

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020

Bc. Vladimír Matula





# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## AUTO POČÍTAČ ZALOŽENÝ NA RASPBERRY PI

CAR COMPUTER BASED ON RASPBERRY PI

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Vladimír Matula

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Münster, Ph.D.

BRNO 2020



# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Vladimír Matula

**ID:** 186139

**Ročník:** 2

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Auto počítač založený na Raspberry Pi

**POKYNY PRO VYPRACOVÁNÍ:**

Cílem práce je teoretický rozbor problematiky realizace počítačů do automobilů pomocí mikropočítačů Raspberry Pi. Student navrhne a realizuje univerzální řešení auto počítače založeného na Raspberry Pi s možností vzdáleného určení polohy vozu, a dále čtení a grafické zobrazení informací z palubního počítače. K auto počítači bude vytvořena aplikace pracující pod systémem Android. Práce bude rovněž obsahovat návrh možného rozšíření auto počítače o LCD a kamerový systém.

**DOPORUČENÁ LITERATURA:**

[1] NORRIS, Donald. Raspberry Pi projects for the evil genius. New York: McGraw Hill Education, [2014]. Evil genius series. ISBN 978-0071821582.

[2] BLUM, Richard a Christine BRESNAHAN. Sams teach yourself Python programming for Raspberry Pi in 24 hours. Indianapolis, Indiana: Sams, [2014]. Sams teach yourself. ISBN 978-0789752055.

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 1.6.2020

**Vedoucí práce:** Ing. Petr Münster, Ph.D.

**prof. Ing. Jiří Mišurec, CSc.**  
předseda oborové rady

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.



## ABSTRAKT

Hlavnou témou diplomovej práce je rozbor problematiky týkajúcej sa inteligentných vozidiel. Daná problematika zahŕňa možnosti automatizácie a personalizácie funkcií automobilu spolu s diaľkovým ovládaním rôznych komponentov.

Cieľom práce je poskytnúť všeobecný prehľad o možnostiach rozšírenia vozidla o auto-počítač založený na zariadení Raspberry Pi Zero W.

Práca je aplikačného charakteru, štrukturovaná do dvoch častí. V teoretickej časti autor približuje problematiku Internetu vecí a komunikácie dvoch zariadení, ako aj popisuje rôzne riešenia inteligentných vozidiel.

Praktická časť ponúka náhľad na autorom vybraný spôsob implementácie auto-počítača založeného na Raspberry Pi Zero W, návrh vlastného zabezpečovacieho systému vozidla s možnosťou detekcie pohybu a získavanie aktuálnych informácií o polohe vozidla. V predloženej práci je tiež uvedený postup vytvorenia vlastnej mobilnej aplikácie pre zobrazovanie stavových informácií z palubného počítača vozidla a ovládanie niekoľkých funkcií auto-počítača.

## KLÚČOVÉ SLOVÁ

Raspberry Pi, Inteligentné auto, AutoPi, auto-počítač, IoT, Internet vecí, M2M, MQTT

## ABSTRACT

The main goal of this diploma thesis is to investigate the smart cars field together with the technologies linked closely to it. Further, the thesis offers an overview of options to extend the vehicle capabilities with the car computer created with the Raspberry Pi Zero W.

The thesis provides hands-on experience and is structured into two parts. The first, theoretical section focuses on Internet of Things and Machine to Machine communication. It also describes the currently available smart car solutions on the market, their pros and cons.

The practical part of this thesis describes design and creation of a smart computer based on Raspberry Pi Zero W with draft of self-made security system from vehicle containing a motion sensor and a camera. It also includes a development and integration steps with a mobile application capable of viewing the static information regarding the car state, such as Rotations per minute. Vehicle speed, etc. The created system also has the capability of informing the user about its current location on request.

## KEYWORDS

Raspberry Pi, Smart Car, AutoPi, Car Computer, IoT, Internet of Things, M2M, MQTT

MATULA, Vladimír. *Chytré auto založené na Raspberry Pi Zero*. Brno, Rok, 65 s. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Petr Münster, Ph.D.





## VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Chytré auto založené na Raspberry Pi Zero“ som vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora



## POĎAKOVANIE

Rád by som podakoval vedúcemu semestrálnej práce panu Ing. Petru Münsterovi , Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.



# Obsah

Úvod	19
<b>1 IoT</b>	<b>21</b>
1.1 História	22
<b>2 M2M komunikácia</b>	<b>23</b>
2.1 História	24
2.2 Bezpečnosť v M2M	24
<b>3 Problematika v teórii</b>	<b>27</b>
3.1 Výhody automatizácie	27
3.2 Nevýhody automatizácie	28
<b>4 Technológia inteligentného vozidla</b>	<b>29</b>
4.1 Carberry	29
4.2 AutoPi Dongle	29
4.3 Xmarton	30
4.4 Vlastné riešenie	30
<b>5 Návrh vlastného riešenia</b>	<b>31</b>
5.1 AutoPi.io	32
5.1.1 Raspberry Pi Zero W	32
5.2 OBD	34
5.3 Použité technológie	35
5.3.1 Protokol MQTT	35
5.4 Postup práce	36
5.4.1 Inštalácia operačného systému	36
5.4.2 Návrh pripojenia LCD displeja a kamery	37
5.4.3 Pripojenie PIR senzoru	38
5.4.4 Konfigurácia 4G/LTE pripojenia	39
5.4.5 Konfigurácia bezdrôtového rozhrania	40
5.4.6 Nastavenie SMS komunikácie	42
5.4.7 Konfigurácia MQTT servera a klienta	44
5.4.8 Komunikácia s OBD II	44
5.4.9 Komunikácia s mobilnou aplikáciou	46
5.4.10 Vytvorenie mobilnej aplikácie	48
5.4.11 Ovládanie komponentov vozidla	50
5.5 Popis priebehu činnosti systému	52

5.5.1	Prvé spustenie . . . . .	52
5.5.2	Úsporný režim . . . . .	53
5.5.3	Spotreba energie . . . . .	53
	<b>Záver</b>	<b>55</b>
	<b>Literatúra</b>	<b>57</b>
	<b>Zoznam symbolov, veličín a skratiek</b>	<b>61</b>
	<b>Zoznam príloh</b>	<b>63</b>
	<b>A Obsah prílohy</b>	<b>65</b>

# Zoznam obrázkov

1.1	Predpoveď Cisco o využití mobilnej siete do roku 2021 . . . . .	21
2.1	Diagram M2M komunikácie . . . . .	23
4.1	Carberry . . . . .	29
4.2	AutoPi Dongle generácia 1 . . . . .	30
5.1	Diagram očakávaného priebehu finálnej komunikácie . . . . .	31
5.2	Raspberry Pi Zero . . . . .	33
5.3	a) OBD konektor (samica) b) OBD konektor (samec) . . . . .	34
5.4	Program balenaEtcher . . . . .	36
5.5	Obsah suboru wpasuplicant.conf . . . . .	37
5.6	Poloha CSI rozhrania . . . . .	38
5.7	Schéma zapojenia PIR senzoru HC - SR501 . . . . .	39
5.8	Mobilná aplikácia CarPi. . . . .	49
5.9	Protokoly podporované zariadením AutoPi Dongle . . . . .	50
5.10	Záznam kódov pre odomknutie zámkov vozidla . . . . .	51





# Zoznam tabuliek

5.1	Najdôležitejšie periférie zariadenia AutoPi Dongle . . . . .	32
5.2	Technické parametre Raspberry Pi Zero . . . . .	33
5.3	Spotreba energie v závislosti na aktívnom režime . . . . .	54



# Zoznam výpisov



# Úvod

V súčasnej dobe sa automobilový priemysel mení ako nikdy predtým a to ako aj vďaka nástupu elektromobilov tak aj možnosťou implementácie rôznych elektronických rozšírení umožňujúcich automatizovať a pridávať funkcionality jednotlivých komponentov obsiahnutých v automobiloch.

Diplomová práca sa bude venovať vytvoreniu systému pre zobrazovanie stavových informácií motorového vozidla pomocou mobilnej aplikácie ako aj popisu základných termínov týkajúcich sa danej problematiky a popisu použitých technológií a protokolov.

Súčasťou práce je návrh pripojenia LCD displeja, kamery a pasívneho infračerveného senzora pre zabezpečenie vozidla proti krádeži.

Jedným z vedľajších cieľov práce je rozšíriť vytvorený auto-počítač o možnosť ovládať nekritické systémy vozidla, ako napríklad zámky dverí, klimatizáciu, a pod.

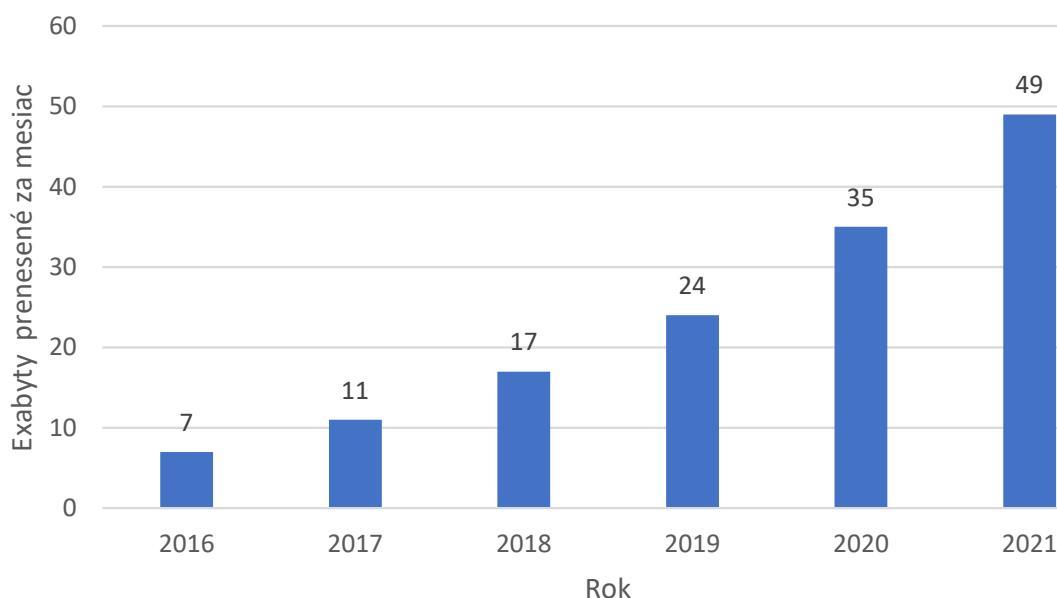
Na dosiahnutie vymedzených cieľov bude použitých niekoľko technológií a komunikačných protokolov, predovšetkým komunikačnú zbernicu OBD-II, ktorá je v automobilovom priemysle využívaná na diagnostiku vozidiel a vyčítanie chybových hlások motora.

Ako hardvérové riešenie bolo vybrané zariadenie AutoPi Dongle, ktoré je nadväzujúcou nad mikrokontrolérom Raspberry Pi Zero W.



# 1 IoT

Pojem Internet vecí (IoT - angl. Internet of Things) hrá veľkú rolu v rôznych odvetviach priemyslu, ako aj v ostatných ekonomických sektoroch. Daný koncept má obrovský vplyv na našu prácu, ale aj na náš každodenný život. IoT zariadenia sú stále viac rozširované a používané ako v domácom, tak aj v pracovnom prostredí pri používaní rôznych senzorov a kamier. V roku 2016 bolo pridaných takmer pol miliardy (429 miliónov) mobilných zariadení, čím sa počet zariadení na svete vyšplhal na osem miliárd. Spolu s počtom mobilných zariadení stúpa aj počet prenesených dát mobilnou sieťou a tým aj jej zaťaženie. V roku 2016 vzrástol počet prenesených dát z 4,4 exabytov mesačne, nameraných na konci roku 2015, na 7,2 exabytov mesačne, čo je v percentách nárast o 63%. Podľa odhadu spoločnosti Cisco bude nárast počtu dát prenesených sieťou za 1 mesiac počas štyroch rokov o 38 exabytov (viz. Obr. 1.1). Od roku 2016 by to bol nárast o 700% [1].



Obr. 1.1: Predpoveď Cisco o využití mobilnej siete do roku 2021<sup>1</sup>

IoT je koncept hovoriaci o tom, že akékoľvek zariadenie, ktoré môže byť pripojené k Internetu, je súčasťou IoT. IoT zariadenia sú pomocou Internetu alebo inej služby ovládané a využívané pre získavanie a spracovávanie informácií, spínanie zariadení alebo aj komunikáciu s človekom. Podľa spôsobu komunikácie účastníkov konverzácie sa delí IoT na komunikáciu dvoch zariadení (D2D – angl. Device to Device Communication) a komunikáciu dvoch strojov (M2M – angl. Machine to machine

<sup>1</sup><https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>, 2017

communication) – M2M komunikácii je venovaná kapitola 2. Koncept IoT zahŕňa inteligentné kávovary, mobilné telefóny, reproduktory, žiarovky, ale aj prúdové motory, drony, vŕtačky či digitálne vodomery, elektromery a pod. [2].

## 1.1 História

História IoT sa započala vznikom komunikácie pomocou elektronických zariadení. Prvé zariadenie pre spomenutý druh komunikácie bolo vytvorené v roku 1832, kedy Baron Shilling v Rusku skonštruoval elektromagnetický telegraf. Avšak prvé zariadenie, ktoré bolo zaradené do skupiny IoT, bolo vytvorené v roku 1990. Bol to topinkovač, ktorý mohol byť zapnutý a vypnutý pomocou Internetu [3].

Hoci zariadenia, ktoré sú radené do skupiny IoT, boli vytvárané už skôr, pojem „Internet of Things“ bol prvýkrát použitý Kevinom Ashtonom, riaditeľom Auto-ID Center, ktorý je jeden z popredných svetových priekopníkov IoT. Uvedený technologický priekopník ho použil vo svojej prezentácii pre firmu Procter & Gamble (tiež známa pod svojou skratkou P&G) v roku 1999 [4].

V roku 2005 nastal veľký rozmach IoT, vplyvom Medzinárodnej Telekomunikačnej Únie (ITU - angl. International Telecommunication Union), uverejnením správy, v ktorej sa uvádza, že kedykoľvek a z akéhokoľvek miesta na svete sa bude možné pripojiť k akémukoľvek zariadeniu v inej časti sveta. Uvedené pripojenia vytvoria novú dynamickú sieť sietí - Internet Vecí [5].

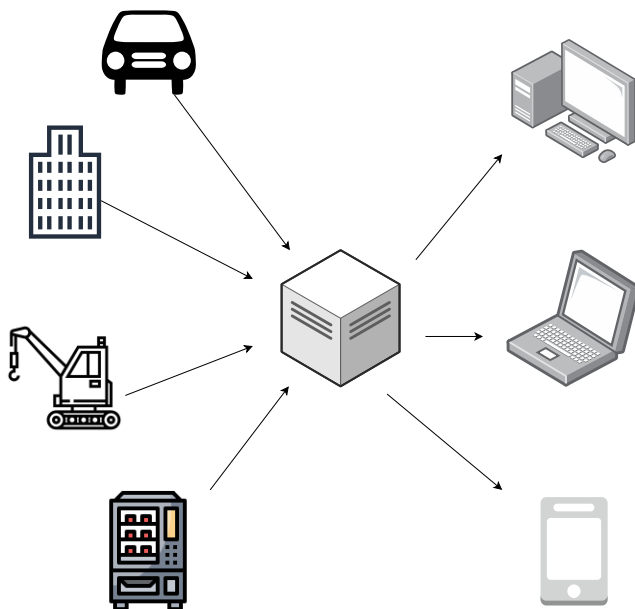
V roku 2006 boli na trh uvedení Nabaztag, mechanické zariadenia pripojené k Internetu pomocou technológie Wi-Fi, ktoré dokázali so svojim užívateľom komunikovať o nových pohyboch na trhu akcií, titulkoch správ, budíkoch, a iné [6].



## 2 M2M komunikácia

V minulosti boli správy a informácie podávané ústne alebo písomne v podobe listov, oznámení, a pod. Daný spôsob komunikácie je nazývaný komunikáciou človeka s človekom (H2H – angl. Human to Human communication). Ľudia sa však vždy snažia nájsť si jednoduchšiu cestu, z toho dôvodu boli vytvárané zariadenia, ktoré dokážu zaslať požiadavky pre získanie údajov iným zariadeniam, ktorými sú požadované informácie zaobstarávané (senzory, kamery, apod.) a zasielané späť pôvodnému zariadeniu pre následné spracovanie.

M2M je technológia, pomocou ktorej sú získavané informácie využitím vzájomne prepojených zariadení a sú pomocou nich vykonávané akcie bez fyzického zásahu ľudskej osoby. Často je používaná pri vzdialenom monitorovaní, využívanom najmä vo výrobných továrňach (sledovanie teploty materiálu, tlaku v nádobách, a iné), ale aj v domácnostiach, či na verejných priestranstvách (kamerové systémy, informačný dopravný systém, a iné). M2M komunikácia je základným komponentom IoT technológie. Často je M2M komunikácia zamieňaná s komunikáciou dvoch zariadení, ktorá ale slúži na zjednodušenie prenosu dát v celulárnej internetovej sieti, pričom dáta zasielané medzi zariadeniami vo vzájomnom dosahu, nemusia prechádzať cez základnú stanicu, následkom čoho je zníženie zataženia siete [7]. Všeobecne je D2D priama komunikácia dvoch zariadení vo vzájomnom dosahu, pričom M2M je komunikácia medzi strojmi, neobmedzená vzdialenosťou.



Obr. 2.1: Diagram M2M komunikácie

Každé M2M zariadenie pozostáva z prijímača informácií, zariadenia pre vytvorenie kanálu pre prenos dát, úložného priestoru, procesnej jednotky (CPU – angl.

Central Processing Unit), softvérového systému (firmvér alebo operačný systém), jedinečného čísla pre identifikáciu jednotlivého zariadenia a autonómneho počítačového softwaru naprogramovaného na interpretáciu dát z pripojeného zariadenia a vykonávanie rozhodnutí na základe získaných dát [8].

## 2.1 História

Začiatok vývoja M2M technológie siaha až do roku 1845, kedy ruská armáda prvýkrát použila systém pre posielanie dát. Bol to úplne najjednoduchší spôsob posielania dát využitím káblovej siete, kedy na oboch koncoch komunikácie bolo pripojené elektronické zariadenie. Nasledovalo vynájdenie dvojcestných rádiových sietí, telefónnych liniek, vzdušných balónov pre leteckú telemetriu, a pod. [9].

V roku 1973 boli vydané prvé počítače spoločnosti IBM (IBM 3650 a 3660), čo viedlo k novej vlně inovácií, ako napríklad merače spotreby elektrickej energie, procesorové čipy pre jeden počítač, a iné.

Ďalšia generácia M2M zariadení prišla s predstavením Globálneho systému pre mobilnú komunikáciu (GSM – angl. Global System for Mobile Communications) pripojenia pre mobilné telefóny, vďaka ktorej si bezdrôtové pripojenie našlo mnoho využití, od automatov až po telemedicínu – poskytovanie zdravotnej starostlivosti a zdravotníckych služieb na veľké vzdialenosti. Vývoj M2M riešení na základe analógového pripojenia dosiahol svojho vrcholu v roku 2010, kedy bola väčšina<sup>1</sup> technológií pripravená na digitálnu revolúciu, ktorá so sebou priniesla rozvoj IoT [10].

## 2.2 Bezpečnosť v M2M

Spoločnosť Gartner predpokladá počet pripojených zariadení v roku 2020 na 20 miliárd [11]. Z daného vyplýva, že výpočtová sila, veľkosť sietí a množstvo dát prenášaných sieťou rýchlo rastie a otvára nové možnosti pre množstvo bezpečnostných hrozieb.

Existuje niekoľko spôsobov ako sa brániť voči útokom na inteligentné zariadenie:

### 1. Zabezpečenie samotného zariadenia

- (a) *Účastnícka identifikačná karta* (SIM – angl. Subscriber Identity Module) optimalizované pre M2M - prostredie odolné voči sfaľšovaniu, ktoré predstavuje silný spôsob autentifikácie pre globálne siete. Šifrujú a autentifikujú dáta pre bezpečnú identifikáciu zariadenia pripojeného do globálnej

---

<sup>1</sup>Výnimkou sú napríklad mobilné siete, ktoré boli z dôvodu preťaženia komunikačných kanálov nestabilné a okrem Long-Term Evolution (LTE) pripojenia problémy s IoT komunikáciou pomocou mobilných sietí stále pretrvávajú.

mobilnej siete. Telo karty je obyčajne viacvrstvé (môže mať až 9 vrstiev), obsahujúc niekoľko laserových obrazov, mikročiarovú potlač, giloše, neviditeľné fluorescentné farbivo, a pod, čo slúži ako ochrana proti sfalšovaniu. Dané SIM karty je možné aktualizovať vzdialene, bez nutnosti ich výmeny, čím je umožnené aj zníženie nákladov na údržbu zariadenia [12].

- (b) *Hardvérové bezpečnostné moduly* - majú rozsiahle využitie pri citlivých IoT zariadeniach. Centrálne ukladajú a generujú autorizačné kľúče v zabezpečených úložiskách, čím je zabezpečená väčšia bezpečnosť pri povolení prístupu k dátam. Dané moduly zabezpečujú transakcie, identity a aplikácie pomocou zabezpečovania kryptografických kľúčov a sprostredkovaním šifrovania, dešifrovania, autentifikácie a služby digitálnych podpisov [13].
  - (c) *Dôveryhodný manažér kľúčov* - pomerne nový spôsob autentifikácie IoT zariadení a zabezpečenia prenosu dát pre mobilné aj pevné siete, pričom je zabráňované neautorizovaným zariadeniam prístupit' do siete. Zabezpečuje silnú digitálnu ochranu pomocou jednoduchého mechanizmu poskytovania bezpečnostných kľúčov a vzdialeného pridelenia poverení.
  - (d) *IP zabezpečenie* - zabezpečenie vstavaných softvérových aplikácií a dátových súborov pred reverzným inžinierstvom alebo sfalšovaním, vytvorením bezpečnostného obalu pre zabezpečenie aplikačného kódu počas cesty ku koncovému užívateľovi. Pre spomenuté sa využíva šifrovanie, zamiešanie kódu a anti-aliasing na systémovej úrovni [14].
2. **Cloudové zabezpečenie** - pre možnosť výskytu napadnutia cloudového prostredia je potrebné upriamiť pozornosť aj na zabezpečenie uvedenej časti siete:
- (a) *Ovládacie prvky cloudovej bezpečnosti* - cloudové zabezpečenie je efektívne iba v prípade, že sú implementované správne bezpečnostné opatrenia. Ovládacie prvky sú implementované do cloudových serverov pre minimalizáciu útokov a tiež efektu prípadných útokov [15].
    - i. Odstrašujúci prvok - informuje prípadných útočníkov o následkoch ich činov v prípade, že budú vo svojom počínaní ďalej pokračovať.
    - ii. Preventívne opatrenia - ochrana cloudu pred útokmi na základe zníženia počtu, prípadne eliminácie slabých miest v zabezpečení cloudového serveru.
    - iii. Detekčné prvky - sú navrhnuté pre detekciu a vhodnú reakciu na prípadný útok. Detekčný prvok v prípade útoku informuje preventívnu alebo korekčnú vrstvu, ktoré ďalej reagujú na daný útok.
    - iv. Korekčné prvky - znižujú dopad útoku, najčastejšie minimalizovaním škôd napáchaných útočníkom. Jeden zo spôsobov minimalizácie škôd

je obnovenie systému zo systémových záloh.

- (b) *Bezpečnosť a utajenie*
- i. Manažér identít - poskytuje zachovanie súkromia aj naprieč viacerými podnikmi využitím biometrickej identifikácie. Prepája dôverné informácie užívateľov s ich biometrickými údajmi a ukladá ich v zašifrovanej podobe [16].
  - ii. Fyzické zabezpečenie - zabezpečenie voči nepovolenému vstupu, krádeži, požiaru alebo povodniam.
  - iii. Utajenie - zabezpečuje, že všetky citlivé dáta sú maskované, šifrované a sú prístupné iba oprávneným osobám.
- (c) *Zabezpečenie dát* - existujú útoky, ktoré sa nezameriavajú na odpočúvanie siete alebo nepovolené vniknutie do systému, ale na útoky na postranný kanál alebo využívajú napadnuteľnosť virtualizácie [17]. Následovné prvky môžu spomenuté druhy útokov obmedziť:
- i. Diskrétnosť - vlastnosť, pri ktorej obsah dát nie je prístupný nepovoleným užívateľom. Nahrané dáta sú uložené v cloude, dokonca bez priamej možnosti kontrolovania majiteľom. Iba autorizovaný užívateľia majú prístup k citlivým dátam.
  - ii. Kontrola prístupu - vlastník údajov môže vykonať obmedzenie prístupu ku svojim externým dátam na cloude.
  - iii. Integrita - vyžaduje udržiavanie a zaistenie presnosti a celistvosti údajov. Údaje by nemali byť ilegálne manipulované, upravované, mazané alebo falšované. V prípade, že akákoľvek nežiadúca operácia naruší alebo vymaže dáta, ich vlastník by mal byť schopný detekovať poškodenie alebo stratu.

## 3 Problematika v teórii

Modernizácia automobilové priemyslu prináša so sebou v praxi množstvo nových možností automatizácie činností vykonávaných predovšetkým vodičom motorového vozidla. Všeobecne je snaha o minimalizáciu rušivých vplyvov na vodiča počas jazdy.

Za týmto účelom sa využívajú rôzne nové technológie, ktoré je však nutné optimalizovať, nakoľko pri príliš veľkom množstve nových prvkov môže vzniknúť opačný efekt ako ten požadovaný.

V súčasnej dobe sa automobilový priemysel mení ako nikdy predtým, a to ako aj vďaka nástupu elektromobilov, tak aj možnosťou implementácie rôznych elektronických rozšírení umožňujúcich automatizovať a pridávať funkcionality jednotlivým komponentom obsiahnutých v automobiloch.

Dnes sú takmer všetky automobily ovládané pomocou počítačov vo forme riadiacich jednotiek (RJ). Jedným z príkladov takéhoto počítača je riadiaca jednotka motora, ktorá v reálnom čase prijíma a spracováva dáta z rôznych senzorov nachádzajúcich sa v motorovom priestore a tým zabezpečuje optimálny výkon a správnu funkciu motora. [20]

S pomocou mikrokontrolérov ako napríklad Raspberry Pi, Arduino, a pod., je dnes možné zautomatizovať činnosti, ktoré človek inak musí vykonávať manuálne a tým si uľahčiť vykonávanie niektorých akcií, spríjemniť cestovanie alebo jednoducho vytvoriť niečo nové, zaujímavé, čím zaujme okolie.

### 3.1 Výhody automatizácie

Pomocou rôznych technológií je dnes človek schopný upraviť si všetky systémy svojho vozidla podľa vlastných predstáv. Či už ide o vytvorenie nového palubného počítača, premenenie svojho starého tabletu na multimediálny systém alebo vzdialené ovládanie auta.

Existuje mnoho projektov automobilových a technických nadšencov, ktorí si vytvorili systém interaktívneho hlasového asistenta umožňujúceho hlasové ovládanie funkcií vozidla, čítanie informácií z palubného počítača, vyhľadávanie informácií na platformách Wikipedia a Google, a pod. Výhodou automatizovania niektorých aktivít je udržanie pozornosti vodiča na ceste. Na ovládanie rôznych aktivít stačia hlasové povely, pričom ruky vodiča zostávajú po celý čas na volante.

V prípade nehlasových asistentov a multimediálnych systémov je výhodou možnosť personalizácie vzhľadu zobrazenia a zjednodušenie prístupu k rôznym informáciám a funkciám vozidla.

## 3.2 Nevýhody automatizácie

Jednou z nevýhod automatizácie je rozrušovanie vodiča pri riadení motorového vozidla v prípade, kedy nie je systém určený na používanie v aute a vodič tak musí venovať pozornosť danému systému počas riadenia.

Veľkou nevýhodou je taktiež bezpečnostné riziko pripájania externých zariadení ku systému vozidla. V prípade neprofesionálneho zaobchádzania s prídavným zariadením môže vzniknúť nespočetné množstvo problémov, ktoré môžu ovplyvniť bezpečnosť vozidla na ceste a tiež správnu funkciu motora.

Od vzniku povinnosti integrácie OBD rozhrania v motorových vozidlách nastalo množstvo nebezpečných situácií zapríčinených práve takýmto nezodpovedným zaobchádzaním s podomácky vyrobeným prídavným systémom. Príkladom sú havárie zapríčinené nesprávnou funkčnosťou riadiacej jednotky motora, samovoľné vznietenie elektroinštalácie automobilu zapríčinené skratom, krádeže, a iné.

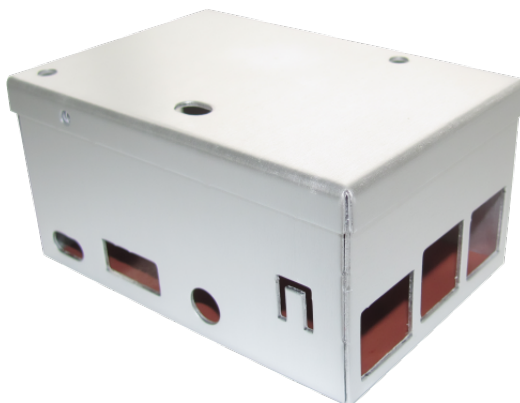
## 4 Technológia inteligentného vozidla

Na trhu je niekoľko riešení poskytujúcich možnosť ovládania a diagnostiky motorového vozidla. Každé z týchto výrobkov sa však zameriava predovšetkým iba na jedno špecifické využitie. Niektoré z týchto výrobkov a možných riešení budú v nasledujúcom texte predstavené.

Naším cieľom je vytvoriť zariadenie, ktoré bude komunikovať so systémom vozidla a podávať stavové informácie o motore a danom vozidle, ako napríklad rýchlosť, otáčky motora, teplota motora a iné.

### 4.1 Carberry

Carberry je zariadenie vytvorené taliaskou spoločnosťou Paser, ktorá sa zaoberá personalizovaním motorových vozidiel podľa požiadaviek zákazníka a vyrába množstvo zariadení slúžiacich práve na daný účel. Ako základ pre vytvorenie Carberry sú použité rôzne modely Raspberry Pi.



Obr. 4.1: Carberry<sup>1</sup>

Carberry sa zameriava predovšetkým na získavanie stavových informácií o vozidle ako napríklad rýchlosť vozidla, otáčky motora, spotreba paliva, a pod. Môže tiež slúžiť ako multimediálny prehrávač alebo ako palubný počítač. Toto zariadenie však nie je určené na ovládanie jednotlivých systémov vozidla.

### 4.2 AutoPi Dongle

AutoPi Dongle je zariadenie vytvorené spoločnosťou dánskych inžinierov a automobilových nadšencov, ktorí sa zaujímajú o riešenie inteligentných áut, či už pre jednotlivcov alebo veľké spoločnosti, ktoré chcú mať prehľad a kontrolu nad svojimi

vozidlami. Spoločnosť AutoPi.io vytvorila nadstavbu pre zariadenie Raspberry Pi Zero obsahujúcu niekoľko modulov a čipov potrebných pre všestrannú personalizáciu vozidla.



Obr. 4.2: AutoPi Dongle<sup>2</sup>

AutoPi Dongle umožňuje ako aj čítanie stavových informácií vozidla, tak aj ovládanie jednotlivých systémov vozidla, napríklad svetiel, zámkov dverí, okien a mnoho ďalších komponentov. Umožňuje tiež pripojenie k internetu pomocou 4G/LTE modulu, určenie aktuálnej polohy vozidla využitím GPS modulu a tiež aktuálnej rýchlosti pohybu určenej akcelerometrom.

### 4.3 Xmarton

Xmarton je česko-slovenská start-upová spoločnosť so sídlom v Prahe, ktorá vydala svoj produkt v roku 2019. Spomenutý produkt umožňuje ovládanie auta, informovanie o polohe vozidla, zabezpečenie vozidla a mnoho ďalších funkcionalít, ktoré zatiaľ ale podporujú iba v Európe.

Ich mobilná aplikácia, ktorou je zariadenie ovládané, je prehľadná a ľahko ovládateľná. Nakoľko svoj produkt vydali počas procesu tvorby tejto diplomovej práce, nie je dostupných veľa informácií o použitých technológiách.

### 4.4 Vlastné riešenie

Na trhu sa tiež nachádza množstvo zariadení umožňujúcich komunikáciu s vozidlom pomocou rozhrania OBD-II, určených pre technologických nadšencov, ktorí si radi vytvárajú vlastné "DIY" projekty.

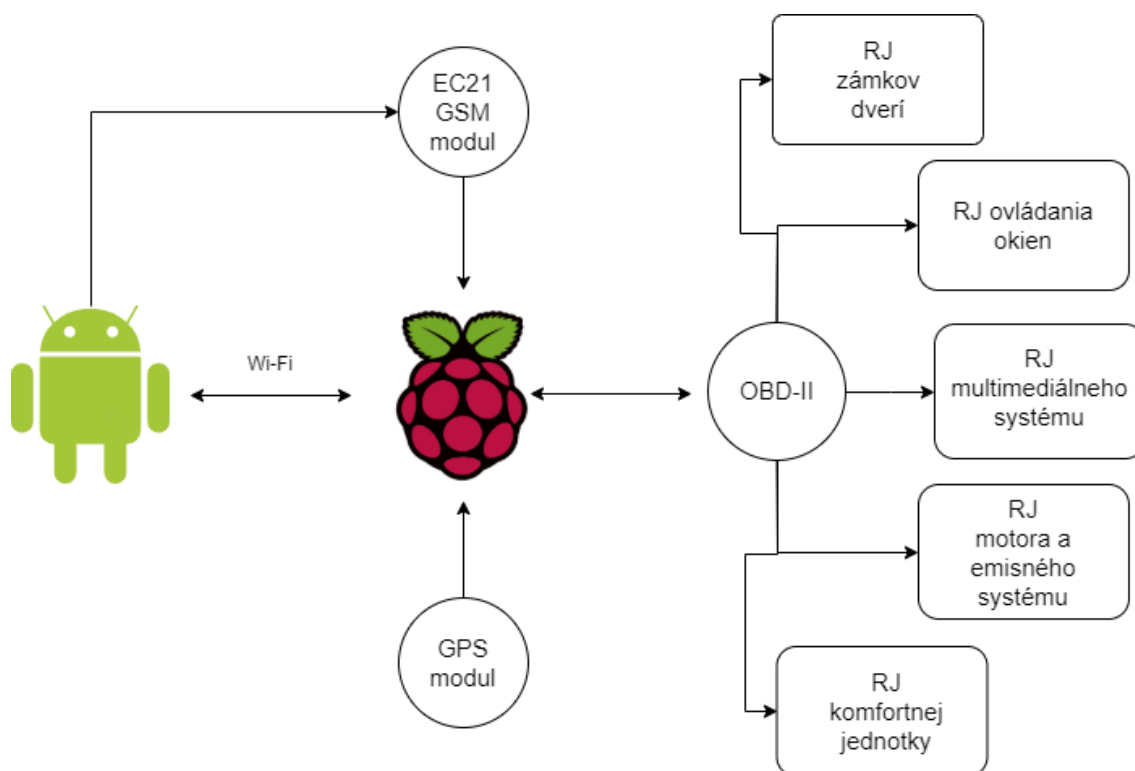
Dané zariadenia sú často príliš pomalé na komunikáciu s vozidlom pomocou portu OBD-II, ale existujú aj funkčné a jednoduché riešenia.



## 5 Návrh vlastného riešenia

Práca sa zaoberá problematikou vytvorenia auto-počítača komunikujúceho s riadiacou jednotkou motora a mobilnou aplikáciou.

Pre dosiahnutie vymedzeného cieľa bolo nutné oboznámenie sa so systémom odovzdávania si informácií medzi jednotlivými kontrolérmi elektronického vybavenia auta a s tým spojených protokolov. Taktiež bolo nutné nájsť technológiu, ktorá umožňuje tieto správy zaznamenávať.



Obr. 5.1: Diagram očakávaného priebehu finálnej komunikácie

Na obrázku 5.1 je možné vidieť diagram očakávaného priebehu komunikácie po dokončení predloženého projektu. Mobilný telefón s operačným systémom Android bude posilať požiadavky a prijímať údaje o aute prostredníctvom Raspberry Pi komunikujúceho s jednotlivými systémami auta pomocou zbernice OBD-II.

Snahou diplomovej práce je zaoberať sa vývojom mobilnej aplikácie na platforme Android a serverovej aplikácie na platforme Raspberry Pi, komunikujúcimi s jednotlivými systémami vozidla, ich ovládaním a spracovávaním údajov získaných z jednotlivých ECU (elektronická ovládacia jednotka - angl. Electronic Control Unit).

## 5.1 AutoPi.io

AutoPi.io je spoločnosť, ktorá spája zariadenie AutoPi Dongle, ktorá je nadstavbou nad zariadením Raspberry Pi Zero W, so službou AutoPi Cloud do IoT platformy umožňujúcej užívateľovi rozšíriť možnosti ovládania a monitorovania svojho vozidla.

Komunikácia užívateľa so zariadením prebieha prostredníctvom služby AutoPi Cloud, ktorá preposiela požiadavky užívateľa do zariadenia AutoPi Dongle.

AutoPi Cloud ponúka široké už prednastavené možnosti monitorovania stavu vozidla prístupné aj menej technicky zdatným užívateľom.

Dôvodom nevyužitia služby AutoPi Cloud je nízka prispôsobiteľnosť aplikácie pre potreby predloženej práce. Komunikácia s užívateľom prostredníctvom cloudovej služby taktiež spôsobuje vyššie dátové prenosy, ktoré sú nežiadúce.

Zariadenie Autopi Dongle obsahuje množstvo periférií, z ktorých tie najdôležitejšie sú obsiahnuté v Tab. 5.1

Použitie	Zariadenie
Základná doska	Raspberry Pi Zero W
OBD	ELM 327 a STN1100
4G/LTE	Quectel EC21
DAC	MCP4725

Tab. 5.1: Najdôležitejšie periférie zariadenia AutoPi Dongle

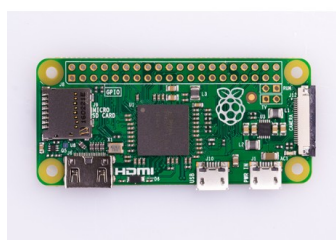
### 5.1.1 Raspberry Pi Zero W

Raspberry Pi (Obr. 5.2) je nízko-rozpočtový malý počítač, ktorý je možné pripojiť k monitoru alebo televízii. K jeho ovládaniu sa používa štandardná klávesnica a myš. Raspberry Pi pôvodne vzniklo ako zariadenie pre výuku programovania a rôzne vedecké práce, avšak stalo sa populárnym medzi vynálezcami a elektronickými nadšencami, predovšetkým kvôli svojej veľkosti a výkonu. Dané zariadenie je vhodné pre projekty, ktoré vyžadujú viac výkonu a jednoduchšie zaobchádzanie s hardvérom v porovnaní s tým, čo ponúka mikrokontrolér ako napríklad Arduino [21]. Technické parametre Raspberry Pi Zero je možné nájsť v tabuľke 5.2.

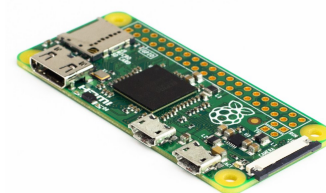
Raspberry Pi je menej výkonné ako moderné notebooky alebo stolné počítače, napriek tomu je to plne funkčný počítač s operačným systémom Linux. Pôvodne

<sup>2</sup><https://kitskart.com/product/raspberry-pi3-model-b-made-in-uk/>

<sup>2</sup><https://www.linux.com/news/survey/2017/6/hacker-board-survey-results-more-raspberry-pi-please>, 2017



(a) Pohľad z hora<sup>1</sup>



(b) Pohľad z boku<sup>2</sup>

Obr. 5.2: Raspberry Pi Zero

<b>SoC</b>	Broadcom BCM2835
<b>CPU</b>	ARM1176JZF-S 1 GHz
<b>GPU</b>	Broadcom VideoCore IV @ 250 MHz
<b>RAM</b>	512 MiB
<b>Sieťové rozhrania</b>	2,4 GHz IEEE 802.11n wireless
<b>Bluetooth</b>	Bluetooth 4.1 Classic, BLE
<b>Úložný priestor</b>	microSD
<b>GPIO</b>	40 pinov
<b>Porty</b>	mini-HDMI, 2× micro-USB, Ethernet, Kamerové sériové rozhranie (CSI) mini

Tab. 5.2: Technické parametre Raspberry Pi Zero

bolo Raspberry Pi vytvorené pre Linux, v súčasnosti sa už mnohé distribúcie Linuxu vyvíjajú aj pre potreby Raspberry Pi. Najpopulárnejšie distribúcie Linuxu, používané v Raspberry Pi sú Raspbian, založený na operačnom systéme Linux Debian, a Pindora, vyvinutá z operačného systému Linux Fedora.

## GPIO

Všeobecný vstup/výstup (GPIO –angl. General-purpose input/output) je pin na integrovanom obvode alebo počítačovej doske. Správanie sa týchto pinov (výber vstupnej alebo výstupnej funkcie) je definované užívateľom.

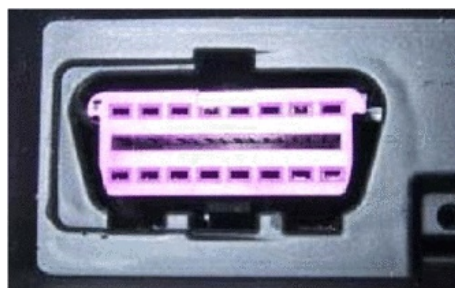
Využitie GPIO pinov je v oblasti programovateľných logických zariadení, zabudovaného hardvéru alebo iných aplikácií, ako napríklad čítanie dát zo senzorov alebo zápis údajov na LC display [22].

## 5.2 OBD

OBD (angl. On-board diagnostics) je termín, ktorý popisuje diagnostické a oznamovacie možnosti vozidla. Daný systém umožňuje majiteľovi alebo servisnému technikovi prístup ku stavu rôznych systémov vozidla. V počiatočných štádiách systému OBD sa množstvo informácií dostupných pomocou daného systému líšilo od informácií podávaných systémom OBD-II. Dnes je možné získať pomocou štandardizovanej digitálnej komunikácie dáta o stave vozidla v reálnom čase, pričom v počiatočných štádiách boli využité iba kontrolky indikujúce široký rozsah možných chýb.

Od roku 2001 je v EÚ systém OBD povinnou súčasťou všetkých benzínových vozidiel triedy M1 a od roku 2003 je tak aj pri dieselových motoroch.

Systém OBD-II pracuje so štandardizovanými diagnostickými chybovými kódmi (DTC, angl. Diagnostic trouble codes), ktoré umožňujú rýchlu a jednoduchú identifikáciu a opravu chýb na vozidle.



(a) Female connector



(b) Male connector.

Obr. 5.3: a) OBD konektor (samica) b) OBD konektor (samec)

## 5.3 Použité technológie

Pre vytvorenie inteligentného auto-počítača existuje niekoľko možných riešení, z ktorých niektoré budú popísané v ďalšom pokračovaní predloženej práce.

Pripojenie k jednotke OBD vozidla je možné pomocou OBD konektora využívajúceho technológiu Bluetooth. Nakoľko je spomenutá technológia málo zabezpečená, tejto možnosti sa vyhneme.

Navrhujeme ju preto nahradiť OBD konektorom pripojeným priamo k zariadeniu Raspberry Pi Zero pomocou sériového portu a GPIO pinov.

Použité prístroje:

- AutoPi Dongle
- SD karta s čítačkou
- počítač
- zdroj 12V2, 5A
- 2x krokosvorky
- prepojovacie káble
- Raspberry Pi Camera Noir 2. generácie
- PIR senzor HC-SR501

### 5.3.1 Protokol MQTT

MQTT (angl. MQ Telemetry Transport) je jednoduchý sieťový protokol typu "publikuj-odoberaj", slúžiaci na prenášanie informácií medzi zariadeniami. Má veľké využitie v IoT, nakoľko prenášané dáta majú minimálnu veľkosť.

Zväčša používa protokoly vyššej vrstvy TCP/IP, avšak je možné ho využívať na akomkoľvek obojsmernom sieťovom protokole.

MQTT protokol definuje 2 základné entity:

- MQTT Broker, ktorý slúži ako server pre posielanie a prijímanie informácií. Všetky správy od klientov putujú najskôr k MQTT Brokerovi a následne sú presmerované ku koncovému zariadeniu.
- MQTT Client, ktorý správy generuje a spracúva, môže byť akékoľvek zariadenie so schopnosťou sieťového prenosu informácií.

Informácie sú zoradené v hierarchii tém, ktoré koncové zariadenia odoberajú, a do ktorých publikujú.

## 5.4 Postup práce

Uskutočneniu vývoja predloženého systému predchádzalo zbieranie informácií o použitých technológiách, študovanie potrebných materiálov a zaobstarávanie si použitých komponentov. Po ich získaní bolo možné začať praktickú realizáciu navrhnutého systému.

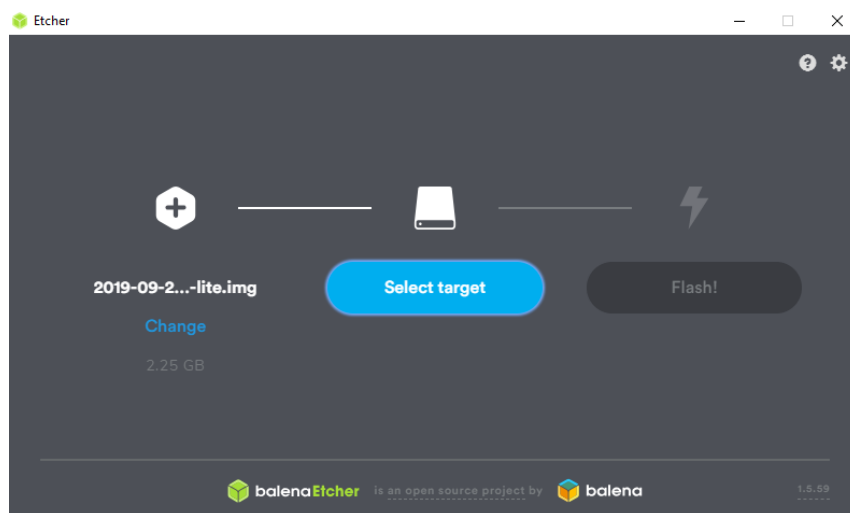
### 5.4.1 Inštalácia operačného systému

V tento práci použijeme operačný systém Raspbian Stretch, ktorý je dostupný na oficiálnej stránke Raspberry Pi.

Z dôvodu šetrenia energie a úložného priestoru stiahneme inštalačný súbor operačného systému bez grafického prostredia. Spolu s ním si stiahneme aj softvér balenaEtcher, ktorý nám umožní vytvoriť z SD karty inštalačný disk operačného systému.

Do počítača vložíme čítačku SD kariet s SD kartou vo vnútri a počkáme, kým sa SD karta načíta. O formátovanie karty sa postará balenaEtcher.

Spustíme program balenaEtcher a vyberieme inštalačný súbor operačného systému, zariadenie, na ktoré chceme daný súbor nahrať (v našom prípade SD karta) a klikneme na tlačidlo "Flash!".



Obr. 5.4: Program balenaEtcher

Pred vložením SD karty do Raspberry Pi Zero ešte vytvoríme na SD karte 2 súbory umožňujúce nám konfiguráciu zariadenia bez nutnosti pripojenia zariadenia k monitoru pomocou káblu.

```
GNU nano 3.2 temp Modified
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=CZ

network={
  ssid="<Nazov siete WiFi>"
  psk="<Heslo siete WiFi>"
}

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^I To Spell  ^_ Go To Line
```

Obr. 5.5: Obsah suboru wpasupplicant.conf

Otvoríme zložku s obsahom SD karty. Najskôr vytvoríme súbor s názvom "wpa-supplicant.conf", ktorý umožní automatické pripojenie Raspberry Pi ku sieti Wi-Fi nastavenej v danom súbore. Obsah súboru je možné vidieť na obr. 5.5

Uistíme sa, že nami vytvorený súbor nemá príponu ".txt".

Ďalej vytvoríme prázdny súbor s názvom "ssh" bez prípony. To aktivuje SSH port na Raspberry Pi a umožní nám konfigurovať zariadenia bez fyzického pripojenia.

Vložíme SD kartu do Raspberry Pi a počkáme približne 60 sekúnd, kým sa operačný systém nainštaluje. Následne sa pomocou programu Putty pripojíme pomocou protokolu SSH ku konzole zariadenia.

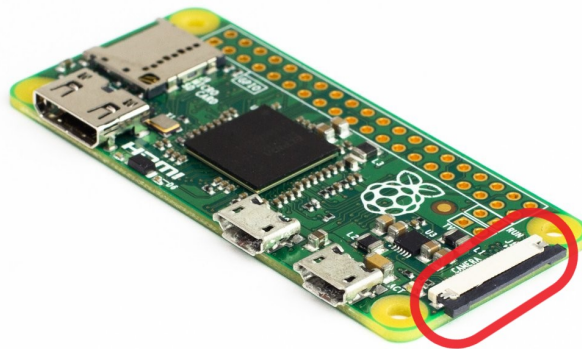
## 5.4.2 Návrh pripojenia LCD displeja a kamery

Zariadenie Raspberry Pi Zero W disponuje iba jedným CSI rozhraním, avšak AutoPi Dongle rozširuje možnosti pripojenia zariadení o dve ďalšie USB rozhrania. Pomocou daných rozhraní je možné pripojiť LCD displej a až 2 USB kamery.

Dôvodom pre zapojenie až dvoch kamier je vytvorenie väčšieho prehľadu o dianí v interiéri aj exteriéri vozidla v prípade narušenia.

Na pripojenie LCD displeja použijeme CSI rozhranie, pričom je potrebné použiť originálny Raspberry Pi displej, nakoľko je ako jediný podporovaný firmvérom mikrokontroléru.

Následovne pripojíme 2 USB kamery využitím oboch USB rozhraní zariadenia AutoPi Dongle. Takto zapojený systém nakonfigurujeme po inštalácii operačného systému.



Obr. 5.6: Poloha CSI rozhrania

V predloženej práci bude vytvorený prototyp kamerového systému s 1 kamerou a pasívnym infračerveným senzorom pohybu (PIR).

### 5.4.3 Pripojenie PIR senzoru

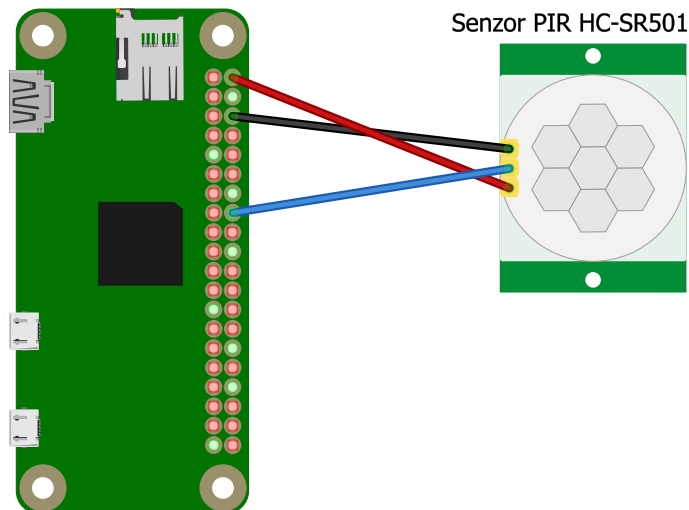
PIR senzor je pasívny infračervený senzor merajúci množstvo infračerveného žiarenia. Pri zmene jeho množstva je senzor zopnutý.

Model nami použitého senzora je HC SR501, ktorého súčasťou sú 3 piny, pričom všetky je potrebné pripojiť na riadiacu dosku. PIR senzor pracuje na napätovej úrovni 3.3V.

Použitý senzor taktiež obsahuje 2 otočné hlavice pre nastavenie citlivosti zopnutia a tiež dĺžku logickej jednotky v prípade zaznamenania pohybu. Minimálna doba zopnutia je 8 sekúnd a minimálna senzitivita je 2.5 metra. Pod tieto úrovne nie je možné PIR senzor nastaviť.

V našom prípade je nastavenie minimálnych hodnôt postačujúce. Otočíme teda obe hlavice na minimum a senzor zapojíme podľa schémy 5.7. Zemiaci pin GND je pripojený na zem RPi (čierny spoj), napájací pin VCC pripojíme na pin 3.3V (červený spoj) a výstup OUT pripojíme na pin GPIO23 na doske Raspberry Pi Zero W.





fritzing

Obr. 5.7: Schéma zapojenia PIR senzoru HC - SR501

#### 5.4.4 Konfigurácia 4G/LTE pripojenia

Zariadenie AutoPi Dongle už obsahuje pripojený 4G/LTE modul, avšak k jeho ovládaniu je potrebné qmi rozhranie, ktoré poskytuje požadované služby.

Najskôr nainštalujeme Raspberry Pi kernelové hlavičky

```
sudo apt install raspberrypi-kernel-headers
```

a skontrolujeme, akú verziu hlavičiek máme nainštalovanú:

```
ls /usr/src/
```

Následne otvoríme internetový prehliadač a prejdeme do githubového repozitára rpi-firmware. V ňom nájdeme commit, ktorý odpovedá verzii našich hlavičiek a prejdeme do daného commitu. Z URL adresy v prehliadači skopírujeme hash a prejdeme späť do konzoly, kde aktualizujeme operačný systém pomocou skopírovaného hashu.

```
sudo rpi-update <hash>
```

Zariadenie reštartujeme. Po reštarte stiahneme inštalačný skript qmi rozhrania.

```
wget https://raw.githubusercontent.com/sixfab/Sixfab_RPi_3G-4G-LTE_Base_Shield/master/tutorials/QMI_tutorial/qmi_install.sh
```

Súboru pridáme exekúčne práva a skript spustíme.

```
sudo chmod +x qmi_install.sh
```

```
sudo ./qmi_install.sh
```

Skript nainštaluje všetky potrebné knižnice a vykoná všetku konfiguráciu za nás. Po dokončení inštalácie stlačíme ENTER a zariadenie sa reštartuje. Po opätovnom spustení sa môžeme pripojiť k internetu s využitím siete LTE spustením príkazu

```
sudo ./files/quectel-CM/quectel-CM -s internet -p 1234
```

, kde argument -s je APN prístupového bodu operátora a -p je PIN karty SIM. APN je možné nájsť na internete alebo je možné ho získať od mobilného operátora.

Ak chceme, aby sa Raspberry Pi pripojilo k internetu vždy po spustení, vložíme predchádzajúci príkaz na koniec súboru rc.local, pričom na koniec príkazu pridáme znak & , aby sa pripojenie odohrávalo na pozadí a nezdržovalo spustenie operačného systému.

### 5.4.5 Konfigurácia bezdrôtového rozhrania

V tejto časti vytvoríme prístupový bod a vlastnú sieť, ku ktorej sa neskôr budeme pripájať.

Na vytvorenie prístupového bodu budeme potrebovať 3 knižnice, ktoré nainštalujeme nasledujúcim príkazom:

```
sudo apt install hostapd dnsmasq
```

Zastavíme služby dnsmasq a hostapd.

```
sudo systemctl stop dnsmasq
```

```
sudo systemctl stop hostapd
```

Aby bolo možné sa k sieti pripojiť, musí v nej existovať akýsi server so statickou IP adresou. V našom prípade bude rozhranie wlan0 slúžiť ako bezdrôtový port na pripojenie k sieti.

Prejdeme do súboru dhcpd.conf

```
sudo nano /etc/dhcpd.conf
```

a upravíme ho, aby vyzeral nasledujúco:

```
interface wlan0
    static ip_address=192.168.4.1/24
    nohook wpa_supplicant
```

Ďalším krokom je konfigurácia DHCP servera. Prejdeme do súboru dnsmasq.conf

```
sudo nano /etc/dnsmasq.conf
```

a nastavíme pre rozhranie wlan0 dhcp pool, z ktorého budú pridelené IP adresy pripojeným zariadeniam:

```
interface=wlan0
dhcp-range=192.168.4.100, 192.168.4.150, 255.255.255.0, 24h
```

Reštartujeme službu dnsmasq, aby sa zmeny uložili:

```
sudo systemctl reload dnsmasq
```

Ďalej nastavíme parametre bezdrôtovej siete v súbore hostapd.conf:

```
sudo nano /etc/hostapd/hostapd.conf
```

Do súboru vložíme nasledujúce parametre:

```
interface=wlan0
driver=nl80211
ssid=<Nazov_siete>
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=<Heslo_siete>
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Spustíme všetky nastavené služby:

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
sudo systemctl restart dnsmasq
```

A nakoniec pridáme routing do súboru sysctl.conf a NAT, aby bolo možné pripojiť sa k internetu využitím karty SIM:

```
sudo nano /etc/sysctl.conf
net.ipv4.ip_forward=1
```

```
sudo iptables -t nat -A POSTROUTING -o wwan0 -j MASQUERADE
sudo sh -c ''iptables-save > /etc/iptables.ipv4.nat''
```

```
sudo nano /etc/rc.local
iptables-restore < /etc/iptables.ipv4.nat
```

```
sudo restart
```

Teraz je možné pripojiť sa k sieti Wi-fi vytvorenej na Raspberry Pi Zero W a následne sa pripojiť pomocou protokolu SSH na IP adresu 192.168.4.1.

### 5.4.6 Nastavenie SMS komunikácie

AutoPi Dongle používa pre pripojenie do mobilnej siete modul EC21 od spoločnosti Quectel. Tento modul využijeme pre vytvorenie možnosti získavať aktuálnu polohu vozidla z akejkoľvek vzdialenosti.

Pre danú úlohu nám bude slúžiť program `gammu-smsd`, umožňujúci automatické spracovávanie prichádzajúcich správ SMS.

Inštalácia programu `gammu-smsd` prebehne po zadaní nasledujúceho príkazu do konzoly:

```
sudo apt install gammu-smsd
```

Po inštalácii službu nakonfigurujeme upravením súboru `/etc/gammu-smsdrc`.

```
[gammu]
port = <port>
connection = at
logformat = textalldate

[smsd]
pin = <SIM PIN>
service = files
logfile = /var/log/gammu-smsd
debuglevel = 4
ReceiveFrequency = 300

inboxpath = /var/spool/gammu/inbox/
outboxpath = /var/spool/gammu/outbox/
sentsmspath = /var/spool/gammu/sent/
errorssmspath = /var/spool/gammu/error/
```

Následne potrebujeme vytvoriť malý skript, ktorý z GPS modulu získa a spracuje dáta o polohe vozidla. Pre vytvorenie skriptu použijeme jazyk Python. Jedinou použitou knižnicou bude knižnica `serial`, umožňujúca komunikáciu so sériovo pripojeným GPS modulom.

GPS modul vypisuje každú sekundu množstvo dát. Z tohto dôvodu je nutné dané správy filtrovať a hľadať iba tie správy, ktoré obsahujú validné dáta o polohe. Taktiež je nutné sa po každom získaní dát uistiť, že GPS modul úspešne získava dáta o polohe a teda, že dáta prijaté z modulu obsahujú informáciu o polohe.

Správa obsahujúca informáciu o polohe nesie označenie "\$GPRMC" a pokiaľ sa v nej validné dáta nachádzajú, ich 3. pole obsahuje znak "A".

```
import serial

gps = serial.Serial("/dev/ttyUSB1", baudrate = 9600)

def get_loc():
    running = True

    while running:
        line = gps.readline()
        data = line.decode().split(",")
        if data[0] == "$GPRMC":
            if data[2] == "A":
                running = False
                return "{:3.4f},{:3.4f}".format((float(data
                    ↪ [3])/100), (float(d$
```

Posledným krokom danej časti je vytvorenie a nastavenie skriptu, ktorý bude spustený službou gammu-smsd v prípade prijatia správy SMS a odpovie na požiadavku o zaslanie polohy vozidla.

```
if [ "$SMS_1_NUMBER" != "<telefónne číslo užívateľa" ] ;
    ↪ then
        exit
fi

case "$SMS_1_TEXT" in
    "Loc")
        python3 /home/pi/geolocsms.py | sudo gammu-smsd-
            ↪ inject TEXT "00$;;
    "")
        echo "$SMS_1_TEXT"
        ;;
esac
```

V uvedenom výpise je vidieť možnosť obmedzenia telefónnych čísel, z ktorých bude povolené získavať polohu vozidla pomocou správ SMS a taktiež zabezpečiť danú funkcionálnosť aj obsahom odoslanej SMS správy.

## 5.4.7 Konfigurácia MQTT servera a klienta

V nasledujúcej časti nainštalujeme a nakonfigurujeme MQTT server, ktorý bude slúžiť ako prostredník medzi mobilnou aplikáciou a Raspberry Pi. Použijeme serverovú aplikáciu Mosquitto, ktorá je dostupná z vlastného repozitára. Z toho dôvodu je nutné daný repozitár importovať do Raspberry Pi.

```
wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
sudo apt-key add mosquitto-repo.gpg.key
cd /etc/apt/sources.list.d/
sudo wget http://repo.mosquitto.org/debian/mosquitto-
stretch.list
sudo apt update
```

Následne je možné Mosquitto nainštalovať pomocou služby apt.

```
sudo apt install mosquitto
```

Budeme tiež potrebovať Mosquitto klienta, ktorý nám umožní odoberať témy ako aj do nich publikovať.

```
sudo apt install mosquitto-clients
```

Z bezpečnostných dôvodov vytvoríme aj užívateľa, ktorého budeme používať na odoberanie a publikovanie správ do tém. Urobíme tak nasledujúcim príkazom:

```
mosquitto_passwd -c passwordfile <uzivatelske_meno>
```

Konzola nás požiada o zadanie hesla. Po zadaní hesla je vytvorený súbor "passwordfile" obsahujúci užívateľské meno a heslo. Mosquitto server však ešte daný súbor nepozná. Upravíme konfiguráciu servera tak, aby pri každom prihlásení požadoval používateľské údaje.

```
sudo nano /etc/mosquitto/mosquitto.conf
allow_anonymous false
password_file /etc/mosquitto/passwordfile
```

## 5.4.8 Komunikácia s OBD II

Ďalším krokom je konfigurácia komunikácie Raspberry Pi s komunikačným portom OBD-II pomocou AutoPi Dongle. Spoločnosť AutoPi.io vytvorila knižnicu<sup>3</sup> vhodnú pre použitie ich zariadenia bez predinštalovaného softvéru AutoPi.

---

<sup>3</sup><https://github.com/autopi-io/py-obd>

Skôr ako si knižnicu stiahneme a použijeme, nainštalujeme si potrebné podporné knižnice. Pre komunikáciu pomocou sériového portu prostredníctvom programovacieho jazyka Python potrebujeme knižnicu python-serial, dostupnú z raspberry repozitára. Navyše budeme potrebovať aj knižnicu python-pint.

Obe knižnice nainštalujeme príkazom:

```
sudo apt install python3-serial python3-pint
```

Pred stiahnutím spomínanej knižnice potrebujeme službu git, ktorá nám umožní naklonovať si repozitár do zariadenia. Daná služba je taktiež dostupná z raspberry repozitára.

```
sudo apt install git
```

Následne prejdeme do githubového repozitára AutoPi.io a otvoríme repozitár py-obd.

```
git clone https://github.com/autopio-io/py-obd.git
```

Teraz je všetko pripravené pre vytvorenie aplikácie na komunikáciu s OBD portom vozidla. Vytvoríme script, ktorý bude komunikáciu ovládať a otestujeme funkcionálnosť.

V zložke py-obd vytvoríme súbor s ľubovoľným názvom a vložíme doň kód na ovládanie komunikácie.

```
import obd

conn=obd.Async(
    portstr='/dev/serial0',
    baudrate=500000
)

def get_rpm(r):
    print(r.value)
    return r.value

conn.watch(obd.commands.RPM, callback=get_rpm)
conn.start()
```

Súbor uložíme. Pripojíme zariadenie AutoPi Dongle k OBD portu vozidla a vozidlo naštartujeme. Následne otestujeme nami vytvorený program príkazom

```
python3 <nazov_suboru>
```

Pri úspešnom priebehu program vypíše aktuálne otáčky motora pri každej zmene počtu otáčok za minútu.

## 5.4.9 Komunikácia s mobilnou aplikáciou

Následujúcim úkonom je upravenie predchádzajúceho programu na komunikáciu s mobilnou aplikáciou pomocou protokolu MQTT. Pre odosielanie správ do tém MQTT servera s využitím programovacieho jazyka Python použijeme knižnicu paho-mqtt. Danú knižnicu nainštalujeme do prostredia Python pomocou služby pip:

```
pip3 install paho-mqtt
```

Otvoríme predchádzajúci súbor s vytvoreným programom pre čítanie otáčok motora a upravíme ho do podoby uvedenej v súbore CarPi.py, obsiahnutom v prílohe.

Aplikácia obsahuje tri paralelne bežiacie vlákna, pričom každé vlákno má rozličnú úlohu.

Vlákno securityThread zabezpečuje nepretržité sledovanie pohybu vo vozidle. V prípade, že PIR senzor zaznamená vo vozidle pohyb dvakrát po dobu deviatich sekúnd pri vypnutom motore, okamžite je aktivovaná funkcia pre vytvorenie fotografie a zaslanie e-mailu s polohou vozidla a vytvorenou fotografiou majiteľovi daného vozidla.

```
class securityThread(threading.Thread):
    def __init__(self, ID, name):
        threading.Thread.__init__(self)
        self.threadID = ID
        self.threadName = name
        self.secure = threading.Event()

    def run(self):
        print("Initializing □thread□" + self.threadName, "\n")
        while True:
            if self.secure.isSet():
                print("Security □Running")
                security()
        print("Security □loop□end. □Terminating□process□"
              + self.threadName)

    def pause(self):
        self.secure.clear()

    def resume(self):
        self.secure.set()
```



V prípade, že PIR senzor zaznamená pohyb a otáčky motora nie sú nulové, serverová aplikácia sa uistí, že Wi-Fi rozhranie wlan0 je zapnuté a teda je možné sa k Raspberry Pi pripojiť pomocou mobilného telefónu. Predpokladá sa, že pokiaľ je v aute pohyb a motor beží, sedí v ňom oprávnená osoba. Nie je preto dôvod na zhotovovanie fotografií a odosielanie ich pomocou e-mailu. Pokiaľ však oprávnená osoba nenašartuje do 9 sekúnd, fotografia bude napriek tomu zhotovená a odoslaná.

Danú funkcionálnosť je možné vypnúť pomocou mobilnej aplikácie, pre prípad návštevy autoservisu, a pod.

Vlákno `monitorThread` umožňuje čítanie dát z riadiacej jednotky vozidla, spracovávanie a odosielanie daných dát do mobilnej aplikácie.

Vždy, keď sa zmení hodnota rýchlosti alebo otáčok motora, vlákno danú zmenu zaznamená a odošle protokolom MQTT správu s novou hodnotou, ktorú následne spracuje mobilná aplikácia a hodnotu vypíše.

```
class monitorThread(threading.Thread):
    def __init__(self, ID, name):
        threading.Thread.__init__(self)
        self.threadID = ID
        self.threadName = name
        self.monitor = threading.Event()

    def run(self):
        print("Initializing□thread□" + self.threadName, "\n")
        while True:
            if self.monitor.isSet():
                print("Monitoring□Running")
                monitoring()
            print("Monitoring□loop□end.□Terminating□process□"
                  + self.threadName)
```

`ServiceThread` je vlákno slúžiace na servisné účely. V prípade potreby je možné odoslať z telefónu správu do MQTT témy, ktorú `serviceThread` odoberá, čím je možné zabezpečiť rôzne prídavné funkcionality systému. V súčasnosti však `serviceThread` slúži iba pre prijímanie užívateľom zvolených nastavení, v našom prípade telefónne čísla pre získanie polohy vozidla pomocou služby SMS a privolanie pomoci v prípade núdzovej situácie.

```
class serviceThread(threading.Thread):
    def __init__(self, ID, name):
        threading.Thread.__init__(self)
        self.threadID = ID
```

```

self.threadName = name

def run(self):
    print("Starting SERVICE thread", "\n")
    client.on_message = on_message
    client.loop_forever()
    print("Service loop ending. Terminating"
          + self.threadName)

```

### 5.4.10 Vytvorenie mobilnej aplikácie

Mobilná aplikácia bude slúžiť na zobrazovanie aktuálnych stavových informácií o vozidle ako napríklad aktuálna rýchlosť vozidla, otáčky motora, teplota chladiacej kvapaliny a pod. Pre vytvorenie aplikácie použijeme programovacie prostredie Android Studio a programovací jazyk Java.

Android Studio nainštalujeme z oficiálnej stránky programu.

Aplikácia sa bude skladať z troch aktivít:

- **Main Activity** - hlavná aktivita, v ktorej bude prebiehať hlavná komunikácia s MQTT Brokerom, spracovávanie a zobrazovanie informácií a vykonávanie hlavných funkcií aplikácie ako získavanie polohy vozidla využitím technológie SMS alebo volanie pomoci v prípade výnimočnej situácie.
- **Login Activity** - prihlasovacia aktivita. V danej aktivite prebieha inicializácia spojenia s MQTT serverom, nastavenie prihlasovacích údajov a ich odovzdávanie do hlavnej aktivity.
- **Settings Activity** - aktivita nastavení, kde je možné nastaviť telefónne čísla a úsporný režim.

Výsledný kód mobilnej aplikácie je možné nájsť v prílohe práce.

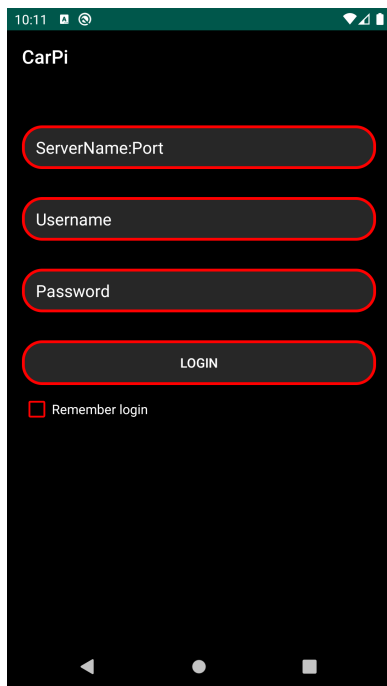
Na obrázkoch 5.8a až 5.8d je možné vidieť vzhľad výslednej mobilnej aplikácie.

Aplikácia obsahuje menu nastavení, v ktorom je možné zvoliť nastavenie úsporného režimu, nastaviť telefónne číslo pre získanie polohy a tiež telefónne číslo na odťahovú službu alebo predvolený núdzový kontakt.

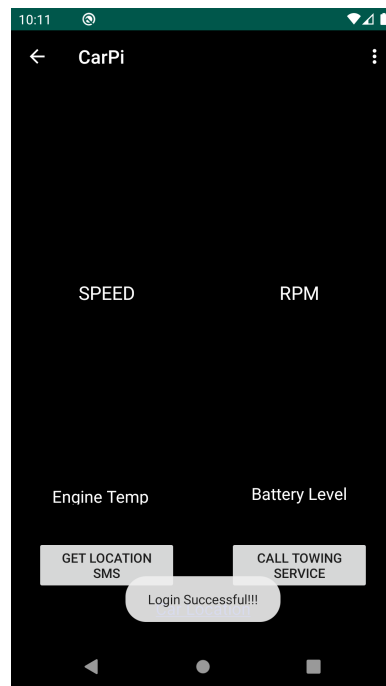
Na hlavnej obrazovke je možné vidieť 4 polia obsahujúce hodnoty získané z Raspberry Pi pomocou protokolu MQTT spolu s ich označeniami, 2 tlačidlá slúžiace na komunikáciu a odkaz pre zobrazenie aktuálnej polohy vozidla.

Prvé tlačidlo (vľavo) slúži na odoslanie SMS pre získanie polohy z Raspberry Pi. Tlačidlo je vhodné využiť napríklad v prípade, že auto nie je na blízku (nie je možné sa s RPi spojiť pomocou Wi-Fi).

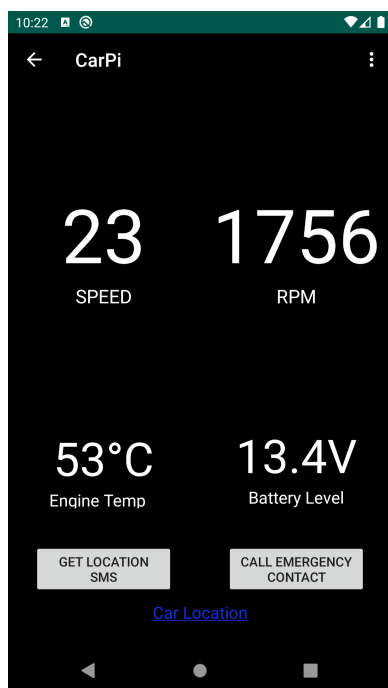
Druhé tlačidlo slúži pre rýchle privolanie pomoci alebo odťahovej služby v prípade núdze. Volané číslo je možné zmeniť v nastaveniach.



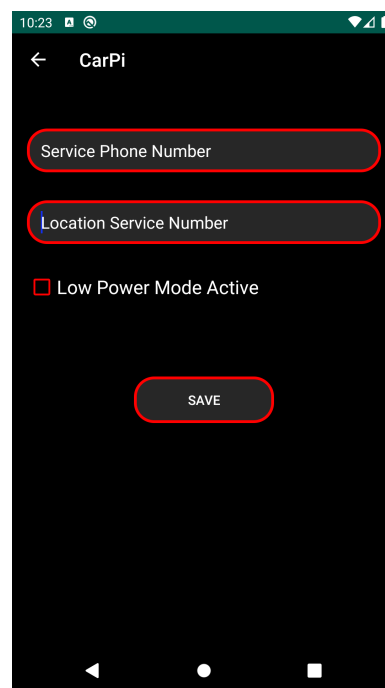
(a) Prihlasovacia obrazovka aplikácie



(b) Aplikácia pred prijatím dát



(c) Aplikácia po prijatí dát



(d) Obrazovka nastavení aplikácie

Obr. 5.8: Mobilná aplikácia CarPi.

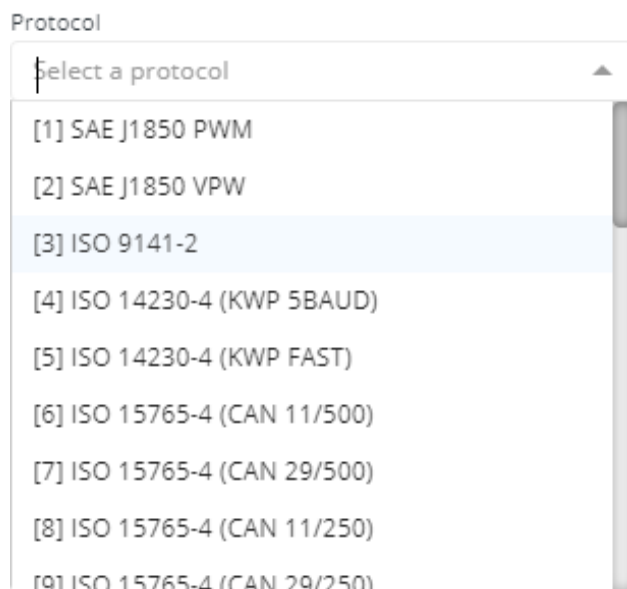
Po kliknutí na odkaz s polohou je používateľ presmerovaný do aplikácie Google Maps (ak je nainštalovaná) a zobrazenie je automaticky nastavené na polohu vozidla podľa GPS súradníc z Raspberry Pi.

### 5.4.11 Ovládanie komponentov vozidla

Spoločnosť AutoPi.io reklamuje, že s ich zariadením je možné ovládať niektoré komponenty vozidla, ako napríklad otváranie a zatváranie okien a zámkov dverí, manipulácia s klimatizáciou, a pod.

Na začiatku tejto časti práce bola snaha o potvrdenie spomenutého tvrdenia replikovaním krokov, ktoré jeden z administrátorov a developerov danej spoločnosti zverejnil na fóre spoločnosti. [23]

Pre komunikáciu s riadiacimi jednotkami vozidla je najskôr nutné zistiť, pomocou ktorého protokolu spolu navzájom tieto jednotky komunikujú. Pôvodný softvér AutoPi ponúka na výber množstvo automobilovými spoločnosťami najčastejšie používaných protokolov. Príklad spomenutých protokolov je zobrazený na obrázku 5.9.



Obr. 5.9: Protokoly podporované zariadením AutoPi Dongle

Najjednoduchšia identifikácia protokolu používaného testovaným vozidlom je vyhľadanie si informácií na internete. Z niekoľkých zdrojov (napr. [24]) sme sa dozvedeli, že prvé testované auto, Mitsubishi ASX 1.8 DI-D rv. 2014, používa protokol CAN 11-bit 500kbps, čo je v AutoPi pojímaní protokol 6. Nastavili sme teda tento protokol v AutoPi softvéri.

Ďalším krokom bolo nahratie správ posielaných medzi riadiacimi jednotkami. AutoPi Dongle umožňuje danú činnosť už od prvej verzie, avšak je potrebné si túto funkcionality doinštalovať na ich webovom portáli.

Pred samotným spustením nahrávania správ je nutné otočiť kľúčik v zapalovaní do polohy "ACC", kedy je elektrický prúd vo všetkých systémoch vozidla. Následne bolo spustené nahrávanie správ po dobu 10 sekúnd, počas ktorých sme opakovane

odomykali a zamykali zámky dverí, aby dané správy boli obsiahnuté v zaznamenaných správach.

Po 10 sekundách bol vrátený prázdny záznam. Ani po niekoľkých pokusoch a ovládaní rôznych komponentov nebola zaznamenaná žiadna správa.

Prešli sme k testovaniu iných protokolov podporovaných zariadením AutoPi Dongle. Výstupom z každého z nich bola prázdna návratová množina alebo chyba o neúspešnom pripojení k vozidlu.

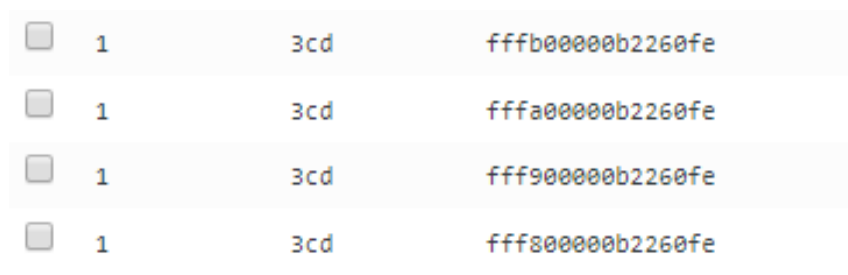
Ani po niekoľkých týždňoch sa nepodarilo získať záznam správ z komunikácie s vozidlom. Predchádzajúci postup bol replikovaný aj pri vypnutom a naštartovanom motore, no bez úspechu.

Následne prebehli testy ďalších vozidiel. Všetky predchádzajúce kroky boli otestované na automobiloch Škoda Roomster 1.4FSI rv. 2012 a Citroen Xsara Picasso 1.6 HDI rv. 2004. Výsledok bol pri oboch rovnaký.

Ako posledné bolo testované vozidlo Peugeot Expert rv.2018. Pri danom vozidle sa podarilo nadviazať spojenie pomocou protokolu "6" a taktiež bolo počas 10-sekundového nahrávania zaznamenaných viac ako 5000 správ.

Softvér AutoPi dovoľuje zaznamenané správy odoslať postupne späť do vozidla. Využili sme spomenutú možnosť, no nič sa nestalo.

Následne sme začali zaznamenané správy skúmať. Počas testu sme 2-krát odomkli a zamkli zámky dverí. Hľadali sme najskôr dve správy s rovnakým identifikátorom riadiacej jednotky, avšak takéto správy sa v zázname nenachádzali. Napokon sme prešli k hľadaniu štyroch správ s rovnakým identifikátorom. Podobný záznam sa v získanej množine nachádzal, ako je možné vidieť na obrázku 5.10.



<input type="checkbox"/>	1	3cd	ffffb00000b2260fe
<input type="checkbox"/>	1	3cd	fffa00000b2260fe
<input type="checkbox"/>	1	3cd	fff900000b2260fe
<input type="checkbox"/>	1	3cd	fff800000b2260fe

Obr. 5.10: Záznam kódov pre odomknutie zámkov vozidla

Záznam ukazuje logickú spojitosť medzi jednotlivými správami. Mení sa štvrtý bit v odoslanej správe od f do 0, následne sa cyklus opakuje. Predpokladali sme, že ide o bezpečnostnú funkciu, kedy riadiaca jednotka zámkov dverí očakáva ďalšiu správu nasledujúcu v poradí, aby nebolo možné auto odomknúť vždy rovnakou správou.

Skúsili sme odoslať iba dané 4 správy a čakali sme na výsledok. Znovu sa nič nestalo.

Po niekoľkých neúspešných pokusoch a rôznych spôsoboch odosielania správ sme usúdili, že je v aute použitý bezpečnostný protokol, ktorý neumožňuje ovládanie komponentov externým zariadením.

Kontaktovali sme teda výrobcu automobilov Mitsubishi so žiadosťou o pomoc a zaslanie materiálov, ktoré by nám pomohli s vyriešením daného problému. Mitsubishi z časti vyhovel a zaslalo schému zapojenia zámkov dverí.

Po preštudovaní schémy bolo zistené, že zapojenie zámkov je také ako sme pôvodne predpokladali, tzn. problém v samotnom zapojení zámkov nie je.

Mitsubishi však odmietlo podať informácie o protokoloch a zabezpečení riadiacej jednotky, čím sa potvrdila teória, že riadiace jednotky sú zabezpečené proti manipulácii externým zariadením.

Po množstve neúspešných pokusov o nadviazanie spojenia s RJ sme dospeli k záveru, že na prelomenie spomínaného zabezpečenia je potrebné špeciálne zariadenie alebo špeciálny autorizačný kľúč. Obe z týchto možností však nie sú verejne dostupné.

Ďalším študovaním danej problematiky bolo zistené, že existuje tzv. "Mód 23", ktorým sa upravujú mapy riadiacich jednotiek pre športové autá, tzv. "profi chip-tuning". Hoci ani jedno z testovaných vozidiel takúto riadiacu jednotku neobsahuje, rozhodli sme sa otestovať aj tento spôsob komunikácie s RJ, avšak ani po niekoľkých pokusoch s rôznymi nastaveniami spomenutého módu nebolo nadviazané ani základné spojenie medzi Raspberry Pi a hlavnou riadiacou jednotkou vozidla.

Záverom nášho snaženia je konštatovanie, že po množstve incidentov, ktoré sa uskutočnili počas rokov existencie nariadenia o OBD konektoroch a ich povinnosti implementácie vo vozidlách, sa automobilové spoločnosti poučili a zabezpečili riadiace jednotky proti externým vplyvom a zmenám informácií v samotnej RJ.

## 5.5 Popis priebehu činnosti systému

Nasledovná kapitola sa venuje popisu funkcionalít systému, behu aplikácie, vlastností systému a krokov potrebných k úspešnej počítačovej konfigurácii vytvoreného systému.

### 5.5.1 Prvé spustenie

Po zhotovení aplikácie je možné ju nainštalovať do mobilného telefónu alebo tabletu ako apk súbor alebo priamo pomocou Android Studia, pričom je však potrebné mať v telefóne alebo tablete povolený debugovací mód.

Zariadenie AutoPi Dongle s našim novým systémom pripojíme do OBD-II zbernice vozidla a otočíme kľúčom. Zariadenie sa zapne a automaticky spustí všetky tri

vlákna. Následne je možné pripojiť sa pomocou mobilnej aplikácie k MQTT serveru, ktorý pri naštartovanom motore začína odosielať aktuálne hodnoty.

Po spustení aplikácie užívateľa privíta prihlasovacia obrazovka, kde je potrebné zadať IP adresu a port servera MQTT (IP Socket) a prihlasovacie meno a heslo. Užívateľ má možnosť si dané údaje v aplikácii uložiť. Po potvrdení tlačidlom "Login" je užívateľ presunutý na hlavnú obrazovku, kde vidí základné informácie o vozidle spolu s tlačidlami popísanými v kapitole 5.4.10.

Pre správne fungovanie aplikácie si užívateľ nastaví požadované telefónne čísla v záložke "Settings" a zvolí, či v prípade klesnutia napätia pod hodnotu 12.2V má Raspberry Pi prejsť do režimu šetrenia energie.

K systému je pripojená aj kamera Raspberry Pi camera Noir 2. generácie spoločne s PIR senzorom HC SR501. Tieto sú využité pre detekciu pohybu a oznámenie narušenia bezpečnosti vozidla počas neprítomnosti majiteľa. V prípade, že PIR senzor zaznamená pohyb, počká 9 sekúnd. Ak po 9 sekundách znovu zaznamená pohyb, okamžite sa vytvorí fotografia narušiteľa pomocou kamery a odošle sa na e-mailovú adresu majiteľa vozidla.

Detekciu pohybu je možné manuálne deaktivovať v nastaveniach aplikácie. Systém taktiež berie do úvahy situáciu, kedy v prípade zapnutej detekcie a naštartovaného motora pravdepodobne v aute sedí oprávnená osoba. Fotografie sa v tomto prípade nebudú zhotovovať, majiteľ vozidla teda nebude rušený neoprávnenými upozorneniami o narušení vozidla.

### **5.5.2 Úsporný režim**

Úsporný režim zabezpečuje nižšiu spotrebu elektrickej energie v prípade poklesu napätia pod istú úroveň. V našom prípade je táto úroveň nastavená na 12.2V, nakoľko minimálne bezpečné napätie auto-batérie je 12V.

V prípade zníženia napätia na batérii sa vypnú všetky prebytočné procesy a bežať zostane iba zabezpečovací systém ovládajúci kameru, PIR senzor a GSM/GPS modul pre možnosť komunikácie a získavania informácií o polohe vozidla.

### **5.5.3 Spotreba energie**

Energetické nároky systému sú prispôbené na dlhodobejšie pripojenie zariadenia AutoPi Dongle k autu s minimalizáciou nebezpečenstva vybitia batérie. Namerané hodnoty spotreby energie je možné nájsť v Tab. 5.3

Základom zariadenia AutoPi Dongle je nízko-výkonový počítač Raspberry Pi Zero W, ktorý je veľmi úsporný. Vďaka použitiu daného zariadenia sa spotreba energie pri plnom výkone, pri zhotovovaní fotografií a ich odosielaní e-mailom nedostala nad úroveň 5W celkového výkonu.

V úspornom režime je dokonca možné dostať sa na úroveň pohybujúcu sa v rozmedzí medzi 80-200mA.

Bez úsporného režimu, resp. za štandardného chodu, kedy prebieha iba čítanie informácií z RJ a ich odosielanie do aplikácie je spotreba energie v rozmedzí od 120-250mA.

<b>Režim</b>	<b>Spotreba</b>
Štandard	~120-250mA
Plný výkon + foto	~0.9A
Úsporný režim	~80-200mA

Tab. 5.3: Spotreba energie v závislosti na aktívnom režime

Všetky hodnoty sú v závislosti na sile GSM signálu a počte zariadení pripojených ku Raspberry Pi pomocou rozhrania Wi-Fi.



## Záver

Pôvodný zámer práce, ktorým bolo vzdialené ovládanie nekritických systémov vozidla, sa nepodarilo uskutočniť. Ani po niekoľko týždňovom snažení sa nepodarilo nadviazať komunikáciu s riadiacimi jednotkami vozidla, ktorými sú ovládané jednotlivé nekritické systémy vozidla zahŕňajúce napríklad ovládanie zámkov dverí. Výskum bol uskutočnený na štyroch testovacích vozidlách rôznych značiek a rokov výroby, napr. Mitsubishi ASX rv. 2014, Citroen Xsara Picasso rv. 2004, Peugeot Expert rv. 2018 a Škoda Roomster rv. 2012. Počas práce prebehol rozsiahly prieskum v oblasti elektrotechnického vybavenia a zapojenia jednotlivých systémov vozidla. Taktiež bol kontaktovaný výrobca vozidiel Mitsubishi so žiadosťou o poskytnutie informácií o zberniciach vozidla a použitých protokoloch komunikácie.

Jediné vozidlo, s ktorým sa podarilo nadviazať akúsi pokročilejšiu komunikáciu a teda vyčítať správy zo zbernice OBD-II bol Peugeot Expert, avšak ani s týmto vozidlom sa nepodarilo komunikovať až do štádia, kedy by bolo systémy vozidla možné ovládať.

Nakoľko pôvodný zámer práce nebolo možné zrealizovať z dôvodu zabezpečenia komunikačnej zbernice zo strany automobilových výrobcov, sa cieľ predloženej diplomovej práce preorientoval na vytvorenie systému slúžiaceho na získavanie a zobrazovanie stavových informácií z riadiacej jednotky motora automobilu.

Výstupom diplomovej práce je funkčná mobilná aplikácia so schopnosťou pripojenia sa k MQTT serveru bežiacom na zariadení Raspberry Pi Zero W a zobrazovania aktuálnych stavových informácií vozidla. Vytvorený systém je možné využiť aj ako prídavné zabezpečovacie zariadenie automobilu, vďaka kamere s nočným videním, pasívnym infračerveným pohybovým senzorom schopnosťou odosielať fotografie pomocou e-mailovej služby.

Aplikácia umožňuje rýchle privolanie pomoci v prípade núdzovej situácie alebo získanie aktuálnej polohy vozidla využitím služby SMS.

Nami vytvorený systém poskytuje možnosť k ďalšiemu rozvoju a dáva priestor pridávaniu nových funkcionalít.



# Literatúra

- [1] Spoločnosť Cisco: *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper* [online] 2017. Dostupné z: <<https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>>
- [2] MORGAN, Jacob, časopis Forbes: *A Simple Explanation Of 'The Internet Of Things'* [online] Dostupné z : [online]. Dostupné z: <<https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#2e0b61b81d09>>
- [3] POSTSCPAES: *Internet of Things (IoT) History* [online] Dostupné z:[online]. Dostupné z: <<https://www.postscapes.com/internet-of-things-history/>>
- [4] ASHTON, Kevin, RFID Journal: *That 'Internet of Things' Thing* [online] Dostupné z: [online]. Dostupné z: <<http://www.rfidjournal.com/articles/view?4986>>
- [5] International Telecommunications Union: *ITU Internet Reports 2005: The Internet of Things* Dostupné z:[online]. Dostupné z: <[http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings\\_summary.pdf](http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf)>
- [6] HUANG, Chung-Ching; BARDZELL, Jeffrey; TERRELL, Jennifer: *978-1-4503-1280-6 Can Your Pet Rabbit Read Your Email?: A Critical Analysis of the Nabaztag Rabbit* Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces, 2011
- [7] Wireless Networking and Communications Group: *Device-to-Device (D2D) Communication: Fundamentals with Applications to LTE*. Dostupné z: [online] Dostupné z: <<https://wncg.org/research/briefs/device-device-d2d-communication-fundamentals-applications-lte>>
- [8] ROUSE, Margaret, IoT Agenda: *Machine-to-machine (M2M)* [online] 2010. Dostupné z: <<http://internetofthingsagenda.techtarget.com/definition/machine-to-machine-M2M>>
- [9] GALININA, Olga; ANDREEV, Sergey ; BALADIN, Sergey; KOUCHE-RYAVY, Yevgeni: *978-3-319-67380-6 Internet of Things, Smart Spaces, and Next Generation Networks and Systems* [online] Dostupné

- z: <https://books.google.cz/books?id=P3o4DwAAQBAJ&printsec=frontcover&hl=sk#v=onepage&q&f=false>
- [10] PLASMACOMP: *The Evolution of M2M Technology – From M2M to IoT* Dostupné z: [online]. Dostupné z: <http://www.plasmacomp.com/blogs/evolution-of-m2m-technology-from-m2m-to-iot>
- [11] Spoločnosť Gartner: *Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016* Dostupné z: [online]. Dostupné z: <https://www.gartner.com/newsroom/id/3598917>
- [12] MAYES, Keith: *Smart Cards, Tokens, Security and Applications* ISBN: 978-3-319-50500-8 Dostupné z: [online]. Dostupné z: <https://link.springer.com/content/pdf/10.1007/978-3-319-50500-8.pdf>
- [13] Gemalto: *Hardware Security Modules* [online] Dostupné z: [online] Dostupné z: <https://safenet.gemalto.com/data-encryption/hardware-security-modules/>
- [14] GEMALTO: *Software IP protection* [online] Dostupné z: <https://sentinel.gemalto.com/software-monetization/ip-protection/>
- [15] KRUTZ, Ronald L.; VINES, Russell Dean: *Cloud security: a comprehensive guide to secure cloud computing*. Indianapolis, IN: Wiley Pub., c2010. ISBN 0470589876.
- [16] HAGHIGHAT, M., Zonouz, S., and Abdel-Mottaleb, M.: *Cloudid: Trustworthy cloud-based and cross-enterprise biometric identification*. Expert Systems with Applications 42, 21 (2015), 7905 – 7916.
- [17] JUN Tang, Yong Cui, Qi Li, Kui Ren, Jiangchuan Liu, Rajkumar Buyya. 2016.: *Ensuring Security and Privacy Preservation for Cloud Data Services*. ACM Comput. Surv. 49, 1, Článok 13 2016., 39 strán. Dostupné z: <https://doi.org/10.1145/2906153>
- [18] PESTANES Philippe, GAUTIER Benjamin, Wavestone. 2017.: *The rise of intelligent voice assistants: new gadget for your living room or window of opportunity to reshuffle the cards in the web economy?* [online] 2017, 8 strán. Dostupné z: <https://www.wavestone.com/app/uploads/2017/09/Assistants-vocaux-ang-02-.pdf>
- [19] BAY Scott: *AI assistants are poised for major growth in 2018*[online] Dostupné z: <https://venturebeat.com/2018/01/22/ai-assistants-are-poised-for-major-growth-in-2018/>

- [20] MTE: *Ridiaca jednotka motora* [online] Dostupné z: <[mte.sk](http://mte.sk), [cit. 2017-03-25].
- [21] OPENSOURCE.COM: *What is a Raspberry Pi?* [online] Dostupné z: <<https://opensource.com/resources/raspberry-pi>>
- [22] ORACLE: *General purpose Input/Output* [online] Dostupné z: <<https://docs.oracle.com/javame/8.0/me-dev-guide/gpio.htm>>
- [23] AutoPi.io *Using the CAN bus commands* [online] Dostupné z: <<https://community.autopi.io/t/using-the-can-bus-commands/571>>
- [24] *Mitsubishi compatible OBD2 and ELM327* [online] Dostupné z: <<https://www.outilsobdfacile.com/vehicle-list-compatible-obd2/mitsubishi>>



# Zoznam symbolov, veličín a skratiek

**IoT** Internet vecí – angl. Internet of Things

**M2M** Komunikácia stroja so strojom – angl. Machine to machine communication

**H2H** Komunikácia človeka s človekom – angl. Human to Human communication

**D2D** Komunikácia zariadenia so zariadením – angl. Device to device communication

**SoC** Systém na chipe – angl. System on a chip

**CPU** Centrálna procesorová jednotka – angl. central processing unit

**GPU** Grafická procesorová jednotka – angl. graphical processing unit

**RAM** Pamäť s náhodným prístupom – angl. Random-access memory

**LED** Svetlo-vyžarujúca dióda – angl. Light-emiting diode

**HD** Vysoké rozlíšenie – angl. High-definition

**SIM** Účastnícka identifikačná karta – angl. Subscriber Identity Module. Karta slúžiaca na identifikáciu účastníka v sieti.

**GSM** (Globálny systém pre mobilnú komunikáciu – angl. Global System for Mobile Communications)

**RJ** (Riadiaca jednotka – angl. Control Unit)

**MQTT** (MQ Telemetry Transport)

**PIR** (Pasívny infračervený senzor – angl. Passive Infrared sensor)

**APN** (Sieťový prístupový bod – angl. Network Access Point)





# Zoznam príloh

A Obsah prílohy

65



## A Obsah prílohy

/	.....	koreňový adresár priloženého CD		
	carpij	.....Zložka obsahujúca kód mobilnej aplikácie		
		LoginActivity.java	.....	Kód prihlasovacej aktivity aplikácie CarPi
		MainActivity.java	.....	Kód hlavnej aktivity aplikácie CarPi
		SettingsActivity.java	.....	Kód aktivity nastavení aplikácie CarPi
	AndroidManifest.xml	.....	Základné nastavenia aplikácie CarPi	
	build.gradle	.....	Súbor obsahujúci nastavenia knižníc aplikácie CarPi	
	carpi.py	.....	Kód programu bežiaceho na Raspberry Pi	