

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra technologií a měření

DIPLOMOVÁ PRÁCE

Monitorování a sběr dat z IoT systémů

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická
Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Jaroslav PODESTÁT**
Osobní číslo: **E18N0011P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Komerční elektrotechnika**
Téma práce: **Monitorování a sběr dat z IoT systémů**
Zadávací katedra: **Katedra technologií a měření**

Zásady pro vypracování

1. Analyzujte aktuální trendy ve sběru dat s využitím IoT.
2. Analyzujte a navrhněte vhodná řešení pro IoT uzly.
3. Navrhněte a zrealizujte řešení pro zadaný IoT systém.
4. Vyberte vhodné postupy pro sběr, uchovávání a vizualizaci získaných dat.
5. Zhodnoťte navržené a zrealizované řešení.



Rozsah diplomové práce: **40 – 60 stran**
Rozsah grafických prací: **podle doporučení vedoucího**
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Systém pro sběr technologických dat v koncepci IoT Diplomová práce ČVUT [online]. (c) 2017 [cit. 16.4.2019]. Dostupné z: https://dspace.cvut.cz/bitstream/handle/10467/68593/F3-DP-2017-Hofman-Jan-System_pro_sber%20technologickych_%20dat_v_koncepci_%20IoT.pdf?sequence=-1&isAllowed=y
2. BURDA, Karel. Úvod do kryptografie. Vydání první. Brno: Akademické nakladatelství CERM, 2015. 108 stran. ISBN 978-80-7204-925-7.
3. BURDA, Karel. Základy elektronických zabezpečovacích systémů. Vydání první. Brno: Akademické nakladatelství CERM, 2017. 123 stran. ISBN 978-80-7204-967-7.
4. GIEBL, Jan. Elektronický zabezpečovací systém pro rodinný dům, diplomová práce, Západočeská univerzita v Plzni, 2019. Internetové zdroje, Univerzitní knihovna ZČU.
5. SmartCAMPUS tým. Projekt KETCube. Západočeská univerzita v Plzni [online]. (c) 2019 [cit. 16.4.2019]. Dostupné z: <https://github.com/SmartCAMPUSZCU/KETCube-docs>

Vedoucí diplomové práce: **Ing. Petr Kropík, Ph.D.**
Katedra teoretické elektrotechniky

Datum zadání diplomové práce: **4. října 2019**
Termín odevzdání diplomové práce: **28. května 2020**


Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan




Doc. Ing. Aleš Hamáček, Ph.D.
vedoucí katedry

V Plzni dne 4. října 2019

Abstrakt

Předkládaná diplomová práce je zaměřena na téma monitorování a sběru dat z IoT systémů. Začíná analýzou toho, v jakých oblastech se v současnosti používá. Dále rozebírá, jaké jsou vhodné možnosti pro IoT uzly. Následně řeší realizaci zadaného IoT systému, konkrétně chytrého parkování, prostřednictvím spolehlivého řešení. Nakonec provádí zhodnocení vybraného aplikovaného provedení.

Klíčová slova

chytré parkování, monitorování dat, sběr dat, Internet věcí, LoRaWAN síť, MySQL databáze, PHP webová stránka

Abstract

The presented diploma thesis is focused on the topic of the data monitoring and data collection from the IoT systems. It starts with the analysis of which fields it is currently used in. Then it depicts what are the appropriate options for IoT nodes. After that, it deals with the realization of the assigned IoT system, specifically smart parking, with the use of a reliable solution. At the end, it evaluates the used chosen application.

Key words

smart parking, data monitoring, data collection, Internet of Things, LoRaWAN network, MySQL database, PHP webpage

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

podpis

V Plzni dne 17.6.2020

Jaroslav Podestát

Poděkování

Na tomto místě bych rád poděkoval vedoucímu diplomové práce panu Ing. Petru Kropíkovi, Ph.D. za vedení, praktickou pomoc a směrování při tvorbě koncepce této práce. Dále bych chtěl poděkovat panu Ing. Karlu Šimovi za vytvoření a přidělení MySQL databáze a zapůjčení zařízení Arduino MKR WAN 1300. Nemalý dík patří též panu Milanu Holubovi z firmy CITIQ s.r.o. za poskytnutí technických parametrů detektorů a konzultaci.

Obsah

OBSAH	8
SEZNAM SYMBOLŮ A ZKRATEK	10
ÚVOD	11
1 SBĚR DAT S VYUŽITÍM IOT	12
1.1 TRENDY V OBLASTI INTERNET OF THINGS	12
1.2 SMART SENZORY	13
1.3 PLATFORMY ARDUINO A RASPBERRY PI	14
1.4 ROZŠÍŘENÍ PRO RASPBERRY PI A ARDUINO	15
1.5 VÝZNAM SBĚRU DAT	17
1.5.1 Domácí aplikace	18
1.5.2 Průmyslové aplikace Raspberry Pi	19
1.6 ROZDĚLENÍ MÍST POUŽITÍ SBĚRU DAT	20
2 MOŽNOSTI ŘEŠENÍ PRO IOT UZLY	22
2.1 IOT PROTOKOLY	22
2.1.1 Protokoly D2D	23
2.1.2 Protokoly D2S	23
2.1.3 Protokoly S2S	25
2.2 VÝVOJOVÉ PROSTŘEDÍ NODE-RED	26
3 NÁVRH A REALIZACE ŘEŠENÍ ZADANÉHO IOT SYSTÉMU	28
3.1 MONITORING PRŮJEZDU DOPRAVY	28
3.2 PŘETÍŽENOST SILNIC V OKOLÍ ŠKODY AUTO V KVASINÁCH POMÁHÁ ŘEŠIT CITIQ, ČLEN IQRF ALLIANCE	29
3.3 TECHNICKÁ SPECIFIKACE	30
3.4 APLIKACE SYSTÉMU PRŮJEZDOVÝCH SENZORŮ U ŠKOLNÍHO PARKOVIŠTĚ	32
3.5 OPEN DATA	34
3.6 LORAWAN	35
3.6.1 Pokrytí sítě LoRaWAN	36
3.6.2 Zabezpečení LoRaWAN	38
3.6.3 Způsoby připojení zařízení do LoRaWAN	39
4 SBĚR, UCHOVÁVÁNÍ A VIZUALIZACE OBDRŽENÝCH DAT	40
4.1 NASTAVENÍ LORA PŘIPOJENÍ	40
4.2 KONFIGURACE V LORATECH	41
4.2.1 Aktivace detektorů	41
4.2.2 Parsování v prostředí LORATECH	42
4.2.3 Callback	44
4.2.4 LORATECH a The Things Network	45
4.3 PARSOVÁNÍ A PŘENOS DAT PROSTŘEDNICTVÍM NODE-RED	45
4.4 UKLÁDÁNÍ DO MYSQL DATABÁZE	47
4.5 TVORBA WEBOVÉ STRÁNKY	48
4.5.1 Základní nastavení	48
4.5.2 Nastavení data a času	49
4.5.3 Získání dat pro grafy	50
4.5.4 Sestavení tabulky	52
4.5.5 Umístění grafů	53

5	ZHODNOCENÍ REALIZOVANÉHO ŘEŠENÍ	55
5.1	POROVNÁNÍ DETEKOVANÝCH A FYZICKY ZAZNAMENANÝCH HODNOT	55
5.2	CELKOVÝ PROCES PŘENOSU DAT	56
	ZÁVĚR	57
	SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	58

Seznam symbolů a zkratek

AMQP - Advanced Message Queuing Protocol

CISS - Connected Industrial Sensor Solution

CoAP - Constrained Application Protocol

GDS - Global Data Space

HAT - Hardware Attached on Top

HTTP - Hypertext Transfer Protocol

IoT - Internet of Things

JSON - JavaScript Object Notation

MQTT - Message Queuing Telemetry Transport

NFC - Near Field Communication

OMG - Object Management Group

ORTE - Open Real-Time Ethernet

PHP - Hypertext Preprocessor

REST - Representational State Transfer

RF - Radio Frequency

RTPS - Real-Time Publish-Subscribe

RFID - Radio Frequency Identification

SQL - Structured Query Language

Wi-Fi - Wireless Fiber

XMPP - Extensible Messaging and Presence Protocol

Úvod

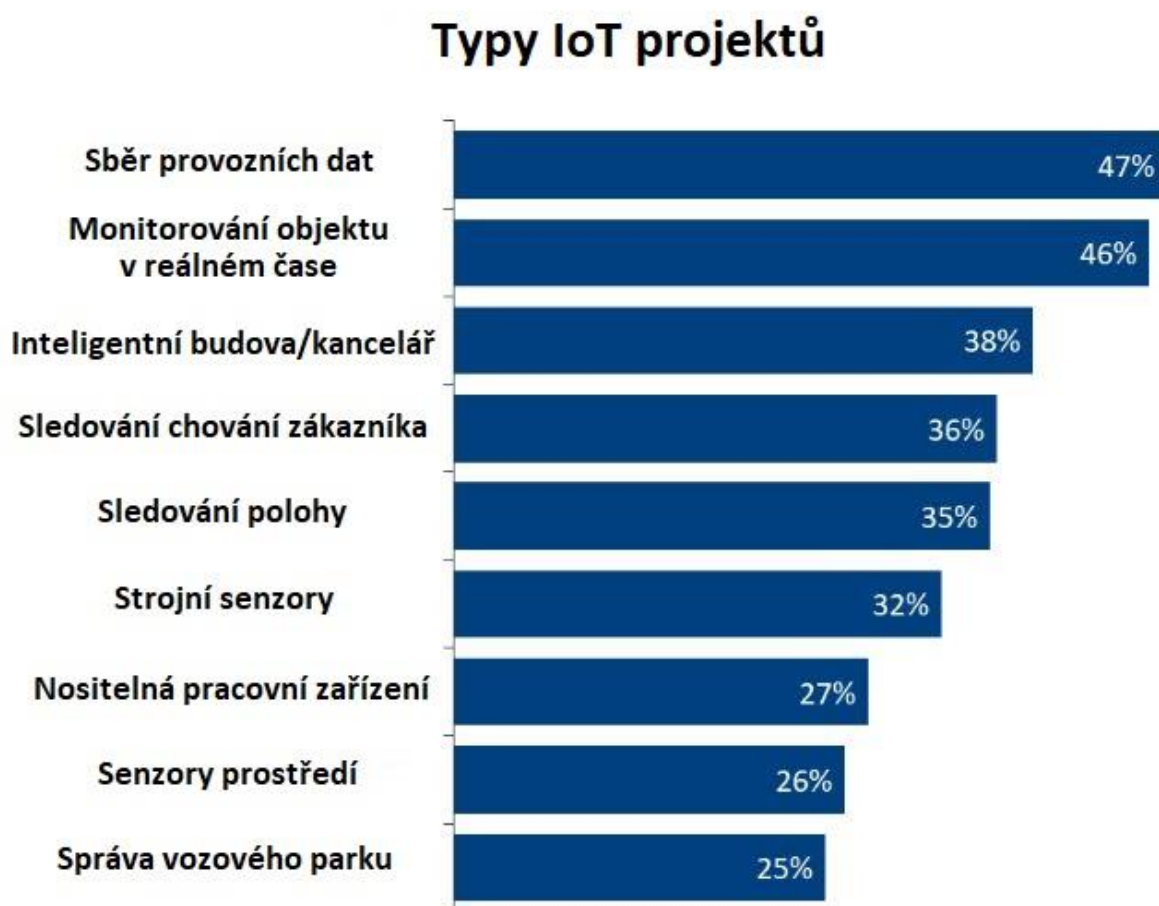
Cílem předkládané práce je vysvětlení významu sběru dat s využitím IoT formou rešerše, spoluúčast na zavedení takového systému, s rozbořením takového procesu a vytvoření webové aplikace pro zobrazení zpracovaných dat.

První část diplomové práce je rešerše věnující se tomu, jakými způsoby se dnes využívá sběr dat s využitím IoT a co je nezbytné pro zprovoznění takové technologie. Druhá část se již podrobně zabývá konkrétním řešením pro IoT uzly. Třetí část obsahuje návrh a realizaci reálně řešeného případu, tedy zavedení nového systému senzorů u parkoviště Fakulty elektrotechnické u Západočeské univerzity v Plzni. Ve čtvrté části je vysvětleno, jak se postupovalo při sběru dat z nainstalovaných senzorů, jejich následnému zpracování a konečnému zobrazení prostřednictvím webové aplikace odpovídající specifickým požadavkům. Na závěr jsou zhodnoceny údaje z nově zprovozněného sensorického systému a porovnány s reálnou situací na místě měření.

1 Sběr dat s využitím IoT

1.1 Trendy v oblasti Internet of Things

IT společnost CompTIA v roce 2019 provedla průzkum u firem v oblasti IT [1] a zaměřila se na informace ohledně IoT. Na prvním obrázku *Obr. 1.1* je graf popisující, jakými typy IoT se dotazované firmy zabývají. Vyplývá zde, že největší podíl je u sběru provozních dat a monitorování objektu v reálném čase. Některé projekty však spíše patří mezi hlavní činnosti firem nepatřící do IT odvětví jako například správa vozového parku nebo sledování chování zákazníka. Spoluprací s IT oddělením se jim tak ale právě otevírají nové možnosti.



Obr. 1.1 Porovnání, jakými typy IoT projektů se firmy zabývají, překresleno z [1]

V této samé studii [1] je dále vytvořený přehled viditelný na obrázku *Obr. 1.2* zabývající se tím, co společnosti označují jako zásadní dovednost potřebnou v IoT a co z toho by u nich bylo potřeba zlepšit. Objevují se zde s nezanedbatelným poměrem právě takové věci jako je správa a analýza dat či cloud computing. Zmíněné zlepšení je pro ně důležité, protože mají

tyto znalosti a dovednosti pouze na základní úrovni. Tím se pak mohou vystavovat riziku nedostatečného zabezpečení údajů, které je proto v žebříčku na prvním místě.

Dovednosti potřebné v IoT

	Označeno společnostmi jako zásadní	Potřebné zlepšit u společností
IT zabezpečení	63%	42%
Správa dat	61%	38%
Networking	59%	37%
Analýza dat	58%	36%
Správa a podpora zařízení	55%	61%
Cloud computing	51%	36%
Umělá inteligence	36%	40%

Obr. 1.2 Porovnání, jaké druhy dovedností jsou u společností nejvíce potřebné v IoT, překresleno z [1]

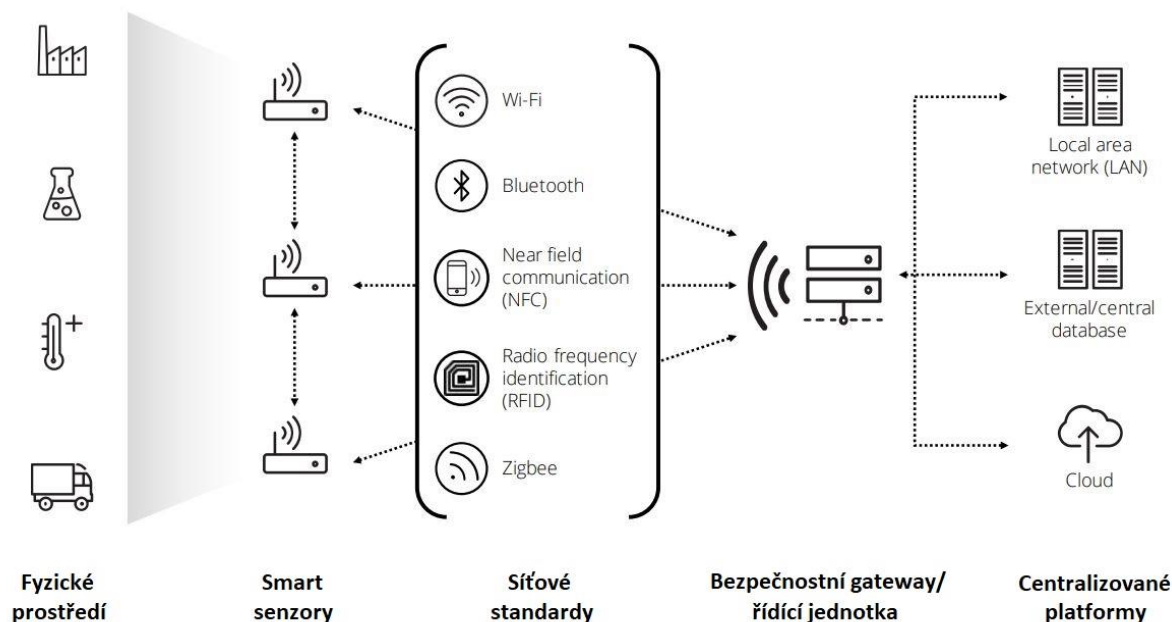
1.2 Smart senzory

Smart senzor je podle [2] definován jako kombinace senzoru, mikroprocesoru a komunikační technologie, jež převádí vstupní veličiny z prostředí (např. vlhkost, váhu, teplotu) na data ke čtení a přenáší je do centralizovaného repozitáře. Tradiční senzory jsou přednastaveny výrobcem, aby prováděli konkrétní operace a zpracovávali určitá data. Smart senzory jsou pak vhodnější ke komplexnějším aplikacím.

Ekosystém smart senzoru

Ze studie [3] vychází následující popis ekosystému smart senzoru, neboli co se všechno děje s odebíranými daty z fyzického prostředí. Lépe pochopitelné je to z obrázku Obr. 1.3. Tyto informace vstupují do smart senzorů, kde jsou převedena do digitální podoby a dále

zpracovávána. Následně jsou prostřednictvím přenosových standardů (např. Wi-Fi, Bluetooth, NFC, RFID) přenesena dalším sensorům či řídicím jednotkám. Z nich to nakonec putuje do databází a do platforem určených pro další zpracování či analýzu.



Obr. 1.3 Znárodnění ekosystému pro přenos dat ze smart senzoru, překresleno z [3]

1.3 Platformy Arduino a Raspberry Pi

V současnosti se vývoji aplikací IoT nejčastěji setkáváme se dvěma typy platforem, Arduino a Raspberry Pi. Základní rozdíl je, že Arduino je mikrokontrolér na rozdíl od Raspberry Pi, které je v podstatě verze mini PC s operačním systémem [4].

Raspberry Pi

Raspberry Pi je podle výrobce [5] počítač o velikosti platební karty, ke kterému vám k standardní obsluze stačí připojit obrazovku a myš s klávesnicí. Pracovat s ním je možné stejně jako u stolního počítače, takže nepoznáte rozdíly při používání internetového prohlížeče, přehrávání videí, vytváření tabulek a dokumentů nebo i hraní her. Zároveň je to přístupné pro začátečníky v programování.

Jak uvádí [6] Raspberry Pi bylo totiž vyvinuto nadací The Raspberry Pi Foundation se záměrem poskytnutí velmi levných počítačů a software přístupného zdarma. Cílené je to totiž pro studenty za účelem rozšíření vzdělávání v počítačové informatice. Nízké ceny je dosaženo tím, že kromě absence periférií nebude v zakoupeném balení ani ochranná krabička, čili se zde nachází pouze vývojová deska.

1.4 Rozšíření pro Raspberry Pi a Arduino

Pro Raspberry Pi i Arduino existují i rozšíření v podobě rozšiřujících desek, dodatečných modulů či případně celých setů, obsahujících větší počet různých modulů od konkrétního výrobce. Na výběr nabízí několik oficiálních modulů i samotný výrobce Raspberry Pi.

Camera Module V2

Kamerový modul je kompatibilní se všemi modely Raspberry Pi a může být použit pro pořizování videa ve vysokém rozlišení a to buď 1080p při 30 FPS (snímčích za sekundu) nebo 720p při 60 FPS. Tento modul je populární zejména v prostředí domácího zabezpečení či jako fotopast pro zachycení pohybu zvěře. [7]

Pi NoIR Camera V2

Pi NoIR (zkráceně pro No Infrared) kamera nabízí všechno, co obyčejný kamerový modul výše, ovšem s jednou odlišností, nemá totiž infračervený filtr. Využít se dá ke sledování noční zvěře nebo po aplikaci přiloženého speciálního gelu i kontrole zdraví zelených rostlin. [8]

Raspberry Pi PoE HAT

Raspberry Pi PoE HAT je deska, která je určena pouze jen pro modely Raspberry Pi 3B+ a Raspberry Pi 4B. PoE je zkratka pro „Power over Ethernet“ a znamená to, že to poskytuje napájení pro Raspberry Pi skrze Ethernet síť prostřednictvím Ethernet kabelu. Také to obsahuje malý větráček, jenž je ovladatelný přes I2C a zapíná se automaticky v případě vysoké teploty procesoru. [9]

Raspberry Pi Sense HAT

Přídavná deska Sense HAT má na sobě 8x8 RGB LED matici, pět tlačítkový joystick, magnetometr, akcelerometr, gyroskop a senzory pro měření teploty, barometrického tlaku a vlhkosti [10]. Původně to bylo vyrobené pro výukový program Astro Pi, během něhož studenti pro Raspberry Pi s touto deskou vyvinuli software a to bylo pak v roce 2015 vysláno na mezinárodní vesmírnou stanici ISS, kde to pod speciálním dohledem sbíralo data [11]. Byly pro to navíc od výrobce vytvořeny i snadno přístupné knihovny, zjednodušující přístup pro využití všech možností desky [12].

Raspberry Pi TV HAT

Raspberry Pi TV HAT má využití jako televizní tuner. Podporuje jak starý standard DVB-T, tak i nový postupně se rozšiřující DVB-T2. Po instalaci vhodného programu jako například Tvheadend lze přehrávat a nahrávat digitální televizi a rádia. HD kanály dokážou ovšem

přebrát jen modely s podporou kodeku HEVC/H.265, v současnosti je to pouze Raspberry Pi 4 model B. [13]

Raspberry Pi Touch Display

Jedná se o obrazovku se sedmipalcovou úhlopříčkou a rozlišením 800x480. Displej je multidotykový a v tomto případě rozpozná v jednom momentu až 10 dotyků zároveň, kvůli tomu je ovšem potřeba doinstalovat potřebné ovladače. Tyto drivery však zpřístupní ještě navíc další funkce jako je zobrazení virtuální klávesnice na obrazovce, přístupnou v případě, že uživatel nedisponuje fyzickou klávesnicí a používá operační systém Raspbian. [14]

UniPi 1.1

UniPi 1.1 zpřístupňuje možnost pro Raspberry Pi fungovat jako PLC (programovatelný logický kontrolér a využít se dá například k ovládání automatizačních systémů. Digitální vstupy slouží ke čtení logických stavů danými úrovní stejnosměrného napětí a využívají se pro dvoustavové senzory a zařízení. Pro spínání zařízení s vysokým proudovým odběrem jsou určeny reléové výstupy a používají se pro ovládání žárovek termoelektrických hlavic či bojlerů. Analogové vstupy umožňují měření stejnosměrného napětí v rozmezí 0 až 10 voltů z analogových senzorů. Analogovým výstupem pak ovládáme stejnosměrným napětím externí zařízení, u nichž se je potřeba plynulé řízení. Zároveň deska disponuje porty UART a I2C pro sériovou komunikaci a také sběrnici 1-Wire pro sběr dat ze stejnojmenných senzorů. [15]

DockerPi IoT Node(A)

Tato deska přidá Raspberry Pi možnost komunikace s LoRa, GSM/GPRS, GPS/BDS přes rozhraní I2C. Je vhodná pro IoT aplikace dále i díky tomu, že má velmi nízkou spotřebu (zejména v klidovém stavu) a je také kompatibilní s řadou dalších mini PC. Pro využití mobilní sítě vyžaduje SIM kartu standardní velikosti. [16]

BPiDuino UNO Board Module

Tento modul zprostředkuje propojení s platformou Arduino Uno včetně jejích dalších rozšíření. Deska je osazena čipem ATmega328, takže je modul kompatibilní s Arduino Uno kódem a periferiemi. Typická aplikace je pro tento produkt u automatizace, kontrola senzorů, 3D tisk, kontrola laserového řezání či ovládání pro inteligentní domácnost. [17]

3D Gesture & Tracking Shield

Toto rozšíření umožní rozpoznávání 3D gest a sledování pohybu. Zachycuje to informace o pozici v kartézských souřadnicích na osách x,y,z a zároveň to zaznamenává i přiblížení, dotyk

nebo dvojité poklepání na povrch. Tohoto dosahuje díky rozlišení změny magnetického pole. [18]

GrovePi+

GrovePi je přídatná deska pro Raspberry Pi, jež podle výrobce [19] nabízí podporu pro více než stovku levných modulů. Je to provedeno prostřednictvím plug-and-play, čili není třeba dalších fyzických úprav, stačí jen zařízení připojit kabelem s 4pinovým konektorem k desce. Tyto moduly komunikují přes sběrnici I2C s mikrokontrolérem ATMEGA328 na desce GrovePi, která slouží jako prostředník při práci s RaspberryPi.

Postup pro zapojení a instalaci popisuje oficiální návod, přibalený k zakoupenému GrovePi nebo také volně přístupný na Internetu [20]. Pro instalaci je doporučen mít na RaspberryPi operační systém Raspbian nebo výrobcem upravenou variantu Raspian for Robots, jež obsahuje veškerý potřebný software. Po osobních zkušenostech druhou zmíněnou možnost však nemohu doporučit. Bylo zde totiž mnoho nefunkčních systémových modulů a knihoven a některé z problémů se nezdálo být možné opravit. Za zmínku stojí hlavně nefunkční prostředky pro používání sběrnice I2C. To je s podivem, protože ta je právě potřeba ke komunikaci, jak již bylo výše zmíněno.

V manuálu je pro instalaci software na RaspberryPi sepsána série příkazů, určených k vkládání do příkazového řádku. Znovu jsem ovšem po dokončení narazil na další problémy, některé moduly byly nefunkční, chyběly důležité knihovny, a firmware dokonce ani nešel aktualizovat. Nejlepší řešením je v příkazovém řádku použít jediný příkaz: „sudo curl -kL dexterindustries.com/update_grovepi | bash“. Celý proces instalace proběhne bez problémů a vše je poté plně funkční. Nejvíce zajímavé je na tom to, že tohle pochází z postupu od výrobce [21], kde jsou navíc i rady k ověření, zda nechybí žádné součásti a jestli zařízení pracuje správně.

Předpokládá se zde totiž, že každý nemusí desku aplikovat se stejným záměrem. To je však podpořeno tím, že jsou k dispozici v prodeji i dodatečné moduly [22].

1.5 Význam sběru dat

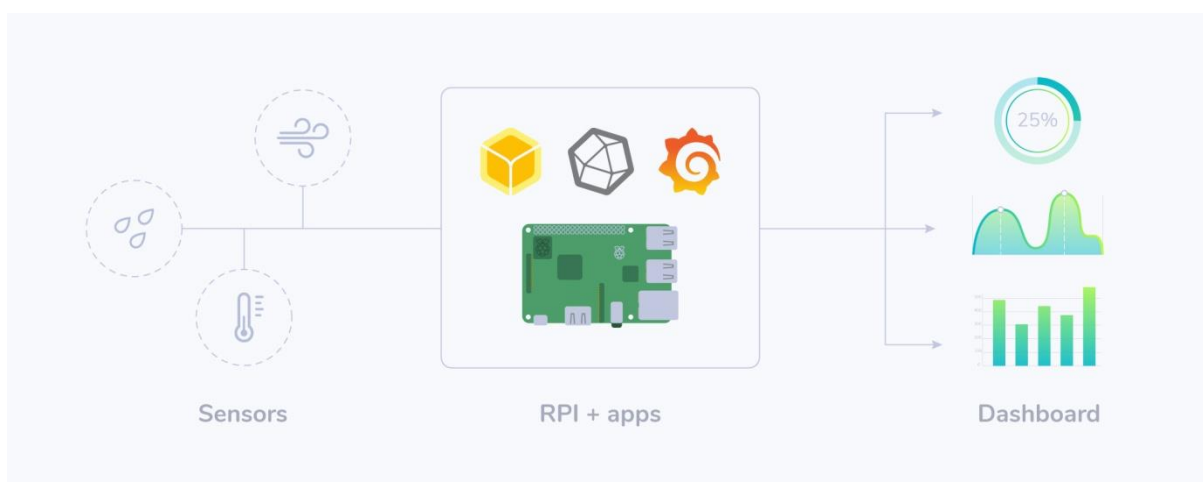
Sběr dat usnadňuje velká řada smart senzorů a modulů, ale před jejich výběrem je zásadní si ujasnit, jaký bude hlavní cíl použití. Všechny komponenty jsou totiž sice levné, ale v součtu to pro některé jedince nebude nezanedbatelná částka. Možnosti aplikace platformy Raspberry Pi se nabízejí v mnoha oblastech. Podle rozsáhlosti provedení lze výsledné systémy řadit do dvou skupin. První z nich je domácí použití, jež bývá v menším měřítku a hlavně se do něj investuje menší částka než pro druhou skupinu, použití v průmyslu. Tam je Raspberry Pi

kupováno s cílem šetřit zdroje (materiálové i lidské) a někdy se tam počítá i s brzkou návratností investic. Níže je vybráno několik příkladů využití v obou skupinách.

1.5.1 Domácí aplikace

Meteostanice

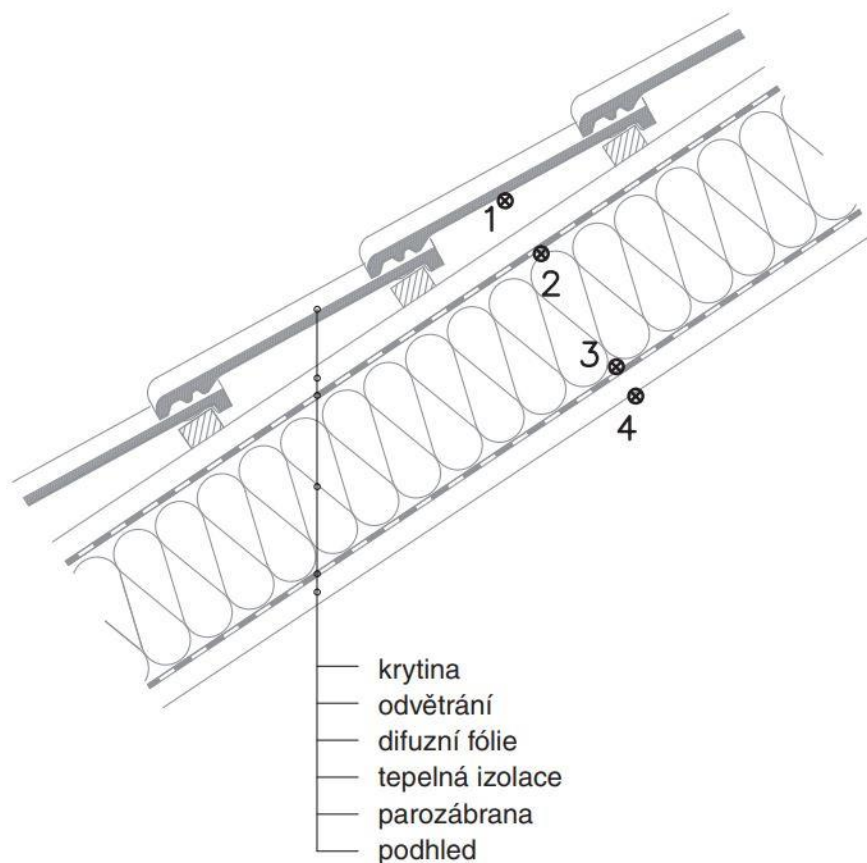
Mezi jednodušší aplikace v domácím prostředí je zařízení meteostanice. V případě projektu [23] pro to vybrali senzor, který samotný zvládá sledovat teplotu, vlhkost, barometrický tlak a kvalitu vzduchu. Tento senzor se propojí s Raspberry Pi, aniž by bylo potřeba cokoliv nutně pájet. Údaje pak putují do databáze a zobrazují se v grafickém prostředí (v tomto případě Grafana), toto bývá značeno jako Dashboard. Celý tento znázorněný proces, jež je naznačený na obrázku *Obr. 1.4*, je společný pro značnou část projektů ve stejné oblasti.



Obr. 1.4 Přenos dat ze senzorů meteostanice do grafického prostředí Dashboard, převzato z [23]

Sběr teplotních údajů u domovních střešních pláštů

Mezi další projekty patří sbírání teplotních údajů střešních pláštů domu [24]. Na obrázku *Obr. 1.5* schematickeho průřezu lze vidět, že tam byly zavedeny čtyři snímače teploty. Takovéto rozmístění je z důvodu zjištění teploty pro střešní pláště různé skladby. Snímaly se tím klimatické podmínky a záměr bylo ověřit, zda jim odpovídají hodnoty matematicky vypočtené. Ty jsou také tímto způsobem stále aktuální včetně následně dalších dopočtených informací, jako je teplotní útlum, tepelné toky a fázový posun teplotního kmitu.



Obr. 1.5 Umístění senzorů u domovních střešních pláštů, převzato z [24]

1.5.2 Průmyslové aplikace Raspberry Pi

System pro řízení mikroklimatu ve stájích

Zajímavým uplatněním je kontrola mikroklimatu v prostředí stáje, tady se jedná konkrétně o chov skotu [25]. Cílem je tu nejen zlepšení životních podmínek pro zmíněná zvířata, ale je s tím dále i spojená produkce mléka, u které by se v určitém rozsahu měla zvýšit jednak kvalita i množství. Opět mezi hlavní sledované veličiny patří teplota, vlhkost, atmosférický tlak, a zde navíc také osvětlení. V grafickém prostředí počítačového programu na obrázku *Obr. 1.6* jsou viditelné dané informace a podle toho ihned nevhodné podmínky regulovat.



Obr. 1.6 Grafické prostředí počítačového programu pro řízení mikroklimatu, převzato z [25]










Sběr dat z CISS senzorů

V tomto případě [26] se pracovalo se senzory CISS ve firmě CISS. Data z nich se přenáší prostřednictvím Bluetooth Low Energy a bylo vyžadováno, aby byla požadovaná data v konkrétních intervalech sbíraná, ukládaná a zpětně dostupná prostřednictvím zobrazení jejich historie. Podobně jako předchozí příklad, i zde se zobrazují v uživatelském prostředí aktuální hodnoty. Mezi zmíněné příklady využití těchto senzorů jsou pak uvedeny následující aplikace jako například měření teploty chladicího oleje, sledování vibrací u strojů a sledování teploty v rozvaděčích.

1.6 Rozdělení míst použití sběru dat

IoT senzorické systémy se dle [27] dají uplatnit v mnoha oblastech, jak je zřetelné u tabulky, kde jsou místa uplatnění rozdělena do devíti skupin, detailněji rozebraných v tabulce *Tab. 1.1*. Tyto skupiny jsou významné z toho důvodu, že v nich aplikací IoT senzorů lze dosáhnout největších zlepšení. Nejčastěji jsou společné vlastnosti zvýšení bezpečnosti lidí, automatizace práce a snížení spotřeby používaných prostředků. Zejména poslední zmíněná výhoda je spojena s (v dokumentu podrobněji rozvedenou) ekonomickou úsporou, jež je důvodem, proč jsou do těchto technologií firmy ochotny investovat.

Tab. 1.1 Místa uplatnění sběru IoT dat pomocí senzorů, převzato a přeloženo z [27]

Místo použití	Popis místa	Příklady	
	Člověk	Zařízení uvnitř nebo vně lidského těla	Zařízení pro monitorování lidského zdraví nebo použití v oblasti fitness. Nositelné nebo vložené do těla.
	Domov	Dlouhodobě obývaná obydlí	Automatizace domácnosti a domácí zabezpečení.
	Kamenné pobočky	V komerčních prodejnách	V obchodech, bankách, restauracích, kdekoliv kde zákazníci kupují produkty. Příkladem jsou samoobslužné pokladny.
	Kanceláře	Pracovní prostředí pro zaměstnance	Úspora energií a zabezpečení pracoviště. Zvyšování produktivity, včetně zaměstnanců pracujících z domova.
	Továrny	Prostředí pro průmyslovou výrobu	Místa s časově se opakující pravidelnou pracovní rutinou. Lze sem zařazovat i nemocnice i farmy. Slouží pro optimalizaci zdrojů a vybavení.
	Pracoviště	Použití při stavbě, těžbě	Údržba zařízení, bezpečnost pracovníků, usnadnění činností jako kopání, těžby, získávání prvků jako je olej či zemní plyn.
	Dopravní prostředky	Systémy uvnitř dopravních prostředků	Vozidla zahrnující automobily, kamiony, lodě, letadla, vlaky. Používané pro snižování spotřeby paliva.
	Města	Součást městského prostředí	Adaptivní semaforey, snížení spotřeby energií, monitorování okolního prostředí.
	Venkovní prostředí	Venkovní umístění mimo městské prostředí	Monitorování leteckého provozu, vlakového provozu, jejich navigace. Sledování zásilky přepravní společností.

Dalším důležitým poznatkem v analýze [27] je, že většina informací ze sběru IoT dat není vůbec využita. Příklad uvádí na systému pro těžbu oleje, z jehož činnosti generuje data 30 000 senzorů, použito je z nich ovšem pouze 1 % dat. Zmíněná data jsou používána pro kontrolu v reálném čase a detekci neobvyklých anomálií. Ze senzorů by se v tomto případě

však dalo získat mnohem více užitečných informací a rozšířit možnosti uplatnění jako je prediktivní údržba nebo změna určitých částí postupu, aby bylo dosaženo vyšší efektivity. Naopak se ale poté musí řešit, jak takové množství dat zpracovávat.

2 Možnosti řešení pro IoT uzly

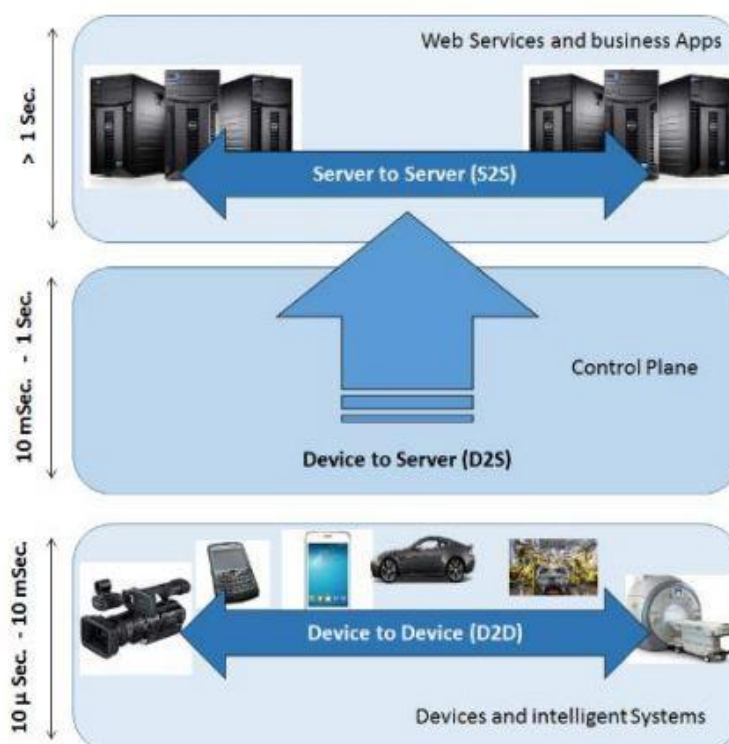
2.1 IoT protokoly

IoT protokoly popisují vlastnosti a pravidla týkající se přenosu zpráv v digitální podobě, to zahrnuje velikost paketů, rychlost přenosu či způsoby kontroly přenosu a synchronizace [28].

Tyto protokoly se podle [28] dělí do tří kategorií v následujícím rozdělení:

- *Device to Device (D2D)*
- *Device to Server (D2S)*
- *Server to Server (S2S)*

Jejich chování popsané ve [29] je lépe srozumitelné na grafickém znázornění na obrázku *Obr. 2.1*. U D2D spolu dokáží komunikovat pouze koncová zařízení, ale dochází k tomu s okamžitou reakcí v rámci desítek mikrosekund až milisekund. K přenosu dat ze zařízení do serveru už je potřeba D2S a odezva se stále očekává v řádech milisekund. S2S je pak prostředkem pro samotné servery pro webové služby a nástroje, počínaje s rychlostí začínajících na jednotkách sekund.



Obr. 2.1 Kategorie IoT protokolů s dobou jejich odezvy, převzato z [29]

2.1.1 Protokoly D2D

DDS RTPS

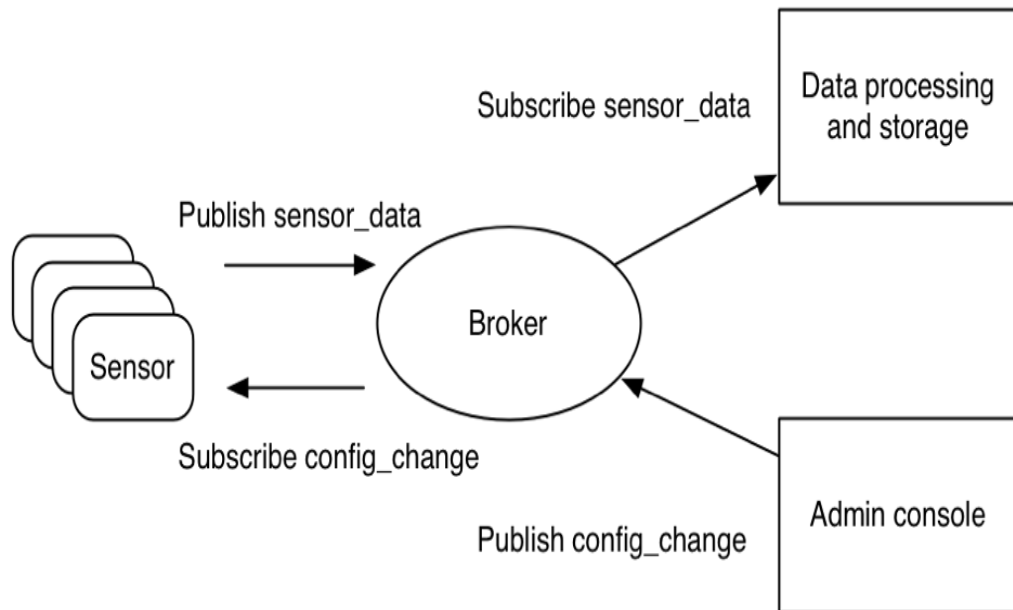
DDS je standard specifikovaný společností Object Management Group (OMG). Zabývá se požadavky distribuovaných systémů, které pracují v reálném čase. Jako systém komunikace to používá DCPS postavený na principu rozšiřování užitečných informací i k zařízením, jež o ně nemají zájem. Místo centrálního řídicího prvku zde působí Global Data Space (GDS). GDS je prostor, do kterého přispívají a ze kterého získávají data zařízení, která jsou jeho součástí. [30]

Real-Time Publish-Subscribe je protokol pro implementování datově řízených v reálném čase distribuovaných aplikací. Smyslem RTPS protokolu je zajistit spolupráci u aplikací založených na DDS API. Vyskytují se implementace komerční i svobodné a je zaváděn hlavně u průmyslových aplikací. Jedna ze zmíněných svobodných implementací je například multiplatformní Open Real-Time Ethernet (ORTE). [31]

2.1.2 Protokoly D2S

MQTT

Následující popis MQTT protokolu vychází z [32]. MQTT protokol je definován dvěma typy objektů v síti, a tedy počtem klientů a Brokerem. Broker je server přijímající zprávy od klientů, jež následně doručí jiným cílovým klientům. Klientem je označováno cokoliv, co zvládne interakci s Brokerem, například IoT senzor nebo aplikace v datovém centru zpracovávající IoT data. Zprávy jsou v MQTT jsou označovány tématy (topics) a klient může nastavit požadavek (subscribe), aby od ostatních klientů obdržel (publish) zprávy požadovaného tématu. Pro případ přenosu soukromých zpráv je možné nastavit zašifrovanou komunikaci. Názorný příklad je uveden na obrázku *Obr. 2.2*. Sensory vyžadují (subscribe) téma „config_change“ a poskytují (publish) zprávy pod tématem „sensor_data“, o které má zájem úložiště se zpracováváním dat. Administrátorská konzole pak získává data „config_change“, která obsahují konfigurační údaje, u nichž kontroluje případné změny.



Obr. 2.2 Komunikace v MQTT vedená prostřednictvím Brokeru, převzato z [32]

Další popis je čerpaný ze článku o protokolu MQTT [33]. U MQTT nejsou u samotného obsahu zprávy (payload) žádné konkrétní požadavky a jedná se pouze o binární data. Omezená je však velikost zprávy na necelých 256 MB. Nejčastěji je zde využíván JSON, BSON nebo textová zpráva. Dále se tu setkáváme s úrovněmi Quality of Service (QoS), jež souvisí s potvrzováním zpráv. U nejnižší úrovně dojde k poslání zprávy bez potvrzení, aniž bychom měli jistotu, že bude doručena. Další úroveň má zajistit, že se zpráva doručí minimálně jednou a u nejvyšší právě jednou. Podpora jednotlivých úrovní QoS se u klienta ovšem může lišit.

CoAP

CoAP pro komunikaci koncových zařízení podobně jako HTTP používá model žádost a odpověď a metody GET, PUT, POST a DELETE. Nevychází však z protokolu TCP, ale je založen na předávání datagramů pomocí protokolu UDP. Binární hlavička zprávy má délku byty a mohou za ní následovat dodatečné informace jako je binární možnosti a zátěž. Součástí zprávy je její ID, aby se lokalizovaly duplikáty a zvýšila se spolehlivost. To je provedeno tím způsobem, že zpráva označená jako potvrzitelná (CON). Tento typ zprávy je odeslán s předem danými intervaly opakovaně, dokud není potvrzeno příjemcem, že byla doručena. To proběhne tak, že zpět odešle potvrzovací (ACK) zprávu se stejným ID. Pokud dojde k tomu, že příjemce nedokáže zpracovat příchozí zprávu, odešle zpět resetovou zprávu (RST). [34]

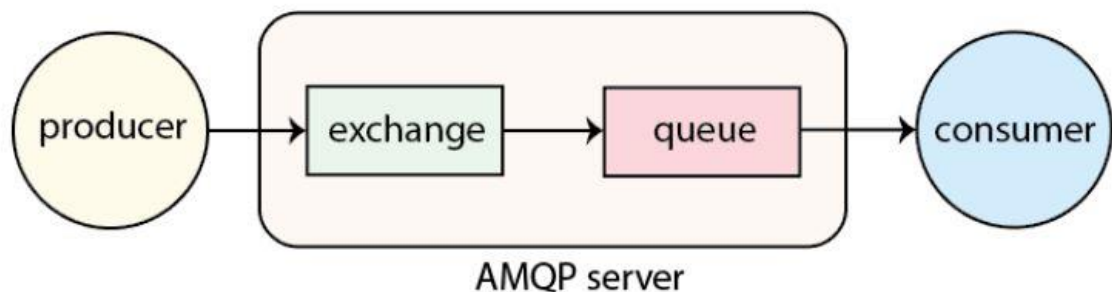
XMPP

Protokol XMPP má architekturu typu klient-server a je otevřený, takže je zdarma volně přístupný. Komunikace klientů neprobíhá přímo, ale decentralizovaně skrze server. Přesněji vysvětleno tu není centrální server, ale uživatel zde má možnost zvolit preferovaný (a případně i vlastní) server. V síti je nutné, aby měli uživatelé jednoznačnou identifikaci označovanou jako Jabber ID, která obsahuje uživatelské jméno, za nímž následuje název serveru, kde došlo k založení účtu. Tyto dva identifikační prvky se oddělují zavináčem ve stylu: „user@server.com“. [35]

2.1.3 Protokoly S2S

AMQP

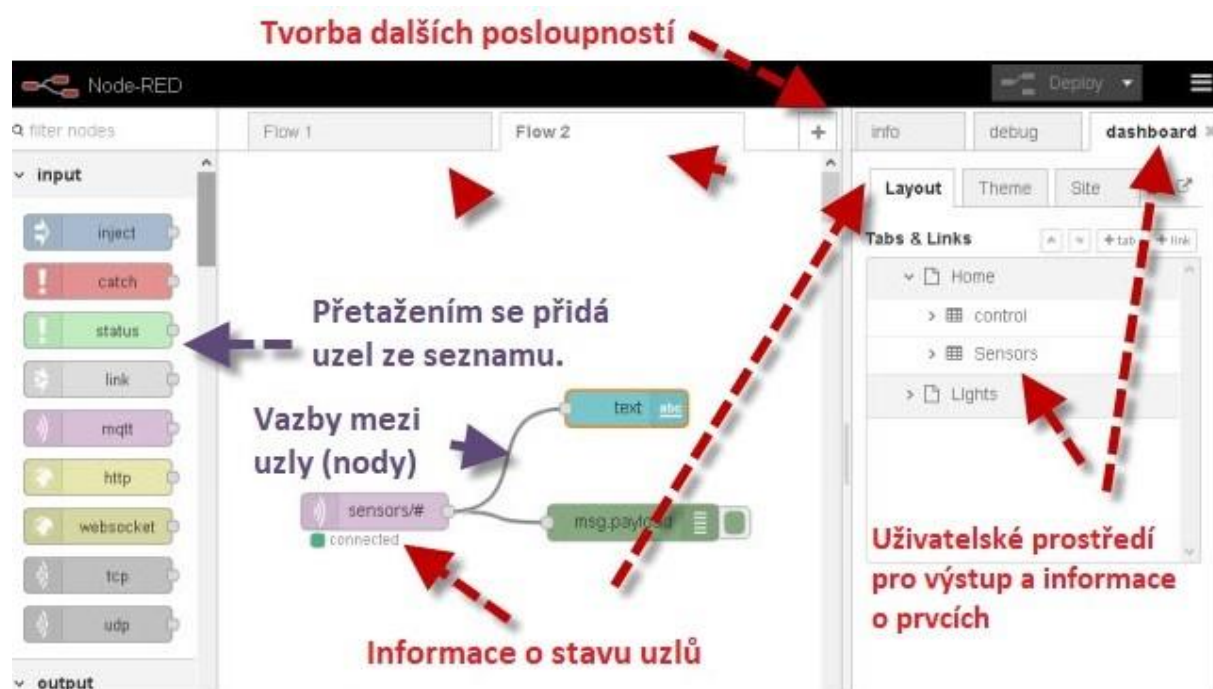
Protokol AMQP je postavený na přenosu zpráv. Funguje na principu na Obr. 2.3, kde publisher je odesílatelem zprávy, kterou consumer přijímá. Při odesílání publisher neví, zda je v síti consumer, jež by zprávu přijal a consumer zároveň netuší, jaký je původ zprávy a ani kolik je publisherů. AMQP umožňuje připojení k serveru a definování exchange, kam jsou zprávy odesílány a rozdělovány do front dle daných pravidel. U jednoho exchange neexistuje omezení týkající se počtu publisherů a consumerů. Producerem je při odesílání zprávy kromě určení exchange také definován routing key, pomocí kterého exchange určí, do jaké fronty zprávu předá. Jsou 3 druhy exchange a odlišují se tím, jak nakládají se zprávami. Exchange typu default přiřadí každé frontě routing key se stejným jménem jako má fronta. Direct je druhým typem, jenž už využívá specifický routing key a zpráva dojde do fronty pouze tehdy, je-li shodný routing key zprávy i fronty. Fanout je třetím druhem ignorujícím routing key a posílajícím zprávy do všech dostupných front. Pokud u fanout dorazí zpráva do exchange, tak je zkopírována a všechny fronty obdrží jednu kopii. [36]



Obr. 2.3 Přenos zprávy od producera ke consumerovi, převzato z [36]

2.2 Vývojové prostředí Node-RED

Jak je uvedeno v [37], Node-RED je vývojové prostředí pocházející od firmy IBM, nejčastěji se používající v oblasti IoT. Node-RED bylo už na počátku vydáno v roce 2013 jako open-source projekt. I kvůli tomu se utvořila velká komunita vývojářů, sdílející své výsledky i s postupy, a také diskutující své pokusy. Programování je zde založeno na flow modelu, kde se v grafickém prostředí spojují bloky předdefinovaného kódu (nody) pomocí vazeb určujících posloupnost jejich vykonávání (flow). Pokročilejší uživatelé mají navíc možnost vytvořit si vlastní nody s kódem v programovacím jazyce JavaScript. Všechny zmíněné věci jsou viditelné na obrázku *Obr. 2.4*.



Obr. 2.4 Uživatelské prostředí u Node-RED, překresleno z [38]

Přístup k Node-RED má uživatel přes webový prohlížeč. Jak popisuje článek [39], k přenosu zpráv bývá ve spojení s Node-RED používán protokol MQTT, jež je vhodný i pro komunikaci s Cloudem. Opět se zde využívá flow modelu k dosažení přijímání a odesílání požadovaných dat. Podpory se to v této oblasti dostalo například i od firem jako Amazon či Microsoft.

Podle [40] je Node-RED programovací softwarový nástroj pro navázání ethernetové komunikace a ovládání s různými IoT zařízeními. Na jedné straně umožňuje realizovat vyžádání a přijetí JSON, XML či emailových zpráv a následně "dekódovat" pro grafické zobrazení na vizualizační obrazovce. Na druhé straně pak realizovat odesílání JSON či XML nebo e-mailových zpráv buď časově automaticky nebo na stisk ikony tlačítka na vizualizační

obrazovce. Software Node-RED je k dispozici primárně určen pro populární IoT destičky Raspberry Pi a jejich OS Linux Raspbian, ale stejně dobře jej lze provozovat prakticky na jakémkoliv běžném počítači s OS Linux nebo případně i OS Windows.

JSON

JSON (zkratka pro JavaScript Object Notation) popisuje [41] jako textový formát sloužící pro uložení a výměnu hierarchicky strukturovaných dat. Stal se alternativou pro formát XML, oproti kterému působí jako odlehčenější varianta. Už podle názvu vychází z programovacího jazyka JavaScript, ale dá se použít i v jiných jazycích.

Další popisované vlastnosti jsou ze článku [40]. Pro přijímání a čtení zpráv v JSON je dostatečný pouhý webový prohlížeč, pro odesílání zpráv pomocí metody POST je však nezbytné použít specifický klient. Ovládání je možné si zjednodušit implementací do některého z řídicích vizualizačních software. Jedním z jednodušších nástrojů pro tuto aplikaci je právě Node-RED.

3 Návrh a realizace řešení zadaného IoT systému

3.1 Monitoring průjezdu dopravy

Společnost CITIQ nabízí [42] v této kapitole rozebíranou možnost monitorování průjezdu dopravy pomocí geomagnetického detektoru, který bývá umístěn pod povrchem vozovky, například v dlažební kostce. Ve výsledku se dá měřit nejen kolik projelo aut, ale i jak rychle. Z toho pak lze určit třeba délku dopravních kolon a podle toho reagovat uzavírkami či změnou intervalů na semaforech.

Na obrázku *Obr. 3.1* je fotografie se získanými informacemi ze systému v Praze v ulici Dvořákovo nábřeží [43]. Během dne 1. července roku 2017 zde bylo detekováno 20278 vozidel, přičemž 204 z nich překročilo rychlost 90 km/h. Z toho je tak možné i určit, kolik vozidel a v jakou dobu překročilo povolenou rychlost.

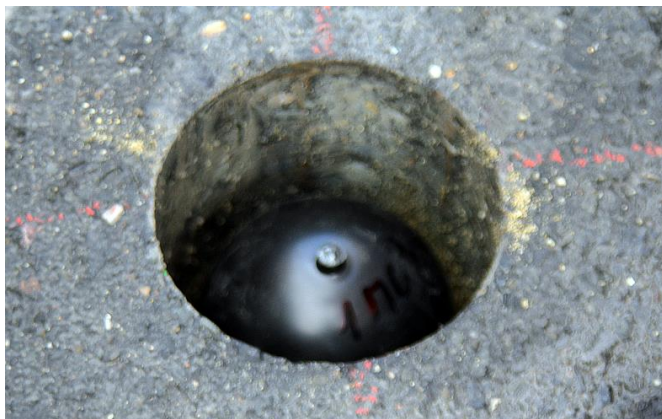


Obr. 3.1 Informace o projíždějících autech získané systémem, převzato z [43]

Realizace systému detektoru vozidel

Následující postup vycházející z dokumentu od firmy CITIQ [44] začíná vyvrtáním válcové díry specifických rozměrů do určené části vozovky, buď v místě průjezdu aut nebo na parkovacím místě. Po vložení magnetického detektoru do díry jak je vidět na *Obr. 3.2*, je otvor zalit asfaltovou směsí tak, že upravený povrch nepůsobí rušivým dojmem. Napájení datového kolektoru může být zajištěno ze sítě, prostřednictvím solárního panelu nebo z baterie. Průjezd vozidla zaregistruje magnetický detektor díky způsobené změně magnetického pole. Zároveň je speciálním algoritmem ošetřeno zjištění toho, jestli se skutečně jednalo o dopravní vozidlo. Tyto informace detektor rádiově předává kolektoru, jenž bývá obvykle na sloupu či dopravní značce. Datový kolektor následně obdržená data odesílá

na centrální web, kde dochází k dodatečnému zpracování, jež pak umožní práci s informacemi a například vystavit údaje o obsazenosti na webové stránky.



Obr. 3.2 Detektor vložený do vyvrtané díry ve vozovce, převzato z [45]

3.2 Přetíženost silnic v okolí Škody Auto v Kvasinách pomáhá řešit CITIQ, člen IQRF Alliance

Jednu z aplikací firmy CITIQ popisuje i článek [45] z webových stránek zaměřených na inovace v oblasti Smart Cities. V okolí Škoda Auto v Kvasinách nejsou dostatečné parkovací prostory pro více než 8000 pracovníků. Systém detektorů byl tedy zaveden s úmyslem monitorování průjezdu, délku a rychlost dopravních vozidel v okolních vesnicích. V nich totiž dochází kvůli tomuto problému s nadměrným množstvím projíždějících aut, ale právě také k nežádoucímu parkování. V tomto případě bylo provedeno napájení pomocí baterií. U nich se předpokládá životnost minimálně 10 let, protože přenos dat je prostřednictvím technologie IQRF s nízkým odběrem energie. Data jsou uchovávána v Cloudu, kam se dá přistupovat i ke starším údajům. Ulehčení dopravy na parkovištích je pak dosaženo informační cedulí na Obr. 3.3, na níž je zřetelné množství obsazených míst.



Obr. 3.3 Informační cedule zobrazující počet volných parkovacích míst, převzato z [45]

3.3 Technická specifikace

Magnetický detektor

Níže následuje z webových stránek firmy CITIQ [46] přímo převzatá podrobná specifikace pořízených magnetických detektorů. Instalace byla provedena do vozovky, ale na výběr je i povrchová varianta. Technické údaje jsou:

„Magnetický detektor DM-21xxx slouží k monitoringu parkování nebo dopravy (počet vozidel, délka a rychlost vozidel). Detektor je určen pro měření přítomnosti vozidla.

- Princip detekce: kontinuální měření geomagnetického pole
- Napájení: baterie, lithium-thionyl chlorid (Li-SOCl₂)
- Komunikace: RF 868/916 Mhz (všeobecné oprávnění č. VO-R/10/09.2010-11),
- LoRA, SIGFOX
- Provozní teplota: -40 °C / +85 °C
- Instalace:

a) do vozovky (intruzivní, pomocí jádrového vývrtu, průměr 100 mm), na Obr. 3.4



Obr. 3.4 Magnetické detektory umístované pod povrch vozovky, převzato z [46]

b) povrchová (upevněná na povrch nebo v podobě zámkové dlažby apod), na Obr. 3.5“



Obr. 3.5 Magnetické detektory umístované na povrch vozovky, převzato z [46]

LoRaWAN Gateway

Od firmy CITIQ byly pořízeny pouze detektory bez datových kolektorů. Prostředí ZČU je totiž již pokryto LoRaWAN sítí v rámci projektu SmartCAMPUS ZČU [47]. Od firmy CITIQ byly pořízeny pouze detektory bez datových kolektorů, jelikož je již prostředí ZČU pokryto LoRaWAN sítí. Jedná se o gateway zapojenou do sítě LORATECH. Následuje přímo převzatý technický popis ze stránek [48]:

„LORATECH Outdoor Gateway je venkovní gateway pro profesionální budování IoT (internet věcí) sítí na technologii LoRaWAN – 868MHz. Plně outdoorová jednotka zajistí pokrytí signálem v rádiusu několika kilometrů a s vhodnou anténou lze pokrýt jednou jednotkou menší obec. Veškerá elektronika je osazena v metalovém šasi s krytím IP66. Gateway je založena na koncentrátoru od firmy IMST IC880A, který používá chip firmy Semtech SX1301. Koncentrátor komunikuje přes SPI rozhraní s Raspberry PI3. Jako mezičlánek je použita redukční deska, speciálně vyvinutá firmou RVTech s.r.o. pro použití

GW na přístupových bodech ISP. Každý bod umožňuje připojení až desítkám tisíc věcí. Přes gateway procházejí data šifrovaně a proto je komunikace bezpečná.“

V internetovém obchodě [49] jsou k dispozici 3 varianty:

- LORATECH Outdoor Gateway = Je to venkovní varianta, která je popsána výše, je viditelná na obrázku *Obr. 3.6*.
- LORATECH GW_ALU_4G Outdoor Gateway = Jde o provedení se 4G modemem.
- LORATECH Indoor Gateway = Jedná se o vnitřní variantu



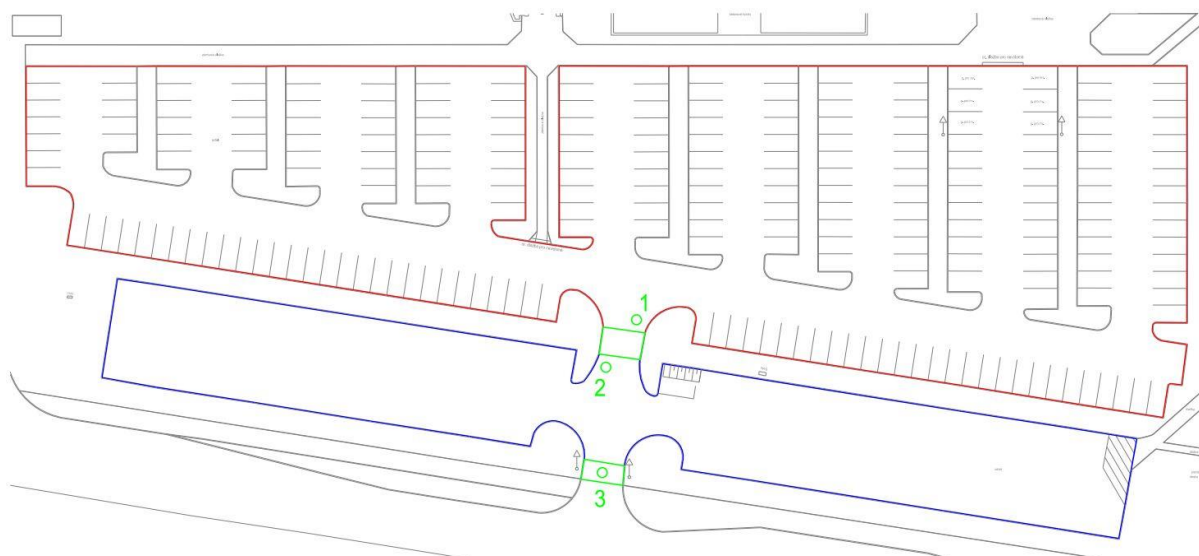
Obr. 3.6 Vzhled LORATECH Outdoor Gateway, převzato z [48]

3.4 Aplikace systému průjezdových senzorů u školního parkoviště

Jedním z míst, kde je zaveden senzorový systém, je parkoviště Fakulty elektrotechnické Západočeské univerzity v Plzni. Provozní doba je zde uváděna v čase od 5:30 do 22:30 a je určena podle toho, kdy se mohou otevírat a zavírat vjezdové závory brány parkoviště a mimo tento časový interval tak může uživatel z parkoviště pouze vyjždět [50]. Celkem je na parkovišti 234 míst k parkování, z nichž je 218 parkovacích míst vyhrazeno pro osobní automobily, 8 parkovacích míst je určeno pro motocykly a 10 míst k parkování je vymezeno pro držitele průkazu ZTP [50]. Parkoviště je v současnosti rozděleno na dvě části, jak je zřetelné na obrázku *Obr. 3.7* níže. Do veřejné části, na obrázku vyznačené modře, má volný přístup kdokoliv. Do neveřejného prostoru, na obrázku červeně zvýrazněné oblasti mohou projet pouze studenti a zaměstnanci univerzity.

Popis funkce systému v tomto odstavci vychází z technické zprávy o závorách na parkovištích ZČU [51]. Přístup do neveřejné části parkoviště je řešen tím způsobem, že jsou na místě

přítomné závory, jež vpustí dovnitř auto v případě, že se u kontaktního RFID čidla řidič identifikuje přiložením příslušné univerzitní karty JIS ZČU. Uživatel se v ten moment navíc také nachází na detekční smyčce umístěné ve vozovce a závora se otevře pouze za splnění obou podmínek, tedy že auto stojí na smyčce a je zároveň ověřena karta JIS ZČU. Závora se poté opět zavře, až když není na místě zaznamenáno auto. To, že řidič bezpečně opustil prostor, je zajištěno tím, že se pod osou ramene závory nachází bezpečnostní smyčka. Nehrozí tedy, že by došlo k zavření závory, pod kterou by byl automobil. Identifikace kartou platí pouze ve směru pro vjezd, u opačného směru je zvednuta závora, jakmile je detekční smyčkou zaregistrováno vyjíždějící auto.



Obr. 3.7 - Návrh na umístění tří nových průjezdových senzorů na parkovišti FEL ZČU

Sice je zde již přítomen dříve instalovaný systém senzorů, u něj jsou však zjištěny problémy způsobující nepřesnost údajů. K tomu dochází z toho důvodu, že není některými návštěvníky dodržováno zamýšlené používání. Jako první příklad lze uvést, že těsně za vjíždějícím autem rychle projede zároveň ještě druhé auto, ale zaznamenáno je však pouze jedno vozidlo. Podobný je pak případ, kdy po výjezdu automobilu okamžitě vjede na parkoviště další vozidlo v protisměru, to ovšem ale opět není systémem zaregistrováno.

Nově zavedené senzory zaznamenávají mnohem více informací a rozeznávají tak například i jakým směrem se automobil pohybuje, což zamezuje výše zmíněným potížím. Jak je vidět na výše se nacházejícím obrázku Obr. 3.7, jedná se celkem o tři senzory, přičemž první z nich je za vjezdovou závorou, druhý za výjezdovou a třetí je umístěn ještě před vstupem do volně přístupné části. První dva senzory fungují jako přesná počítadla průjezdů a poslední je potom určen zejména k sledování úrovně hustoty dopravy. Nepočítá se u něj totiž s tím, že by s ním bylo možné naprosto spolehlivě monitorovat vjezdy a výjezdy vozidel.

Ze začátku realizace bylo potřeba zajistit dokumentaci stávajícího systému závor a inženýrských sítí. Nové senzory totiž nesměli zasahovat do původního systému, případně dalších inženýrských sítí. Všechny prvky z dokumentací byly zaznamenány do jednoho výkresu pomocí počítačového programu AutoCAD. Tento program byl zvolen zejména z toho důvodu, že dokáže z PDF souborů importovat výkresy, které lze následně okamžitě upravovat. Zároveň zde došlo k věrnému převedení bez zkrácení originálu a ztráty údajů (např. chybějící text, číslice nebo kóty). Finální úprava, která byla na závěr exportovaná do formátu PDF, byla poté předána s žádostí na Správu budov ZČU za účelem povolení instalace. Po jejím provedení firmou přišlo na řadu zavedení a nastavení LoRaWAN a integrace do Cloudu s realizací Open Data.

3.5 Open Data

Pokud není uvedeno jinak, tak následující text vychází z příručky „jak otevírat data?“ [52]. Tato konkrétní příručka je vhodná jak pro začátečníka s minimálními znalostmi v oblasti IT, tak pro specialistu v tomto oboru. V úvodu je uveden diagram, jenž čtenáře navede na kapitoly s potřebnými informacemi. Dotyčný tak nepřijde o důležité informace a zkušení jedinci nemusí zbytečně číst kompletně celou příručku.

Standard otevřených dat je zde definován několika požadavky:

K datům lze přistupovat na Internetu a je možné je vyhledat běžnými nástroji a prostředky informačních a komunikačních technologií (zkráceně ICT).

Data jsou čitelná jak pro člověka, tak jsou i strojově čitelná pro počítačový program, jsou tedy ve vhodném formátu. Software pro jejich zpracování (jenž podporuje právě ten formát) by měl být k dispozici volně a bezplatně.

Data jsou kompletní. Jsou tedy zveřejněná v největším možném rozsahu.

K datům je možný přístup a to s nejnižším možným počtem technických nebo legislativních překážek.

Co jsou přesně otevřená data, se dá potom ještě vymezit podle následujících vlastností:

Data mohou pocházet odkudkoliv, zde je hlavně podstatná forma a nikoliv obsah. Může jít o záznamy fyziologického stavu těla (např. údaje získané při sportování), informace vytvářené samosprávou a státní správou nebo data z univerzitního výzkumu.

Data jsou k dispozici online a to v podobě textového dokumentu, geodat nebo v grafickém formátu. Často také bývají v podobě tabulek, kde jako příklad lze uvést jízdní řády, statistiku kriminality nebo městský rozpočet.

Pro všechna otevřená data nejsou jednotné požadavky a liší se podle nároků každého autora. Jediným společným rysem je zde pouze to, že je zde otevřenost definována technickými parametry daného datasetu (datového souboru).

Umístěním otevřených dat na Internet dochází k znásobování jejich hodnoty s každým použitím. Zvyšuje se tak pravděpodobnost, že se najde někdo s novým praktickým nápadem, jak je využít. Předem se nedá předpovědět, kdo bude stát za dalším způsobem použití, ať už to bude firma, výzkumný tým nebo státní instituce. Každopádně tak ale soukromý i veřejný sektor získává možnost daná dostupná otevřená data zhodnotit.

Velmi se doporučuje, co nejdříve sestavit pracovní skupinu obsahující všechny relevantní osoby. Tito účastníci jsou především takoví lidé, kteří jsou v pozici, aby mohli klást co nejvíce požadavků. Tato pracovní skupina musí dále řešit několik zásadních věcí. Po úspěšném získávání dat je podstatné určit, které z dostupných datasetů je vhodné zpřístupnit ke zveřejnění. Zároveň se také může řešit, jaké z datasetů je konkrétní útvar, jako je například vedení nějakého úřadu, ochotný odsouhlasit ke zveřejnění. Nelze taktéž opomenout, jaké mohou být požadavky budoucích uživatelů ohledně poskytovaných dat. Vhodné je případně provést i analýzu souvisejících nákladů a přínosů.

S procesem publikace datasetů se často pojí technické problémy. Nejčastěji jsou to hlavně soubory, jež jsou příliš složité nebo nadměrně překračují požadovanou velikost. Nedostatek úložné kapacity je nutné řešit pořízením vhodného serveru nebo ukládat data na cloudové úložiště.

Součástí konečného vyhodnocení je zjišťování, jak velký je zájem o poskytovaná data, minimálně určování návštěvnosti. Pro tento účel existují i analytické nástroje, jež poskytují mnoho užitečných informací. Dozvědět se tak můžete, zda návštěvníci webovou stránku navštěvují přímo, anebo jestli jsou přeměrováni z jiného odkazu. Když jsou pro uživatele dostupné ke stažení různé druhy datových formátů, tak lze zjistit, který z nich stahují nejčastěji. Je také možné určit, jak často se přistupuje na stránku z mobilních zařízení. V případě velkého poměru je tak vhodné se zaměřit na to, aby u nich navigace byla jednoduchá a přehledná. Na závěr se nabízí umožnit uživatelům podat zpětnou vazbu, která po nich ovšem nebude nutně vyžadována.

3.6 LoRaWAN

LoRaWAN je síťový protokol patřící do kategorie LPWAN (Low Power Wide Area Network) navržený pro bezdrátová IoT bateriová zařízení, kterým umožňuje připojení k Internetu. Je založen na nízké energetické spotřebě a je použitelný jak pro malé sítě s jednou

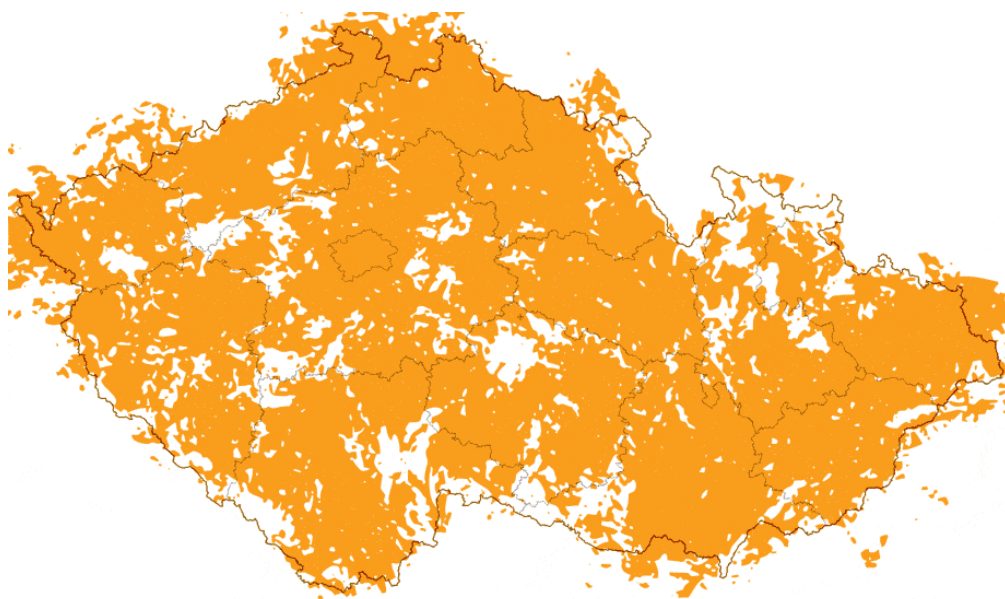
bránou, tak pro rozsáhlé globální síť. Specifikace LoRaWAN je k dispozici v podobě dokumentů od společnosti LoRa Alliance. Obsahem dokumentů je detailní popis síťového protokolu LoRaWAN, a také infrastruktury a parametrů fyzické vrstvy označované jako LoRa. [53]

Pokud není uvedeno jinak, tak text čerpá z nejnovějšího z dokumentů [54] zabývajících se specifikací. Protokol LoRaWAN je optimalizován pro baterií napájená zařízení a to mobilní i pevně umístěná. Síťová topologie je zde hvězda, ve které brány předávají zprávy mezi koncovými zařízeními a servery. Komunikace mezi bránou a serverem probíhá přes IP připojení, ale mezi bránou a koncovými zařízeními je to pomocí radiové modulace FSK. Komunikace je obousměrná a je rozprostřena mezi různé frekvenční kanály a přenosové rychlosti, přičemž zde nedochází k interferenci. Rychlost přenosu se pohybuje od 0,3 kb/s do 50 kb/s. Pro zvýšení výdrže baterie a celkové kapacity sítě je možné upravovat rychlost přenosu a RF výstup pro jednotlivá koncová zařízení.

V souvislosti s LoRaWAN jsou koncová zařízení dělena do tří tříd, lišících se energetickou spotřebou a latencí odezvy. Do třídy A se zahrnují nízkospotřebová zařízení, jež přijímají signál jen během dvou krátkých přijímacích oken, která jsou v náhodných časech. Třída B se odlišuje tím, že má pak ještě okna k přijímání v pevně daných časech. Předání informace o tom, kdy nastane přijímací časový interval, dochází při synchronizaci brány se zařízením, aby bylo zajištěno, že zařízení bude připraveno server poslouchat. U třídy C jsou přijímací okna otevřená prakticky neustále, k jejich uzavření totiž dochází, pouze když zařízení samo vysílá. Třída C je kvůli tomu sice nejvíce energeticky náročná, ale výhodou je naprosto minimální odezva při komunikaci mezi serverem a zařízením.

3.6.1 Pokrytí sítě LoRaWAN

Na obrázku *Obr. 3.8* od společnosti České Radiotelekomunikace ze stránek portálu porovnávajícího pokrytí IoT sítí v ČR [55] je zřetelné, že už v říjnu 2018 LoRaWAN pokrývala většinu České republiky. Vzhledem k tomu, že se jedná o starší údaj (a novější není k dispozici), je tak pravděpodobné, že je dnes pokryto ještě více území.



Obr. 3.8 Stav pokrytí LoRaWAN v České republice k říjnu 2018, převzato z [55]

O pokrytí se ovšem nestará pouze jedna společnost, jak je uvedeno v následujícím rozdělení vycházejícím z [56]:

- *O nekomerční pokrytí se v ČR staralo sdružení ISP přes přístupové body (POP). Autor původního textu uváděl aktivitu „free.things.cz“, ta je však už nyní nedostupná. Místo toho je například možné využít plzeňského projektu „Plzeň LoRa“, jehož cílem je dát studentům prostředky pro vývoj a podpořit podnikání v Plzni [57].*
- *Komerční pokrytí poskytují v ČR v největším měřítku České Radiokomunikace a pro připojení nabízí několik balíčků, u kterých se cena liší podle počtu připojených zařízení. Konkrétní ceny zde nejsou uvedeny a je nutné společnost kontaktovat. Dalším poskytovatelem je pak STARNET účtující podle počtu přenesených zpráv. Zde je nabídka včetně konkrétních cen, jak je vidět na obrázku Obr. 3.9.*

počet odeslaných / přijatých zpráv z jednoho zařízení za den (velikost jedné zprávy jsou maximálně desítky bajtů)	do 200 připojených zařízení (cena za jedno připojené zařízení za rok)	nad 200 připojených zařízení
500 / 50	329 Kč	smluvně (dohodou)
250 / 25	299 Kč	
100 / 5	229 Kč	

Obr. 3.9 Ceník služeb pro IoT síť na technologii LoRa u společnosti STARNET, převzato z [58]

- *Kdokoliv si případně může zařídit vlastní síť, za předpokladu, že dodrží podmínky týkající se použité vhodné frekvence a vysílacího výkonu. Nejdostupnější a nejjednodušší na obsluhu jsou v současnosti výrobky s označením The Things Network. Jedním z nich je síťová LoRaWAN brána, která je schopná obsluhovat tisíce uzlů [59]. Další možností je pak The Things Uno, což je s LoRaWAN kompatibilní varianta jednodeskového počítače Arduino Uno [60]. Oba zmíněné výrobky pracují na frekvenci 868 Mhz, určeny jsou tedy pouze pro použití v Evropě.*

3.6.2 Zabezpečení LoRaWAN

Pro zabezpečení sítě LoRaWAN se používají u síťových zařízení níže definované identifikátory a klíče převzaté ze [61]:

- DevEUI (End-device identifier):

Identifikátor vycházející z adresního rozsahu IEEE EUI-64. Je unikátní a je přiřazen už během výroby. Lze ho přirovnat k MAC adrese u zařízení v Ethernetu.

- DevAddr (End-device address):

Jedná se o dynamickou adresu zařízení s délkou 32 bitů. Nejvyšších 7 bitů má funkci identifikátoru sítě a zbylé bity reprezentují identifikátor koncového zařízení.

- AppEUI (Application identifier):

Jedná se o unikátní globální aplikační identifikátor (IEEE EUI-64). Jeho smyslem je identifikovat provozovatele aplikace.

- NwsSKey (Network Session Key):

Jde o unikátní klíč koncového zařízení. V případě, že přenášená zpráva obsahuje jen MAC příkazy, je pak klíčem zašifrován rámec i s hlavičkou. Pokud přenáší data, slouží klíč k výpočtu MIC položky pro podepsání paketu. Jeho smyslem je ověření odesílatelovy identity NetServerem.

- AppSKey (Application Session Key):

AppSKey je unikátním klíčem koncového zařízení, určeným pro zašifrování payloadu rámce. Dešifrování poté probíhá až v daných aplikačních serverech.

- AppKey (Application Key):

Je to 128bitový klíč, který je unikátní u každého koncového zařízení. Přiděluje ho provozovatel aplikace a vychází z hlavního aplikačního klíče. S jeho pomocí lze šifrovat přenos provozních klíčů a konfiguraci u OTAA aktivace.

3.6.3 Způsoby připojení zařízení do LoRaWAN

Zařízení musí při připojování do sítě projít tzv. „Aktivací“. Pro připojení existují dvě následující metody:

1. OTAA (Over-The-Air Activation):

Pro aktivování je nutné ihned při připojování projít přihlašováním. V případě, že proces proběhl v pořádku, dojde ke generaci relačních klíčů NwkSKey a AppSKey. Při ztrátě spojení nebo jeho ukončení se musí při dalším připojení opět provést přihlášení. Požadavek k připojení „join-request“ posílá zařízení serveru obsahující údaje DevEUI, AppEUI a AppKey. Pokud server dovolí, aby se zařízení připojilo, reaguje na to zprávou „join-accept“, která obsahuje AppNonce, NetID a DevAddr, jež jsou potřebné k vytvoření právě výše zmíněných relačních klíčů. [61]

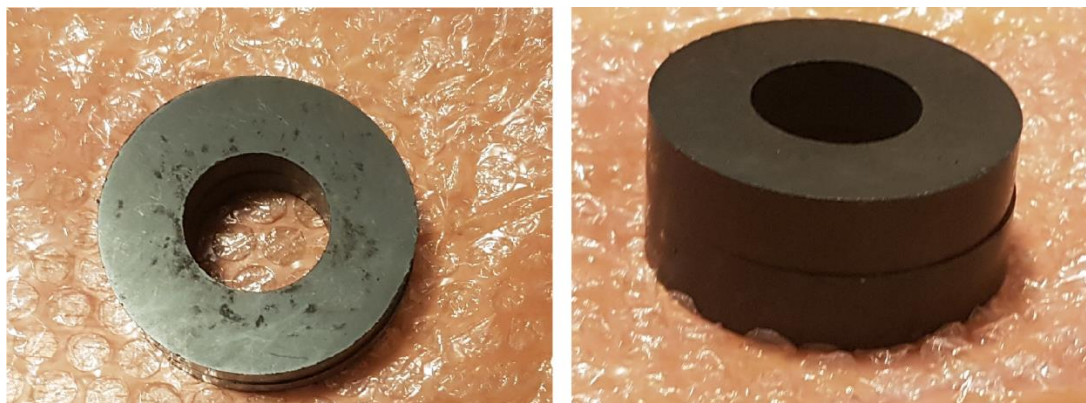
2. ABP (Activation By Personalization):

U této metody se vyhýbáme procesu výměny zpráv, u zařízení ale musíme ručně provést statické nastavení DevAddr, NwkSKey a AppSKey. Jelikož je staticky nastavená adresa DevAddr specifická pro konkrétní síťový server, není tak možné se se zařízením připojit do jiné sítě. [62]

4 Sběr, uchovávání a vizualizace obdržených dat

4.1 Nastavení LoRa připojení

LoRa Aktivace detektorů je podle poskytnutého manuálu k detektorům [63] metodou OTAA a detektory se pokusí připojit právě po aktivaci magnetem, který je možné vidět na fotografii na *Obr. 4.1*. Pro základní konfiguraci připojení potřebujeme znát tři údaje: DevEUI, AppEUI, AppKey. Tyto údaje jsou v paměti zařízení a musí být shodné u serveru. Ze zmíněného důvodu, že se očekává provedení konfigurace v cílové aplikaci ještě před aktivací senzorů, bylo vhodné si předem postup nastavení nejdříve otestovat na jiném zařízení. K tomuto účelu se nabízelo Arduino MKR WAN 1300, varianta Arduino Zero s kompatibilitou komunikačního protokolu LoRa [64].



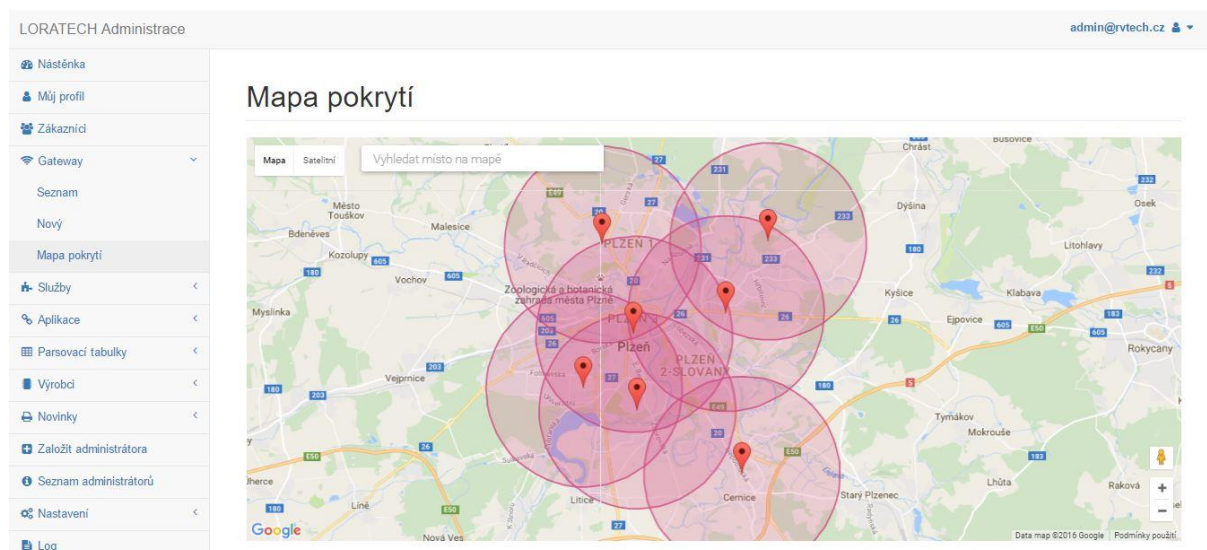
Obr. 4.1 Dva pohledy na aktivační magnet pro detektory

Při zkoušení připojení se vycházelo z postupu [65] ze stránek Arduino. Pro programování desky je nezbytné mít v počítači přítomné vývojové prostředí „Arduino“ a ručně doinstalované příslušné knihovny pro konkrétní desku. Zpřístupní se tím zároveň i několik ukázkových kódů, ze kterých jsou pro naše použití užitečné FirstConfiguration [66] nebo LoraSendAndReceive [67].

Při nahrání kódu FirstConfiguration terminál vypíše, jaké je DevEUI u Arduino desky a zeptá se, jestli chce člověk použít OTAA nebo ABP. Bylo zadáno OTAA, protože tato metoda bude také pak použita u senzorů. Terminál nás tak následně vyzve k zadání údajů APPEUI a APPKEY. V případě úspěšného připojení je do aplikace odeslána textová zpráva: „HeLoRA world!“.

V případě LoraSendAndReceive se před nahráním musí do kódu napsat konkrétní APPEUI a APPKEY. Po spuštění následně napíšeme do terminálu obsah zprávy, který chceme odeslat. Po odeslání zprávy poté dojde ke kontrole, zda naopak nějaké zprávy nedostalo koncové zařízení.

Úspěšné připojení závisí na tom, zda jsou všechny tři údaje nastavené ve vybrané aplikaci, a také zda je dostatečná síla signálu, aby se zařízení připojilo k bráně. Návod navrhuje použití The Things Network [68], se kterým však během testování nešlo navázat komunikaci kvůli síťově nepokryté oblasti použití. V horní části hlavní stránky [68] je možné si na mapě najít umístění přístupových bodů. Proto byl místo toho vybrán LORATECH [69], jelikož v jeho případě je pokryté prakticky celé město Plzeň [70], jak je vidět na obrázku Obr. 4.2. Pro práci v obou prostředích je od uživatele vyžadována bezplatná registrace.



Obr. 4.2 Pokrytí LORATECH sítě ve městě Plzeň k listopadu 2016, převzato z [70]

4.2 Konfigurace v LORATECH

Pro vysvětlení funkcí všech nástrojů v LORATECH je dostupná dokumentace [71]. Po vytvoření aplikace a přiřazení OTAA nodu fungoval úspěšně jak uplink, tak downlink. V případě uplinku je třeba ještě doručenou zprávu dekodovat např. na stránce [72], protože dorazí v kódování Base64. Na stejném místě se pak dá i zakódovat, pokud máme v úmyslu odeslat zprávu přes downlink.

4.2.1 Aktivace detektorů

Po nastavení údajů poskytnutých dodavatelem do LORATECH pro všechny detektory bylo možné začít s aktivací senzorů na parkovišti. Při provádění následujících operací pro nastavení detektorů je podmínkou, že se nesmí v jejich blízkosti nacházet feromagnetické předměty, permanentní magnety či jiné zdroje magnetického pole. Proces aktivace byl proveden 6. května v 13 hodin podle postupu uvedeného v manuálu k detektorům od firmy CITIQ [63]. Na místo, kde byl pod vozovku umístěn magnet byl přiložen aktivační magnet po dobu 90 sekund. Po uplynutí této doby byl přemístěn na jiné místo vozovky, jehož vzdálenost od původní pozice bylo větší než 2 metry a zůstal tam opět po minimální dobu 90 sekund.

Dalším krokem byla kalibrace, která je potřebná k tomu, aby bylo zajištěno, že se detektor přizpůsobí místním magnetickým podmínkám. Až poté mohou být naměřené hodnoty s jistotou považovány za přesné. Detektory mají schopnost provést kalibraci samy automaticky, odhadem by to mělo proběhnout nejdéle do 30 minut. I přesto je silně doporučováno udělat kalibraci ručně. Postup je podobný jako při aktivaci, jen s tím rozdílem, že se po položení místo nad detektorem čeká 30 sekund místo 90 sekund, před tím než se magnet přemístí. Poté se opět oddálí na místo na zemi na alespoň 90 sekund, čímž se dokončí ruční kalibrace.

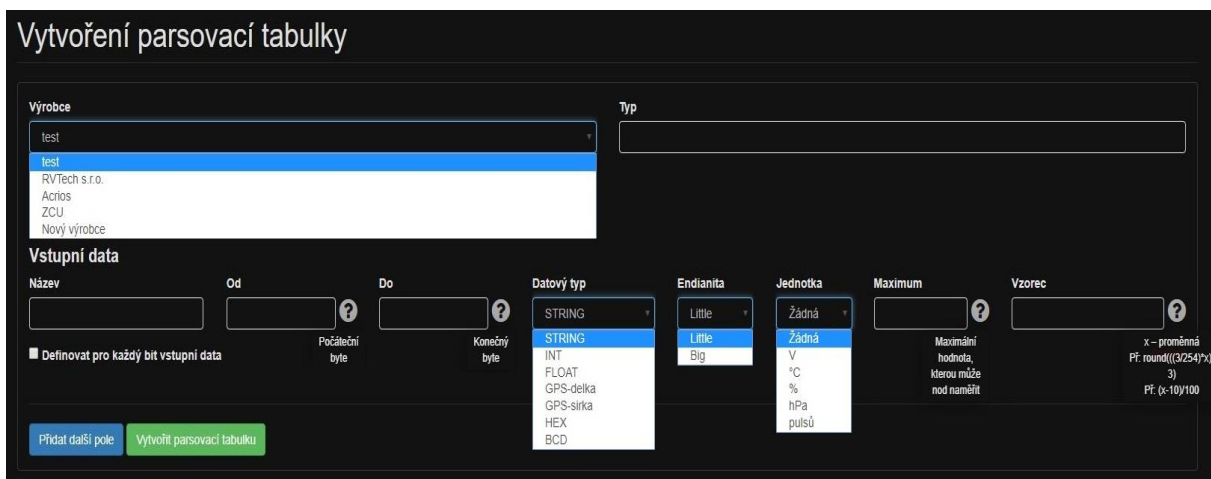
Po aktivaci byla od detektoru na LORATECH doručena zpráva o délce 5 bytů jež oznamovala: „TX_ACK_REPORT“: true“. Jednalo se o potvrzovací paket, vyjadřující, že bylo úspěšně navázáno spojení. Po kalibraci potom přišla zpráva s délkou 7 bytů obsahující proměnné s kalibračními příznaky cal0, cal1 a cal_success. Pokud má jako v našem případě cal_success hodnotu „1“, znamená to, že byla kalibrace úspěšná.

4.2.2 Parsování v prostředí LORATECH

Další činností, která by se měla provést, je dekodování payloadu. Tím dojde k převedení posloupnosti bytů (např. v podobě [225, 230, 255, 0]) na objekt, který pak může vypadat následným způsobem: {"temperature": 22.5}.

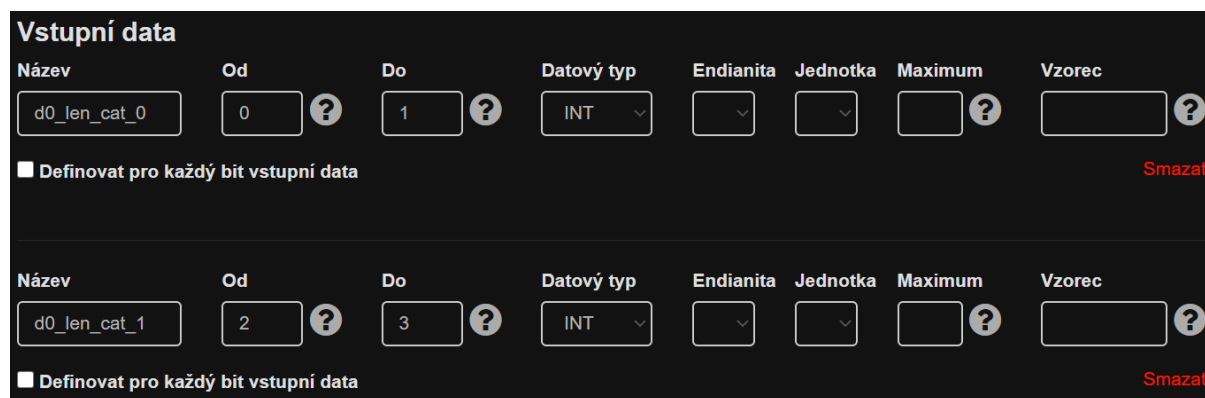
Dekodér je možný použít v prostředí The Things Network k příslušné aplikaci vytvořené uživatelem. Aby však správně fungoval, je nutné si ověřit, zda je hlavička funkce ve správném tvaru, a tedy: „function Decoder(bytes, fPort)“. Obdržený kód získaný z prostředí ChirpStack, určený pro dekodování payloadu, byl totiž v podobě: „function Decode(fPort, bytes)“. Původně to bylo nefunkční, protože zde byl odlišný název funkce a navíc bylo zároveň prohozené pořadí proměnných (pro vstupní byty a číslo portu).

Testování dekodování vypadá tak, že se zadají vstupní byty a port. Po spuštění dekodování daného kódu se poté zobrazují údaje v podobě čitelné pro člověka a přizpůsobené pro ukládání do databáze či jiné aplikace. U LORATECH je dekodování řešeno vytvořením parsovací tabulky, do které uživatel přidává pole podle vyplněných vlastností, jak je vidět na *Obr. 4.3*. Po jejím vytvoření se zpřístupní ukládání vybraných 100 paketů do formátu .csv.



Obr. 4.3 Vytvoření parsovací tabulky pro interpretaci dat

Příklad části parsovací tabulky v LORATECH vytvořeného na základě parsovacího kódu je vidět na obrázku Obr. 4.4.



Obr. 4.4 Příklad vytvoření parsovací tabulky podle parsovacího kódu

U interpretace dat nelze vybrat naráz zároveň více vytvořených parsovacích tabulek, ale pouze jen jedna (takže např. jen pro 20 bytů). Výsledná tabulka příchozích paketů je vidět na obrázku Obr. 4.5 níže, kde nalevo je původní obsah zprávy a napravo jsou pak místo toho přímo vidět hodnoty proměnných.

Čas a Datum	Data	Čas a Datum	Data
28.05.2020 16:07:20	GrYDPABjCocB/gA9FREAKgQFD/s=	28.05.2020 16:07:20	d0_len_cat_0: 6838 ; d0_len_cat_1: 828 ; d0_len_cat_2: 99 ; d1_len_cat_0: 2695 ; d1_len_cat_1: 510 ; d1_len_cat_2: 61 ;
28.05.2020 16:02:20	GrUDPABjCocB/gA9FREAKgMEA2g=	28.05.2020 16:02:20	d0_len_cat_0: 6837 ; d0_len_cat_1: 828 ; d0_len_cat_2: 99 ; d1_len_cat_0: 2695 ; d1_len_cat_1: 510 ; d1_len_cat_2: 61 ;

Obr. 4.5 Původní podoba přijatých dat (nalevo) a data po aplikovaném parsování (napravo)

Ověření shodného výsledku parsování bylo použito The Things Network, kde se dá v záložce Payload Function provést simulace funkce. Vstupními proměnnými, které jdou do parsovací funkce jsou číslo portu „fPort“ a pole bytů „bytes“. Port má tak v našem případě hodnotu: „1“ a bytes je příchozí zpráva dekovávaná z Base64 a převedená do hexadecimálního pole bytů, k čemuž je určena funkce base64ToHex. Mimo toto prostředí však funkce očekává pole bytů v desítkové soustavě, k potřebnému převodu tak slouží dodatečná funkce toByteArray. Kontrola správnosti převodu byla provedena pomocí převodníku RapidTables [73].

4.2.3 Callback

Předávání dat z LORATECH do dalšího umístění umožňuje použití Callback, kde si pomocí HTTP funkcí lze přeměřovat data z LORATECH do vlastní aplikace. V The Things Network to je také k dispozici pod záložkou Integrations. Tímto způsobem se dají také data předávat do aplikací jako např. ThingSpeak [74], v jehož prostředí je pak lze zpracovávat pomocí jazyka MATLAB.

V případě výběru možnosti Callback v LORATECH odchází pomocí metody HTTP POST zprávy včetně aplikovaného parsování. Přeneseny jsou ty údaje, u kterých v nastavení uživatel uvede, že mají být obsaženy v těle zprávy. Použité nastavení je na obrázku *Obr. 4.6*. V době psaní textu neměl jeho autor k dispozici vlastní vhodné úložiště, kam by to ukládal, a tak byla pro účely zkoušení použita jako náhrada Requestbin. Jedná se o službu, kde je možné sbírat data předávaná HTTP metodami. Základní použití je zdarma, ale pro nastavení aplikace jako privátní je nutná registrace. Z něj si to pak Node-RED vezme přes GET (HTTP request node).

Název
http_prenos

URL
https://enqpxpuqklozk.x.pipedream.net/ ?

Metoda POST
POST

Hlavička

název	hodnota

Content type

Tělo

```
{
  "server": "rvtech",
  "hardware_serial": "{devEUI}",
  "payload_raw": "{data}",
  "d0_len_cat_0": "{userData#d0_len_cat_0}",
  "d0_len_cat_1": "{userData#d0_len_cat_1}",
  "d0_len_cat_2": "{userData#d0_len_cat_2}",
}
```

Upravit callback

Obr. 4.6 Nastavení Callback pro přenos dat z LORATECH

4.2.4 LORATECH a The Things Network

Z textu v této kapitole je zřetelné, že ačkoliv jako prostředí pro nastavení LoRaWAN bylo zvoleno LORATECH, v průběhu bylo používáno i The Things Network. Druhá varianta sloužila zejména pro kontrolu stejného chování koncového zařízení Arduino a ověření, zda byl v obou případech výsledek parsování stejný. Kromě odlišné formy aplikace parsování nebyly zaznamenány zásadní rozdíly. Ve prospěch LORATECH nakonec hlavně pomohlo značně lepší pokrytí sítě v oblasti Plzně, také krok za krokem popsaná dokumentace a kladem byla i česká lokalizace, která u TTN chybí.

4.3 Parsování a přenos dat prostřednictvím Node-RED

V Node-RED byl vytvořen flow pro přenos dat z LORATECH do MySQL databáze včetně aplikovaného parseru. Node-RED flow byl původně nastaven a spuštěn na osobním notebooku. Po dokončení posledních úprav byl řetězec exportován a importován do Node-RED na Raspberry Pi 3 Model B+. Nahrazení notebooku za Raspberry Pi mělo význam zejména kvůli značné energetické úspoře. Zaheslování pro zamezení přístupu neoprávněným uživatelům bylo provedeno podle oficiálního návodu [75].

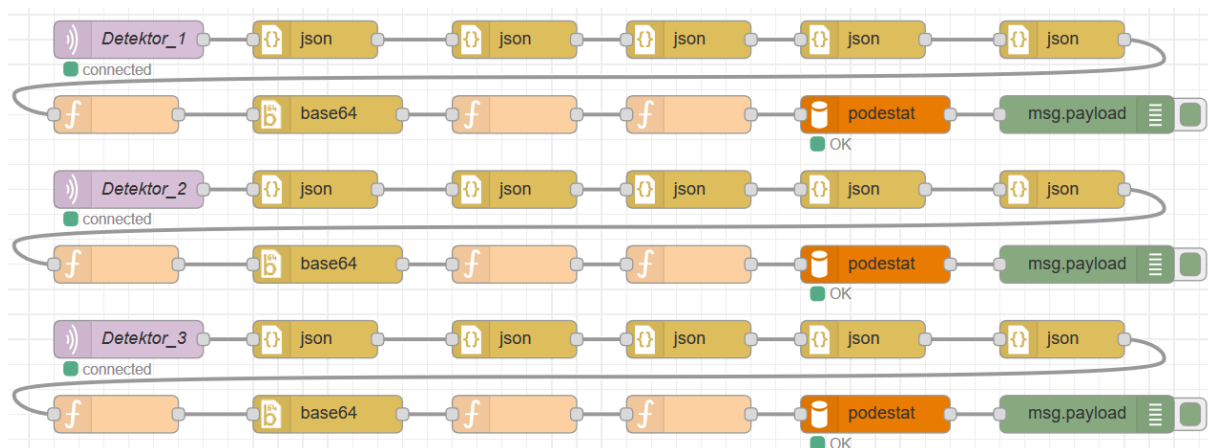
Vstupním nodem byl „mqtt in“, kde byly vyplněny údaje, k jakému serveru se to má připojit (test.loratech.cz:1883), přihlašovací jméno a heslo uživatelského účtu a Topic, tedy od jakého

zařízení bude zprávy odebírat. Tímto vstupním nodem odebíráme data přímo a vyhneme se přenosu HTTP metodami přes Requestbin. Dále byl přidán json node, zajišťující, že bude příchozí zpráva obsažená v „msg.payload“ vždy v podobě JavaScript objektu (Object).

Následně jsou vybrané položky konvertovány do JavaScript řetězce (String): fPort, data, rxInfo, txInfo. Číslo portu „fPort“ a obsah zprávy „data“ jsou převedeny, protože budou použity jako vstupní proměnné do parsovací funkce. Kdyby nebyl implementován tento blok, tak by byl obsah proměnných rozpoznán jako: [Object object]. Na metadata „rxInfo“ a „txInfo“ to bylo aplikováno z toho důvodu, že obsahují mnoho informací a bylo by zbytečné vytvářet navíc spoustu dalších sloupců v databázi, ačkoliv by to nemělo pro koncové uživatele význam.

V první funkci, která následovala, byl napsán příkaz: „msg.payload.puvodni_data = msg.payload.data;“. Slouží ke zkopírování původních dat do nové proměnné, protože bylo úmyslem je uložit do databáze také. Proměnná „data“ je totiž dále upravována, nejdříve došlo blokem „base64“ k dekodování zprávy z base64 a v dalším funkčním bloku převedena nejprve na hexadecimální, a poté na decimální pole bytů. Node „base64“ byl k dispozici až po nainstalování balíčku „node-red-node-base64“. Po parsování je přepsán celý obsah původní zprávy a tak jsou některé důležité části zprávy před použitím parsovací funkce uloženy do dočasných proměnných: devEUI, puvodni_data, fCnt, rxInfo a txInfo. Poté je možné je opačným postupem „vrátit“ zpět.

V posledním funkčním bloku bylo pomocí příkazu INSERT uvedeno, do které tabulky v databázi a do kterých sloupců konkrétně měly být vybrané údaje uloženy. Jedním z nich bylo i vygenerované datum a čas, ale u toho bylo nezbytné přidat offset o jednu hodinu, protože funkce Date() vracela Greenwichský čas, který je v jiném časovém pásmu. Příkazy pro nastavení offsetu byly převzaty z [76]. Na konec byl po nainstalování balíčku „node-red-node-mysql“ přidán node „mysql“, kde bylo nakonfigurováno připojení k cílovému serveru s vybranou databází. Za to je možné ještě přidat node „debug“, díky kterému se pak v debug okně vypisují všechny pokusy o zápis do databáze a lze tak zkontrolovat, že to proběhlo v pořádku. Jelikož měl každý detektor pro data vyhrazenou vlastní tabulku, musel mít také samostatný řetězec jak je vidět na obrázku *Obr. 4.7*. V jednom řetězci by to mohlo být za předpokladu, že by zprávy ze všech senzorů putovaly do jediné tabulky.



Obr. 4.7 Řetězec v Node-RED pro přenos dat z LORATECH do MySQL včetně aplikovaného parsování

4.4 Ukládání do MySQL databáze

MySQL databáze je umístěna na serveru ket-nas.fel.zcu. Pro každou z tabulek byla použita následující struktura znázorněná na tabulce Tab. 3.1 níže, která je doplněna podle od firmy CITIQ poskytnutého vysvětlení významu proměnných.

Tab 3.1 Výsledná tabulka v MySQL databázi včetně popisu významu proměnných

#	Název	Typ	Význam
1	ID	int(11)	automatické číselné označení záznamu
2	d0_len_cat_0	int(11)	délková kategorie ve „správném“ směru
3	d0_len_cat_1	int(11)	délková kategorie ve „správném“ směru
4	d0_len_cat_2	int(11)	délková kategorie ve „správném“ směru
5	d1_len_cat_0	int(11)	délková kategorie v „opačném“ směru
6	d1_len_cat_1	int(11)	délková kategorie v „opačném“ směru
7	d1_len_cat_2	int(11)	délková kategorie v „opačném“ směru
8	len_cat3	int(11)	počet nekategorizovaných vozidel
9	overspeed_cnt	int(11)	počet vozidel, která překročila rychlost (30 km/h)
10	avg_speed	int(11)	průměrná rychlost vozidel
11	d0	int(11)	celkový počet vozidel ve „správném“ směru
12	d1	int(11)	celkový počet vozidel v „opačném“ směru
13	total_counter	int(11)	celkový počet detekovaných vozidel
14	ref1	int(11)	referenční a průběžné magnetické hodnoty
15	ref2	int(11)	referenční a průběžné magnetické hodnoty
16	m1	int(11)	referenční a průběžné magnetické hodnoty
17	m2	int(11)	referenční a průběžné magnetické hodnoty
18	bat1	int(11)	napětí baterií
19	bat2	int(11)	napětí baterií
20	bat3	int(11)	napětí baterií
21	bat4	int(11)	napětí baterií

22	rfsup	int(11)	napětí baterií
23	ti_temp	int(11)	teplota křemíku procesoru
24	vsup	int(11)	napětí baterií
25	cal0	int(11)	kalibrační příznaky
26	cal1	int(11)	kalibrační příznaky
27	cal_success	int(11)	kalibrační příznaky
28	TX_ACK_REPORT	varchar(5)	aktivační příznaky
29	datum	datetime	datum odeslání zprávy
30	devEUI	varchar(16)	identifikace zařízení
31	puvodni_data	varchar(255)	původní obsah zprávy před parsováním
32	fCnt	int(11)	počítadlo rámců (frame counter)
33	rxInfo	text	informace týkající se přenosu zprávy
34	txInfo	text	informace týkající se přenosu zprávy

4.5 Tvorba webové stránky

Vizualizace získaných dat byla provedena vytvořením webové stránky, jež byla napsána s pomocí skriptovacích programovacích jazyků JavaScript [77] a PHP [78] v bezplatném textovém editoru PSPad [79]. Prostřednictvím PHP bylo pak možné přistupovat k datům v MySQL databázi [80]. PHP však vyžadoval ke své správné funkci webový server, který skripty zpracovává [81]. Z toho důvodu byl použit programový balíček XAMPP [82], díky kterému lze zprovoznit lokální Apache server a MySQL databázi. Daný vytvořený webový projekt se pak musí nacházet v podsložce, kde je umístěný nainstalovaný XAMPP: „xampp\htdocs\“. Poté se dá spustit v prohlížeči přes URL řádek při náhradě cesty umístění končící částí „xampp\htdocs/“ za „localhost/“.

4.5.1 Základní nastavení

Pro pochopení základního chování při práci s databází byl využit kód [83]. Na úvod je nezbytné navázat spojení s databází, zde k tomu dochází po přiložení (require) externího souboru s přihlašovacími údaji. Tam se funkce `mysqli_connect` pokusí navázat spojení s MySQL databází s přihlašovacími údaji obsahujícími adresu serveru, přihlašovací jméno, heslo a jméno databáze. Pokud se podařilo připojit a tabulka není prázdná, tak se přes while cyklus vypíší požadované řádky a sloupce, zde jsou to všechny. Při neúspěchu to oznámí, že nebyly nalezeny žádné výsledky. Aby se ukončilo spojení s databází, použije se na závěr skriptu funkce `close()`;

Pro porozumění toho, jakým způsobem pomocí PHP předávat data z MySQL databáze grafům `morris.js` byl využit zdrojový kód z [84]. Konkrétně šlo o PHP stránku „chart.php“ a SQL příkazy pro vytvoření zdrojových dat v MySQL databázi „bar_chart.sql“.

Zvolené grafy a formulář na volbu data s časem vyžadují pro běh několik knihoven:

- jQuery (jquery-3.4.0.js) = Javascriptová knihovna usnadňující práci s HTML dokumenty [85].
- Bootstrap (bootstrap.min.css) = Šablona upravující vzhled a rozložení prvků stránky [86], není nezbytná.
- morris.js (morris.js, raphael-min.js, morris.css) = Knihovna pro vykreslování grafů [87].
- jQuery Date and Time picker (jquery.datetimepicker.full.js, jquery.datetimepicker.css) = jQuery plugin pro formulář s datem a časem [88].

4.5.2 Nastavení data a času

Na začátku PHP kódu jsou pomocí funkce „date“ definovány proměnné týkající se data:

- \$dnesni_datum - aktuální datum (s „nulovým“ časem)
- \$pocatecni_datum - od kterého data a času se načtou záznamy pro tabulku
- \$skoncove_datum - do kterého data a času se načtou záznamy pro tabulku
- \$dnesni_datum - aktuální datum (včetně času)
- \$pocatecni_datum_provozni - od kterého data a času se načtou záznamy pro graf
- \$skoncove_datum_provozni - do kterého data a času se načtou záznamy pro graf
- \$pocatecni_datum_upravene - úprava data pro zobrazení ve formátu: „den. měsíc. rok“

S datem se pracuje ve stejné podobě jako je v MySQL databázi, tedy „Y-m-d H:i:s“ neboli „Rok-měsíc-den hodiny:minuty:sekundy“. Pomocí funkce „strtotime“ lze pak data mezi sebou porovnávat a také u nich změnit časový offset, detailně to popisuje oficiální dokumentace [89]. Pokud převádíme datum a čas ze tvaru „Y-m-d H:i:s“ na „Y-m-d“, je datum ponecháno beze změny, ale čas je vynulován na „00:00:00“.

Dnešní datum je ve dvou proměnných, protože jedno vymezuje časové rozpětí pro tabulku a druhé pro grafy. \$dnesni_datum má nastavený čas na půlnoc, protože z něj vychází rozsah pro tabulku, který je od půlnoci na začátku dne (\$pocatecni_datum) až do půlnoci na konci dne (\$skoncove_datum). \$dnes vrátí přesný čas v daném momentě, který je v prvním případě posunutý o hodinu zpět (\$pocatecni_datum_provozni) a v druhém případě je ponechán aktuální čas. \$pocatecni_datum_upravene je vytvořeno proto, aby uživatel vždy viděl, z kterého dne vidí údaje. Den je pak místo „rok-měsíc-den“ zobrazen jako „den. měsíc. rok“.

Následuje reakce na obrázku Obr. 4.8 viditelný formulář nacházející se pod PHP skriptem, prostřednictvím kterého si uživatel vybere, ze kterého data a hodiny chce zobrazit informace a

svůj výběr potvrdí stisknutím tlačítka „Vyhledat“. Tím dojde k předání požadovaného data metodou POST a přepíše to původní hodnotu v proměnné \$pocatecni_datum. Poté dochází ke kontrole, zda uživatel nevybral nežádoucí datum, ze kterého neexistují výsledky, tedy dobu v minulosti (\$prvni_datum) nebo v budoucnosti, kde nesmí překročit dnešní den (\$dnes). Za předpokladu, že je výběr vyhovující, jsou nastaveny odpovídající časové rozsahy, v opačném případě jsou vráceny zpět k vypisování dnešního dne. U vyhovující varianty má \$pocatecni_datum přepsaný čas na půlnoc až po přiřazení hodnot do \$pocatecni_datum_provozni a \$skoncove_datum_provozni, jinak by grafy pokaždě vykreslovaly dobu od půlnoci do jedné hodiny.

Následující příkaz pod zmíněným formulářem pod PHP skriptem slouží pro grafické zobrazení pro výběr data po kliknutí na textové pole jako je na Obr. 4.8: „\$(\"#pocatecni_datum\").datetimepicker({step:60});“. Část v hranatých závorkách „{step:60}“ specifikuje, po kolika minutách je rozdělen posuvník s časem. Přepnutí na českou lokalizaci se pak dosáhne příkazem: „jQuery.datetimepicker.setLocale('cs');“. Jelikož se jedná o javascriptové příkazy, musí být mezi „<script></script>“.

Zobrazit údaje ze dne:

Ne	Po	Út	St	Čt	Pá	So
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

15:00	366 vozidel
16:00	363 vozidel
17:00	789 vozidel
18:00	1 vozidel
19:00	
20:00	

Obr. 4.8 Formulář pro výběr data a času

4.5.3 Získání dat pro grafy

Do proměnných označených „\$sql“ je vždy ukládán SQL příkaz sloužící pro manipulaci s tabulkami v databázi. Zde je používán příkaz SELECT pro načtení dat z určených sloupců, za tím následuje FROM se jménem tabulky a podmínkami WHERE a AND. V tomto kódu jsou tak vybrány pouze záznamy, u kterých je datum v rozmezí počátečního a koncového data. Výsledek je přiřazen proměnné „\$result“. Při splnění podmínky, že není počet řádků

získané tabulky nulový, se pomocí cyklu foreach prochází každý řádek, jinak je oznámeno: „0 záznamů“.

Názvy tabulek pro detektory odpovídají detektorům následovně:

- detek1 = hlavní průjezdový detektor
- detek2 = vjezdový detektor
- detek3 = výjezdový detektor

Jelikož detektory posílají ve zprávách celkový počet detekovaných vozidel, je potřeba si z určit rozdíl mezi po sobě jdoucími záznamy. Toho je docíleno posláním SQL požadavku databázi. Tam se výběr po obdržení hodnoty z konkrétní řady posune podle data o jedno zpět a starší údaj odečte. Výsledek je uložen nového dočasného sloupce „pocitadlo“, který po dokončení operací přestane existovat. Podoba tohoto výběru byla odvozena na základě příkladu řešení zmíněného v [90]. Buňka v prvním řádku je ale pokaždé prázdná, protože nelze provést matematický výpočet, když předchozí řádek není v rozsahu. Proto je zahrnuto „limit 300 offset 1“, kde hodnota za „offset“ udává, kolik prvních řádků chceme vynechat z výběru a to vždy musí být doplněno o „limit“, tedy kolik záznamů chceme načíst maximálně. První cyklus foreach projde požadované údaje pro průjezdy ve správném směru a v něm vnořený druhý cyklus udělá to samé pro průjezdy opačném směru. Průjezdy do grafů a tabulky jsou konkrétně počítány takto:

- Vjezd = „průjezd ve správném směru u vjezdového detektoru“ + „průjezd v opačném směru u výjezdového detektoru“
- Výjezd = „průjezd ve správném směru u výjezdového detektoru“ + „průjezd v opačném směru u vjezdového detektoru“
- Hlavní průjezd = „průjezd ve správném směru u hlavního průjezdového detektoru“ + „průjezd v opačném směru u hlavního průjezdového detektoru“

Tento součet je spolu s datem přidáván do řetězce „\$chart_data“, kde je předdefinovaná podoba vstupních dat, kterou graf očekává. Po poslední změně je funkcí „substr“ odebrána na konci čárka, aby nedošlo k problému se syntaxí. Rozdíl je v načítání dat pro graf hlavního průjezdového detektoru, kde stačí jediný foreach cyklus. Důvod je ten, že se filtrovaná data u proměnných pro oba směry průjezdu nachází ve stejné tabulce v databázi.

4.5.4 Sestavení tabulky

Kvůli údajům pro tabulku na webové stránce je získán první a poslední záznam. Jelikož není mezi složenými závorkami cyklu foreach žádný kód, zůstane v proměnné \$row poslední načtený řádek. Dvakrát po sobě prostřednictvím „prázdného“ cyklu foreach najdeme poslední záznam. Napoprvé příkazem „ORDER BY datum ASC“ projedeme tabulku vzestupně a dostaneme nejnovější záznam a napodruhé se přes „ORDER BY datum DESC“ postupuje opačným směrem k nejstaršímu záznamu. Součty nejnovějších údajů jsou poté odečteny od nejstarších. Opět je to podle vzoru výše definovaných matematických součtů.

Pomocí rozdílu nejnovějších a nejstarších hodnot je určen počet aut, která překročila rychlost. Překročené rychlosti jsou u jednotlivých detektorů v \$prekroceni (s rozdílnými indexy) a nakonec jsou sečteny v \$prekroceni_celkem. Rychlosti jsou ukládány do \$dovnitř_za_den pro vjezdový detektor, \$ven_za_den pro výjezdový a \$hlavni_prujezd_za_den pro hlavní průjezdový. Počet volných míst (\$volna_mista) parkoviště je určen výchozím stavem ze dne 6.6. 2020 ve 22:00, kdy bylo na parkovišti jedno auto. Rozdíl jednoho auta je odečten od kapacity parkoviště, která je 234 vozidel. Zároveň je ještě odečtena konstantní hodnota 110 reprezentující odchylku v počtu míst ve zmíněném výchozím stavu. K tomu je přičítán rozdíl mezi nejnovějším údajem o počtu vozidel, která vjela s počtem vozidel, která vyjela. V závěrečné verzi webové stránky není tento údaj obsažen z důvodu popsaného v prvním odstavci v kapitole 5.1.

Tabulka se vymezuje pomocí „<table></table>“, řádek tabulky „<tr></tr>“, buňka tabulky „<td></td>“, hlavičková buňka tabulky „<th></th>“ a úroveň velikosti nadpisu tabulky je dána výběrem „<h3></h3>“. Vysvětlení toho, jak se zapisuje tabulka bylo nalezeno v přehledu na [91]. Tabulka vypisuje údaje k vybranému dni, konkrétně se jedná o:

- Počet vjezdů do parkoviště
- Počet výjezdů z parkoviště
- Počet průjezdů hlavním průjezdem
- Počet vozidel, která překročila rychlost

4.5.5 Umístění grafů

Definované „ohraniceni“ kromě jednoznačného zvýraznění hranic uvnitř se nacházejícího grafu (se šířkou 1000 pixelů) pomáhá při přidávání názvů os grafu, jelikož je samotný graf postrádá. Text pro osu y „osay“ je umístěn nad vykresleným grafem se zarovnáním doleva s pozicí danou hranicí „ohraniceni“. Text pro osu x „osax“ je stejným způsobem umístěn pod grafem, akorát je naopak zarovnán doprava. Nadpis grafu má úroveň velikosti „h2“ a datum je vypsáno pomocí PHP příkazu „echo“. K pochopení správného způsobu zápisu stylů (zejména kvůli ohraničení) pomohly ukázky použití z výukových stránek [92].

Javascriptový skript „`<script></script>`“ udává vlastnosti grafu:

- element - označení grafu
- data - řetězec obsahující vstupní data k vykreslení
- xkey - veličina na ose x
- ykey - veličina na ose y
- ymin - minimální hodnota na ose y
- labels - popisek, zobrazující se po najetí na hodnotu
- hideHover - chování toho, kdy se zobrazí popisek
- gridIntegers - zda budou hodnoty na osách celá čísla

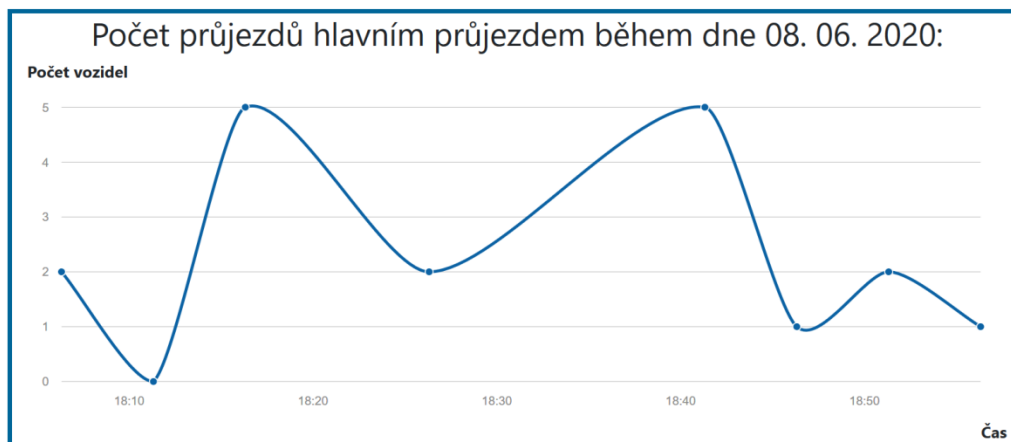
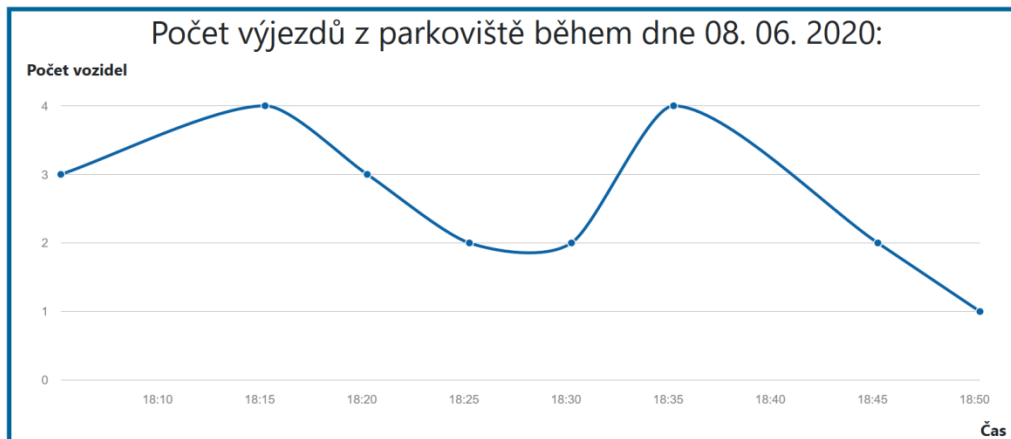
Na závěr se opět nesmí zapomenout na uzavření spojení (close) s databází. Konečný vzhled celkové stránky je vidět na obrázku *Obr. 4.9*.

Informace získané detektory průjezdů:

Zobrazit údaje ze dne:

Nejaktuálnější údaje ze dne 08. 06. 2020

Počet vjezdů do parkoviště	366 vozidel
Počet výjezdů z parkoviště	363 vozidel
Počet průjezdů hlavním průjezdem	789 vozidel
Počet vozidel, která překročila rychlost	1 vozidel



Obr. 4.9 Vzhled vytvořené webové stránky

5 Zhodnocení realizovaného řešení

5.1 Porovnání detekovaných a fyzicky zaznamenaných hodnot

Dne 11.6. v čase mezi 14:20 až 15:00 bylo provedeno fyzické zaznamenávání průjezdů přímo na parkovišti. Detektory u závor pro vjezd a výjezd z neveřejné části parkoviště lze považovat za přesné. Zaznamenaný počet prakticky odpovídal detekovaným hodnotám, nepatrně se to lišilo pouze v řádu jednotek. S tím také souvisí po jednom týdnu ověřený počet volných míst zmíněný v předchozí kapitole. V době 22 hodin bylo na parkovišti znovu obsazeno pouze jedno parkovací místo, takže volný míst by správně měl být zase 233. Výsledek vypočtených údajů jich však ukazoval pouze 227. To znamená, že velikost odchylky za jeden týden odpovídá 8 vozidlům. Kvůli této nepřesnosti byl z tabulky ve vytvořené webové stránce odstraněn údaj o počtu volných míst.

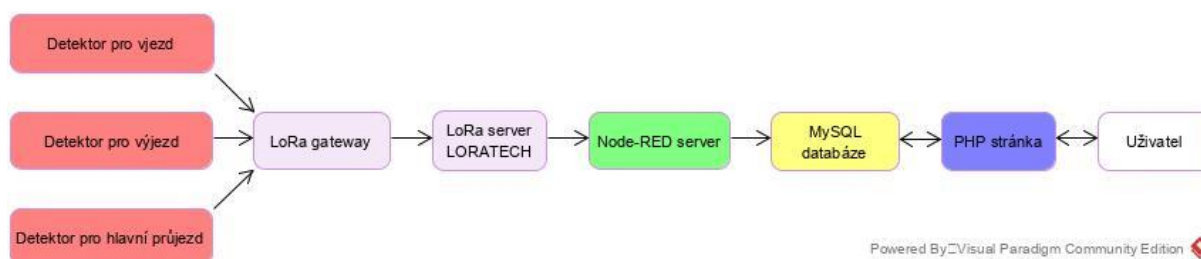
Rozdíl při fyzickém zaznamenávání nastal u hlavního průjezdového detektoru, kde bylo napočítáno více aut, než kolik jich bylo detekováno. Jedním z hlavních důvodů je ten, že řidiči, kteří chtěli opustit parkoviště čekali, dokud je ostatní auta na hlavní silnici nepustí vyjet ven. Na hlavním průjezdu tak v době zaznamenávání v jeden moment velmi těsně za sebou čekala tři vozidla na výjezd. Detektor tak tento jev zřejmě zařadil do „počtu nekategorizovaných“ jako jedno vozidlo. V této situaci se zároveň pokoušela o vjezd do parkoviště další vozidla, což mohlo výsledky také ovlivnit.

Je důležité brát v potaz, že tyto nepřesnosti nastaly v čase intenzivního provozu. Poslední zmíněný detektor se nachází uprostřed hlavního průjezdu, který je velmi široký. Spousta aut často neprojíždí přímo uprostřed nad detektorem jako je to u detektorů u závor. Zejména k tomu dochází v nejméně navštěvovaných hodinách, kde si řidiči navzájem dávají prostor, aby mohli všichni naráz vjíždět i vyjíždět. Navíc průjezdy v obou směrech snímá právě pouze jediný detektor. Stojí za zvážení, že by na místě průjezdu byly místo jednoho detektoru dva, které by byly umístěny dále od středu (a blíže k okraji) vozovky.

Pro jednoznačného určení nepřesností by bylo nezbytné provést aktualizaci detektorů. V současnosti odesílají informace jednou za 5 minut a ne úplně vždy dorazí každá zpráva. Tento stav pravděpodobně souvisí s tím, že pokud není příchozí zpráva na serveru vyhodnocena jako zcela v pořádku, není uložena, ale zahozena. Najednou tak není vypočítaný rozdíl za posledních 5 minut, ale místo toho je rozmezí desetiminutové. Nepochybně by tedy pomohla změna intervalu odesílání dat, ať už na menší časový úsek (např. 1 minuta) nebo po každém detekovaném průjezdu. Zvýšila by se tím časová přesnost záznamů a usnadnilo by to další analýzu nestandardních situací.

5.2 Celkový proces přenosu dat

Pro uskutečněné řešení byl pro lepší znázornění v programu Visual Paradigm vytvořen diagram na obrázku *Obr. 5.1*. Diagram ukazuje, jakou cestou putují data snímaná detektory. Proces začíná tím, že jsou všechna odesílaná data prostřednictvím LoRa gateway v blízkém dosahu doručena na LoRa server, konkrétně se jedná o plzeňský LORATECH. Node-RED server je nastaven, aby ukládané zprávy odebíral. Provede na nich úpravu v podobě parsování a zároveň udává, kam budou dále putovat. Cílovým umístěním je MySQL databáze, kde má každý z detektorů pro příchozí data svoji vlastní tabulku. Z tabulek si potřebné záznamy vybírá a zobrazuje PHP stránka. Výběr požadovaných údajů k vizualizaci je ovlivněn koncovým uživatelem volbou časového rozmezí na webové stránce.



Obr. 5.1 Diagram procesu přenosu dat z detektorů až ke koncovému uživateli

Předností celkového řešení je, že při existenci nového IoT zařízení je možné jej snadno a rychle integrovat do stávajícího systému s minimem nezbytných úprav. Další výhodou je případná možnost nahradit část procesu alternativou, jako příklad lze uvést výměnu MySQL databáze za jiný typ databáze. Zároveň se dá proces obohatit rozšířením o další prvek, např. na zvýšený důraz ohledně zálohování reagovat přidáním dodatečného úložného serveru.

Závěr

Stanoveným cílem diplomové práce bylo provést rozbor IoT aplikací pro sběr dat, realizovat příslušnou část IoT systému na tomto principu včetně vysvětlení celého procesu a zajistit vizualizaci získaných dat v navržené webové aplikaci.

První část práce se věnovala tomu, jak důležitou pozici sběr dat zastává v oblasti IoT. Popisuje platformy, jež jsou součástí systému. Zároveň ukazuje přínosy použití na několika příkladech.

Ve druhé části diplomové práce proběhla analýza prostředků pro manipulaci s IoT uzly. Zabývala se rozdílnými IoT protokoly a vývojovým prostředím, které je vhodné pro jejich správu a komunikaci.

Třetí část řešila implementaci IoT senzorů v místě parkoviště a výběr spolehlivého prostředí za účelem integrace do systému pro sběr a monitorování dat.

Čtvrtá část obsahuje postup pro dosažení úspěšné aktivace detektorů a doručení užitečných dat až ke koncovému uživateli. Součástí je průběh nastavení LoRaWAN sítě, dekodování a přenos dat Node-RED prostředím a vzájemné chování cílové databáze a vytvořené webové stránky.

V závěrečné části bylo provedeno ověření přesnosti provedeného řešení vycházející z hodnot naměřených detektory a fyzicky zaznamenaných hodnot. Prací byl vytvořen základ systému pro monitoring parkoviště na principu IoT s možností dalšího jeho rozšiřování.

Seznam literatury a informačních zdrojů

- [1] 2019 TRENDS IN INTERNET OF THINGS. *CompTIA* [online]. 2019 [cit. 2019-10-12]. Dostupné z: https://comptiacdn.azureedge.net/webcontent/docs/default-source/research-reports/research-report---comptia-iot-2019.pdf?sfvrsn=176d2d0_0
- [2] MEERMAN, Lukas. Sensors as drivers of Industry 4.0. *EY* [online]. 2019 [cit. 2019-10-12]. Dostupné z: [https://www.ey.com/Publication/vwLUAssets/ey-at-studie-sensoren-treiber-industrie-40-2019/\\$FILE/EY Studie - Sensors as drivers of Industry 4.0.pdf](https://www.ey.com/Publication/vwLUAssets/ey-at-studie-sensoren-treiber-industrie-40-2019/$FILE/EY%20Studie%20-%20Sensors%20as%20drivers%20of%20Industry%204.0.pdf)
- [3] Using smart sensors to drive supply chain innovation. *Deloitte* [online]. 2018 [cit. 2019-10-16]. Dostupné z: <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/process-and-operations/us-cons-smart-sensors.pdf>
- [4] MCFADDEN, Christopher. Raspberry Pi and Arduino: What's the Difference and Which Is Best for Your Project? *Interesting Engineering* [online]. 2018 [cit. 2019-10-16]. Dostupné z: <https://interestingengineering.com/raspberry-pi-and-arduino-whats-the-difference-and-which-is-best-for-your-project>
- [5] What is a Raspberry Pi? *Raspberry Pi* [online]. [cit. 2019-10-20]. Dostupné z: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- [6] VERRY, Tim. What is the Raspberry Pi? - ExtremeTech. *ExtremeTech* [online]. 2012 [cit. 2019-10-20]. Dostupné z: <https://www.extremetech.com/computing/124317-what-is-raspberry-pi-2>
- [7] Camera Module V2 – Raspberry Pi. *Raspberry Pi Foundation* [online]. [cit. 2019-11-18]. Dostupné z: <https://www.raspberrypi.org/products/camera-module-v2/>
- [8] Pi NoIR Camera V2. *Raspberry Pi Foundation* [online]. [cit. 2019-11-18]. Dostupné z: <https://www.raspberrypi.org/products/pi-noir-camera-v2/>
- [9] Raspberry Pi PoE HAT. *Raspberry Pi Foundation* [online]. [cit. 2020-02-16]. Dostupné z: <https://www.raspberrypi.org/products/poe-hat/>
- [10] Sense HAT. *Raspberry Pi Foundation* [online]. [cit. 2020-02-08]. Dostupné z: <https://www.raspberrypi.org/products/sense-hat/>
- [11] HONESS, David a Oliver QUINLAN. Astro Pi: Running your code aboard the international space station. *Acta Astronautica* [online]. 2017 [cit. 2020-02-08]. Dostupné z: <https://www.raspberrypi.org/app/uploads/2017/03/Astro-Pi-Running-your-code-aboard-the-international-space-station.pdf>
- [12] Python module to control the Raspberry Pi Sense HAT. *Sense HAT* [online]. [cit. 2020-02-08]. Dostupné z: <https://pythonhosted.org/sense-hat/>
- [13] HŮNA, Jaroslav. Raspberry Pi TV HAT: kompletní průvodce. *blog zonepi* [online]. 2019 [cit. 2019-11-18]. Dostupné z: <http://blog.zonepi.cz/raspberry-pi-tv-hat-kompletni-pruvodce/>
- [14] Raspberry Pi Touch Display. *Raspberry Pi Foundation* [online]. [cit. 2019-11-19]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-touch-display/>
- [15] Unipi 1.1 - kompletní sada. *unipi technology* [online]. [cit. 2019-11-19]. Dostupné z: <https://www.unipi.technology/cs/unipi-1-1-kompletni-sada-p123>
- [16] HORSEY, Julian. Raspberry Pi Docker Pi IoT Node(A) enables LoRa, GPRS and GPS. *Geeky Gadgets* [online]. 2019 [cit. 2019-11-25]. Dostupné z: <https://www.geeky-gadgets.com/raspberry-pi-docker-pi-iot-nodea-enables-lora-gprs-and-gps/>
- [17] Banana Pi BPiDuino UNO Board Module, compatible with for Arduino UNO, available for Raspberry Pi. *Raspberry Pi Projects* [online]. 2015 [cit. 2019-11-25]. Dostupné z: <https://projects-raspberrypi.com/banana-pi-bpiduino-uno-board-module->

- compatible-with-for-arduino-uno-available-for-raspberry-pi/
- [18] 3D Gesture & Tracking Shield for Raspberry Pi (MGC3130). *Seeed Studio* [online]. [cit. 2019-11-25]. Dostupné z: <https://www.seeedstudio.com/3D-Gesture-Tracking-Shield-for-Raspberry-Pi-MGC3130-p-4073.html>
- [19] GrovePi+ Starter Kit for Raspberry Pi A+,B,B+&2,3 (CE certified) - Seeed Studio. *Seeed Studio* [online]. 2019 [cit. 2019-10-23]. Dostupné z: <https://www.seeedstudio.com/GrovePi-Starter-Kit-for-Raspberry-Pi-A-B-B-2-3-CE-certified-p-2572.html>
- [20] Getting Started with the GrovePi+. *Dexter Industries* [online]. [cit. 2019-10-23]. Dostupné z: <https://drive.google.com/file/d/0B0WChwP4CnLBbIJMZWM3MzhIT0k/view>
- [21] Get Started with GrovePi. *Dexter Industries* [online]. 2019 [cit. 2019-10-24]. Dostupné z: <https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/>
- [22] NIELD, David. The Beginner's Guide to the Raspberry Pi. *Gizmodo* [online]. 2016 [cit. 2019-10-20]. Dostupné z: <https://gizmodo.com/the-beginners-guide-to-the-raspberry-pi-1782608097>
- [23] CROCKER-WHITE, Chris. Build an air quality monitor with InfluxDB, Grafana, and Docker on a Raspberry Pi. *Balena* [online]. 2019 [cit. 2019-10-22]. Dostupné z: <https://www.balena.io/blog/build-an-environment-and-air-quality-monitor-with-raspberry-pi/>
- [24] ZUTH, Daniel. Sběr teplotních údajů při použití jednodeskového počítače. *Automa* [online]. 2015 [cit. 2019-10-23]. Dostupné z: http://automa.cz/Aton/FileRepository/pdf_articles/53340.pdf
- [25] KRÁKORA, Jiří. *Vytvoření modulu regulace pro systém řízení stáje za pomoci RaspBerry Pi* [online]. 2017 [cit. 2019-10-23]. Dostupné z: https://theses.cz/id/6w8e4q/BP_krakora.txt
- [26] MAREK, Josef. *Vizualizace dat ze senzorů CISS pomocí RaspBerry PI* [online]. 2017 [cit. 2019-10-23]. Dostupné z: <https://is.vspj.cz/bp/get-bp/student/42041/thema/6746>
- [27] THE INTERNET OF THINGS: MAPPING THE VALUE BEYOND THE HYPE. *McKinsey Global Institute* [online]. 2015 [cit. 2019-10-23]. Dostupné z: https://www.mckinsey.com/~media/McKinsey/Industries/Technology Media and Telecommunications/High Tech/Our Insights/The Internet of Things The value of digitizing the physical world/Unlocking_the_potential_of_the_Internet_of_Things_Executive_summary.ashx
- [28] COSTA, Pedro. *IoT for Efficient Data Collection from Real World Resources* [online]. 2016 [cit. 2019-11-03]. Dostupné z: https://run.unl.pt/bitstream/10362/21531/1/Costa_2016.pdf
- [29] ACHARY, Rathnakar a Jenith LALSETA. Internet of Things: Essential Technology, Application Domain, Privacy and Security Challenges. *International Journal of Computer Applications* [online]. 2017 [cit. 2019-11-03]. Dostupné z: <https://pdfs.semanticscholar.org/77c0/22cc81f043680408bdfc3563bf712043724e.pdf>
- [30] FROLLO, Martin. *Implementace komunikačního scénáře využívajícího DDS a hodnocení bezpečnosti* [online]. 2019 [cit. 2020-01-08]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=190576
- [31] VAJNAR, Martin. *ORTE communication middleware for Android OS* [online]. 2014 [cit. 2020-01-08]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/61469/F3-BP-2014-Vajnar-Martin-thesis.pdf?sequence=2&isAllowed=y>
- [32] YUAN, Michael. What is MQTT? Why use MQTT? – IBM Developer. *IBM Developer* [online]. 2017 [cit. 2019-11-04]. Dostupné z: <https://developer.ibm.com/articles/iot->

- mqtt-why-good-for-iot/
- [33] MALÝ, Martin. Protokol MQTT: komunikační standard pro IoT. *Root.cz* [online]. [cit. 2020-02-06]. Dostupné z: <https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/>
- [34] SIXTA, Pavel. *Datové protokoly v IoT* [online]. 2016 [cit. 2019-11-29]. Dostupné z: https://dk.upce.cz/bitstream/handle/10195/68081/SixtaP_DatoveProtokkoly_SN_2017.pdf?sequence=3&isAllowed=y
- [35] HAVLENA, Petr. *Protokol XMPP Jabber a jeho využití pro přenos multimediálních dat* [online]. 2009 [cit. 2019-11-29]. Dostupné z: <https://dspace.vutbr.cz/bitstream/handle/11012/11147/diplomPrace.pdf?sequence=3&isAllowed=y>
- [36] KRUML, Jiří. *Integrace nástroje Esper a fronty zpráv RabbitMQ* [online]. 2016 [cit. 2020-01-10]. Dostupné z: <https://is.muni.cz/th/u0rm5/bcthesis.pdf>
- [37] LEA, Rodger. Node-RED: Lecture 1 – A brief introduction to Node-RED – Node RED Programming Guide. *Node-RED Programming Guide* [online]. 2016 [cit. 2019-10-27]. Dostupné z: <http://noderedguide.com/nr-lecture-1/>
- [38] COPE, Steve. Creating Flows- Node-Red Admin Basics. *Steve's internet Guide* [online]. 2019 [cit. 2019-11-03]. Dostupné z: <http://www.steves-internet-guide.com/node-red-admin-basics/>
- [39] PEŠKA, Robert. Node Red slibuje snadné programování IoT aplikací | Vývoj.HW.cz. *vyvoj.hw.cz* [online]. 2019 [cit. 2019-10-27]. Dostupné z: <https://vyvoj.hw.cz/node-red-slibuje-snadne-programovani-iot-aplikaci.html>
- [40] VOJÁČEK, Antonín. Node-RED = PC aplikace pro JSON ovládání IoT zařízení. *automatizace.hw.cz* [online]. 2019 [cit. 2019-11-25]. Dostupné z: <https://automatizace.hw.cz/netio-node-red/node-red-pc-software-pro-json-ovladani-iot-zarizeni.html>
- [41] LHOTKA, Ladislav. Nové standardy pro JSON. *Root.cz* [online]. 2013 [cit. 2019-11-29]. Dostupné z: <https://www.root.cz/clanky/nove-standardy-pro-json/>
- [42] Monitoring průjezdu dopravy. *CITIQ* [online]. [cit. 2020-01-08]. Dostupné z: <http://www.citiq.cz/monitoring-prujezdu-dopravy.html>
- [43] Transport detector. *Sigfox Partner Network* [online]. [cit. 2019-12-20]. Dostupné z: <https://partners.sigfox.com/products/traffic-and-parking-detection>
- [44] Systém chytrého parkování. *CITIQ* [online]. [cit. 2019-12-20]. Dostupné z: <http://www.citiq.cz/images/obsah/desatero.pdf>
- [45] SPURNÁ, Ivona. Přetíženost silnic v okolí Škody Auto v Kvasinách pomáhá řešit CITIQ, člen IQRF Alliance. *Synopcity* [online]. 2018 [cit. 2019-12-20]. Dostupné z: <https://www.synopcity.com/en/articles/ivona-spurna-iqrf-alliance-2147497235/pretizenos-silnic-v-okoli-skody-auto-v-kvasinach-pomaha-resit-citiq-clen-iqrf-alliance-399>
- [46] Detektory pro městskou dopravu a parkování. *CITIQ* [online]. [cit. 2019-12-21]. Dostupné z: <http://www.citiq.cz/technologie-doprava.html>
- [47] O projektu SmartCAMPUS ZČU. *SmartCAMPUS ZČU* [online]. [cit. 2020-06-16]. Dostupné z: <https://www.smartcampus.cz/>
- [48] LoRaWAN Gateway LORATECH. *LORATECH* [online]. 2017 [cit. 2020-06-16]. Dostupné z: <http://chytra-obec.cz/services/lorawan-gateway-loratech/>
- [49] Gateway - Loratech Shop. *LORATECH* [online]. [cit. 2020-06-16]. Dostupné z: http://eshop.loratech.cz/index.php?id_category=13&controller=category&id_lang=1
- [50] NASLER, Antonín. Provozní řád parkoviště ZČU v Plzni - parkoviště č. 4, Univerzitní 26. *Západočeská univerzita v Plzni* [online]. 2015 [cit. 2020-03-02]. Dostupné z: [60](https://ps.zcu.cz/export/sites/ps/dokumenty/provozni-rady-parkovist/provozni-rad-</p></div><div data-bbox=)

- parkoviste-c_4.pdf
- [51] VITUŠKO, Alexandr a Zbyněk BENEŠ. Závory na parkovištích ZČU: FEL, FUD a Kaplířova ulice - ELEKTROINSTALACE. *Západočeská univerzita v Plzni*. 2014.
- [52] MRÁČEK, Jakub. jak otevírat data? *Fond Otakara Motejla* [online]. 2014 [cit. 2020-03-02]. Dostupné z: https://osf.cz/wp-content/uploads/2014/06/Jak_otevirat_data.pdf
- [53] About the LoRaWAN® Specification. *LoRa Alliance* [online]. [cit. 2020-03-23]. Dostupné z: <https://loro-alliance.org/lorawan-for-developers>
- [54] LoRaWAN™ 1.0.3 Specification. *LoRa Alliance* [online]. 2018 [cit. 2020-03-23]. Dostupné z: <https://loro-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf>
- [55] Mapy pokrytí. *IoT portál* [online]. 2018 [cit. 2020-03-23]. Dostupné z: <https://www.iot-portal.cz/mapa-pokryti/>
- [56] LoRaWAN. *onlinetechnology.cz* [online]. 2019 [cit. 2020-03-23]. Dostupné z: <http://wiki.merenienegie.cz/index.php/LoRaWAN>
- [57] Internet věcí Plzeň. *Plzeň LoRa* [online]. 2017 [cit. 2020-03-23]. Dostupné z: <https://iot.plzen.eu/>
- [58] CENÍK TELEKOMUNIKAČNÍCH SLUŽEB pro IoT síť pro Internet věcí na technologii LoRa. *STARNET* [online]. 2018 [cit. 2020-03-23]. Dostupné z: <https://www.starnet.cz/download/cenik-iot.pdf>
- [59] TTN-GW-868 - Síťová Brána, The Things Network, LoRaWAN, IoT Vývoj, Pouze EU (868MHz). *Farnell* [online]. [cit. 2020-03-23]. Dostupné z: <https://cz.farnell.com/the-things-network/ttn-gw-868/the-things-gateway-eu/dp/2675813>
- [60] TTN-UN-868 - Jednodeskový Počítač, The Things Network, LoRaWAN, Arduino UNO, IoT Vývoj, Pouze EU (868MHz). *Farnell* [online]. [cit. 2020-03-23]. Dostupné z: <https://cz.farnell.com/the-things-network/ttn-un-868/the-things-uno-eu/dp/2675815>
- [61] POSPÍŠIL, Jan. *Lokalizace v rámci technologie LoRaWAN* [online]. 2019 [cit. 2020-03-23]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=190150
- [62] POSPÍŠILÍK, Jiří. *Lokalizace senzorů v síti LoRaWAN* [online]. 2018 [cit. 2020-03-23]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/76355/F8-DP-2018-Pospisilik-Jiri-thesis.pdf>
- [63] CITIQ LORA traffic counting detector. *CITIQ*. 2017.
- [64] Arduino MKR WAN 1300, Arduino Zero s LoRaWAN. *Arduino-shop.cz* [online]. [cit. 2020-04-12]. Dostupné z: <https://arduino-shop.cz/arduino/3229-arduino-mkr-wan-1300-arduino-zero-s-lorawan.html>
- [65] How to connect MKRWAN 1300 and MKRWAN 1310 to The Things Network (TTN). *Arduino* [online]. 2019 [cit. 2020-04-14]. Dostupné z: <https://www.arduino.cc/en/Tutorial/MKRWANTTNFirstConfiguration>
- [66] First Configuration. *Arduino* [online]. 2017 [cit. 2020-04-14]. Dostupné z: <https://www.arduino.cc/en/Tutorial/MKRWANFirstConfiguration>
- [67] LoRa Send And Receive. *Arduino* [online]. 2017 [cit. 2020-04-14]. Dostupné z: <https://www.arduino.cc/en/Tutorial/MKRWANLoraSendAndReceive>
- [68] *The Things Network* [online]. [cit. 2020-04-12]. Dostupné z: <https://www.thethingsnetwork.org/>
- [69] Přihlášení do systému LORATECH-TEST. *LORATECH* [online]. [cit. 2020-04-14]. Dostupné z: <https://test.loratech.cz/login>
- [70] Nový LORATECH NETWORKSERVER. *LORATECH* [online]. 2016 [cit. 2020-04-12]. Dostupné z: <http://chytra-obec.cz/2016/11/28/novy-loratech-networkserver/>
- [71] BĚLOHOUBEK, Jan a Radek VOZÁK. LORATECH dokumentace. *LORATECH* [online]. 2018. Dostupné z: <https://app.loratech.cz/manual.pdf>

- [72] Decode from Base64 format. *Base64 Decode and Encode* [online]. [cit. 2020-04-14]. Dostupné z: <https://www.base64decode.org/>
- [73] ASCII text,Hex,Binary,Decimal,Base64 converter. *RapidTables* [online]. [cit. 2020-06-03]. Dostupné z: <https://www.rapidtables.com/convert/number/ascii-hex-bin-dec-converter.html>
- [74] ThingSpeak for IoT Projects. *ThingSpeak* [online]. [cit. 2020-06-12]. Dostupné z: <https://thingspeak.com/>
- [75] Securing Node-RED. *Node-RED* [online]. [cit. 2020-06-16]. Dostupné z: <https://nodered.org/docs/user-guide/runtime/securing-node-red>
- [76] Why does new.Date() differs 1 hour from new.Date().toISOString()? - Stack Overflow. *Stack Overflow* [online]. 2015 [cit. 2020-06-14]. Dostupné z: <https://stackoverflow.com/questions/29477072/why-does-new-date-differs-1-hour-from-new-date-toisostring>
- [77] JavaScript. *JavaScript.com* [online]. [cit. 2020-06-03]. Dostupné z: <https://www.javascript.com/>
- [78] What is PHP? *PHP.net* [online]. [cit. 2020-06-03]. Dostupné z: <https://www.php.net/manual/en/intro-what-is.php>
- [79] Textový editor PSPad. *PSPad* [online]. [cit. 2020-06-03]. Dostupné z: <http://www.pspad.com/cz/>
- [80] What can PHP do? *PHP.net* [online]. [cit. 2020-06-03]. Dostupné z: <https://www.php.net/manual/en/intro-whatcando.php>
- [81] ZAJÍC, Petr. PHP (2) - Jak to funguje. *Linuxsoft.cz* [online]. 2004 [cit. 2020-06-03]. Dostupné z: http://archiv.linuxsoft.cz/article.php?id_article=172
- [82] XAMPP Apache + MariaDB + PHP + Perl. *Apache Friends* [online]. [cit. 2020-06-03]. Dostupné z: <https://www.apachefriends.org/index.html>
- [83] SHAH, Nawaraj. How to Fetch Data from Database in PHP and display in HTML Table? *CodeAndCourse.com* [online]. [cit. 2020-06-03]. Dostupné z: <https://www.codeandcourse.com/how-to-display-data-from-mysql-database-into-html-table-using-php/>
- [84] Create-Dynamic-Moris-Charts-with-Database. *GitHub* [online]. 2019 [cit. 2020-06-12]. Dostupné z: <https://github.com/Abdullahmasood553/Create-Dynamic-Moris-Charts-with-Database>
- [85] What is jQuery? *jQuery* [online]. [cit. 2020-06-12]. Dostupné z: <https://jquery.com/>
- [86] Getting started - Bootstrap. *Bootstrap* [online]. [cit. 2020-06-12]. Dostupné z: <https://getbootstrap.com/docs/3.4/getting-started/>
- [87] SMITH OLLY. morris.js. *morris.js* [online]. 2013 [cit. 2020-06-12]. Dostupné z: <https://morrisjs.github.io/morris.js/>
- [88] jQuery Date and Time picker. *jQuery Plugin Registry* [online]. 2014 [cit. 2020-06-12]. Dostupné z: <https://plugins.jquery.com/datetimetypepicker/>
- [89] PHP: strtotime. *PHP.net* [online]. [cit. 2020-06-12]. Dostupné z: <https://www.php.net/manual/en/function.strftime.php>
- [90] MySQL - Subtracting value from previous row, group by. *Stack Overflow* [online]. 2002 [cit. 2020-06-12]. Dostupné z: <https://stackoverflow.com/questions/13196190/mysql-subtracting-value-from-previous-row-group-by>
- [91] DUŠAN JANOVSKEJ. Tabulky. *Jak psát web* [online]. [cit. 2020-06-12]. Dostupné z: <https://www.jakpsatweb.cz/html/tabulky.html>
- [92] CSS Layout - Horizontal & Vertical Align. *W3Schools* [online]. [cit. 2020-06-12]. Dostupné z: https://www.w3schools.com/css/css_align.asp