí jednotky.

Mapové podklady jsou od společnosti Google. Doplnění informací do mapy bylo realizováno pomocí „Google Maps JavaScript API“, které umožňuje přizpůsobit a vytvořit vlastní zobrazení mapy pro vložení na webové stránky. Nejdříve je potřeba mapu inicializovat. V procesu inicializace se nastaví výřez, střed, funkce, elementy a importuje se obsah mapy. Pro import obsahu se musí vygenerovat XML soubor s daty, které balík Google Maps JavaScript funkcí umí číst. Generování je provedeno PHP

skriptem, který vytáhne data z MySQL databáze a vloží je do nově vzniknuvšího XML souboru. Po inicializaci je mapa vyvolána HTML div tagem, který má specifické id *map*.



Obrázek 48 - Stránka Mapa

## 7 Ekonomická analýza projektu

V této části se bude věnováno ekonomické analýze projektu – tj. vyhodnocení nákladů, potřebná počáteční investice a určení cash flow za dobu projektu.

Životnost projektu je počítána na čtyři roky a bylo by vyrobeno a osazeno 400 měřicích jednotek. Měřicí jednotky by se neprodávaly, ale pouze pronajímaly a s internetovou aplikací by byla provozována služba za měsíční poplatek. Výdaje na práci by byly nulové – veškeré práce bych dělal sám bez nároku na mzdu a na konci účetního období bych si vyplatil případný zisk z nově vzniklé firmy.

### 7.1 Náklady

Před zahájením realizace projektu jsem již vlastnil 3D tiskárnu, počítač a drobné nářadí, které je potřeba k montáži měřicích jednotek. Tyto položky bych jako fyzická osoba nechal užívat nově vzniklé firmě formou výprosy. Čas vynaložený na vývoj celého systému byl zároveň čas, který jsem věnoval škole, a proto ho také nebudu brát jako náklad.

### 7.1.1 Provozní náklady

Provozní náklady jsou v tabulce 7. Systém monitoringu odpadových nádob by měl fungovat zcela autonomně bez jakýchkoliv zásahů, nicméně by se počítalo s 5% kazivostí měřicích jednotek za dobu životnosti projektu a tím by vznikaly náklady na servis (výměnu jednotek). Náklad na výměnu jedné měřicí jednotky je vyčíslen na 100 Kč. Další provozní náklady se skládají z položek provozu serveru a poplatky za SIM kartu. Položky serveru jsou fixní náklady, protože nezáleží, kolik měřicích jednotek bude provozováno, položka SIM je variabilní náklad – čím více jednotek je provozováno, tím více SIM karet je potřeba. Hosting by byl zajištěn firmou WEDOS a SIM karty firmou Vodafone. Obě společnosti nabízejí nejpříznivější ceny na trhu.

Tabulka 7 - Provozní náklady

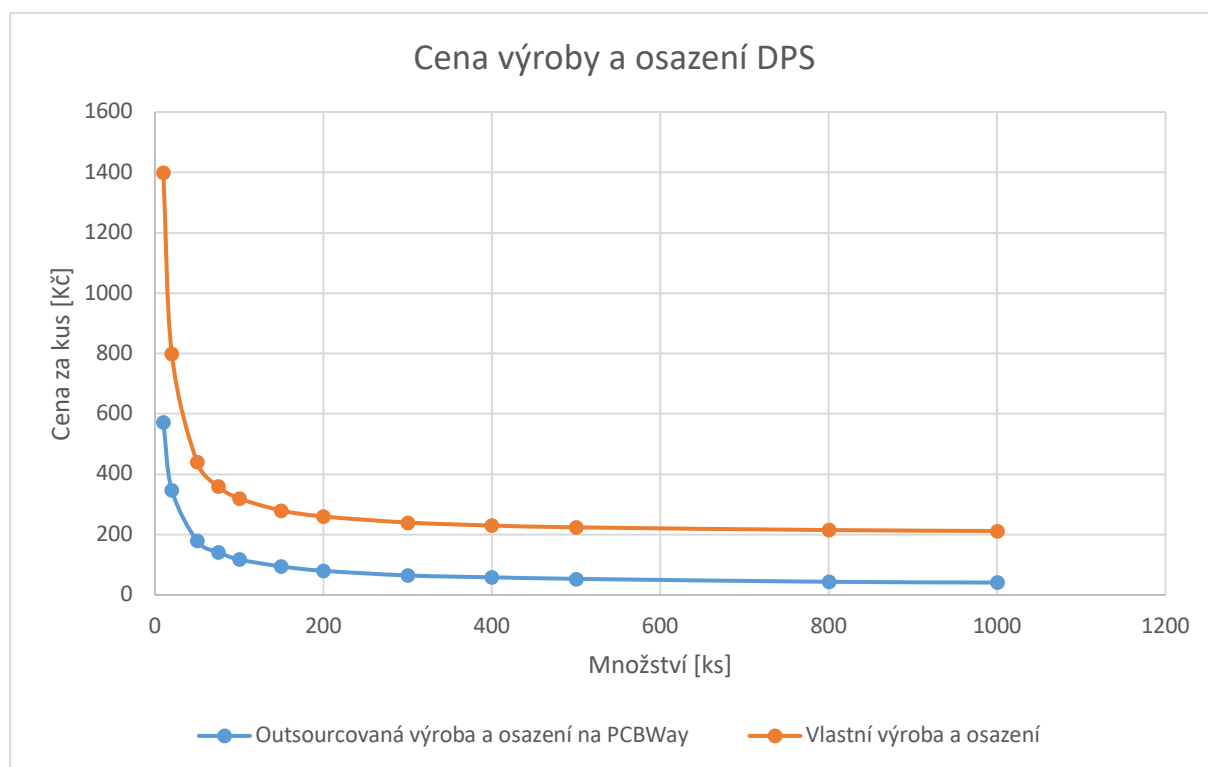
Položka	Cena/měsíc/jednotka [Kč]	Celková cena/rok/400 jednotek [Kč]
<b>Server</b>		
Hosting	25	300
Doména	10	125
<b>Měřicí jednotka</b>		
SIM	30	144000
Servis	0,1	500
<b>Celkové provozní náklady na rok</b>		<b>144925</b>

### 7.1.2 Výrobní náklady

Výroba desky plošného spoje a osazení komponenty bude outsorcováno čínské firmě PCBWay. Pokud by se výroba DPS a osazení komponenty dělalo ručně ve vlastní výrobě, bylo by potřeba nakoupit vhodné vybavení a zaměstnat lidi na zvětšení výrobní kapacity, což se při jakémkoliv množství nevyplatí viz obr. 49. Jediná vlastní výroba bude 3D tisk ochranného obalu a následná kompletace jednotky. Tyto úkony budu dělat já.

Cena outsourcované výroby DPS a osazení jsou určeny kalkulátorem cen na webu [www.pcbway.com](http://www.pcbway.com). Cena vlastní výroby a osazení DPS zahrnuje odhadovaný přímý materiál, mzdy + odvody a pořízení čtyř pájecích stanic (dvě na SMT, dvě na through hole pájení) v ceně 12 000 Kč.





Obrázek 49 - Porovnání cen vlastní a outsourcované výroby a osazení DPS

V tabulce 8 jsou zahrnuty veškeré náklady na výrobu měřících jednotek a instalaci do odpadových nádob. Položky výrobních nákladů obsahují náklady včetně poštovního, balného a cel. Položka *Náklady na instalaci* v sobě zahrnuje náklady na instalaci měřící jednotky do odpadové nádoby včetně půjčení automobilu od rodičů za cenu 4 Kč/km a předpokládanou vzdálenost odpadových nádob od sebe 300 m. Všechny položky jsou variabilní náklady, tento projekt nemá žádné fixní výrobní náklady.

Tabulka 8 - Výrobní náklady na měřící jednotku

Položka	Cena/ks (100 ks) [Kč]	Cena/ks (400 ks) [Kč]	Cena/ks (1000 ks) [Kč]
RTC	22,00	22,00	22,00
GSM modul	75,00	75,00	75,00
Ultrazvukový senzor	18,00	18,00	18,00
Baterie do RTC	3,00	3,00	3,00
Držák baterií	31,84	31,84	27,99
C1	0,21	0,21	0,16
C2	0,21	0,21	0,16
R1	0,05	0,05	0,04
Y1	2,95	2,95	2,51
C3	0,18	0,18	0,14
C4	0,18	0,18	0,14
R2	0,06	0,06	0,04

LED1	1,17	1,17	1,05
Q1	2,93	2,93	1,82
R3	1,17	1,17	1,05
LED2	1,17	1,17	1,14
C6	1,47	1,47	1,47
C7	1,32	1,32	0,96
U1	36,72	36,72	35,95
ICSP DIP-6	0,80	0,80	0,59
3x alkalické baterie	24,41	24,41	24,41
Drobný spotřební materiál	10,00	10,00	10,00
Výroba a osazení DPS	118,00	58,00	41,00
Náklady na výrobu ochranného obalu	58,00	58,00	58,00
<b>Celková cena nákladů přímého materiálu</b>	<b>410,84</b>	<b>350,84</b>	<b>326,62</b>
Náklady na instalaci	2	2	2
Mzdy	0,00	0,00	0,00
<b>Celkové náklady na měřící jednotku</b>	<b>412,84</b>	<b>352,84</b>	<b>328,62</b>

## 7.2 Cena služby

Plánovaná konkurenční výhoda je mít nižší cenu pronájmu služby než konkurence. Analýza trhu a kontaktování konkurenčních firem ukázala, že nejlevnější provozovatel z firem, které byly ochotny poskytnout informace o ceně, je firma Enevo s cenou 10 euro/měřící jednotka/měsíc při objednání 400 kusů viz tabulka 9.

Tabulka 9 - Ceny konkurence

Provozovatel	Požizovací cena	Cena/měsíc/jednotka při 400 ks
Enevo	Nebylo zjištěno	10 euro
O2 IT Services	3000 Kč [6]	764 Kč [6]
Ecube labs	Nebylo zjištěno	15 euro

Minimální cena za měsíční pronájem měřící jednotky při počtu 400 kusů, aby na konci životnosti projektu byl nulový zisk je 38 Kč. Z konzultace s potenciálním zákazníkem – klientem firmy Ing. Pavel Novák s.r.o. – vyplynulo, že maximální přijatelná cena za měsíční pronájem měřící jednotky by byla 100 Kč. Jelikož je snaha o maximalizaci zisku, měsíční cena za pronájem jedné měřící jednotky včetně internetové aplikace při počtu 400 kusů byla stanovena na 100 Kč.

## 7.3 Cash flow

Cash flow (tab. 10) je v tisících a je vypočítáno přímou metodou – tj. příjmy mínus výdaje.

Tabulka 10 - Cash flow projektu

Položka	Rok 0	Rok 1	Rok 2	Rok 3	Rok 4
Výdaj na výrobu a instalaci	-148				
Provozní výdaje		-145	-145	-145	-145
Příjmy		480	480	480	480
CF	-148	335	335	335	335
DCF	-148	319	304	289	276

CF je cash flow projektu v jednotlivých letech. Pro zohlednění budoucí ceny peněz je také vypočítán DCF – diskontovaný cash flow – ze vzorce 6, kde  $CF_t$  je cash flow v jednotlivých obdobích,  $t$  je dané období a  $r$  je diskont. Diskont byl zvolen na 5 %. Součet diskontovaných peněžních toků je 1040 tis. a je to zároveň NPV projektu.

$$DCF_t = \frac{CF_t}{(1+r)^t} \quad (6)$$

*Výpočet diskontovaného cash flow*

## 8 Přínosy projektu

Aby zákazník investoval do sledování odpadových nádob, musí mít investice nějaké pozitivní přínosy. Přínosy tohoto projektu se dělí na dvě části – na ekonomické a na neekonomické.

### 8.1 Neekonomické přínosy projektu

Jak bylo podotknuto v úvodní kapitole, hlavní neekonomický důvod k pořízení systému sledování odpadových nádob, je zabránění znečišťování okolí při plných nádobách. Pokud je provozovatel odpadových nádob zároveň majitelem pozemku (typicky město), tak znečištění nepůsobí dobře a dělá špatnou vizitku danému místu. Pokud provozovatel odpadových nádob není majitelem pozemku, majitel pozemku může při znečištění v okolí odpadových nádob z důvodu přeplnění udělit provozovateli pokutu, případně vypovědět smlouvu na umístění odpadových nádob.

### 8.2 Ekonomické přínosy projektu

Ekonomické přínosy projektu spočívají v ušetřených nákladech při chytrému svozu oproti konvenčnímu svozu. Ekonomický svozový model je vypočítán pro svoz 100 odpadových nádob o objemu 2 m<sup>3</sup>, produkci odpadu 100 m<sup>3</sup> týdně a předpokládá se, že lidé při zaplnění odpadové nádoby odnesou odpad do nejbližší prázdné nádoby. Klient na svoz používá běžný užitkový vůz (dodávka) typu Fiat Ducato či

Ford Transit s nákladovým prostorem 20 m<sup>3</sup>. Průměrná zaplněnost vyvážené odpadové nádoby je 50 %, vyprázdnění nádoby trvá 15 minut, vyskladnění zaplněného auta trvá 30 minut a průměrná vzdálenost do skladu při plném autě je 11 km. Souhrnné informace jsou v tabulce 11.

*Tabulka 11 - Základní informace o svozu*

Objem odpadové nádoby	2 m <sup>3</sup>
Objem nákladového prostoru	20 m <sup>3</sup>
Počet nádob	100 ks
Čas na výběr nádoby	15 min
Průměrná zaplněnost nádob	50 %
Vyskladnění zaplněného auta	30 min
Průměrná vzdálenost do skladu	11 km

Posádka auta se skládá z dvou lidí a jejich hodinový náklad je 500 Kč. V nákladech na posádku se počítá mzda, odvody a režie. Náklad na provoz auta je 5 Kč/km. Všechny výše zmíněné údaje byly získány od klienta firmy Ing. Pavel Novák s.r.o.

*Tabulka 12 - Náklady na posádku a auto*

Práce svozové posádky	500 Kč/h
Auto (dodávka)	5 Kč/km

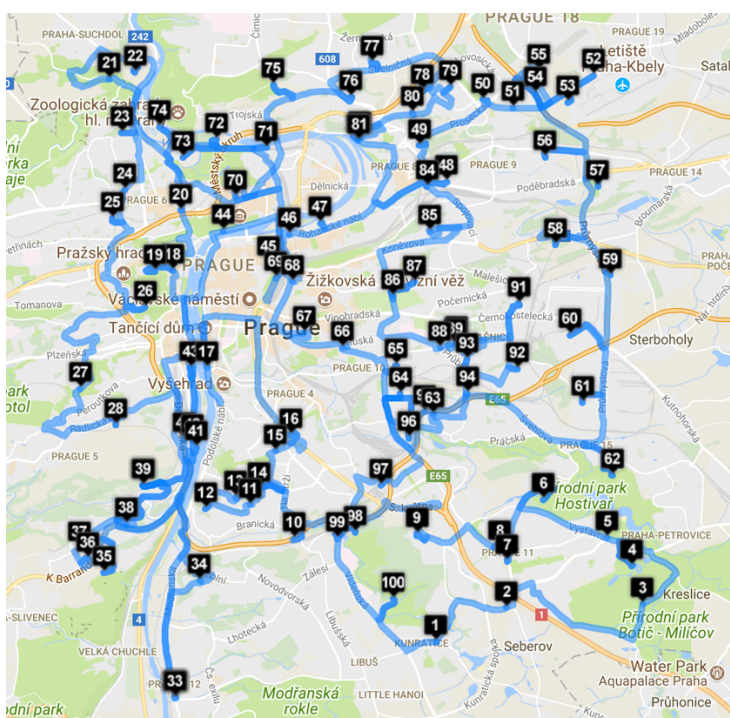
Pro výpočet nákladů na jednotlivé svozy, bylo náhodně vygenerováno 100 bodů na území Hlavního města Prahy a ty se použily jako umístění odpadových nádob. Výchozí bod a zároveň sklad odpadu je v Praze – Kunraticích a je na mapě viz obrázek 50 označen špendlíkem č. 1.

### **8.2.1 Běžný svoz**

Při běžném svozu, majitel odpadových nádob vyváží pravidelně jednou týdně všechny nádoby bez ohledu na zaplnění. Optimální trasa na objety všech odpadových nádob byla vypočítána pomocí online aplikace na optimalizaci tras OptiMap (<https://www.gebweb.net/optimap/>) (obr. 50). Položka *Vzdálenost – vyskladňování* v tabulce 13 je celková vzdálenost mezi trasou a skladem, kterou posádka ujede pro vyskladňování vozu. Vypočítané náklady na jeden týdenní běžný svoz všech odpadových nádob jsou 21 991 Kč viz tabulka 13.

Tabulka 13 - Kalkulace běžný svoz

Vzdálenost – trasa	264 km
Vzdálenost – vyskladňování	55 km
Celková vzdálenost	319 km
Čas na ujetí celkové vzdálenosti	13,3 h
Čas na výběr kontejnerů	25,0 h
Čas na vyskladnění vozu	2,5 h
Celkový čas	40,8 h
Náklady na auto	1595 Kč
Náklady na mzdy	20 396 Kč
<b>Celkové náklady na jeden svoz</b>	<b>21 991 Kč</b>



Obrázek 50 - Svozová trasa běžného svozu

## 8.2.2 Chytrý svoz

Informace z měřících jednotek se dají použít k optimalizaci svozu. Chytrý svoz může mít dvě úrovně optimalizace. První úroveň spočívá v pouhém vynechání nezaplněných nádob na svozové trase, druhá úroveň optimalizuje trasu samotnou. Obě optimalizace počítají s ideálním případem, kdy se vyváží pouze zcela zaplněné nádoby.

### 8.2.2.1 Stejná trasa

Zachování stejné svozové trasy jako při běžném svozu a vynechání zcela nezaplněných odpadových nádob, vyústí v snížení potřebného času na ruční výběr odpadu. Při plném zaplnění a objemu 2 m<sup>3</sup> jedné odpadové nádoby, je potřeba na pokrytí týdenní produkce odpadu vyvézt 50 odpadových nádob. Oproti běžnému svozu se ušetří 28 % nákladů, tj. 6250 Kč týdně viz tabulka 14.

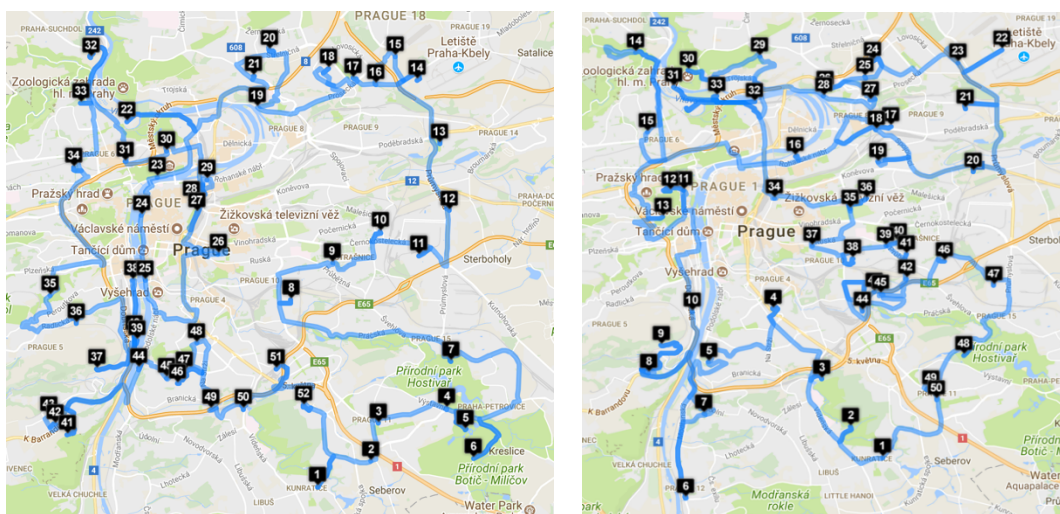
Tabulka 14 - Kalkulace chytrý svoz – stejná trasa

Vzdálenost – trasa	264 km
Vzdálenost – vyskladňování	55 km
Celková vzdálenost	319 km
Čas na ujetí celkové vzdálenosti	13,3 h
Čas na výběr kontejnerů	13 h
Čas na vyskladnění vozu	2,5 h
Celkový čas	28,3 h
Náklady na auto	1595 Kč
Náklady na mzdy	14 146 Kč
<b>Celkové náklady na jeden svoz</b>	<b>15 741 Kč</b>
<b>Rozdíl oproti běžnému svozu</b>	<b>-6250 Kč</b>

### 8.2.2.2 Optimalizace trasy

Při chytrém vývozu s optimalizací trasy je stejný předpoklad jako při chytrém vývozu se stejnou trasou – k vyvezení týdenní produkce odpadu je potřeba vybrat 50 zcela naplněných nádob.

Svoz všech sto nádob se rozdělí na dvě části. V první týdnu svozu se vyveze prvních 50 plných nádob, v druhém týdnu se vyveze druhých 50 plných nádob. Pro každý svoz se naplánuje optimální trasa. V modelu bylo náhodně vybráno prvních i druhých 50 nádob viz obrázek 51.



Obrázek 51 - Optimální trasa první a druhý týden

Tabulka 15 - Kalkulace chytrý svoz s optimalizací trasy

	<b>Týden 1</b>	<b>Týden 2</b>
Vzdálenost – trasa	164 km	154 km
Vzdálenost – vyskladňování	55 km	55 km
Celková vzdálenost	219 km	209 km
Čas na ujetí celkové vzdálenosti	9,1 h	8,7 h
Čas na výběr kontejnerů	13 h	13 h
Čas na vyskladnění vozu	2,5 h	2,5 h
Celkový čas	24,1 h	23,7 h
Náklady na auto	1095 Kč	1045 Kč
Náklady na mzdy	12 063 Kč	11 854 Kč
<b>Celkové náklady</b>	<b>13 158 Kč</b>	<b>12 899 Kč</b>
<b>Celkové průměrné náklady na svoz</b>	<b>13 028 Kč</b>	
<b>Rozdíl oproti běžnému svozu</b>	<b>-8833 Kč</b>	

Celkové průměrné náklady na týdenní optimalizovaný svoz jsou 13 028 Kč (tab. 15), což je o 40 % méně než při používání běžného svozu. Roční snížení nákladů na svoz je tedy o 107 550 Kč.

### 8.2.3 Závěr ekonomických přínosů projektu

Oba případy chytrého svozu byly kalkulovány při ideálních podmínkách, kdy týdenní produkce je rozložena do plně zaplněných odpadových nádob. V reálné aplikaci by se musely vyvážet i odpadové nádoby, které by zcela zaplněné nebyly, aby se vývozem pokryla celá týdenní průměrná produkce odpadu. Trend zaplňování se nedá dost dobře aproximovat a musí být tedy použita ideální data.

Při chytrém svozu s optimalizovanou trasou je předpokládané 40% snížení nákladů. Velmi podobný výsledek vypočetala i společnost O2 IT Services, která kalkulovala ušetřené náklady obdobného projektu pro město Kolín. Jejich vypočítané předpokládané snížení nákladů bylo 30 – 50 % [39].

Cena pronájmu jedné měřicí jednotky je 100 Kč za měsíc. Při pronájmu 100 měřících jednotek, roční cena pronájmu je 120 000 Kč. Ušetřené roční náklady při chytrém svozu s optimalizací jsou 107 550 Kč, což ale nedokáže pokrýt ani cenu pronájmu služby. Je tedy vidět, že projekt by pro klienta čistě z ekonomického hlediska neměl smysl. Projekt by klientovi dal ekonomický smysl, pokud by cena pronájmu jedné měřicí jednotky byla nižší než 89 Kč.

## 9 Závěr

Cílem bakalářské práce bylo provést rozbor současného systému pro dálkový sběr dat pro tříděný odpad, navrhnout vlastní koncept zařízení pro monitoring, vytvořit obslužný software a provést rozvalu přínosů projektu. Všechny body se mi podařilo splnit.



V první části jsem analyzoval trh a zjistil, jaké technologie jsou nejvhodnější pro dálkové sledování zaplněnosti odpadových nádob. Na základě tohoto rozboru jsem zvolil měření ultrazvukem a přenášení dat pomocí GSM sítě. Důvod ke zvolení GSM technologie bylo pokrytí celého území České republiky veřejně dostupnou sítí. V době, kdy jsem začínal navrhovat celý systém monitoringu, specializované bezdrátové sítě pro IoT neměly 100% pokrytí celého území České republiky, což se ale ke květnu 2018 změnilo. Použití GSM sítě je poměrně drahé – měsíčně jedna SIM pro M2M aplikace stojí 30 Kč, což při velkém počtu měřících jednotek je největší provozní náklad celého projektu. Z tohoto důvodu by při reálné aplikaci bylo vhodné přejít na komunikační technologie SigFox, NB-IoT nebo Lora, které jsou speciálně vyvinuté pro IoT a jejich cena začíná na 7,50 Kč/měsíc za jedno zařízení.

V druhé části jsem se zabýval realizací a testováním mého navrženého konceptu měření zaplněnosti odpadových nádob. Koncept se skládá z dvou částí – měřících jednotek umístěných v odpadových nádobách a serveru, na kterém běží internetová aplikace, která se stará o příjem, zpracování a interpretaci dat. Dlouhodobý test, který spočíval v umístění měřící jednotky do popelnice na bio odpad, potvrdil funkčnost konceptu. Pro reálnou aplikaci by však bylo ještě potřeba v internetové aplikaci naprogramovat funkci optimalizace svozových tras a drobně upravit HW měřící jednotky, aby se nepoužívaly neznačkové moduly s neznámou kvalitou.

V poslední části byla provedena analýza přínosů projektu. Přínosy projektu se dělí na dvě části – na neekonomické a na ekonomické. Neekonomické přínosy spočívají v nepřeplnění odpadových nádob, čímž je zvýšená spokojenost zákazníků a zabraňuje se tím vzniku nelegální skládky kolem nádob. Ekonomický přínos je snížení nákladů na vývoz při optimalizaci svozové trasy na základě zaplnění odpadových nádob. Ze svozového ekonomického modelu vyplynulo, že při použití chytrého svozu s optimalizací tras, je možno snížit náklady na svoz až o 40 %. Podobným výsledkům se také dopočítala firma O2 IT Services, která spočítala pro město Kolín, že při použití chytrého svozu se ušetří 30 – 50 % celkových nákladů na svoz [39]. Tyto modely však počítají s ideálním případem, kdy se vyvázejí pouze zcela zaplněné odpadové nádoby. V reálném provozu na pokrytí týdenní produkce odpadu se budou muset vyvážet i nádoby ne zcela zaplněné a tím se sníží ekonomický přínos. Přesný ekonomický přínos by se musel zjistit empiricky po určité době používání systému.

Pro mě samotného měla bakalářská práce obrovský přínos, protože s hardwarovým a softwarovým vývojem jsem se ve svém dosavadním studiu nesetkal a neměl jsem tedy téměř žádné zkušenosti. Práce mi pomohla rozšířit a prohloubit teoretické a praktické znalosti z oblasti programování mikrokontrolérů, hardware návrhu a programování v jazycích PHP, HTML, CSS a JavaScript.

Můj navržený koncept sledování zaplnění odpadových nádob byl potvrzený jako funkční. Projekt nicméně v nynějším stádiu ještě není připravený na uvedení na trh. V následujících měsících mám v plánu dovyvinout měřící jednotku a internetovou aplikaci ke komerčnímu použití a poté uvést produkt na trh a pokusit se uspět v rychle se rozvíjejících odvětvích internetu věcí a smart odpadového hospodářství.



# Literatura

---

- [1] OICT. Smart Prague do roku 2030. In: *Smartprague.eu* [online]. 6. 2017 [cit. 2018-02-09]. Dostupné z: [https://www.smartprague.eu/files/koncepce\\_smartprague.pdf](https://www.smartprague.eu/files/koncepce_smartprague.pdf)
- [2] KAŠPAR, Michael. Smart city Kolin. In: *Mukolin.cz* [online]. [cit. 2018-03-12]. Dostupné z: <http://www.mukolin.cz/cz/o-meste/smart-city-kolin/odpady/>
- [3] Components Choosing an Ultrasonic Sensor for Proximity or Distance Measurement Part 2: Optimizing Sensor Selection. In: *Sensormag.com* [online]. 1. 3. 1999 [cit. 2018-02-21]. Dostupné z: <https://www.sensormag.com/components/choosing-ultrasonic-sensor-for-proximity-or-distance-measurement-part-2-optimizing>
- [4] Technology & Trash: The Winning Combination. In: *Medium.com* [online]. 28. 11. 2017 [cit. 2018-03-29]. Dostupné z: <https://medium.com/@compology/technology-trash-the-winning-combination-582a6f41216>
- [5] MU Kolín - odpady. In: *Mukolin.cz* [online]. 31. 8. 2016 [cit. 2018-04-09]. Dostupné z: <http://www.mukolin.cz/cz/o-meste/smart-city-kolin/odpady/>
- [6] Kolín: Senzory v kontejnerech nic neušetří, město je ale uklizenější. In: *Lupa.cz* [online]. 6. 9. 2017 [cit. 2018-04-14]. Dostupné z: <https://www.lupa.cz/clanky/kolin-senzory-v-kontejnerech-nic-neusetri-mesto-je-ale-uklizenejsi/>
- [7] Frekvenční přiděl na pásmech GSM, DCS, UMTS a LTE v České republice. In: *Gsmweb.cz* [online]. 2005 [cit. 2018-04-17]. Dostupné z: <https://www.gsmweb.cz/clanky/freq2.htm>
- [8] POOLE, Ian. *GSM EDGE tutorial* [online]. In: . [cit. 2018-04-17]. Dostupné z: <http://www.radio-electronics.com/info/cellulartelecomms/gsm-edge/basics-tutorial-technology.php>
- [9] STRINGFELLOW, Angela. What is 4G LTE?. In: *Tccrocks.com* [online]. 11. 9. 2017 [cit. 2018-04-17]. Dostupné z: <https://www.tccrocks.com/blog/what-is-4g-lte/>
- [10] Mapa pokrytí v ČR veřejnou mobilní sítí LTE. In: *Ctu.cz* [online]. [cit. 2018-04-17]. Dostupné z: <https://digi.ctu.cz/lte-pokryti/>
- [11] POOLE, Ian. Bluetooth radio interface, modulation, & channels. In: *Radio-electronics.com* [online]. [cit. 2018-05-06]. Dostupné z: <http://www.radio-electronics.com/info/wireless/bluetooth/radio-interface-modulation.php>
- [12] POOLE, Ian. Wi-Fi / WLAN Channels, Frequencies, Bands & Bandwidths. In: *Radio-electronics.com* [online]. [cit. 2018-05-06]. Dostupné z: <http://www.radio-electronics.com/info/wireless/wi-fi/80211-channels-number-frequencies-bandwidth.php>
- [13] The evolution of WiFi standards: a look at 802.11a/b/g/n/ac. In: *Actiontec.com* [online]. [cit. 2018-04-17]. Dostupné z: <https://www.actiontec.com/wifihelp/evolution-wi-fi-standards-look-802-11abgnac/>
- [14] The ZigBee vs Wifi battle for M2M communication. In: *Link-labs.com* [online]. 11. 2. 2015 [cit. 2018-04-17]. Dostupné z: <https://www.link-labs.com/blog/zigbee-vs-wifi-802-11ah>
- [15] POOLE, Ian. Sigfox for M2M and IoT. In: *Radio-electronics.com* [online]. [cit. 2018-04]. Dostupné z: <http://www.radio-electronics.com/info/wireless/sigfox/basics-tutorial.php>

- [16] LoRAWAN. In: *Iot-portal.cz* [online]. 29. 2. 2016 [cit. 2018-04-17]. Dostupné z: <https://www.iot-portal.cz/2016/02/29/lorawan/>
- [17] NB-IoT - enabling new business opportunities. In: *Huawei.com* [online]. 2015 [cit. 2018-04-18]. Dostupné z: [http://www.huawei.com/minisite/iot/img/nb\\_iot\\_whitepaper\\_en.pdf](http://www.huawei.com/minisite/iot/img/nb_iot_whitepaper_en.pdf)
- [18] ATmega328P microcontroller. In: *Microchip.com* [online]. [cit. 2018-04]. Dostupné z: <http://www.microchip.com/wwwproducts/en/ATmega328p>
- [19] Battery performance characteristics. In: *Mpoweruk.com* [online]. London, 23. 2. 2007 [cit. 2018-04-18]. Dostupné z: <http://www.mpoweruk.com/performance.htm>
- [20] Elevating Self-discharge. In: *Batteryuniversity.com* [online]. 4. 4. 2017 [cit. 2018-04]. Dostupné z: [http://batteryuniversity.com/learn/article/elevating\\_self\\_discharge](http://batteryuniversity.com/learn/article/elevating_self_discharge)
- [21] HC-SR04 Datasheet. In: *Micropik.com* [online]. June 2014 [cit. 2018-03-17]. Dostupné z: <http://www.micropik.com/PDF/HCSR04.pdf>
- [21] HC-SR04 Datasheet. In: *Micropik.com* [online]. June 2014 [cit. 2018-03-17]. Dostupné z: <http://www.micropik.com/PDF/HCSR04.pdf>
- [22] SIMCOM SIM800L module. In: *4gltemall.com* [online]. [cit. 2018-03-17]. Dostupné z: <https://www.4gltemall.com/simcom-sim800l.html>
- [23] LPC2148 UART Tutorial. In: *Electronicshub.org* [online]. 3. 12. 2016 [cit. 2018-03-17]. Dostupné z: <https://www.electronicshub.org/lpc2148-uart-tutorial/>
- [24] RTC DS3231 datasheet. In: *Maximintegrated.com* [online]. 3. 2015 [cit. 2018-04-01]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>
- [25] DS3231 RTC High Precision Real-Time Clock Module with AT24C32 EEPROM. In: *Makerlab-electronics.com* [online]. [cit. 2018-04-06]. Dostupné z: <https://www.makerlab-electronics.com/product/ds3231-rtc-high-precision-real-time-clock-module/>
- [26] NXP. I2C bus specification. In: *Nxp.com* [online]. 4. 4. 2014 [cit. 2018-04-02]. Dostupné z: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [27] How to install LAMP stack on Ubuntu 17.04. In: *Unixmen.com* [online]. 25. 5. 2017 [cit. 2018-04-03]. Dostupné z: <https://www.unixmen.com/how-to-install-lamp-stack-ubuntu-17-04/>
- [28] Sim800l hardware design datasheet. In: *Simcom.ee* [online]. 3. 5. 2015 [cit. 2018-04-04]. Dostupné z: [http://simcom.ee/documents/SIM800/SIM800\\_Hardware%20Design\\_V1.08.pdf](http://simcom.ee/documents/SIM800/SIM800_Hardware%20Design_V1.08.pdf)
- [29] *Atmega 328p datasheet* [online]. In: . 2016 [cit. 2018-04-04]. Dostupné z: [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf)
- [30] SMD QUARTZ CRYSTAL RESONATOR datasheet. In: *Gme.cz* [online]. [cit. 2018-04-04]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.131-075.1.pdf>
- [31] Atmega 328p datasheet. In: *Microchip.com* [online]. 2016 [cit. 2018-04-08]. Dostupné z: [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf)

- [32] LIU, Allen. KP-1608SGC Datasheet. *Farnell.com* [online]. Únor 2012 [cit. 2018-04-09]. Dostupné z: [http://www.farnell.com/datasheets/2045827.pdf?\\_ga=2.86789853.414364001.1524743601-238257970.1522182431&\\_gac=1.182900308.1522689833.Cj0KCCQjwqYfWBRDPARIsABjQRYy04j3Z51SXqKy5pcQ60PEfCT11Yv-2rGTzV4ut55D8O6WoOTB2fCwaAnJOEALw\\_wcB](http://www.farnell.com/datasheets/2045827.pdf?_ga=2.86789853.414364001.1524743601-238257970.1522182431&_gac=1.182900308.1522689833.Cj0KCCQjwqYfWBRDPARIsABjQRYy04j3Z51SXqKy5pcQ60PEfCT11Yv-2rGTzV4ut55D8O6WoOTB2fCwaAnJOEALw_wcB)
- [33] Serial Peripheral Interface (SPI). In: *Sparkfun* [online]. [cit. 2018-04-10]. Dostupné z: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>
- [34] Arduino DS3232RTC Library. In: *Github* [online]. Březen 2013 [cit. 2018-04-15]. Dostupné z: <https://github.com/JChristensen/DS3232RTC>
- [35] Pull-up Resistors. In: *Sparkfun.com* [online]. [cit. 2018-04-15]. Dostupné z: <https://learn.sparkfun.com/tutorials/pull-up-resistors>
- [36] SIM800 Series\_Serial Port\_Application Note\_V1.01. *SIMCOM* [online]. 8. 4. 2014 [cit. 2018-04-19]. Dostupné z: [https://cdn-shop.adafruit.com/product-files/2637/SIM800+Series\\_Serial+Port\\_Application+Note\\_V1.01.pdf](https://cdn-shop.adafruit.com/product-files/2637/SIM800+Series_Serial+Port_Application+Note_V1.01.pdf)
- [37] Arduino reference - millis(). *Arduino.cc* [online]. 2015 [cit. 2018-04-20]. Dostupné z: <https://www.arduino.cc/reference/en/language/functions/time/millis/>
- [38] ČOŽÍK, Ondřej. POUŽITÍ BLOKOVACÍCH KONDENZÁTORŮ PŘI NÁVRHU DPS [online]. Brno, 2013 [cit. 2018-05-07]. <Http://www.slaboproudyobzor.cz/files/20130308.pdf>. VUT v Brně.
- [39] Kolín s O2 testují chytré kontejnery. Mají ušetřit a ještě se samy zaplatit. *Lupa.cz* [online]. 30. 8. 2016 [cit. 2018-05-13]. Dostupné z: <https://www.lupa.cz/clanky/kolin-s-o2-testuji-chytre-kontejnery-maji-usetrit-a-jeste-se-samy-zaplatit/>

## Seznam tabulek a obrázků

---

Obrázek 1 - Nelegální skládka kolem sběrných nádob na textil. 5.5. 2017, Olbrachtova, Praha 4 .....	10
Obrázek 2 - Ultrazvukové měření odpadové nádoby .....	12
Obrázek 3 - Nerovnoměrné rozprostření odpadu v kontejneru [4] .....	12
Obrázek 4 - Velkoobjemový kontejner s obrazovým senzorem [4] .....	13
Obrázek 5 - Odpadové nádoby s QR kódem v Kolíně [6] .....	14
Obrázek 6 - Blokový diagram konceptu systému .....	18
Obrázek 7 - Blokový diagram měřící jednotky .....	19
Obrázek 8 - Microchip ATmega328P [18] .....	19
Obrázek 9 - Časový diagram komunikace ultrazvukového senzoru [21] .....	22
Obrázek 10 - Ultrazvukový senzor HC – SR04 [21] .....	22
Obrázek 11 - GSM modul SIM800l [22] .....	22
Obrázek 12 - Grafické znázornění UART datového paketu [23] .....	23
Obrázek 13 - RTC DS3231 modul [25] .....	24
Obrázek 14 - Grafické znázornění průběhu I2C komunikace [26]. .....	25
Obrázek 15 - Blokové schéma serveru .....	25
Obrázek 16 - Lamp logo [27] .....	26
Obrázek 17 Relační model databáze .....	27
Obrázek 18 - Elektrické schéma měřící jednotky .....	28
Obrázek 19 - Proudové špičky GSM modulu [28] .....	29
Obrázek 20 - SPI komunikace [33] .....	30

Obrázek 21 - Návrh desky plošného spoje.....	31
Obrázek 22 - Osazená deska plošného spoje.....	32
Obrázek 23 - 3D model ochranného obalu.....	32
Obrázek 24 - Finální podoba měřicí jednotky .....	33
Obrázek 25 - Inicializace.....	34
Obrázek 26 - Funkce setup() .....	34
Obrázek 27 - Funkce nastaveniRTC().....	36
Obrázek 28 - Funkce nastaveniSpanku .....	37
Obrázek 29 - Schéma pull-up resistoru [35].....	37
Obrázek 30 – Smyčka loop().....	38
Obrázek 31 - Funkce zmerit().....	40
Obrázek 32 - Funkce usni() .....	40
Obrázek 33 - Funkce connection().....	41
Obrázek 34 - Funkce BATT().....	42
Obrázek 35 - Funkce CGATT().....	42
Obrázek 36 - Funkce SAPBR() .....	43
Obrázek 37 - Funkce SAPBR2() .....	43
Obrázek 38 - Funkce SAPBR3() .....	44
Obrázek 39 - Funkce HTTPINIT() .....	44
Obrázek 40 - Funkce HTTPPARA().....	45
Obrázek 41 - Funkce HTTPACTION().....	45
Obrázek 42 - Pokles napětí v systému .....	46
Obrázek 43 - Obvod pro postupné sepnutí tranzistoru MOSFET.....	46
Obrázek 44 - Struktura a vzájemné vztahy tabulek databáze „bakalarka“ .....	47
Obrázek 45 - Stránka přihlášení.....	49
Obrázek 46 - Domovská stránka (aktuální data).....	49
Obrázek 47 - Detail odpadové nádoby.....	50
Obrázek 48 - Stránka Mapa.....	51
Obrázek 49 - Porovnání cen vlastní a outsourcované výroby a osazení DPS .....	53
Obrázek 50 - Svozová trasa běžného svozu .....	57
Obrázek 51 - Optimální trasa první a druhý týden.....	58
Tabulka 1 - Operátoři v ČR a jejich frekvenční pásma pro GSM [7] .....	15
Tabulka 2 - Operátoři v ČR a jejich frekvenční pásma pro LTE [7].....	16
Tabulka 3 - Kapacita baterií v jednotlivých letech .....	20
Tabulka 4 - Porovnání ultrazvukových senzorů .....	21
Tabulka 5 - Základní parametry 3D tisku.....	32
Tabulka 6 - Typy alarmů [34].....	35
Tabulka 7 - Provozní náklady.....	52
Tabulka 8 - Výrobní náklady na měřicí jednotku .....	53
Tabulka 9 - Ceny konkurence.....	54
Tabulka 10 - Cash flow projektu .....	55
Tabulka 11 - Základní informace o svozu.....	56
Tabulka 12 - Náklady na posádku a auto .....	56
Tabulka 13 - Kalkulace běžný svoz .....	57
Tabulka 14 - Kalkulace chytrý svoz – stejná trasa.....	58
Tabulka 15 - Kalkulace chytrý svoz s optimalizací trasy.....	59

# Seznam příloh

---

Příloha 1 – Vývojový diagram programu v mikrokontroléru

Příloha 2 – Program mikrokontroléru

Příloha 3 – Zdrojové soubory serveru a internetové aplikace

Veškeré přílohy jsou na přiloženém CD.