

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

Přenos telemetrie přes IoT

Bc. Filip Malý

© 2018 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Filip Malý

Informatika

Název práce

Přenos telemetrie přes IoT

Název anglicky

Transmission of telemetry via IoT

Cíle práce

Hlavním cílem práce je realizovat přenos telemetrie prostřednictvím Internet of Things (IoT) rozhraní. Dílčími cíli diplomové práce jsou analýza IoT rozhraní včetně porovnání se stávajícími nejběžnějšími poskytovateli IoT, dále pak tvorba aplikace pro komunikaci s telemetrickým zařízením prostřednictvím IoT a nakonec implementace komunikace s prototypem dronu.

Metodika

Rešeršní část diplomové práce bude obsahovat analýzu odborné a vědecké literatury. V praktické části bude jako základní metodický postup použita metoda vícekriteriální analýzy variant. Pro charakteristiku přenosu telemetrie prototypu bude využita metoda UML, dále pro tvorbu přenosu a výsledné aplikace budou použity jazyky Python, JavaScript a SQL. Na základě vícekriteriální analýzy variant bude vybrán výchozí poskytovatel IoT rozhraní. Aplikace bude vytvořena pomocí platformy SAP FIORI.

Doporučený rozsah práce

50

Klíčová slova

IoT, DRONY, RPI, Internet, Fiori, Telemetrie

Doporučené zdroje informací

ANTOLOVIC, Miroslav. Getting started with SAPUI5. ISBN 9781592299690.

BURIAN, P. *Internet inteligentních aktivit*. Praha: Grada, 2014. ISBN 978-80-247-5137-5.

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. KATEDRA OPERAČNÍ A SYSTÉMOVÉ ANALÝZY, – ŠUBRT, T.

Ekonomicko matematické metody II : aplikace a cvičení. Praha: ČZU PEF Praha ve vydavatelství Credit, 2001. ISBN 80-213-0721-8.

FOWLER, Martin. Destilované UML. Praha: Grada, 2009. Knihovna programátora (Grada). ISBN 9788024720623.

LAFAY, Mark. Drones for Dummies. –For dummies. ISBN 9781119049784.

Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

Ing. Tomáš Vokoun

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 31. 10. 2017

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 1. 11. 2017

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 18. 03. 2018

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Přenos telemetrie přes IoT" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 29.3.2018

Poděkování

Rád bych touto cestou poděkoval vedoucímu této práce, Ing. Tomášovi Vokounovi, za jeho odborné vedení, rady a vstřícnost. Také bych rád poděkoval vedení a kolegům ze společnosti KCT Data s.r.o za umožnění vzniku této práce a pomoc při jejím zpracování. Dále bych chtěl poděkovat Anetě Koudelkové za vstřícnost a pochopení při tvorbě této diplomové práce.

Přenos telemetrie přes IoT

Abstrakt

Tato diplomová práce se zabývá problematikou přenosu telemetrie přes Internet věcí (Internet of Things), zkráceně IoT. Práce je zejména zaměřena na přenos telemetrie z dronu, využití počítačů Arduino a Raspberry Pi, vybrání nejvhodnějšího IoT poskytovatele a tvorbu obslužné responsivní aplikace.

Práce je rozčleněna do dvou částí – teoretické a praktické. V teoretické části jsou definovány pojmy související s přenosem telemetrie přes IoT a technologie pro uskutečnění přenosu z dronu. Následuje představení potřebných termínů a technologie s vysvětlením řešené problematiky. Dílčí částí teoretické části je popsání aktuálních IoT projektů a současných technologií pro uskutečnění přenosu telemetrie.

Praktická část se pak zabývá analýzou IoT platforem pro nejlepší přenos telemetrie z dronu. Na základě Saatyho a Bodovací metody z vícekritériální analýzy variant je vybrán nejvhodnější poskytovatel IoT: Sap Cloud Platform. Dílčí částí je pak uskutečnění přenosu telemetrie z dvou nezávislých dronů s jednotkami Arduino a Raspberry Pi 2 a 3. Poslední dílčí částí praktické části je tvorba aplikace pro zobrazení telemetrických dat, odesílání příkazů do dronů a správu údajů o dronech. Telemetrie byla pak obousměrně přenesena, dron poslal GPS údaje o poloze, výšce a další letové hodnoty. Dle příkazu z aplikace pak dron zaslal aktuální fotografii z místa a zapnul alarm.

Klíčová slova: Přenos telemetrie přes IoT, IoT, dron, Raspberry Pi, Arduino, SP Racing, Fiori, SAPUI5, cloud, telemetrie, Multiwii, CleanFlight

Transmission of telemetry via IoT

Abstract

This diploma thesis deals with issues of transmission of telemetry via IoT. The work is mainly focused on transferring telemetry from the drone, use of Arduino and Raspberry Pi computers. Work continues selection of the most suitable IoT provider and creation of responsive applications.

The thesis is divided into two parts - theoretical and practical. In the theoretical part are defined the concepts related to IoT telemetry transmission and the technology for transferring telemetry from drones. The following is a presentation of the necessary terms and technologies with an explanation of the main problems. A part of the theoretical part is the description of current IoT projects and present technologies for telemetry transmission.

The practical part deals with the analysis of IoT platforms for choosing the best IoT platform with which author realized transfer telemetry from drone. Choosing is based on Saaty's and Scoring Method of Multiple-criteria decision analysis. Sap Cloud Platform was found out as the most appropriate IoT provider. The next subsection is the telemetry transmission from two independent drones with Arduino and Raspberry Pi 2 and 3. The last part of the practical part is the creation of an application for displaying telemetric data, sending request to drones and managing drone's data. Telemetry was transmitted, the drone sent GPS position, altitude, and other flight data. By the command from the application the drone sent the photo from the location and turned on the alarm.

Keywords: Transmission of telemetry via IoT, IoT, drone, Raspberry Pi, Arduino, SP Racing, Fiori, SAPUI5, cloud, telemetry, Multiwii, CleanFlight

Obsah

1 Úvod.....	13
2 Cíl práce a metodika	14
2.1 Cíl práce	14
2.2 Metodika	14
3 Teoretická východiska	15
3.1 Úvod do IoT	15
3.1.1 Koncepce IoT	15
3.1.2 Síť v IoT.....	15
3.1.3 Věc v IoT	15
3.1.4 Základní prvky v IoT	16
3.1.5 Architektura IoT	16
3.1.6 Tří a pětivrstvá architektura IoT	17
3.1.7 Cloud architektura.....	18
3.1.8 Fog architektura	18
3.1.9 Způsoby propojení v rámci IoT	19
3.1.10 Požadavky na IoT	20
3.1.11 Směry vývoje IoT	21
3.1.12 Technologie IoT v ČR	21
3.1.13 Technologie pro bezdrátový přenos dat.....	26
3.1.14 Mapy pokrytí IoT u nás	29
3.1.15 Významné IoT projekty	30
3.2 Arduino a Raspberry Pi.....	32
3.2.1 Základní informace o Arduino.....	32
3.2.2 Využití Arduino	32
3.2.3 Základní typy Arduino pro drony	33
3.2.4 Multiwii	35
3.2.5 CleanFlight.....	35
3.2.6 BetaFlight.....	35
3.2.7 Raspberry Pi.....	35
3.2.8 Modely a verze Raspberry Pi.....	36
3.2.9 OS pro Raspberry Pi	38
3.2.10 Moduly pro Raspberry Pi.....	39
3.3 Multikoptéry – drony	40
3.3.1 Základní terminologie.....	40
3.3.2 Tricopter.....	40
3.3.3 Quadcopter	40

3.3.4	Hexacopter	41
3.3.5	Octocopter.....	41
3.3.6	Ostatní typy dronů	42
3.3.7	Telemetrie	42
3.4	Fiori.....	43
3.4.1	SAP	43
3.4.2	Fiori.....	43
3.4.3	SAP Fiori designová stránka.....	44
3.4.4	Základní architektura Fiori aplikací.....	44
3.4.5	MVC architektura	45
4	Vlastní práce	46
4.1	Srovnání IoT poskytovatelů	46
4.1.1	Použité metody vícekritériální analýzy variant	46
4.1.2	Nejznámější IoT poskytovatelé pro rok 2017 a 2018	47
4.1.3	Vstupní data	52
4.1.4	Kritéria pro vícekritériální analýzu dat.....	53
4.1.5	Výsledky Saatyho metody	60
4.1.6	Výsledky bodovací metody	60
4.1.7	Výsledky bodovací metody s váhami kritérií	60
4.1.8	Zhodnocení výsledků a vybrání vhodného poskytovatele	61
4.2	Přenos telemetrie přes IOT.....	62
4.2.1	Drony pro přenos telemetrie	62
4.2.2	IoT struktura	63
4.2.3	Schéma komunikace dronu s IoT.....	64
4.2.4	Databáze v SAP Cloud Platform	65
4.2.5	Schéma databáze.....	65
4.2.6	Nastavení xsodata pro aplikaci	66
4.2.7	Komunikace s iot-services mms.....	66
4.2.8	Získání telemetrie z dronu	67
4.2.9	Propojení IoT Serveru s dronem.....	68
4.2.10	Nastavení RPI pro připojení k Internetu	68
4.2.11	Nastavení RPI import skriptů pro čtení dat	69
4.2.12	Nastavení RPI na automatické spuštění.....	70
4.2.13	Nastavení RPI pro display 480*320px	71
4.3	Tvorba aplikace.....	72
4.3.1	Základní požadavky na aplikaci	72
4.3.2	Architektura aplikace	72
4.3.3	Knihovny využité v aplikaci	73
4.3.4	Návrh aplikace	73

4.3.5	Popis základních obrazovek.....	73
4.3.6	Výchozí stav aplikace pro jednotlivá zařízení	76
4.3.7	Postup implementace aplikace.....	76
4.3.8	Princip komunikace aplikace se službou IoT	76
4.3.9	Zdrojový kód aplikace	77
4.3.10	Potenciální rozvoj aplikace.....	77
5	Výsledky a diskuse	78
5.1	Shrnutí vícekritériální analýzy variant	78
5.2	Shrnutí vytvořené aplikace.....	78
5.3	Shrnutí přenesené obousměrné telemetrie	78
5.4	Vytvořené prototypy pro přenos telemetrie	79
5.5	Přílohy na CD disku	79
6	Závěr.....	80
7	Seznam použitých zdrojů	82
8	Přílohy	86

Seznam obrázků

Obrázek 1	rozdíl mezi 3. a 5. vrstvou architekturou IoT (2)	18
Obrázek 2	Fog architektura (7).....	19
Obrázek 3	propojení zařízení v IoT (1 str. 186)	19
Obrázek 4	Rozdíl mezi Fogem a Cloudem (3)	20
Obrázek 5	Směry vývoje IoT (3)	21
Obrázek 6	Logo MQTT (11)	22
Obrázek 7	MiWi (12).....	23
Obrázek 8	NB- IoT (13).....	23
Obrázek 9	Wi-Fi Halow (14).....	24
Obrázek 10	sigfox (15)	24
Obrázek 11	LoRaWan (16).....	24
Obrázek 12	Z-Wave (17)	25
Obrázek 13	Bluetooth SMART (18).....	25
Obrázek 14	ZigBee (19)	26
Obrázek 15	LoRaWAN pokrytá červen 2017 (23).....	29
Obrázek 16	LoRaWan finální pokrytí 2017 (23).....	30
Obrázek 17	Logo Arduino	32
Obrázek 18	Logo Raspberry Pi.....	32
Obrázek 19	Crius Multiwii v2.5	33
Obrázek 20	Crius AOIP 2	34
Obrázek 21	SP Racing F3	34
Obrázek 22	Počítač Raspberry Pi (36).....	36
Obrázek 23	Tricopter	40
Obrázek 24	Quadcoptera.....	41
Obrázek 25	Hexacoptera.....	41

Obrázek 26 Octocoptera	41
Obrázek 27 Reversivní octocoptera.....	42
Obrázek 28 platformy pro SAP Fiori (52 str. 1).....	43
Obrázek 29 Architektura Fiori (52 str. 19)	45
Obrázek 30 Struktura IoT modulu pro přenos telemetrie.....	67
Obrázek 31 architektura MVC struktura aplikace	72

Seznam tabulek

Tabulka 1 srovnání technologií IoT (22)	29
Tabulka 2 kritéria pro hodnocení dle Webu	54
Tabulka 3 kritéria pro hodnocení dle Licence	54
Tabulka 4 kritéria pro hodnocení dle Protokolů	55
Tabulka 5 kritéria pro hodnocení dle Ceny	55
Tabulka 6 kritéria pro hodnocení dle Online školení	56
Tabulka 7 kritérium pro hodnocení dle Vlastní sady aplikací	56
Tabulka 8 kritérium pro hodnocení dle Dema	56
Tabulka 9 kritéria pro hodnocení dle Dema	57
Tabulka 10 kritérium pro hodnocení dle API dokumentace.....	57
Tabulka 11 kritérium pro hodnocení dle Vlastní knihovny	57
Tabulka 12 kritéria pro hodnocení dle Programovacích jazyků	58
Tabulka 13 kritéria pro hodnocení dle Průmyslového odvětví.....	58
Tabulka 14 kritéria pro hodnocení dle Hardwarové podpory.....	59
Tabulka 15 kritéria pro hodnocení dle Konektivity.....	59
Tabulka 16 kritérium pro hodnocení dle Databáze.....	59
Tabulka 17 kritérium pro hodnocení dle Vlastního hodnocení	60

Seznam příloh

Příloha 1 bodování pro kritérium Web	86
Příloha 2 bodování pro kritérium Licence	86
Příloha 3 bodování pro kritérium Protokoly	87
Příloha 4 bodování pro kritérium Cena.....	87
Příloha 5 bodování pro kritérium Online školení	88
Příloha 6 bodování pro kritérium Vlastní sada aplikací	88
Příloha 7 bodování pro kritérium Demo	89
Příloha 8 bodování pro kritérium Git.....	89
Příloha 9 bodování pro kritérium API dokumentace	90
Příloha 10 bodování pro kritérium Vlastní knihovny	90
Příloha 11 bodování pro kritérium Programovací jazyky.....	91
Příloha 12 bodování pro kritérium Průmyslové odvětví.....	91
Příloha 13 bodování pro kritérium Hardware	92
Příloha 14 bodování pro kritérium Konektivita	92
Příloha 15 bodování pro kritérium Databáze	92
Příloha 16 bodování pro kritérium Vlastní hodnocení.....	93
Příloha 17 tabulka preferencí pro Saatyho metodu.....	93
Příloha 18 výsledná tabulka vah kritérií	93
Příloha 19 Tabulka vstupních dat pro Saatyho metodu	94
Příloha 20 Výsledná tabulka pro Bodovací metodu	95

Příloha 21 tabulka pět nejlepších poskytovatelů Bodovací metodou	96
Příloha 22 tabulka pět nejhorších poskytovatelů Bodovací metodou	96
Příloha 23 tabulka pět nejlepších poskytovatelů Bodovací metodou s váhami	96
Příloha 24 tabulka pět nejlepších poskytovatelů Bodovací metodou s váhami	97
Příloha 25 tabulka s body a váhami kritérií	98
Příloha 26 Skript pro nastavení modemu přes wvdial	99
Příloha 27 nastavení Message Types	99
Příloha 28 IoT zpráva DATA_FROM_DRONE	100
Příloha 29 IoT zpráva CMD_TO_DRONE	100
Příloha 30 IoT zpráva PHOTO_FROM_DRONE	101
Příloha 31 funkce na uložení příkazu odeslaného do dronu	101
Příloha 32 funkce na mazání ze seznamu dronů	101
Příloha 33 funkce na aktualizaci záznamu o dronu	102
Příloha 34 iotsenario.xsodata služba pro Fiori aplikaci	102
Příloha 35 Stavový diagram propojení dronu s IoT	103
Příloha 36 schéma databáze pro IoT	103
Příloha 37 Třída Config.py	104
Příloha 38 Třída Main.py	106
Příloha 39 master view pro výběr dronu a zobrazení informací	109
Příloha 40 prvek pro výběr dronu	110
Příloha 41 prvek na aktualizaci dat z databáze	110
Příloha 42 master view pro editaci informací o dronu	111
Příloha 43 potvrzovací dialog pro mazání dronu	112
Příloha 44 dialog přidání nového dronu	112
Příloha 45 detailní obrazovka Monitoring ve výchozím stavu	113
Příloha 46 detailní obrazovka Monitoring s vyznačenou dráhou letou	113
Příloha 47 detailní obrazovka Requesty	114
Příloha 48 detailní obrazovka Fotografie	114
Příloha 49 detailní obrazovka DB tabulka	115
Příloha 50 dialog pro filtrování dle datumu	115
Příloha 51 mazání více záznamů z tabulky a informativní toolbar	116
Příloha 52 detailní obrazovka O Aplikaci	116
Příloha 53 aplikace ve výchozím stavu na desktopu	117
Příloha 54 zobrazení aplikace na zařízení tablet	118
Příloha 55 zobrazení aplikace na mobilním zařízení	119
Příloha 56 stavový diagram popisující aplikaci	120
Příloha 57 odeslání telemetrie z dronu	120
Příloha 58 přijmutí příkazu z aplikace v dronu	121
Příloha 59 odeslání příkazu do dronu	121
Příloha 60 odeslání příkazu do dronu Response	122
Příloha 61 quad dron s jednotkou RPI 2	122
Příloha 62 dron s RPI 2 detail	123
Příloha 63 hexa dron s jednotkou RPI 3	123
Příloha 64 dron s RPI 3 detail	124

1 Úvod

Internet věcí dále jen IoT je v angličtině nazýván Internet of Things. Je to technologie, která se stává stále dostupnější a oblíbenější u širší veřejnosti. Zabývá se přenosem informací z chytrých zařízení, která jsou napojena na celosvětovou síť Internet.

Propojení těchto zařízení by mělo být zejména bezdrátové, nejčastěji se pro tyto účely využívá wifi, GSM modul anebo bluetooth. IoT přináší nejen nové možnosti vzájemné interakce mezi jednotlivými systémy, ale i nové možnosti jejich ovládání, sledování a zajištění pokročilých služeb.

IoT zařízení může být široká škála: od chytrých žárovek přes automatické zalévání až po fotopasti a smart automobily. Jednoduše se dá říci, že vše, co automaticky komunikuje se vzdáleným zařízením, spadá do kategorie IoT.

V posledních letech se objevuje značný technologický boom v odvětví chytrých autonomních zařízení UAV s podkategorií dronů. Tyto létající stroje se staly velmi oblíbeným a užitečným koníčkem tisíců lidí a pomalu pronikají do různých odvětví lidských činností. Tato zařízení mohou například monitorovat těžce dostupná místa, kontrolovat úrodu, hlídat požáry, snímat budovy, natáčet filmy, doručovat zásilky, a mnoho dalších činností, u kterých je potřeba stabilita a mobilita.

Tato práce se bude zabývat obousměrným přenosem telemetrických dat z dronu pomocí principů IoT a bezdrátové sítě. Cílem této práce je vytvořit prototyp bezdrátové komunikace s dronem s využitím poskytovatele IoT. Dron bude odesílat různorodá data přes síť IoT a bude přijímat příkazy z aplikace pro monitorování. Záměrem této práce je propojit dron skrze IoT a připravit tak podklady pro bezdrátové ovládání, volbu dráhy letu a dalších vymožeností.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavním cílem práce je realizovat přenos telemetrie prostřednictvím Internet of Things (IoT) rozhraní. Dílčími cíli diplomové práce jsou analýza IoT rozhraní včetně porovnání se stávajícími nejběžnějšími poskytovateli IoT, dále pak tvorba aplikace pro komunikaci s telemetrickým zařízením prostřednictvím IoT a nakonec implementace komunikace s prototypem dronu.

2.2 Metodika

Rešeršní část diplomové práce bude obsahovat analýzu odborné a vědecké literatury. V praktické části bude jako základní metodický postup použita metoda vícekriteriální analýzy variant. Pro charakteristiku přenosu telemetrie z prototypu bude využita metoda UML, pro implementaci přenosu a tvorbu výsledné aplikace budou použity jazyky Python, JavaScript a SQL. Na základě vícekriteriální analýzy variant bude vybrán výchozí poskytovatel IoT rozhraní. Aplikace bude vytvořena pomocí platformy SAP FIORI s architekturou MVC.

3 Teoretická východiska

3.1 Úvod do IoT

V následující kapitole je představeno IoT, základní termíny, technologie, směry a poslední trendy vývoje.

3.1.1 Koncepce IoT

Termín IoT Internet of Things, v češtině Internet věcí, je vysvětlován mnoha autory různorodě, autor této práce se nejvíce přiklání k této definici: „Internet věcí umožňuje uživatelům a sdíleným objektům sdílení informací hladkým, automatizovaným způsobem. V této souvislosti slibuje Internet věcí vytvoření nové generace na síti Internet, ve které se pohybuje směrem k celosvětové propojitelnosti předmětů, věcí a objektů s výrazným rozšířením oblastí působení aplikací. (1 str. 242)

Koncept IoT byl poprvé zmíněn v roce 1999. Tato technologie se skládá z integrace různých technologií zahrnující senzory, komunikaci, sítě, platformu jako servis (SoA) a inteligentní informační procesy. Dále přináší významný počet změn, jako jsou bezpečnost, integrace hybridních internetových technologií, technologii snímání a další. Podstatnou část v IoT hraje zabezpečená komunikace, která brání útočníkům napadnout zařízení. (2 str. 1)

Důležitým bylo pro IoT období mezi lety 2008 a 2009, kdy podle odhadu společnosti Cisco překročil počet zařízení připojených k síti Internet počet světové populace. V tomto období se datuje vznik Internetu věcí. (3)

3.1.2 Síť v IoT

Síť v IoT nemusí vysvětlovat pouze Internet, jak lze pochopit z překladu názvu. Je to celosvětový systém navzájem propojených počítačových sítí, ve kterých mezi sebou počítače komunikují prostřednictvím rodiny protokolů TCP/IP, ale může to i znamenat komunikaci po lokální síti LAN, v rámci které tyto věci mohou komunikovat. Díky Internetu tato zařízení mohou sdílet výsledky a Internetová síť tak rozšiřuje jejich konektivitu. (3)

3.1.3 Věc v IoT

Věc v popisu IoT znázorňuje neživý objekt, a to buď fyzický nebo virtuální, který obsahuje elektroniku, software a senzory, pomocí kterých snímá určitou veličinu nebo veličiny, a poskytuje jim schopnost sloužit k danému účelu. Do této kategorie tedy spadá

zařízení (systém), které autonomně poskytuje data, která jsou kabelově nebo bezdrátově sdílena s dalšími věcmi nebo systémy. V IoT nejsou základem věci, ale data, která tyto věci poskytují. IoT tedy představuje koncept, v rámci kterého si fyzické a virtuální objekty (věci) vyměňují data přes síť Internet. (3)

3.1.4 Základní prvky v IoT

Základním stavebním kamenem celého IoT jsou data. V první fázi jsou tato data nezpracovaná – například řetězec znaků 123456. Přiřazením souvislosti (kontextu) k datům vzniká informace. Pokud řetězec znaků 123456 obohatíme o atribut „heslo“, dostaneme informaci. Kdybychom přidali více atributů k řetězci 123456, dostaneme znalost a následným aplikováním bychom dostali porozumění. (3) (4 str. 199)

V IoT se velmi často také objevuje termín „Big Data“. Tento výraz lze vysvětlit jako ohromné množství dat, které se stává limitujícím pro nástroje a úložiště, které s těmito daty pracují. Lze předpokládat, že se tento termín bude v budoucích letech více dostávat do podvědomí, a jedním z klíčových prvků bude IoT. (5 str. 13)

Obecně lze strukturu IoT rozdělit do tří komponent: Hardware, Middleware a Software. Komponenty jsou detailněji popsány v následujících bodech.

- Pojmem hardware lze popsat veškeré fyzické součásti systému (senzory, komunikační infrastruktura a datová úložiště).
- Middleware je programové vybavení součástí IoT systému, které vytváří rozhraní mezi systémy.
- Software jsou pak programy, které slouží pro správu, analýzu a ukládání získaných dat. (6 str. 140) (3)

3.1.5 Architektura IoT

Neexistuje přesný předpis, jak má IoT architektura vypadat, a z čeho se má skládat. Různé návrhy architektury byly popisovány mnoha odborníky. V IoT se využívají základní čtyři typy architektury, které jsou popsány v následujících kapitolách. Architektury jsou následující:

- třívrstvá architektura,
- pětivrstvá architektura,
- fog architektura,

- cloud architektura. (7)

3.1.6 Tří a pětivrstvá architektura IoT

Základní architektura, která byla představena v počátcích vývoje IoT, je třívrstvá. Skládá se ze tří vrstev perception (vjemové), network (síťové) a application layer (aplikační vrstvy). Tato architektura definuje základní principy IoT.

Třívrstvá architektura se tedy skládá z těchto částí:

- Perception layer je fyzická vrstva, která má na starosti sbírání informací z okolí.
- Network layer je síťová vrstva, která je odpovědná za spojení s chytrými věcmi, servery a zařízeními. Využívá se zejména pro přenos a zpracování dat.
- Application layer je aplikační vrstva, která je v interakci s uživateli. Definuje aplikace, které se využívají v IoT, jako jsou například chytré domácnosti, chytrá města a chytré zdravotnictví.

Pětivrstvá architektura vychází z třívrstvé, ale jsou zde zahrnuté další vrstvy, které lépe vystihují funkčnost IoT. Z původní architektury zůstaly vrstvy perception a application, naopak přibyly vrstvy transport (transportní), processing (procesní) a business (obchodní).

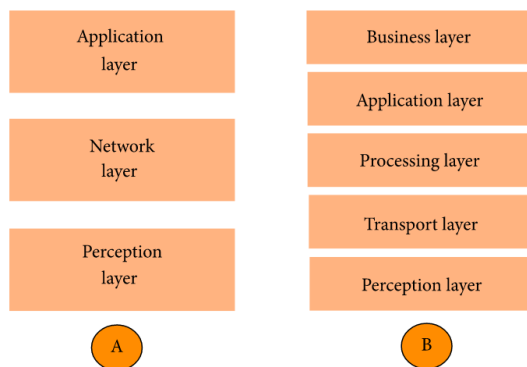
Pětivrstvá architektura se tedy skládá z následujících částí:

- Business layer je obchodní vrstva, která zahrnuje aplikace, obchod, obchodní modely a uživatelské rozhraní. Je to velmi rozsáhlá vrstva, která v této práci nebude více popisována.
- Application layer stejný význam jak u třívrstvé architektury.
- Processing layer je známa pod pojmem prostřední vrstva. Ukládá, analyzuje a zpracovává ohromné množství dat, které přichází z transportní vrstvy. Může ovládat služby nižších vrstev. Využívá mnoho technologií, jako jsou databáze, cloud computing a big data procesní moduly.
- Transport layer je transportní vrstva. Přenáší data z percepiton vrstvy přes Internet. Využívá bezdrátové technologie, jakou jsou například 3G, LAN, Bluetooth, RFID a NFC.
- Perception layer má stejný význam jak u třívrstvé architektury.

Zpracováno dle odborné publikace Journal of Electrical and Computer Engineering

Rozdíl mezi třívrstvou a pětivrstvou architekturou je znázorněný v následujícím obrázku.

Obrázek 1 rozdíl mezi 3. a 5. vrstvou architekturou IoT (2)



Zdroj: Shancang Li, Li Da Xu, 2017, (2)

3.1.7 Cloud architektura

V některých systémových architekturách je zpracování dat prováděno velkým centralizovaným způsobem pomocí výpočetních modelů cloudových typů. Tato cloudová architektura udržuje cloud ve středu aplikace nad ním a síť IoT pod ním. Cloud computing má prvenství v poskytování velké flexibility a škálovatelnosti. Nabízí služby jako je základní infrastruktura, platforma, software a úložiště. Vývojáři mohou prostřednictvím cloudu využívat softwarové nástroje pro rychlý vývoj a snazší práci. (8 str. 169) V této práci bude preferována zejména tato architektura.

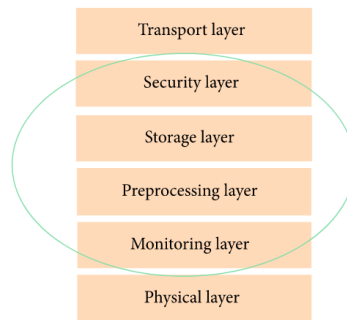
3.1.8 Fog architektura

V poslední době dochází k posunu směrem k jiné systémové architektuře, konkrétně k Fog Computingu, kde senzory a síťové prvky tvoří součást zpracování a analýzy dat. Fog architektura představuje vrstvený přístup, který vkládá vrstvy pro monitorování, předzpracování, ukládání a bezpečnost, mezi vrstvou fyzickou a transportní. (9 str. 122)

- Monitoring layer je monitorovací vrstva, která sleduje výkon, zdroje, reakce a služby.
- Preprocessing layer je vrstva předběžného zpracování, provádí filtrování, zpracování a analýzu údajů ze snímačů.

- Storage layer je dočasná úložná vrstva, která poskytuje funkce pro ukládání dat, jako jsou replikace, distribuce a ukládání dat.
- Security layer je bezpečnostní vrstva, která provádí šifrování nebo dešifrování a zajišťuje integritu a soukromí dat. (7)

Obrázek 2 Fog architektura (7)



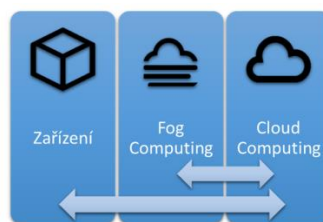
Zdroj: Sarangi, Pallavi Sethi and Smruti R. 2017, (7)

3.1.9 Způsoby propojení v rámci IoT

Propojení mezi prvky IoT systému je vždy závislé na určení systému. Je zde několik způsobů, jakým mohou být prvky mezi sebou propojeny, a jakým způsobem tedy mezi nimi probíhá komunikace. Nejčastější propojení jsou:

- mezi zařízeními navzájem (Fog Computing),
- od zařízení do cloudu,
- mezi cloudy (Cloud Computing). (1 str. 186)

Obrázek 3 propojení zařízení v IoT (1 str. 186)



Zdroj: Pavel, Burian 2014, (1 str. 186)

Komunikace mezi zařízeními (věcmi) se používá v systémech tehdy, kdy využití cloudu pro ukládání, vyhodnocování a sdílení dat není pro dané řešení vyhovující. Může se to stát například v případech, kdy není k dispozici dostatečná kapacita linky pro zaslání všech dat, nebo zaslání velkého množství dat je finančně nákladné. Pro některé systémy

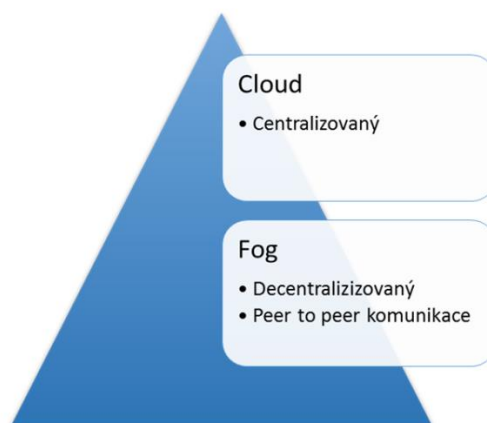
není centralizované cloudové řešení vhodné a místo něho se využívá decentralizovaný Fog computing. (1 str. 198)

Fog computing spolu se zařízeními (věcmi) komunikuje přímo mezi sebou (peer-to-peer). Tato technologie tak doplňuje Cloud Computing o schopnost provádět sběr, ukládání, analýzu a sdílení blíže zařízením a jejich datům (lokální zpracování). Tímto je dosaženo lepší škálovatelnosti, spolehlivosti, rychlejších odezev systému a snížení nákladů. Tento přístup je využíván zejména v Průmyslovém internetu věcí (IIoT), kde je kladen důraz na spolehlivost a definované odezvy systémů. Výsledky zpracování dat jsou poté předávány do cloudu. (1 str. 16)

Další typem komunikace je od zařízení do cloudu. Tento druh se často využívá v rámci spotřebitelského internetu věcí (CIoT). Zde není kladen takový důraz na rychlost odezvy a případný výpadek spojení nezpůsobí žádné ztráty. (3)

Posledním typem komunikace je mezi cloudy (cloudové služby a aplikace a cloudová úložiště). Cloud může být veřejný, privátní, hybridní nebo komunitní. Tento princip komunikace mezi cloudy se používá v případě nutnosti sdílení dat mezi různými doménami. V reálném případě je to tedy mezi privátním a veřejným cloudem. (3)

Obrázek 4 Rozdíl mezi Fogem a Cloudem (3)



Zdroj: Pohanka, Pavel 2017, (3)

3.1.10 Požadavky na IoT

Architektura IoT by měla umožňovat plnit specifické požadavky, které se dělí na hlavní a dílčí. Mezi hlavní požadavky můžeme zařadit tyto:

- Sběr dat, informací a znalostí,
- Uložení dat, informací a znalostí,

- Analýzu dat, informací a znalostí,
- Sdílení výsledků,
- Bezpečnost.

Dále pak mezi dílčí požadavky na IoT můžeme zařadit:

- Efektivní přenos,
- Sdílení dat,
- Volba vhodného přenosového standardu,
- Zpracování velkých objemů dat. (3)

3.1.11 Směry vývoje IoT

Postupem vývoje IoT se vytvořily dva hlavní směry, průmyslový internet věcí (IIoT) a spotřebitelský internet věcí (CIoT).

- IIoT je souhrn IoT zařízení a systémů, které se využívají převážně v průmyslových odvětvích. Patří sem průmyslová automatizace, dopravní průmysl, energetický průmysl a zdravotnictví. (10 str. 41)
- CIoT je naopak IoT, které se primárně zaměřuje na spotřebitele a zařízení, která by mu měla usnadnit život. Do této kategorie spadá automatizace v domácnostech, chytrá zařízení, nebo nositelná elektronika. (3)

Obrázek 5 Směry vývoje IoT (3)



Zdroj: Pohanka, Pavel 2017, (3)

3.1.12 Technologie IoT v ČR

Podle českého portálu „IoT portál“, autor této diplomové práce shrnul nejznámější technologie IoT, které se vyskytují na našem trhu. Autor zpracoval tuto část v druhé polovině roku 2017. Technologie jsou:

- MQTT,
- MiWi,
- NarrowBand IoT,
- Wi-Fi HaLow,
- Sigfox, LoRaWAN,
- Z-Wave,
- Bluetooth Smart,
- ZigBee.

V následujících odstavcích budou shrnuty nejzákladnější informace o těchto technologiích.

MQTT

MQTT je zkratka pro *MQ* Telemetry Transport. Termín je pojmenováním pro mimořádně jednoduchý protokol postavený na rodině TCP/IP protokolů. Byl vyvinut v roce 1999 a je navržený pro jednoduchá zařízení s úzkou šířkou pásma, vysokou latencí, nebo pro nespolehlivé sítě. Základními principy tohoto protokolu jsou minimalizace zatížení sítě a omezení požadavků na zdroje zařízení. (11)

Obrázek 6 Logo MQTT (11)



Zdroj: IoT portál, 2016 (11)

MiWi

MiWi je bezdrátový protokol navržený firmou Microchip Technology na bázi standardu IEEE 802.15.4. Tento protokol je určen pro nízké přenosové rychlosti a krátké vzdálenosti. Najde uplatnění především pro nízkorozpočtové sítě s omezenou pamětí, jako je dálkové řízení a monitoring, automatizované odečty, senzorové sítě atd. Tento protokol pracuje zejména na frekvenci 2,4 GHz a podporuje všechny síťové konfigurace (hvězdice, strom, mesh). (12)

Obrázek 7 MiWi (12)



Zdroj: IoT portál, 2016 (12)

NarrowBand

NarrowBand IoT, dále jen NB-IoT, je nová bezdrátová úzkopásmová LPWA technologie speciálně vyvinutá pro internet věcí. Hlavní předností je možnost nasazení v pásmech GSM a LTE. NB-IoT ctí standardy 3GPP. Tento standard je dohoda o spolupráci mezi telekomunikačními asociacemi z roku 1998 s cílem vyvinout a udržovat 3G sítě, následně také LTE a IMS. (13)

Obrázek 8 NB- IoT (13)



Zdroj: IoT portál, 2016 (13)

Wi-Fi HaLow

Technologie Wi-Fi HaLow je založena na IEEE 802.11ah. Na rozdíl od běžných Wi-Fi sítí pracujících na frekvencích 2,4 a 5 GHz je tato provozována na 900 MHz. Toto pásmo zaručuje širší pokrytí a menší náchylnost k rušení z obvyklého pásma. Datový přenos neprobíhá kontinuálně, ale v pravidelných dávkách, jejichž interval lze nastavit. Nižší vysílací výkon a odlišné schéma tak dovoluje provoz na baterie. Wi-Fi HaLow by měla být přímým konkurentem pro Bluetooth, jen s větším dosahem. (14)

Obrázek 9 Wi-Fi Halow (14)



Zdroj: IoT portál, 2016 (14)

Sigfox

Sigfox je název francouzské firmy, která vytvořila stejně pojmenovanou bezdrátovou technologii určenou ke spojení nízko příkonových zařízení. Těmi mohou být elektroměry, chytré hodinky, automatické pračky apod. Tato zařízení mají být neustále zapnutá, a přitom vysílat malé množství dat. Sigfox využívá technologii ultra nízkého pásma (UNB, Ultra-narrow Band). (15)

Obrázek 10 sigfox (15)



Zdroj: IoT portál, 2016 (15)

LoRaWAN

LoRaWAN (Long Range Wide Area Network) je další z nízko příkonových bezdrátových síťových protokolů vytvořených pro levnou a zabezpečenou obousměrnou komunikaci v internetu věcí. Pracuje v pásmech do 1GHz a rychlost přenosu je pak od 0.3 kb/s do 50 kb/s. Komunikace mezi koncovými segmenty a bránami je rozložena na různá frekvenční pásma a přenosové rychlosti. (16)

Obrázek 11 LoRaWan (16)



Zdroj: IoT portál, 2016 (16)

Z-Wave

Z-Wave je bezdrátová komunikační technologie pro domácí automatizaci. Tato technologie nachází uplatnění v řízení osvětlení, systémech HVAC (Heating, Ventilation, and Air Conditioning), zabezpečovacích systémech, domácích kinech, automatickém ovládání oken a stínění, bazénové technice, či ovládání garáží a přístupových systémech. Z-Wave má přednost v minimalizaci spotřeby energie a je tak vhodný do zařízení napájených bateriemi. Je navržen pro poskytování spolehlivých datových přenosů malých objemů s nízkou latencí a do rychlosti 100 kbit/s. (17)

Obrázek 12 Z-Wave (17)



Zdroj: IoT portál, 2016 (17)

Bluetooth Smart

Bluetooth Smart (BLE, Bluetooth Low Energy, Bluetooth LE) je bezdrátová síťová technologie pro „osobní prostor“, která nachází uplatnění zejména ve zdravotnictví, fitness, bezpečnosti a domácím zábavním průmyslu. Ve srovnání s klasickým Bluetooth má Bluetooth Smart za cíl poskytnout výrazně sníženou spotřebu energie a nákladů při zachování komunikačního dosahu. (18)

Obrázek 13 Bluetooth SMART (18)



Zdroj: IoT portál, 2016 (18)

ZigBee

ZigBee je bezdrátová komunikační technologie vystavěná na standardu IEEE 802.15.4. s novým standardem platným od listopadu 2004. Je určena pro spojení nízko výkonových zařízení v sítích PAN na malé vzdálenosti do 75 metrů. Díky použití multiskokového ad-hoc směrování umožňuje komunikaci i na větší vzdálenosti bez přímé radiové viditelnosti jednotlivých zařízení. Primární určení této technologie směřuje do aplikací v průmyslu a senzorových sítích. (19)

Obrázek 14 ZigBee (19)



Zdroj: IoT portál, 2016 (19)

3.1.13 Technologie pro bezdrátový přenos dat

Komunikace v IoT funguje prostřednictvím zařízení anglicky device nebo bran v překladu gateway. Druhé jmenované jsou nejpoužívanější a umí komunikovat často přes několik rozhraní. Důležitými parametry při výběru přenosové technologie jsou dosah, rychlost přenosu dat, náročnost zpracování, energetická spotřeba či bezpečnost. (20)

Pro bezdrátový přenos se v současnosti používají tyto technologie:

- GSM (Telemetrie, SMART City, dálková ovládání, sběr středního množství dat),
- LTE (Telemetrie, SMART City, dálkové ovládání, sběr většího množství dat),
- Bluetooth (Domácí automatizace – SMART Home),
- WiFi (Domácí automatizace – SMART Home),
- Lora / LoRaWAN (Telemetrie, měřiče energií, posílání dat ze senzorů, SMART city),
- Ad hoc (Domácí automatizace – SMART Home),
- Sigfox (Telemetrie, měřiče energií, posílání dat ze senzorů). (20)

V následujících odstavcích jsou stručně popsány technologie, které se v IoT používají. Technologie 2G, 3G a LTE budou dále využity v praktické části této práce pro přenos telemetrie. Technologie Wifi a Bluetooth (Local Area Network) jsou velmi rozšířené protokoly, které slouží převážně pro komunikaci osobních zařízení. Naopak 2G/3G/4G (Cellular Network) jsou mobilní technologie, které jsou vhodné pro vysokou propustnost dat velkého počtu přístupových zařízení s nutností energetického zdroje.

2G (GSM a GPRS) / 3G

GSM technologie se označuje jako 2. generace bezdrátových telefonních technologií mobilních sítí, známá je pod značkou 2G. Tato technologie je digitální a slouží k přenosu hlasových dat a SMS. Pro přenos dat používá výhradně frekvence 900MHz a 1800MHz. Nástavbou pro 2G sítě je technologie GPRS, která se označuje jako 2.5. GPRS navíc nabízí i možnost přenosu dat na základě protokolů IP a WAP. Přenosová rychlost je nízká – zhruba 20kb/s. Třetí generace je označována jako 3G. Tato technologie zvládá přenos hlasu i elektronických dat. Mobilní technologie 3G pracují v pásmu 2,1 GHz. Technologie umožňuje využívání většího počtu služeb, jako je např. rychlý přenos pomocí nastavy sítě 3G HSDPA. Výhodou těchto technologií je vysoké pokrytí většinou operátorů, nízká cena za přenos dat v rámci EU a spolehlivost. Nevýhodou této technologie je vysoká energetická náročnost, nízká přenosová rychlost a náročnější komunikace mezi moduly. (20)

4G (LTE)

LTE technologie, neboli 4G, je současností nejnovější technologie pro mobilní sítě. Jedná se o 4. generaci mobilních technologií, která je také označována jako LTE (Long Term Evolution). Tato technologie je určena pro vysokorychlostní internet v mobilních sítích. Umožňuje až 10x rychlejší připojení než stávající sítě 2G a 3G. Povoleno rozsah frekvencí pro Evropu je 700 – 2600MHz s přenosovou rychlostí 70-250Mb/s. Výhody této technologie jsou: vysoká přenosová rychlost, rychlé budování sítě, spolehlivost a stejná cena za přenos v rámci EU. Naopak nevýhodou této technologie je vysoká energetická spotřeba a vyšší cena modulů. (20)

Bluetooth

Bluetooth technologie je určena pro bezdrátovou komunikaci více elektronických zařízení na krátkou vzdálenost. Od roku 1994 prošla tato technologie značným vývojem a v současnosti je nejběžnější verze 4.2. Pracuje v omezeném kmitočtovém pásmu od 2,402GHz do 2,483GHz. Tato technologie umožňuje Smart zařízením přístup k internetu bez nutnosti dodatečného zařízení. Maximální přenosová rychlost u Bluetooth je 24Mb/s. Na počátku roku 2017 byla uvedena na trh nová verze Bluetooth 5. Tato verze deklaruje několikanásobně větší dosah a přenosovou kapacitu. Výhodou této technologie je nižší

spotřeba energie, rozšířenost a schopnost komunikovat přímo se smartphony. Nevýhodu této technologie je pak relativně krátký dosah, placená licence, nekompatibilita verzí a přetížené pásmo 2,4Ghz. (20)

WiFi

Poslední technologií v této kapitole je WiFi (Wireless Fidelity). Je to bezdrátová technologie, která využívá pro přenos rádiové vlny v sítích WLAN ve frekvenčním pásmu v rozsahu 2,4 a 5 GHz s rychlostí přenosu až 150MB/s. V současné době je WiFi nejrozšířenější technologií pro bezdrátový přenos dat. Základním prvkem celé sítě je takzvaný přístupový bod (Access point nebo hotspot). Tento bod vysílá signál, který je zařízení schopno rozpoznat a zpracovat. Výhody této technologie jsou: vysoká přenosová rychlost, velmi rozšířená technologie a vysoká bezpečnost. Nevýhodou této technologie je pak vysoká energetická spotřeba, topologie bod-bod. (20)

Ad hoc

Bezdrátová ad hoc síť zvaná WANET nebo MANET je decentralizovaný typ bezdrátové sítě. Místo toho se určité uzly podílí na předávání dat jiným uzlům. Určení je prováděno dynamicky na základě síťové konektivity. Jako ad hoc síť se typicky označuje množina sítí, kde jsou si všechna zařízení rovnocenná a volně spojitelná. Tyto sítě se mohou spojit s jakoukoliv jinou ad hoc sítí v dosahu. (21 str. 15)

V následující příloze je zobrazena tabulka srovnání zmiňovaných sítí v rámci IoT. Nejlepší vlastnosti pro vytvoření připojení mají Lora, LoraWan a Sigfox. Tyto sítě jsou primárně určeny pro posílání malého množství dat na velké vzdálenosti. (22)

Tabulka 1 srovnání technologií IoT (22)

Technologie	Frekvenční pásmo	Šířka pásma	Přenosová rychlost	Maximální vzdálenost	Standard
Bluetooth	2,4GHz	1MHz	cca 1Mb/s	10-100m	vlastní
Wifi	2,4GHz, 5GHz	40MHz	54Mb/s	10-100m	IEEE 802,11
GSM/GPRS (2G)	900/1800MHz	200kHz	172kb/s	10-15km	ETSI
3G	900-2100MHz	5MHz	384kb/s	5-10km	3GPP
LTE (4G)	450-2600MHz	1,4MHz	0,2-1Mb/s	5-11km	3GPP
LoRa	pod 1GHz	125/250kHz	0,25-50kb/s	5-15km	LoRa Aliance
LoRaWAN	pod 1GHz	125/250kHz	0,25-50kb/s	5-15km	LoRa Aliance
Sigfox	pod 1GHz	100Hz	100b/s	10-50km	vlastní
NB-IoT	800MHz	200kHz	20-200kb/s	5-15km	3GPP

Zdroj: IoT portál, 2017, (22)

3.1.14 Mapy pokrytí IoT u nás

Na IoT portále lze nalézt mapy pokrytí pro technologie Sigfox, LoRaWAN a NB-IoT. Všechny tyto technologie plánují do konce roku 2017 pokrýt českou republiku minimálně z 90 %. Jako příklad autor této práce přikládá dvě mapy pokrytí České Republiky s technologií LoRaWAN, kterou využívají České radiokomunikace. Mapy jsou aktuální se stavem k červnu 2017 a prognózou pro konec roku 2017.

Obrázek 15 LoRaWAN pokrytá červen 2017 (23)



Zdroj: IoT portál, 2017, (23)

Obrázek 16 LoRaWan finální pokrytí 2017 (23)



Zdroj: IoT portál, 2017, (23)

3.1.15 Významné IoT projekty

V následujících odstavcích jsou autorem této práce vybrány nejzajímavější IoT projekty převážně za roky 2017 a 2018. Projekty jsou zmíněny jen stručně, pro lepší představu širokého užití IoT.

Chytrá vinice

Chytrá vinice je IoT projektem, který využívá malé monitorovací stanice pro zaznamenávání základních veličin. Tyto veličiny se využívají pro předpovídání rozvoje potenciálních patogenů napadajících vinnou révu. Zařízení monitoruje relativní vlhkost vzduchu, množství srážek a teplotu v půl hodinových intervalech. Přínosem této technologie je lokální postihnutí problému a následná likvidace při použití menšího množství chemie. (24)

Moderní zemědělství

Obdobným projektem jako je chytrá vinice, je využití monitorování rozsáhlejších ploch pro zemědělské účely. Využívá se především model chorob rostlin a jejich matematický popis pro interakci mezi prostředím, rostlinou a proměnlivými patogeny, které mohou vést k rozvoji nemoci. Tyto modely jsou schopny předpovědět dopad nebo závažnost onemocnění a vývoj kultury mikrobů. Tyto modely byly odzkoušeny v různých klimatických podmínkách se stejným úspěchem. Hlavní funkčnost celého systému obstarává soustava speciálních senzorů a čidel. (25)

Hydroponie

Dalším zajímavým IoT projektem ze zemědělství je využití technologie hydroponie, při které se pěstují rostliny ve vodě. Základním principem je zavěšení kořenů v pohybuující

se vodě. Kořeny tak absorbují živiny a kyslík rychleji. IoT zařízení má za úkol hlídat potřebné hodnoty rostliny a na jejich základě najít rovnováhu mezi množstvím vody, živin a kyslíku, a maximalizovat tak výnos a kvalitu. Systém každých 5 minut automaticky kontroluje hodnotu snímače PH a v dalších intervalech kontroluje dle ostatních snímačů stav rostliny. (26)

Airdrop

Poslední zmínka o využití IoT v zemědělství v této práci patří inteligentnímu zavlažovacímu systému od firmy Airdrop. Jedná se o český projekt, který vyhrál Czech IoT summer Jam 2017. Systém se specializuje na inteligentní zavlažování, je využitelný pro oblast smart cities a řízení zavlažování v oblasti veřejné zeleně a dalších zelených ploch. Projekt Airdrop prokázal výrazný byznysový potenciál a výrazné finanční úspory při spotřebě vody. (27)

IoT ve Škodě Kodiaq

Nový vůz od automobilky Škoda využívá službu Care Connect s funkcí nouzového volání eCall. Služba ŠKODA Care Connect zahrnuje vzdálený přístup, proaktivní servis a nouzové volání. Díky zabudované elektronické SIM kartě umožňuje prostřednictvím aplikace pro chytré telefony, nebo webové aplikace, vzdálený přístup k nejrůznějším údajům o vozidle. V případě potřeby lze skrze auto přivolat na cestě pomoc s informací o lokaci a přesným popisem závady či kolize. (28)

Kontrola plynu přes Sigfox

Zajímavým produktem na českém trhu je měření spotřeby plynu přes síť Sigfox. Tuto praktickou věc mohou využít zákazníci společnosti Innogy. Služba Monitoring spotřeby za uživatele automaticky sleduje spotřebu plynu. Systém načítá a posílá hodnoty do mobilní aplikace, porovná skutečnou spotřebu s již uhrazenými i do budoucna nastavenými zálohami, a při hrozbě velkého nedoplatku nebo přeplatku navrhne změnu stanovených záloh. (29)

3.2 Arduino a Raspberry Pi

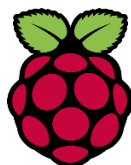
Arduino a Raspberry Pi patří do sekce počítačů s nízkoenergetickou náročností. Pro svou miniaturní velikost a výkonnost jsou tyto počítače využívány v mnoha projektech, kde je kladen důraz na tyto parametry. V praktické části této práce je využito Arduino pro řízení dronu a Raspberry Pi pro zprostředkování komunikace s IoT.

Obrázek 17 Logo Arduino



Zdroj: <https://www.arduino.cc>, 2018

Obrázek 18 Logo Raspberry Pi



Zdroj: www.raspberrypi.org, 2018

3.2.1 Základní informace o Arduino

Arduino je open-source elektronická platforma založená na snadno použitelném hardwaru a softwaru. Desky Arduino dokáží například číst vstupy: světlo na snímači, prst na tlačítku nebo zprávu z Twitteru – a přeměnit je na nějaký výstup (například aktivaci motoru), zapínat LED světla a mnoho dalších funkcí. Deska je plně programovatelná a uživatel může zasílat sady instrukcí mikrokontroleru na desce. K tomu se využívá programovací jazyk Arduino a software Arduino (IDE). (30)

3.2.2 Využití Arduino

Díky své jednoduché a přístupné uživatelské zkušenosti je Arduino používáno v tisících různých projektů a aplikací. Software Arduino je snadno použitelný pro začátečníky, je ale také dostatečně flexibilní pro pokročilé uživatele. Lze je spustit na počítačích Mac, Windows a Linux. (30)

Mezi hlavní přednosti Arduino patří příznivá cena, Cross-platformní využití, jednoduché a přehledné programovací prostředí, otevřený a rozšířitelný software a otevřený

a rozšiřitelný hardware. Pak zejména otevřený hardware dává vývojářům možnost vytvářet klony a desky pro řízení dronů, kterých je v současné době nespočet. (30)

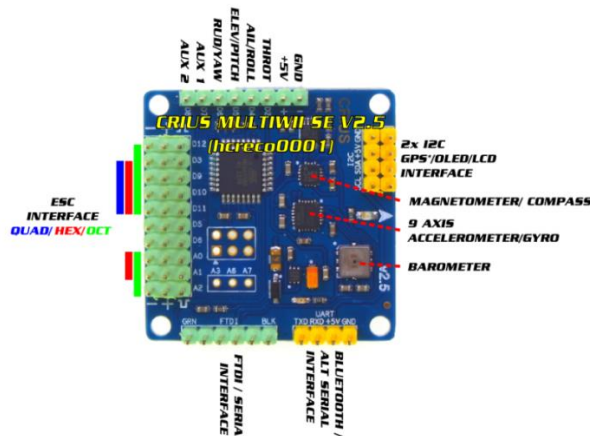
3.2.3 Základní typy Arduino pro drony

Arduino bylo vybráno pro drony z důvodu nízké ceny a velké rozšiřitelnosti. V průběhu vývoje byly nejdříve využívány 8 bit jednotky. V následujících kapitolách budou stručně popsány nejběžnější jednotky pro drony.

8 bit jednotky

Nejstarší 8 bit jednotky byly u zrodu Multiwii programu, který bude popsán dále, a prvních masově rozšířených dronů. Tyto jednotky byly postaveny na jednoduchém mikrokontroleru ATmega 328P, který měl dostatek výkonu pro základní ovládání dronu. Mezi nejznámější jednotky tohoto druhu lze zařadit výrobce Crius Multiwii SE.

Obrázek 19 Crius Multiwii v2.5



Zdroj: www.multiwii.com, 2018

16 bit jednotky

Nástupcem 8 bit jednotek se převážně stal nový čip ATmega 2560 s 16 bit architekturou. Na těchto jednotkách bylo možné využít firmware MegaPirateNG a MultiWii. Tyto jednotky zvládaly zpracovat více instrukcí najednou a dokázaly využívat více I/O zařízení, jako je například přídavný display nebo GPS modul. Autor uvádí jako příklad jednotku Crius AIOP 2. (31)

3.2.4 Multiwii

MultiWii je bezplatný open source projekt pro řízení létajících RC modelů. MultiWiiCopter je historicky založen na rozšíření Wii Motion Plus a mini desce Arduino pro. Projekt vypěl z velmi jednoduchého, levného a minimalistického letového ovladače a nyní podporuje všechny očekávané funkce, včetně GPS navigace.

3.2.5 CleanFlight

CleanFlight je Open-Sourcový letecký software, který je založen na 8 bit MultiWii kódu, ale využívá nové rychlé procesory s architekturou 32 bit. Tento software může být využit na multikoptéry, letadla, létající křídla a mnohé další letecké projekty. (33) V praktické části, bude tento software využit pro ovládání testovacích dronů.

3.2.6 BetaFlight

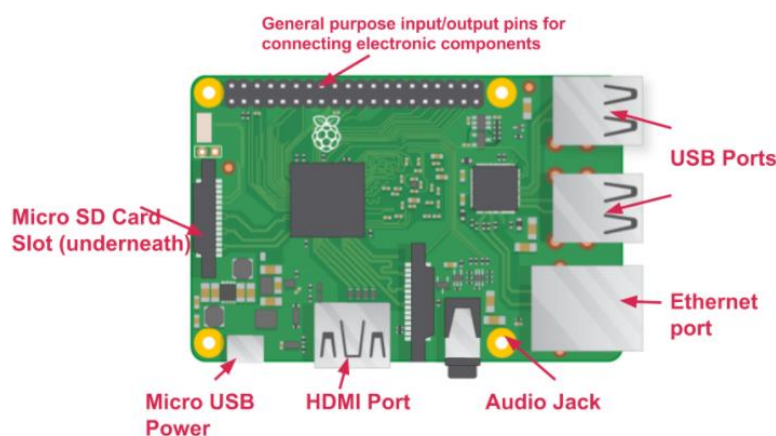
Betaflight je kopie (FORK) softwaru Cleanflight, kterou zahájil BorisB. V minulosti tato verze fungovala jako beta testovací platforma pro Cleanflight. Podle analýzy Google Trends z ledna 2017, byl Betaflight prohlášen za nejlepší open source software pro multirotory. (34)

3.2.7 Raspberry Pi

Raspberry Pi je malý počítač, který umožňuje vytvářet širokou škálu projektů. Pro správné fungování stačí připojit monitor, klávesnici a myš. (35) Počítače Raspberry Pi disponují těmito vstupy a výstupy (I/O):

- USB porty,
- Slot na SD kartu,
- Ethernet port,
- Audio jack,
- HDMI port,
- Micro USB napájení,
- GPIO porty. (36)

Obrázek 22 Počítač Raspberry Pi (36)



Zdroj: PI, redakce Raspberry, 2018, (36)

3.2.8 Modely a verze Raspberry Pi

Počítače Raspberry Pi během vývoje prošly značným zlepšením. Pro realizaci telemetrie bude v této práci využito zařízení RASPBERRY Pi 2 Model B+ a jeho novější verze RASPBERRY Pi 3 Model B+. Nicméně pro realizaci přenosu telemetrie by bylo možné využít jakýkoli model z následujících odstavců.

Model A

Model A je nejlevnější verzí tohoto počítače. Tento model má nižší spotřebu než model B a je výhodný pro projekty, které využívají energie ze slunce, větru či baterií. Srdcem počítače je procesor SoC BCM2835 s kapacitou paměti 256 MB. Nevýhodou modelu A je dispozice jediného USB portu, který se musí rozšiřovat USB hubem, pokud uživatel má více USB periférií. (37 str. 27)

Model B

Model B je nákladnější verzí počítače Raspberry Pi. Oproti modelu A disponuje větší operační pamětí - 512 MB. Procesor je stejný jako u modelu A, tedy SoC BCM2835. Tento model má 2–4 USB porty dle verze a Ethernet port, který je nezbytný pro SSH připojení a vzdálené programování. Model B je tedy nejčastější volba pro koupi a různé projekty. (37 str. 28)

Model B+

Poslední model z řady Raspberry Pi je B+. Tento model obsahuje nejnovější verzi základní desky vyvinuté v nadaci Raspberry Pi Foundation. Model B+ je vybaven stejným procesorem BCM2835 a 512 MB paměti jako jeho předchůdce. Změny jsou následující: přibyly 4 otvory pro montáž desky, prodloužený konektor GPIO, 4 USB konektory a byly přepracovány napájecí obvody, čímž se snížila energetická náročnost a zvýšila spolehlivost. V této diplomové práci autor zvolil pro přenos telemetrie právě tyto modely B+. (37 str. 30)

Raspberry Pi Zero

Raspberry Pi Zero má poloviční velikost než model A, ale dvojnásobnou užitečnost. Je to malý milník v rodině Raspberry Pi. Tento model se skládá z jedno jádrového procesoru s taktem 1GHz, 512 MB RAM, jedním mini HDMI, micro USB OTG portem a micro USB pro napájení. (38)

Raspberry Pi Zero W

Raspberry Pi Zero W rozšiřuje rodinu nízko energeticky náročných čipů Pi Zero. Na konci února 2017 bylo oznámeno, že Pi Zero W má všechny funkce původního Pi Zero, ale přichází s přidanou konektivitou, skládající se z: wifi, bluetooth, bluetooth Low Energy (BLE), jedno jádrového procesoru s taktem 1Ghz a 512 MB RAM. (39) Jedná se o nejmenší Raspberry Pi, pro svou velikost a váhu je velmi dobře využitelný pro projekty s létáním.

Raspberry Pi 2 Model B

Tato verze Raspberry Pi vyhází z modelu B+, má navíc 900MHz quad-core ARM Cortex-A7 procesor a 1 GB RAM. (40) Tento model byl vybrán pro přenos telemetrie z dronu quadcoptery.

Raspberry Pi 3 Model B

Raspberry Pi 3 je třetí generace Raspberry Pi. Byla představena únoru roku 2016 a nahrazuje model Raspberry Pi 2 Model B. Tato verze má silnější procesor Quad Core

1.2GHz Broadcom BCM2837 64bit, 1 GB RAM a nově má integrovaný modul wifi a bluetooth. (41) Tento modul byl zvolen pro testování na hexacopteru v praktické části.

3.2.9 OS pro Raspberry Pi

Na zařízení Raspberry Pi lze spustit většinu verzí Linuxových distribucí, přehled základních distribucí je popsán v následujících odstavcích.

Raspbian

Názvem Raspbian se označuje přizpůsobená varianta linuxové distribuce Debian. Tato distribuce vyniká výkonem i na slabších HW strojích a skvělou stabilitou. Raspbian používá distribuci Debian jako svou základní neboli zdrojovou, distribuci. Je pak dále doplněn o vlastní nástroje a software. Jelikož se jedná o standardní OS pro Raspberry Pi, bude Raspbian použit v praktické části. (37 str. 56)

RaspBMC

Raspbmc je vysoce optimalizované a specializované mediální centrum pro Raspberry Pi. Raspbmc je flexibilní a umožňuje rozšíření prostřednictvím balíčkového systému Debian. Například je možné využít GPIO (univerzální vstupní výstup) pro konfiguraci vlastního infračerveného přijímače. (42)

OpenELEC

OpenELEC (zkratka pro Open Embedded Linux Entertainment Center) je Linuxová distribuce určená pro PC domácího kina, založená na mediálním přehrávači Kodi (dříve XBMC). (43) OpenELEC uplatňuje zásadu "just enough operating system", což znamená použití jen takového množství softwaru, které potřeba pro běh. Je navržen tak, aby konzumoval relativně málo zdrojů, a rychle se spouštěl z flash paměti. (44)

Pidora

Pidora se liší od Raspbianu v několika ohledech. Dodává se s jiným softwarem než Raspbian, a to včetně množství textových editorů, programovacích jazyků a dalších nástrojů. Tento systém je založen na distribuci systému Linux Fedora. (45)

Arch Linux

Arch Linux je nezávisle vyvinutá distribuce GNU / Linux x86-64, jejímž cílem je poskytnout nejnovější stabilní verze většiny softwaru. Výchozí instalace je minimálně omezena jen na základní systém. Ten je nakonfigurovaný uživatelem, aby přidal pouze to, co je záměrně požadováno. (46)

Nástroj NOOBS

NOOBS je jednoduchý instalátor operačního systému, který obsahuje Raspbian. Poskytuje výběr alternativních operačních systémů, které jsou poté staženy z internetu a nainstalovány. Existuje i verze Lite bez předinstalovaného Raspbianu. (47) Tento nástroj je vhodný pro začínající uživatele Linuxu a byl využit v praktické části pro instalaci systému Raspbian na testovací zařízení.

3.2.10 Moduly pro Raspberry Pi

Pro počítač Raspberry Pi existuje mnoho přídavných modulů. Tyto moduly můžeme dělit na originální – vytvořené nadací Raspberry Pi Foundation – a na neoriginální, převážně čínské klony. Pro přenos telemetrie skrze IoT, jsou v praktické části využity zejména moduly pro přídavnou kameru a dotykový LCD display.

Mezi originální příslušenství můžeme zařadit tyto komponenty:

- Raspberry Pi Compute module 3 I/O,
- Raspberry Pi Camera Board,
- Raspberry Pi NoIR Camera Module V2,
- Raspberry Pi Touch display 7.

Mezi neoriginální komponenty můžeme zahrnout:

- Raspberry LCD module Pi TFT 3.5 inch,
- Raspberry Pi 3 Camera Module 1080p 5MP Night Vision Camera,
- Raspberry Pi senzory převzaté od Arduino platformy.

3.3 Multikoptéry – drony

V následující kapitole jsou představeny základní typy dronů, na které se pak autor této práce odkazuje v praktické části.

3.3.1 Základní terminologie

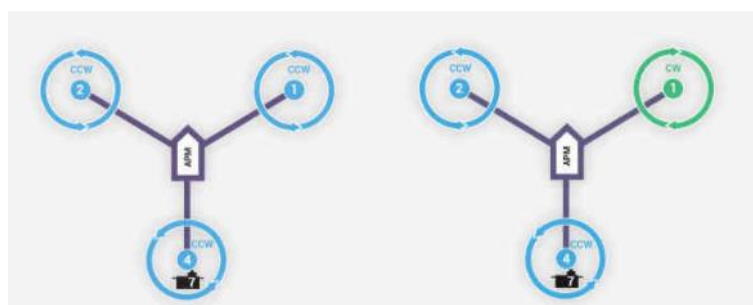
Existuje mnoho rozdílných pojmenování pro létající stroje. Termín dron vyjadřuje zařízení, které nemá pilota a může létat autonomně. (48) Základní rozdíl mezi drony je v počtu vrtulí. Nejjednodušší drony mají dvě vrtule a nejvyužívanější jsou drony se čtyřmi vrtulemi, takzvané quadrocoptery. V následujícím přehledu jsou představeny nejtýpičtější druhy dronů. V popisech obrázků se objeví symboly CCW a CW, používané pro určení směru rotace vrtule:

- CCW- směr otáčení vlevo,
- CW- směr otáčení vpravo.

3.3.2 Tricopter

Tricoptera je druh dronu, který má pouze tři vrtule, a jeden servomotor pro udávání směru pro let vpřed nebo vzad. Tento druh dronu se příliš nepoužívá, patří spíše do odvětví pro nadšené modeláře.

Obrázek 23 Tricopter

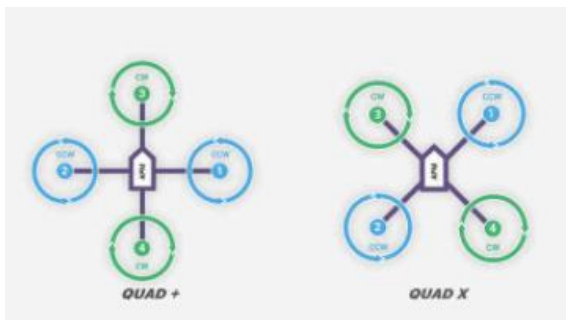


Zdroj: ardupilot.org, 2018

3.3.3 Quadrocopter

Quadrocoptera je nejběžnější typ prodáváného dronu. Vyniká skvělými letovými vlastnostmi a obratností. Tyto drony bývají nejčastěji využívány širší veřejností pro natáčení a focení. Tento typ je charakteristický čtyřmi vrtulemi a bude využit v praktické části.

Obrázek 24 Quadrocoptera

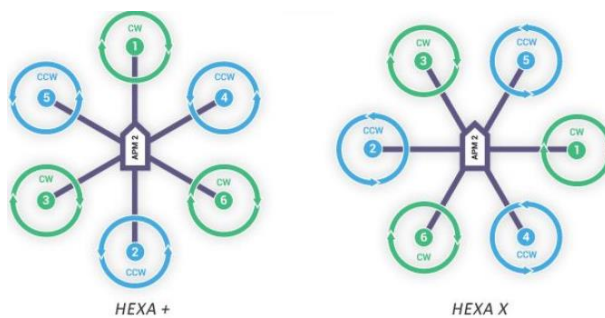


Zdroj: ardupilot.org, 2018

3.3.4 Hexacofter

Hexacofter je druh dronu, který využívá pro let 6 motorů a vrtulí. Výhodu oproti quadcoptere má především ve větší stabilitě a nosnosti. Existuje mnoho variací dronů hexacofter, nejběžnější jsou zobrazeny v následující obrazové příloze. Tento druh dronu je také využit v praktické části.

Obrázek 25 Hexacoftera

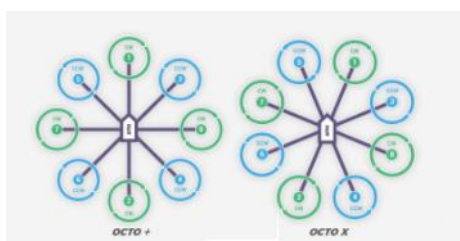


Zdroj: ardupilot.org, 2018

3.3.5 Octocofter

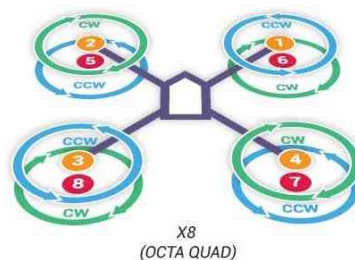
Octocofter je druh dronu, který má osm vrtulí. Vrtule mohou být buď všechny směrem vzhůru, nebo dané proti sobě, takzvané reversivní. Tento druh je velmi stabilní a používá se pro náročnější letecké operace.

Obrázek 26 Octocoftera



Zdroj: ardupilot.org, 2018

Obrázek 27 Reversivní octocoptera



Zdroj: ardupilot.org, 2018

3.3.6 Ostatní typy dronů

Do kategorie ostatní patří všechny druhy dronů, které nebyly vyjmenované v předešlé sekci. Jedná se především o drony s více jak osmi vrtulemi. Obecně o těchto létajících strojích lze říci, že čím je více vrtulí, tím má dron stabilnější letovou dráhu.

V praktické části diplomové práce autor zvolil pro přenos telemetrie rámy HEXA a QUADRO od firmy Tarot. Více o těchto rámech bude popsáno v realizační části této práce.

3.3.7 Telemetrie

Telemetrie je automatizovaný proces komunikace, pomocí něhož jsou měření a další data shromažďována na vzdálených nebo nepřístupných místech. (49) Tato data jsou pak přenášena na zařízení, kde se tato data sledují. Termín se běžně týká bezdrátových mechanismů pro přenos dat, kde mohou být využité rádiové, ultrazvukové nebo infračervené systémy. Dále pak zahrnuje také data přenášená přes jiná média, jako mohou být například telefonní síť, počítačová síť, optické spojení nebo jiné drátové komunikace, například nosiče elektrické energie. Většina moderních telemetrických systémů všudypřítomnosti GSM sítí, kde je telemetrie přenášena pomocí SMS. (50)

V této diplomové práci bude telemetrie přenášena pomocí bezdrátových sítí GSM a LTE. K tomuto přenosu budou využity bezdrátové modemy se sim kartou.

3.4 Fiori

V kapitole Fiori autor této práce představuje platformu Fiori, skrze kterou bude implementováno monitorování přenosu telemetrie.

3.4.1 SAP

SAP je ryze německá firma se sídlem ve Walldorfu. Její produkty jsou z oblasti ERP (Enterprise Resource Planning, česky Plánování podnikových zdrojů). Název firmy je vytvořen ze zkratky „Systeme, Anwendungen, Produkte in der Datenverarbeitung“, což je analogicky anglicky „Systems – Applications - Products in data processing“. SAP je světový lídr v oblasti podnikových aplikací. Z hlediska tržní kapitalizace je třetím největším světovým nezávislým výrobcem softwaru. (51)

3.4.2 Fiori

SAP Fiori je platforma, která umožňuje spuštění aplikací na mobilních zařízeních prostřednictvím webové služby s podporou HTML5. Zveřejněna byla dne 15. května 2013. SAP Fiori je založena na technologii SAP NetWeaver. SAP Fiori umožňuje používání aplikací na stolních počítačích, tabletech či smartphonech. První verze měla několik aplikací pro spuštění a postupně aplikace přibývají. SAP Fiori je k dispozici pro iOS, Android a Windows Mobile platformy. (52 str. 88) Ve Fiori se nejčastěji využívají 3 základní druhy aplikací. Tyto aplikace jsou buď transakční, analytické anebo informační.

Obrázek 28 platformy pro SAP Fiori (52 str. 1)



Zdroj: Mathew, Bince, 2015, (52 str. 1)

3.4.3 SAP Fiori designová stránka

Inspirace pro design Fiori aplikací přichází od webových a grafických návrhářů, kteří využívají standardy pro tvorbu webových stránek, android a IOS aplikací. Při návrhu aplikací se využívá metoda zvaná „flat design“. Tato metoda, neboli styl, je zaměřena na zjednodušení uživatelského view (obrazovky) jen na potřebné prvky pro základní práci s aplikací. V důsledku této skutečnosti se s aplikací pracuje pohodlněji a rychleji. Příkladem pro zjednodušenou aplikaci se využívá klasické kalkulačky od Microsoftu ve verzi 8 a výše. Zde byla ponechána jen základní kalkulačka, rozšířené možnosti pro náročnější výpočty jsou skryty pod tlačítkem se třemi řádky. (52 str. 10)

Primární vlastnosti flat designu pro uživatele jsou:

- jednoduchost,
- čistota,
- barevnost,
- modernost,
- trend.

Hlavní výhody flat designu pak jsou:

- snadné používání,
- moderní vzhled,
- efektivní responsivní design,
- rychlejší načítání a inicializace.

3.4.4 Základní architektura Fiori aplikací

Architektura Fiori aplikací je znázorněna v obrazové příloze níže. V první části jsou zobrazeny zařízení, která jsou podporována (počítače, tablety a mobily). Tato zařízení spouští aplikace prostřednictvím takzvaného launchpadu. Launchpad je výchozí obrazovka, na které jsou ikony s aplikacemi, které jsou zpřístupněné jen uživatelům s odpovídajícími právy.

Aplikace komunikuje skrze HTTPS s aplikačním serverem. Server se stará o běh aplikace a zprostředkovává spojení s databází. Pro připojení do databáze se využívá spojení

4 Vlastní práce

4.1 Srovnání IoT poskytovatelů

První oddíl praktické části této práce je věnována stručné charakteristice IoT poskytovatelů a vybrání nejvhodnějších variant na přenos telemetrie z dronu dle metodiky vícekriteriální analýzy. Zejména bude využita Saatyho metoda (kvantitativního párového srovnávání) pro určení vah kritérií a bodovací metoda pro určení nejlepšího poskytovatele IoT.

4.1.1 Použité metody vícekriteriální analýzy variant

Saatyho metoda

Saatyho metoda (kvantitativního párového srovnávání) slouží k určení vah kritérií, hodnotí-li pouze jeden expert, který porovnává každou dvojici kritérií a hodnocení pak následně vyplní do takzvané Saatyho matice. Tato matice se vyplní následujícím způsobem:

- 1 pokud jsou rovnocenné,
- 3 slabá preference ,
- 5 silná preference,
- 7 velmi silná preference,
- 9 absolutní preference.

Do matice lze využít i mezistupně (2, 4, 6 a 8). Matici vyplníme tak, že na diagonála bude rovná 1, a zbylé párové porovnání se zapisuje preferencí u dominující a převrácenou hodnotou u submisivní například 3 a 1/3. Samotný výpočet pak probíhá tak, že z řádků matice se vytvoří geometrické průměry a vytvoří se k-tá odmocnina. Vytvoří se součet těchto hodnot, kterým se pak vydělí geometrické průměry. Dostaneme pak váhy, jejichž suma je rovná 1. (54 str. 83)

Bodovací metoda

Bodovací metoda se používá při kvantifikaci hodnocení variant podle kritéria. Nejprve se stanoví bodová stupnice, v této práci bude od 0 do 100 bodů. Dále se provádí hodnocení každé z variant podle kritéria, vyjadřuje se počtem bodů tak, aby čím lépe je

hodnocena tím dostala více bodů. Jsou povolena desetinná čísla a stejné hodnoty u variant. Výběr nejlepší varianty se určuje součtem bodů v daných kritériích. (54 str. 83)

Stavový diagram je známá technika používaná k popisu chování systému. Různé formy stavových diagramů se objevují zhruba o šedesátých let. Stavový diagram se skládá z následujících stavů: počáteční pseudostav (initial pseudostate), přechod (transition) a koncový stav (final state). (55 str. 110)

4.1.2 Nejznámější IoT poskytovatelé pro rok 2017 a 2018

V následujícím odstavci jsou popsány nejznámější IoT poskytovatelé pro roky 2017 a 2018. Autorem této práce byly vybrány nejzajímavější platformy, které lze použít pro přenos telemetrie. V následujících podkapitolách jsou stručně charakterizovány vybraní IoT poskytovatelé.

Poskytovatelé vybraní pro srovnání jsou:

- Amazon web services IoT platform (AWS),
- Microsoft azure IoT hub,
- IBM Watson IoT platform,
- Google cloud platform,
- ORACLE,
- Salesforce,
- BOSCH,
- Cisco IoT cloud connect,
- General Electrics Predix,
- SAP IoT Leonardo,
- SAMSUNG Artik cloud,
- ThingWorx,
- Kaa Platform,
- Macchina Platform,
- SiteWhere Platform,
- ThingSpeak Platform,
- Carriots Platform,

- Ubidots,
- MyDevices Cayenne,
- Temboo Platform.

Amazon web services IoT platform (AWS)

Amazon je dominantním hráčem na poli cloudových spotřebitelů. Byl mezi prvními, kteří v roce 2004 přeměnili cloud computing na komoditní způsob. Tato platforma je extrémně škálovatelná. Amazon tvrdí, že dokáže podporovat miliardy zařízení a bilióny interakcí mezi sebou. (56) (57)

Ceny služeb Amazonu jsou založeny na zprávách odeslaných a přijatých do AWS IoT. Každá interakce mezi zařízením a serverem je považována za zprávu. Poplatky Amazonu se platí za milion odeslaných nebo přijatých zpráv. Amazon má sadu pro vývoj softwaru (SDK), která pomáhá vývojářům vytvářet aplikace pro běh na platformě AWS. Amazon má nejkompexnější služby ze všech poskytovatelů cloudů, ale na druhou stranu je také nejdražší. (56) (57)

Microsoft azure IoT hub

Společnost Microsoft má své portfolio cloudových služeb Internet of Things. Disponuje úložištěm v cloudu, adaptivním prostředím a službami IoT. Microsoft disponuje vlastním operačním systémem pro zařízení IoT se jménem Windows 10 IoT Core (58), který dokáže fungovat i na zařízeních Raspberry Pi. (56) (59)

Poplatky za služby se dělí do čtyř úrovní podle množství generovaných dat ze zařízení. Pod 8 000 zpráv na jednotku za den je zdarma, pokud se neintegruje s dalšími službami společnosti Microsoft. (56) (59)

IBM Watson IoT platform

IBM je další IT gigant, který se snaží stát dominantním ve světě IoT. Snaha IBM je poskytnout cloudové služby pro začátečníky s jednoduchými aplikacemi a rozhraními.

Uživatelé tak mohou vyzkoušet ukázkové aplikace, do kterých mohou sbírat data ze senzorů a následně pak porovnávat s daty v historii. Ceny za služby jsou odstupňované do 3 kategorií

cena za výměnu dat, za analýzu dat a cena za ostatní data. Výhodou je, že každý uživatel dostane měsíčně 100 MB prostoru pro testování. (56) (60)

Google cloud platform

Vyhledávač Google má svoji IoT platformu. Google tvrdí, že Cloud Platform je nejlepším místem pro budování IoT podnětů pro zpracování a analýzu dat. Google klade důraz na snadnou a rychlou práci s okamžitými informacemi. Dále Google umožňuje využít jejich soukromé globální optické sítě. Za služby v Google Cloud se platí za využitou minutu. Služby vycházejí levněji než od konkurenčního Amazonu. Google také nabízí vlastní kalkulačku, která spočítá kolik ušetříte. Nevýhodou této služby je, že nemá tak komplexní nástroje a dokumentaci. (56) (61)

ORACLE

Oracle je platforma, která poskytuje služby zaměřené na výrobní a logistické operace. Cílem firmy ORACLE je pomoci firmám rychleji prosadit produkt na trhu. Cena za služby Oracle se vypočítávají dle použitých zařízení, dále uvádějí počet zpráv na zařízení za měsíc, pokud uživatel překročí stanovený limit musí připočítat dodatečné náklady. (56) (62)

Salesforce

Společnost Salesforce se specializuje na řízení vztahů se zákazníky. Cloudová platforma je poháněna Thunderem (speciální engine pro zpracování big data) (63), který je zaměřen na vysokorychlostní rozhodování v reálném čase v cloudu. Hlavní myšlenka společnosti Salesforce je snadné uživatelské rozhraní typu point-and-click pro snadnější propojení se zákazníky. Cena za služby Salesforce je okolo 4000\$ za měsíc nebo dle stanovených tabulek. (56) (64)

BOSCH

Bosch je německá IT firma, která v nedávné době zahájila vlastní služby na poli cloudu. Zaměřují se především na bezpečnost a efektivitu. Platforma IoT je flexibilní a založená na otevřených standardech a open source licencí. Poplatky za služby jsou odstupňované, čím více uživatel službu využívá, tím více zaplatí. (56) (65)

Cisco IoT cloud connect

Cisco je globální lídr v oblasti služeb IT. Cisco věří, že nové možnosti IT leží v cloudu a vyvinuli nový softwarový balíček založený na mobilním cloudu. Jejich hlavním cílem je posílit u klientů vztah se zákazníky. (56) Ceny za služby jsou kategorizované dle využití. (66)

General Electrics Predix

Firma General Electric nabízí služby platform-as-a-service. Je zaměřena především na průmyslový trh, kde nabízí konektivitu a analytiku pro běžná odvětví, jako je například letectví. Nemají cenové tabulky jako velké společnosti. (56) (67)

SAP IoT Leonardo

Platforma SAP Cloud Platform má služby pro IoT, které podporují zpracování Big Data, umělou inteligenci a strojové učení. Velkou výhodou je využití celé řady služeb pro vyzkoušení této platformy. Cena za služby je rozdělená do 3 kategorií: Developers, Media Business a Enterprise. (56) (68)

SAMSUNG Artik Cloud

Další významnou platformou je Artik Cloud od společnosti SAMSUNG, který umožňuje otevřenou výměnu dat pro IoT. Výhodou je využití cloudových konektorů, které umožňují integrovat služby třetích stran. Samsung nabízí 3 cenové kategorie: Free Plan, Small Business, Enterprise. (69) (70)

ThingWorx

ThingWorx je průmyslová inovační platforma. Cílem této platformy je poskytovat IIoT a zkušenosti s rozšířenou realitou (AR), které odhalují hodnoty digitálního a fyzického světa. Zahrnuje technologie a nástroje, které umožňují průmyslovým podnikům rychle a snadno se rozvíjet, nasazovat a rozšiřovat aplikace IoT. (71) Firma ThingWorx nabízí uživatelům vyzkoušení a následné zakoupení platformy, ceny na webu neuvádí.

Kaa Platform

Kaa je víceúčelová middleware platforma pro IoT, která umožňuje vytvářet komplexní řešení IoT, spojené aplikacemi a inteligentní produkty. Platforma Kaa poskytuje otevřený, bohatý nástroj pro vývoj produktů IoT, a tím dramaticky snižuje související náklady, rizika a čas uvedení na trh. Pro rychlý start společnost Kaa nabízí sadu IoT komponent, které lze snadno zapojit a použít k implementaci velké většiny případů používání IoT. Jedná se o open-source platformu, která je zdarma. (72)

Macchina Platform

Macchina.io je rozsáhlou a výkonnou softwarovou platformou pro aplikace IoT, které se připojují k senzorům, zařízením a cloudovým službám. Macchina.io je tedy bohatý framework pro rychlé budování aplikací IoT na zařízení se systémem Linux. (73) Pro vývoj jsou využívány jazyky JavaScript a C++. Existují dva druhy licencí open source a komerční. (74)

SiteWhere Platform

SiteWhere je open source platforma pro IoT s vlastním frameworkem a API pro vlastní vývoj. Platforma umožňuje propojení zařízení přes protokoly MQTT, AMQP, Stomp a další. Výhodou této služby je možnost propojení s dalšími cloudy, jako je například Microsoft Azure. (75)

ThingSpeak Platform

ThingSpeak je IoT platforma pro analýzu, která umožňuje agregovat, vizualizovat a analyzovat živé datové toky v cloudu. ThingSpeak poskytuje okamžitou vizualizaci dat odesílaných ze zařízení. Pomocí kódu MATLAB může uživatel provádět online analýzu a zpracování dat. Tato platforma se často používá k prototypování koncepčních systémů IoT, které vyžadují analýzu. (76) Ceny za platformu jsou rozdělené do pěti kategorií dle náročnosti projektu, Standard, Academic, Student, Home a Free. (77)

Carriots Platform

Carriots je platforma jako služba (PaaS) určená pro projekty Internet of Things (IoT) a Machine to Machine (M2M). Tato platforma umožňuje sbírat a ukládat data ze zařízení, vytvářet aplikace pomocí soukromého SDK enginu a odesílání do tisíců zařízení. Společnost nabízí volný účet pro ovládání dvou zařízení, zbylá řešení jsou komerční. (78)

Ubidots Platform

Společnost Ubidots se v počátcích zabývala strojírenskými službami na specializaci a vývoj hardwaru a softwaru pro IoT v Latinské Americe. V letech 2012 až 2014 uskutečnila stovky projektů internetových věcí v průmyslových odvětvích. Nabízí cloudové řešení s vlastní API. (79) Ubidots nabízí trial verzi a 4 kategorie pro platby: Developer, IoT Lab, Industrial a Scale. (80)

MyDevices Cayenne Platform

MyDevices je zajímavá IoT platforma, která nabízí dva hlavní produkty IoT in a box a IoT Ready. V prvním případě se jedná o plnohodnotné řešení se senzory, vysílači a editovatelnou aplikací. Firma vytváří moduly i na konkrétní potřeby zákazníka. Ceny za služby začínají na 29\$ za měsíc. (81)

Temboo Platform

Platforma Temboo dokáže pracovat s libovolným programovacím jazykem pro komunikaci se zařízeními a cloudovou službou. Společnost nabízí 10 různých SDK pro uspokojení vlastních IoT požadavků. Firma disponuje sadou připravených aplikací jako je například měření úniku plynu nebo monitorování kvality půdy. Temboo nabízí čtrnáctidenní trial verzi na vyzkoušení. (82)

4.1.3 Vstupní data

Autor této práce využil existující srovnání, které zkontroloval, opravil a doplnil o chybějící informace. Autor využil kritéria zmíněná v existujícím srovnání a přidal potřebná kritéria, která jsou důležitá pro tvorbu přenosu telemetrie přes IoT. Tato kritéria jsou popsána

v sekci Kritéria pro vícekritériální analýzu variant. Výchozí existující srovnání autor využil ze stránek: <https://www.postscapes.com/internet-of-things-platforms/>, v sekci Provider List Comparison.

Vstupní data s uvedenými zdroji jsou uvedena v souboru Excel, který je přiložen k této práci. Data s výsledky bodování, bez zdrojů jsou uvedeny v přílohách této diplomové práce.

4.1.4 Kritéria pro vícekritériální analýzu dat

Prvním krokem v analýze bude stanovení vah kritérií Saatyho metodou. Pro vybrání nejvhodnějšího poskytovatele bude použita bodovací metoda. Následně byla bodovací metoda obohacena o váhy kritérií ze Saatyho metody pro dvojí srovnání.

- Web,
- Licence,
- Protokoly,
- Cena,
- Online školení,
- Vlastní sada aplikací,
- Demo,
- Propojení na Github,
- API dokumentace,
- Vlastní knihovny,
- Jazyky,
- Průmyslové odvětví,
- Hardware podpora,
- Konektivita,
- Databáze,
- Vlastní hodnocení.

Web

U kritéria web autor práce zhodnotil celkové stránky a první dojem při výběru poskytovatele IoT. Při seznamování s IoT službami je web jednou z hlavních vypovídajících kritérií, které uživatel vidí. U webu autor hodnotil responsibilitu, přehlednost stránek, jazykovou variaci včetně české lokalizace, online podporu a obsah bez vyskakovacích reklam. Udělené body lze nalézt v Příloha 1. Bodování je znázorněno v následující tabulce:

Tabulka 2 kritéria pro hodnocení dle Webu

Kriterium	Bodování
Responsibilita	0 -20
Přehlednost	0 -20
Čeština	0-10
Vícejazyčnost	0-10
Online podpora	0 -20
Bez reklam	0 -20

Zdroj: vlastní práce autora

Licence

Kritérium licence autor hodnotil podle existencí následujících typů licencí: Trial, Developerská, Komerční a Free (volná). Všechny podkategorie byly ohodnoceny bodovou stupnicí 0-25 bodů dle existence či neexistence licence. Získané body lze nalézt v Příloha 2. Udělování bodů probíhalo dle následující tabulky:

Tabulka 3 kritéria pro hodnocení dle Licence

Kriterium	Bodování
Trial	0 -25
Developerská	0 -25
Komerční	0 -25
Free	0 -25

Zdroj: vlastní práce autora

Protokoly

Kritérium protokoly bylo hodnocené autorem dle uvedené podpory na stránkách poskytovatelů. Za každý podporovaný protokol dostal poskytovatel 10 bodů do maximální výše 100 bodů. Přehled získaných bodů je přiložen v Příloha 3. Hodnocená kritéria jsou pak v následující tabulce:

Tabulka 4 kritéria pro hodnocení dle Protokolů

Kriterium	Bodování
AMQP	0 -10
CoAP	0 -10
MQTT	0 -10
LVM2M	0 -10
IPv4	0 -10
IPv6	0 -10
TCP	0 -10
HTTP(s)	0 -10
Ostatní	0 -10
UDP	0 -10

Zdroj: vlastní práce autora

Cena

Cena je jedním z hlavní kritérií, dle kterých autor hodnotil. Pro přenos telemetrie se autor snažil vybrat nejlépe dostupnou cenovou variantu. U této kategorie autor hodnotil existenci účtu zdarma (Free účet), jednotky, kategorie cen a neomezený přenos dat. Udělené body lze vidět v Příloha 4. Bodové hodnocení je k nahlédnutí v následující tabulce:

Tabulka 5 kritéria pro hodnocení dle Ceny

Kriterium	Bodování
Free účet	0 - 25
Jednotky	0 - 25
Kategorie Cen	0 - 25
Neomezeny přenos	0 - 25

Zdroj: vlastní práce autora

Online školení

Kritérium online školení vybral autor na základě zjištění nabízených služeb dle poskytovatelů platforem IoT. V této kategorii autor hodnotil existenci kurzu, možnou certifikaci a cenu za tento kurz. Udělené body jsou v Příloha 5. Následující tabulka znázorňuje udělování bodů autorem:

Tabulka 6 kritéria pro hodnocení dle Online školení

Kritérium	Bodování
Školení	0 - 25
Certifikace	0 - 25
Kurz zdarma	0 - 50

Zdroj: vlastní práce autora

Vlastní sada aplikací

Vlastní sada aplikací je kritérium, dle kterého lze určit pracnost celého projektu. Při existenci aplikací má uživatel zjednodušenou práci a ovládání celého systému. Autor u tohoto kritéria využil bodování, které se nachází pod tímto odstavcem. Tabulka udělených bodů se nachází v sekci přílohy: Příloha 6.

Tabulka 7 kritérium pro hodnocení dle Vlastní sady aplikací

Kritérium	Bodování
Vlastní aplikace	0 - 100

Zdroj: vlastní práce autora

Demo

Existenci demo autor vybral zejména kvůli snadnější implementace přenosu telemetrie a rychlejší orientaci ve zvolené platformě. Autor zde hodnotil pouze existenci dle dostupných informací od poskytovatelů. Tabulky s udělenými body se nachází v příloze: Příloha 7. Bodová stupnice je znázorněná v následující tabulce:

Tabulka 8 kritérium pro hodnocení dle Dema

Kritérium	Bodování
Existence Dema	0 - 100

Zdroj: vlastní práce autora

Propojení na Github

Propojení na Github je kritérium, které autor našel ve výchozí srovnávací tabulce a zkontroloval její zpracování, které následně doplnil. Toto kritérium je pro autora velmi důležité z důvodu získávání informací o službě. Body udělené pro toto kritérium jsou uvedené v: Příloha 8. Kritéria pro bodování propojení na Github jsou v následující tabulce:

Tabulka 9 kritéria pro hodnocení dle Dema

Kritérium	Bodování
Git - napojeni	0 -50
Git - Demo rep.	0 - 50

Zdroj: vlastní práce autora

API dokumentace

Api dokumentace je velmi důležité kritérium pro programátora. Všichni poskytovatelé udávají dostupnou dokumentaci, proto toto kritérium není rozhodující. Body jsou uvedené v: Příloha 9 . Bodovací kritérium je v následující tabulce:

Tabulka 10 kritérium pro hodnocení dle API dokumentace

Kritérium	Bodování
Dokumentace	0 -100

Zdroj: vlastní práce autora

Vlastní knihovny

Kritérium vlastní knihovny pro komunikaci, je důležité pro výběr poskytovatele. Autor zde využil údaje získané ze srovnání uvedeném ve zdrojových datech. Všichni poskytovatelé s výjimkou jednoho nabízí vlastní knihovny pro komunikaci. Udělené body naleznete v příloze: Příloha 10. Bodové hodnocení je v následující tabulce:

Tabulka 11 kritérium pro hodnocení dle Vlastní knihovny

Kritérium	Bodování
Existence knihoven	0 -100

Zdroj: vlastní práce autora

Programovací jazyky

Kritérium programovací jazyky je důležité pro vývojáře, kteří se raději přikloní k platformě, která podporuje jejich osvojené jazyky. Většina platform podporuje širokou škálu jazyků. Bodování probíhalo dle tabulky pod tímto odstavcem. Výsledné body jsou k nahlédnutí v příloze: Příloha 11.

Tabulka 12 kritéria pro hodnocení dle Programovacích jazyků

Kritérium	Bodování
C	0-10
C++	0-10
C#	0-10
Python	0-10
Node.js	0-10
Java	0-10
Android	0-10
PHP	0-10
JavaScript	0-10
Ostatní	0-10

Zdroj: vlastní práce autora

Průmyslové odvětví

Průmyslové odvětví je kritérium, které bylo autorem převzato a rozšířeno. Toto kritérium může mít potenciál v případě masivnějšího rozšíření přenosu telemetrie přes IoT, kde by se specializovalo jen na určitý okruh. Bodové ohodnocení je v příloze: Příloha 12. Tabulka bodování byla následující:

Tabulka 13 kritéria pro hodnocení dle Průmyslového odvětví

Kritérium	Bodování
Normální použití	0-50
Specializace	0-50

Zdroj: vlastní práce autora

Hardware podpora

Hardware podpora je kritérium, které bylo rozhodující pro výběr poskytovatele. Autor pro svou práci potřebuje, aby poskytovatel podporoval zařízení Raspbbery Pi. Výsledné ohodnocení se nachází v příloze: Příloha 13. Tabulka pro hodnocení kritéria byla následující:

Tabulka 14 kritéria pro hodnocení dle Hardwarové podpory

Kriterium	Bodování
Lora_Ready	0 -10
Raspberry Pi	0 -10
BeagleBone	0 -10
Arduino	0 -10
Ostatní	0 -10
Neomezeno	0 -50

Zdroj: vlastní práce autora

Konektivita

Konektivita je jeden z dalších nezbytných ukazatelů pro přenos telemetrie přes IoT. Autor této práce potřeboval využít platformu, která podporuje mobilní síť (Celluar). Platformy byly hodnocené dle podpory nejrozličnějších sítí. Udělené body jsou zobrazené v: Příloha 14. Rozpis udělování bodů je v následující tabulce:

Tabulka 15 kritéria pro hodnocení dle Konektivity

Kriterium	Bodování
Wi-Fi	0 -10
Cellular (mobilní)	0 -10
Sigfo10	0 -10
LoRaWAN	0 -10
LTE-M	0 -10
NB-IOT	0 -10
Ethernet	0 -10
Bluetooth	0 -10
Sigfox	0 -10
Ostatní	0 -10

Zdroj: vlastní práce autora

Databáze

Kritérium databáze bylo při výběru poskytovatele přenosu telemetrie přes IoT pro autora také nezbytné. Autor potřebuje ukládat data v rámci jedné platformy. Hodnocení bylo podle existence alespoň jedné databáze. Bodové ohodnocení je přiložené v: Příloha 15. Kritérium udělování bodů je v následující tabulce:

Tabulka 16 kritérium pro hodnocení dle Databáze

Kriterium	Bodování
Existuje DB > 0	0-100

Zdroj: vlastní práce autora

Vlastní hodnocení

Vlastní hodnocení autora je poslední kritérium, dle kterého autor hodnotil platformy. Hodnocení probíhalo na základě poznatků při vytváření podkladů pro Bodovací metodu vícekritériální analýzy variant. Subjektivní ohodnocení je přiložené v: Příloha 16. Hodnotící kritérium je přiložené v následující tabulce:

Tabulka 17 kritérium pro hodnocení dle Vlastního hodnocení

Kritérium	Bodování
Subjektivní názor	0 - 100

Zdroj: vlastní práce autora

4.1.5 Výsledky Saatyho metody

Pomocí Saatyho metody dospěl autor k výsledkům vah kritérií, které následně využil v bodovací metodě pro určení nejlepšího poskytovatele. V Saatyho metodě bylo porovnáno 16 kritérií a využitá tabulka preferencí, která je přiložena v Příloha 17. Výsledkem metody je, že největší váhu má Cena (0,180421), následuje Licence (0,147655) a Propojení na Github (0,108668). Nejmenší váhu pak naopak dostalo kritérium Průmyslové odvětví. Jedná se o subjektivní názor autora, který kritéria koncipoval na přenos telemetrie z dronu. Celkové výsledky vah kritérií jsou uvedené v tabulce v sekci přílohy: Příloha 18. Tabulka vstupních dat je uvedena v Příloha 19.

4.1.6 Výsledky bodovací metody

Bodovací metodou autor této práce dospěl k výsledku, že z dvaceti porovnávaných platform se nejlépe umístila platforma od společnosti SAP s 1405 body. Nejhůře se pak umístila platforma MyDevices Cayenne se ziskem 720 bodů. Výsledná tabulka je přiložena v příloze: Příloha 20. Autor této práce přikládá dále pět nejhorších poskytovatelů a pět nejlepších, kteří jsou potenciální pro přenos telemetrie IoT. Srovnání je přiložené v přílohách: Příloha 21 a Příloha 22.

4.1.7 Výsledky bodovací metody s váhami kritérií

Dle výsledků bodovací metody, vyšla nejlépe platforma SAP Cloud Platform IoT Leonardo. Pro upřesnění použité metody, autor přidal ke každému kritérium váhu ze Saatyho metody. Výsledkem tohoto doplnění byla nová data, která určila vítěznou platformu

pro přenos IoT. Tabulka s body a přidanými váhami je přiložená v Příloha 25. Nejlépe se opět umístila platforma SAP Cloud Platform IoT Leonardo s 87,175 body, nejhůře pak dopadl poskytovatel ThingWorx s 30,261 body. Autor dále přikládá srovnání pěti nejlepších a nejhorších poskytovatelů dle výsledků v přílohách: Příloha 23 a Příloha 24.

4.1.8 Zhodnocení výsledků a vybrání vhodného poskytovatele

Výsledkem obou bodovacích metodik vyhrála s předstihem platforma SAP Cloud Platform s modulem IoT Leonardo. Tato platforma byla vybrána pro uskutečnění přenosu telemetrie přes IoT. Výsledek je založen na subjektivním hodnocení u všech platforem. V diplomové práci bude nadále využívána zkratka SCP pro SAP Cloud Platform.

4.2 Přenos telemetrie přes IOT

Pro uskutečnění přenosu telemetrie autor zvolil s použitím bodovací metodiky poskytovatele SAP Cloud Platform s rozšířením o modul SAP Leonardo pro IoT. Pro přenos telemetrie autor nastavil jednotku RPI, Trial účet na SAP Cloud Platform v edici Neon, nastavil službu IoT, nastavil databázi, propojení s databází a vytvořil aplikaci pro přehled telemetrie a odesílání příkazů.

Pro přenos telemetrie byly vytvořeny dva drony s rámy HEXA a QUAD pro lepší demonstraci a testování služby IoT. Tvorba dronů není obsahem této diplomové práce, autor pouze popsal komponenty, ze kterých se dron skládá.

4.2.1 Drony pro přenos telemetrie

Pro přenos telemetrie byly vybrány dva drony s rozdílnými verzemi jednotek RPI a to verze 2 a 3. Rozdílné verze byly zvoleny autorem z důvodu větší nezávislosti vytvořeného programu, mezi. V následující podkapitole autor této diplomové práce rozepsal potřebné komponenty, které jsou třeba pro uskutečnění přenosu a komponenty, ze kterých se dron skládá.

HEXA dron

Dron s typem rámu HEXA pro přenos telemetrie obsahuje následující nezbytné komponenty:

- 1x počítač Raspberry Pi 3 model B,
- 1x TFT display 320*480px pro RPI 3,
- 2x 3A UBEC 5V HOBBYWING (napájení RPI 3 a SP Racing),
- 1x modem Huawei E3372h (LTE),
- 1x SP Racing F3 Evo,
- 1x CRIUS Distribution Board,
- 1x micro USB kabel 20 cm,
- 1x baterie Turnigy Graphene Professional 10000mAh 4S 15C,
- 1x NEO-6 V3.0 GPS NEO-6M.

Dále se dron skládá z těchto částí, kterou jsou nezbytné pro pilotování a létání:

- 1x rám Tarot TL68C01 FY690S,
- 6x ESC Hobbywing SkyWalker 60A,
- 6x motor Turnigy D3536/6 1250KV,
- 4x vrtule 10x5 Black (CW/CCW),

- 1x přijímač Frsky V8FR II,
- 1x PPM Encoder V1.0,
- 1x FPV vysílač TBS Unify Pro HV 25-800mW 40ch,
- 1x FPV anténa TBS Triumph 5.8G anténa RHCP,
- 1x FPV kamera ODAY HD 700TVL CCD,
- 1x Gimbal Storm32 pro kameru GoPro Hero 5 session.

QUAD dron

Dron s typem rámu QUAD pro přenos telemetrie obsahuje následující nezbytné komponenty:

- 1x počítač Raspberry Pi 2 model B,
- 1x TFT display 320*480px pro RPI 2,
- 2x 3A UBEC 5V HOBBYWING (napájení RPI 2 a SP Racing),
- 1x modem Huawei E3131 (GSM),
- 1x SP Racing F3 deluxe,
- 1x CRIUS Distribution Board,
- 1x micro USB kabel 20cm,
- 1x baterie Turnigy Graphene Professional 8000mAh 6S 15C,
- 1x Radiolink M8N GPS.

Dále se dron skládá z těchto částí, kterou jsou nezbytné pro pilotování a létání:

- 1x rám Tarot Iron Man 650,
- 4x ESC dys XSD 30A 3-6S V1,
- 4x motor Tarot 6S 380KV 4008,
- 4x vrtule 1355 Black (CW/CCW),
- 1x přijímač Frsky X8R,
- 1x FPV vysílač TBS Unify Pro HV 25-800mW 40ch,
- 1x FPV anténa TBS Triumph 5.8G anténa RHCP,
- 1x FPV kamera MOBIUS ActionCam V3,
- 1x Tarot Metal TL03FLIR Gimbal pro GoPro Hero 5 Session,
- 2x Konektor xt90.

4.2.2 IoT struktura

V této kapitole IoT struktura autor odkazuje na teoretická východiska, kde byla popsána obecná struktura IoT. Tato struktura se skládá ze 3 částí: Hardware, Middleware a Software. V následujících sub odstavcích autor popsal strukturu přenosu telemetrie přes IoT.

Hardware

Do Hardware struktury autor zařadil nejdůležitější prvky pro přenos telemetrie, jsou to: ovládací jednotky dronů, počítače Raspberry Pi, mobilní modemy a cloudová platforma Sap Cloud Platform, ve které je implementována databáze a IoT server pro komunikaci. Ostatní hardware součásti jsou popsány v kapitole drony pro přenos telemetrie.

Middleware

V sekci Middleware autor zahrnul programové vybavení, které vytváří rozhraní mezi systémy. Autor v této práci do této sekce zahrnul knihovnu pyMultiwii pro čtení dat, vlastní knihovnu pro přenos telemetrie do IoT, rozhraní IoT pro nastavení zařízení a program wvdial pro nastavení mobilního Internetu v prostředí Linux.

Software

V posledním bodu Software, autor vytvořil aplikaci pro správu a analýzu přenesených dat. Tato aplikace je popsána v samostatné kapitole. Aplikace je založena na frameworku SAPUI5, která využívá technologii responzivního designu Fiori.

4.2.3 Schéma komunikace dronu s IoT

Pro vizuální představu komunikace dronu s IoT přikládá autor stavový diagram vytvořený metodikou UML pro charakterizaci přenosu telemetrie z dronu. Diagram popisuje chování v knihovně Main.py pro odesílání a přijímání telemetrie. Schéma popisuje činnosti algoritmu od připojení pohonné jednotky až po odpojení. V programu se nachází nekonečný cyklus omezený časovou prodlevou. Tato prodleva se dá nastavit v Config.py souboru. V cyklu jsou volané dvě funkce, jedna pro získání dat a druhá pro odeslání dat. V diagramu toto rozvětvení vychází ze stavu Zapnutý program. Nekonečný cyklus programu končí zastavením od uživatele nebo odepnutím elektrické energie. Diagram je přiložen v příloze: Příloha 35.

4.2.4 Databáze v SAP Cloud Platform

Autor pro ukládání dat využil dostupnou databázi v rámci trial účtu v platformě SCP. Využil systém HANA MDC (Multitenant Database Containers), který je standard ve světě SAP. Databázi autor pojmenoval droneiot. Nevýhodou databáze v trial účtu je dostupnost databáze pouze na 12 hodin, po této době se musí databáze aktualizovat. Další nevýhodou, kterou autor musel opravit, je tvorba dvou rolí, jedna je pro administrátora a druhá pro programátora na vývoj. Tato vytvořená databáze slouží IoT kokpitu pro ukládání dat, a vytvořené aplikaci pro správu a ukládání aktivity uživatele. V nastavené databázi autor využil sekci Catalog pro tvorbu tabulek a sekci Editor pro tvorbu oData pro Fiori aplikaci.

4.2.5 Schéma databáze

Pro přenos telemetrie, nastavení IoT služby a správnou funkci aplikace autor využil osm tabulek a jedno databázové view.

Použité tabulky jsou:

- DRONCOMMAND (tabulka příkazů do dronu),
- DRONLIST (tabulka s informacemi o dronu),
- T_IOT_408EB93C73C389B430CB (tabulka služby IoT s telemetrií z dronu),
- T_IOT_ACKSTORE (tabulky IoT služby),
- T_IOT_C6CF6810E0F18CBD9536 (tabulka služby IoT pro přenos fotek z dronu),
- T_IOT_CONFIG (tabulky IoT služby),
- T_IOT_HTTP_PUSH (tabulky IoT služby),
- T_IOT_MONITOR_LOG (tabulky IoT služby).

Databázové view je:

- UNS_DEVICES (view pro zobrazení nepřirazených dronů).

Autor práce vytvořil schéma databáze s tabulkami, které využívá v aplikaci a ke komunikaci, jedná se o tabulky:

- DRONCOMMAND,
- DRONLIST,
- T_IOT_408EB93C73C389B430CB,
- T_IOT_C6CF6810E0F18CBD9536.

K vyjmenovaným tabulkám autor přiložil do příloh schéma

Příloha 36.

4.2.6 Nastavení xsodata pro aplikaci

Pro správné fungování aplikace autor této práce vytvořil xsodata, která převádí data z databáze do aplikace. Odata využívají vnitřní tabulky a view SCP databáze. Služba xsodata je pojmenovaná iotscenario.xsodata, pro její obsluhu je vytvořena složka util s podsložkami CREATE, UPDATE a DELETE. Do těchto složek autor této práce implementoval obslužné metody pro práci s databází. Příklady metod lze nalézt v přílohách: Příloha 31, Příloha 32 a Příloha 33. Nastavení xsodata služby je pak přiloženo v Příloha 34.

Služba xsodata potřebuje pro správné fungování primární klíč, autor této práce musel upravit vygenerovanou tabulku z IoT služby o primární klíče, aby se mohla zobrazovat telemetrie v aplikaci. Do tabulky byly vloženy tři primární klíče příkazem ALTER TABLE. Syntaxe příkazu je následující:

```
ALTER TABLE "DRONMASTER"."T_IOT_408EB93C73C389B430CB"  
ADD PRIMARY KEY (G_DEVICE, G_CREATED, C_DEVICEID);
```

4.2.7 Komunikace s iotservices mms

V SAP Cloud Platform Cockpit v záložce Java Applications autor zapnul a nastavil službu pro IoT. Pojmenoval ji iotmms. V této službě autor založil propojení do databáze v záložce Data Source Bindings a nastavil ověření na tuto službu pro duplexní přenos v záložce Destinations.

Po přihlášení do Java aplikace IoT služby, autor pro komunikaci musel založit a zaregistrovat typy zpráv, typy zařízení a konkrétní zařízení. V sekci zprávy byly vytvořeny tři typy pro komunikaci mezi dronem aplikací a služnou IoT. Zprávy jsou:

- DATA_FROM_DRONE,
- CMD_TO_DRONE,
- PHOTO_FROM_DRONE.

Ukázka IoT kokpitu je přiložena v: Příloha 27. Struktury zpráv jsou pak zobrazené v: Příloha 28,

Příloha 29 a Příloha 30. Pro tyto zprávy byly systémem IoT vygenerované id pro správnou komunikaci, id jsou řazena stejně jako zprávy:

- 1a6fa20042f203191551,
- 408eb93c73c389b430cb,
- c6cf6810e0f18cbd9536.

V sekci Device Types autor vytvořil druh zařízení, ke kterému přiřadil vytvořené zprávy (messages). Device type autor vytvořil následující:

- TELEMETRY_DRONE.

Pro tento druh zařízení IoT systém vygeneroval ID a OAuth Token.

Posledním krokem pro komunikaci RPI a IoT službou je nastavení příslušných zařízení. Autor zaregistroval dvě zařízení pro přenos telemetrie, jsou to:

- HEXA_DRONE,
- QUAD_DRONE.

Pro tyto zařízení byly vygenerovány bezpečnostní Tokeny a id zařízení.

4.2.8 Získání telemetrie z dronu

Pro přenos telemetrie autor této práce využívá standardní protokol pro komunikaci s jednotkami, které využívají Multiwii 2.4 knihovnu. Princip komunikace s jednotkou a zasílání kódu je znázorněn na stránkách:

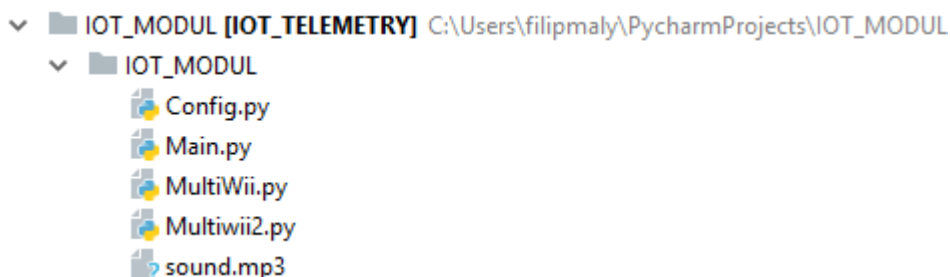
- http://www.multiwii.com/wiki/index.php?title=Multiwii_Serial_Protocol.

Pro získání telemetrie autor využil existující knihovnu pyMultiwii, kterou upravil a rozšířil o vlastní metody pro získání konkrétních informací pro přenos telemetrie. Při tvorbě diplomové práce se knihovna změnila, autor do výsledného přenosu telemetrie využil nejnovější verzi této knihovny. V přiloženém souboru s programem je i zachovaná stará implementace nad starší verzí knihovny. Knihovna je k nalezení na následující adrese:

- <https://github.com/alduxvm/pyMultiWii>.

Výsledný IoT modul pro počítače Raspberry Pi má následující strukturu:

Obrázek 30 Struktura IoT modulu pro přenos telemetrie.



Zdroj: vlastní práce autora

Autor zahrnul starou verzi knihovny MultiWii.py a novou Multiwii2.py, dále vytvořil dvě třídy Config.py a Main.py. Třída Config.py slouží k nastavení zařízení, zde se zadávají hodnoty pro komunikaci s SCP IoT Leonardo serverem. Zdrojový kód je přiložen v Příloha 37. Třída Main slouží v programu pro inicializaci a spuštění celého programu. V této třídě autor implementoval přijímání a odesílání zpráv. Autor této práce knihovnu stále vyvíjí a doplňuje o nové funkčnosti. Třída Main.py ve verzi 1.2 je přiložená v

Příloha 38.

4.2.9 Propojení IoT Serveru s dronem

Propojení dronu a IoT serveru pomocí zpráv, typu zařízení a zařízení bylo již popsáno v předešlých kapitolách. Komunikace probíhá prostřednictvím bezpečnostních tokenů. Ve třídě main je implementovaná metoda „startProgram“, která má na starosti pravidelnou komunikaci s IoT serverem. V programu se nachází nekonečný cyklus, který je omezený funkcí sleep pro pozastavení cyklu a parametr pro délku prodlevy. Funkce „startProgram“ vypadá takto:

```
def startProgram(self):
    counter = 1
    try:
        while True:
            self.sendTelemetry(self.getTelemetryData(),
                               self.config.getDeviceUrl(),
                               self.config.getTelemetryHeader())
            self.getResponse()
            time.sleep(self.config.getFromMetadata('timeSleep'))
            print counter
            counter = counter + 1
    except Exception, error:
        print "Chyba v cyklu programu " + str(error)
```

.

4.2.10 Nastavení RPI pro připojení k Internetu

Pro přenos telemetrie autor této práce nastavil počítač Raspberry Pi. Kroky potřebné k přenosu telemetrie jsou popsány v následujících sekcích. Instalace Raspbian do RPI autor provedl pomocí instalátoru NOOBS, který je dostupný s postupem v následujícím odkazu: <https://www.raspberrypi.org/downloads/noobs/>.

Po instalaci a iniciálních nastavení, bylo potřeba provést update a upgrade systému na novou verzi pomocí dvou následujících příkazů:

- `sudo apt-get upgrade,`
- `sudo apt-get update.`

Nastavení bezdrátového přenosu prostřednictvím modemu bylo potřeba provést pomocí programu `wvdial`. Tento program se nainstaloval pomocí následujícího příkazu:

- `sudo apt-get install wvdial.`

Po úspěšné instalaci autor upravil spouštěcí skript pro ovládání modemu. Ve složce `/etc` upravil soubor `wvdial.conf`. Soubor otevřel přes příkaz `sudo` v editoru `nano` ve znění:

- `sudo nano /etc/wvdial.conf.`

Do souboru zapsal připravenou syntaxi, která je uvedena v příloze: Příloha 26 Skript pro nastavení modemu přes `wvdial`. Spuštění modemu autor ověřil příkazem:

- `sudo wvdial`

Autor narazil na problém v systému Linux u externích modemů, načítaly se jako USB disk. Pro správné fungování připojuje modem před zapnutím RPI.

4.2.11 Nastavení RPI import skriptů pro čtení dat

Kvůli správnému nastavení RPI, autor musel využít v práci skripty pro Shell programování. Shell skript je počítačový program navržený tak, aby mohl být spuštěn pomocí shellu Unixu. Shell je interpret příkazového řádku v operačním systému. Typické operace prováděné pomocí shell skriptů zahrnují základní operace: manipulaci se soubory, spuštění programu a tisk textu. (83 str. 342)

Pro kontrolu nastavení a funkčnosti autor v RPI implementoval dva skripty. Jeden na zapnutí modemu a druhý pro zapnutí programu. Jedná se o jednoduché batch skripty, které jsou dále využívány k automatickému spuštění. Pro oba dva skripty autor nastavil administrátorská práva `rwx` příkazem `chmod 777`. Nastavení vypadalo následně:

- `chmod 777 /pi /start_modem.sh,`
- `chmod 777 /pi/start_program.sh.`

Skript `start_modem`:

```
#!/bin/bash  
sudo wvdial
```

.

Skript `start_program`:

```
#!/bin/bash
cd /home/pi/test/DRONE_SERIAL
python get_data_new.py
.
```

4.2.12 Nastavení RPI na automatické spuštění

Pro automatické spuštění programu autor této práce musel nastavit specificky systém Linux. Pro správnou inicializaci využil dva skripty `start_modem.sh` a `start_program.sh`. V první kroku založil autor skript `my_startup_script.sh` a vytvořil je ve složce `/usr/local/bin/`. Nastavil na něm práva `chmod 755` (vlastník, skupina a ostatní).

```
Skript my_startup_script.sh:
#!/bin/bash
/home/pi/start_modem.sh &
/home/pi/start_program.sh &
exit 0
.
```

V dalším kroku založil autor ve složce `/etc/systemd/system/` soubory `my_startup_script.service` a `my_startup_script.timer`. Na oba soubory autor přidělil práva příkazem `chmod 755`.

```
Skript my_startup_script.service
[Unit]
Description = Skript pro automaticke spusteni
[Service]
Type=forking
ExecStart=/usr/local/bin/my_startup_script.sh
```

```
Skript my_startup_script.timer
[Unit]
Description= Skript který se spusti minutu po spusteni
[Timer]
# Cas který program ceka na spusteni
OnBootSec=1min
Unit=my_startup_script.service
[Install]
WantedBy=multi-user.target
```

V posledním kroku nastavení autor použil příkaz:

- `Systemctl enable my_startup_script.timer`.

4.2.13 Nastavení RPI pro display 480*320px

Autor pro lepší práci s drony a jejich testování využil displeje o rozlišení 480*320px. Postupoval podle návodu z následujících stránek:

[https://www.waveshare.com/wiki/3.5inch_RPi_LCD_\(A\)#Method_1._Driver_installation](https://www.waveshare.com/wiki/3.5inch_RPi_LCD_(A)#Method_1._Driver_installation).

Pro správnou inicializaci autor vykonal sled dále popsanych činností. Stáhl soubor s ovladači, který nakopíroval do RPI a vykonal následující příkazy:

- `tar xvf LCD.tar.gz,`
- `cd LCD-show/,`
- `chmod +x LCD35-show,`
- `./LCD35-show.`

Pro přepínání mezi LDC displejem a HDMI výstupem, využil autor následující příkazy:

- `cd LCD-show/,`
- `./LCD-hdmi` (pro LCD naopak `./LCD35-show`).

Autor této práce zjistil, že displej použitý v RPI má větší odběr ampér, nelze ho napájet z USB portu. V dronu proto je speciální UBEC obvod s odběrem do 3A.

4.3 Tvorba aplikace

V poslední části praktické části se autor zaměřil na tvorbu aplikace. Aplikace je tvořena pomocí jazyku JavaScript a XML s responzivním designem. Pro tvorbu je využita platforma Fiori s knihovny SAPUI5. Pro propojení s IoT serverem byla využita knihovna dodávána poskytovatelem SCP. Jedná se tedy o HTML aplikaci s responzivním designem.

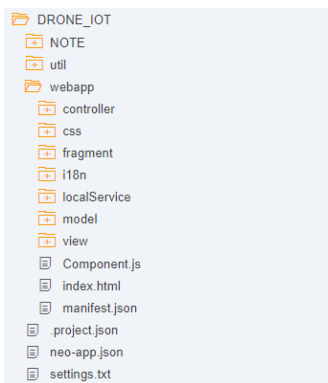
4.3.1 Základní požadavky na aplikaci

Základním požadavkem na aplikaci je zobrazení telemetrie z dronu, dalším požadavkem je odesílání příkazů do dronu a posledním požadavkem je vedení záznamů o dronu a jeho historii. Platforma Fiori díky frameworku SAPUI5 všechny tyto náležitosti splňuje, a proto také bude využita pro tvorbu aplikace.

4.3.2 Architektura aplikace

Aplikace je tvořena pomocí architektury MVC (model, view, controller). Tato architektura je popsána detailněji v metodice této práce. Prvky architektury MVC jsou vidět ve struktuře projektu. Složka controller obsahuje ovládací funkce aplikace, view obsahuje hlavní obrazovku se třídou sap.m.SplitApp pro rozdělený pohled a složka model obsahuje „device“ model pro rozpoznání zařízení. Nastavení aplikace je v deskriptoru aplikace, soubor, manifest.json. Ve složce fragment jsou uloženy pomocné soubory pro tvorbu view. Celá aplikace se spouští souborem index.html.

Obrázek 31 architektura MVC struktura aplikace



Zdroj: vlastní práce autora

4.3.3 Knihovny využité v aplikaci

Pro tvorbu HTML aplikace byly využité následující knihovny. Základní knihovnou pro tvorbu aplikace byla knihovna SAPUI5, která disponuje i otevřenou knihovnou OpenUI5 pro bezplatné použití. SDK knihovny je k nalezení na stránkách:

- <https://sapui5.hana.ondemand.com/#>.

Další knihovny byly přidány z balíku od společnosti SAP, ve kterých se nachází třída pro komunikaci s IoT službou. Třída využitá pro přenos telemetrie je `sap.services.IOT`. Stránky se službou jsou:

- <https://github.com/SAP/iot-starterkit>.

Poslední využitou knihovnou byla knihovna pro integraci Google map. Jedná se verzi pro HTML využití a je integrována ve složce `fragment` v soboru `monitoring.fragment.js`. Knihovna byla převzata z oficiálních stránek Googlu:

- <https://developers.google.com/maps/web-services/>.

4.3.4 Návrh aplikace

Aplikace, jak již bylo zmíněno využívá třídu `sap.m.SplitApp`. Tato třída rozděluje výchozí obrazovku na detailní a master pohled. V master pohledu se nachází výběr a editace dronu s nastavením aplikace. V části detail se pak nachází třída `sap.m.IconTabBar`, která rozděluje pravou část aplikace na jednotlivé pohledy. V této třídě jsou pak zobrazeny pohledy: Monitoring, Requesty, Fotografie, DB tabulka a O Aplikaci. Ve třídě `sap.m.SplitApp` je zapnutý „hideMode“, který skrývá master část a zobrazuje iniciálně jen detail.

4.3.5 Popis základních obrazovek

V následujících odstavcích autor zpracoval popsání funkčnosti a vzhledu obrazovek, které jsou k nalezení v aplikaci. Jsou popsány jen klíčové prvky pro ovládání aplikace. Obrazovky jsou rozděleny na master a detailní.

Master obrazovka výběr dronu

V master obrazovce pro výběr dronu, se nachází drop-down seznam (třída `sap.m.Select`) pro výběr modelu. Po změně modelu se v aplikaci aktualizují všechna data.

Nad tímto prvkem se nachází ovládání aktualizace dat v aplikaci. Uživatel si může nastavit čas ve vteřinách, v jakém cyklu se data budou aktualizovat. Zbylé prvky v master obrazovce jsou vstupní pro zobrazení nebo zadání hodnot. V této obrazovce pro výběr dronu jsou tyto prvky deaktivované pro editaci. V hlavičce obrazovky se nachází tlačítko s ikonou „+“ pro přidání nepřirazeného dronu. V patičce stránky se pak nachází editační tužka pro aktivaci editace zadávacích polí. Prvek výběr dronu je přiložen v příloze: Příloha 40, prvek aktualizace databáze v příloze: Příloha 41, celé detail view je přiloženo v příloze:

Příloha 39.

Master obrazovka editace dronu

V master obrazovce pro editaci dronu, jsou aktivované všechna editační pole. V této obrazovce může uživatel editovat záznam, přidat nový dron anebo vymazat dron z databáze. V hlavičce se opět nachází tlačítko pro přidání modelu. V patičce stránky jsou pak zobrazeny tři tlačítka, tlačítko s ikonou „koše“ pro mazání, tlačítko s ikonou „kazety“ pro uložení změn, a tlačítko se šipkou „zpět“ pro návrat do základní obrazovky. Celá obrazovka je přiložena v příloze: Příloha 42. Při stisku tlačítka pro mazání dronu, se zobrazí vyskakovací potvrzovací dialog, pro zamezení nechtěného zmáčknutí. Dialog je přiložen v příloze: Příloha 43.

Master obrazovka přidání dronu

V obrazovce po kliknutí na tlačítko „+“, se zobrazí dialogové okno pro přidání nepřirazeného dronu. V dialogu se nachází opět drop-down list pro výběr nepřirazeného dronu. Po výběru je nutné vyplnit povinná pole označená v popisu „*“. Dialog jde zavřít nebo potvrdit. Po potvrzení se vybraný dron uloží do databáze a nabídne se k vybrání v master obrazovce pro výběr dronu. V dialogu jsou kontrolována vstupní pole, nevyplněné pole se označí červeně. Dialog je přiložený v příloze: Příloha 44.

Detail obrazovka Monitoring

Detailní obrazovka monitoring obsahuje integrovanou mapu od společnosti Google. Na této obrazovce může uživatel dělat následujících několik akcí. Pomocí tlačítka „Aktuální pozice“, se aplikace pokusí najít pozici uživatele pomocí Internetu. Dále se zde

nacházejí dvě vstupní pole pro zadání zeměpisné šířky a délky. Při správném vyplnění těchto polí se aplikace po kliknutí na tlačítko „šipky“ pokusí zobrazit bod na mapě dle zadaných hodnot. Poslední funkcí této aplikace je zobrazení letu dráhy dronu, kde se vybírají hodnoty z obrazovky DB tabulka. Tato funkce bude více popsána v uvedené detailní obrazovce. Obrazovka Monitoring ve výchozím stavu je přiložena v příloze: Příloha 45. Obrazovku Monitoring se zadanou dráhou letu je pak zobrazená v příloze: Příloha 46.

Detail obrazovka Requesty

Detailní obrazovka requesty slouží uživateli pro odesílání příkazů do zvoleného dronu z master obrazovky. Tato obrazovka je rozdělena do tří částí. V levé části se nachází ovládací panel s tlačítky pro odesílání příkazů do dronu. V pravé části se pak nachází graf s přehledem úspěšných a neúspěšných příkazů do dronu. V poslední spodní části této obrazovky, se pak nachází informativní tabulka s přehledem příkazů. Uživatel může příkazy odesílat jednotlivě anebo tlačítkem Poslat najednou. Obrazovka je zobrazená v příloze: Příloha 47.

Detail obrazovka DB tabulka

Obrazovka DB tabulka slouží pro zobrazení telemetrie z dronu. Tato tabulka má v této diplomové práci zobrazené sloupce Datum, Model, Napětí, Proud mA, Satelity, Zeměpisná Šířka a Zeměpisná délka. Sloupce se dají programově měnit v XML souboru, do tabulky je nahráván celý JSON soubor se všemi daty skrze oData službu. V tabulce je možné mazat záznamy jednotlivě nebo skupinově, pokud uživatel zaškrtně více záznamů. Každý záznam nabízí tlačítko, které po stisku zobrazí v obrazovce monitoring bod měření na mapě. V hlavičce tabulky se nachází třída sap.m.Toolbar, ve které jsou implementovány tři tlačítka. Tlačítko pro otevření filtru pro volbu dne (implicitně je zde nastavován aktuální den), viz příloha: Příloha 50. Následující je tlačítko pro mazání více záznamů, které po kliknutí zobrazí potvrzovací dialog. Posledním tlačítkem v hlavičce tabulky je tlačítko pro zobrazení trasy dronu. Pro aktivaci této funkce, musí uživatel vybrat body z tabulky, které chce spojit. V dalším rozvoji této aplikace bude nad tabulkou rozšířen filtr pro filtrování tras v daný den. Ukázka filtru je přiložena v příloze Příloha 50. Poslední funkcí nad tabulkou, je aktuální výpis zaškrtnutých řádků, tato funkce je vidět

u dialogu pro mazání více záznamů viz příloha: Příloha 51 Detailní obrazovka tabulka je přiložená v Příloha 49. Zobrazení vybraných bodů jako trasa letu je přiložené v Příloha 46.

Detail obrazovka O Aplikaci

Poslední detailní obrazovkou v aktuální verzi aplikace je obrazovka O Aplikaci. V této části je připravený jednoduchý formulář se jménem autora a rokem výroby. U této obrazovky autor počítá s budoucím rozvojem, například přidáním legendy pro ovládání aplikace a dalších údajů. Tato obrazovka je přiložená v příloze: Příloha 52.

4.3.6 Výchozí stav aplikace pro jednotlivá zařízení

Aplikace je vytvořena způsobem, aby byla spustitelná na nejrůznějších typech zařízení. Autor tuto funkčnost otestoval v prostředí prohlížeče Chrome ve vývojářských nástrojích. Pomocí přepínání zařízení otestoval funkčnost na mobilní, desktopové a tablet platformě. Dále autor využil svůj mobilní telefon Asus ZenFone 4 pro spuštění aplikace v terénu a otestoval telemetrii a posílání příkazů do dronu. Aplikace má upravené ovládání pro mobilní telefony. Na těchto zařízeních se zobrazuje pouze jedna obrazovka, buď detail nebo master. K přepínání obrazovek je uživateli zobrazeno tlačítko pro navigaci zpět. Výchozí stavy obrazovek pro jednotlivá zařízení jsou přiloženy v přílohách: Příloha 53, Příloha 54, Příloha 55.

4.3.7 Postup implementace aplikace

Aplikace byla autorem této práce tvořena paralelně s implementací přenosu telemetrie z dronu. V aplikaci byla využita nejnovější dostupná sada knihoven SAPUI5 ve verzi 1.52. Pro tvorbu autor využil architekturu MVC, jak již bylo zmíněno. Při integraci Google map autor musel zaregistrovat aplikaci v prostředí Google kvůli získání developerského klíče. Dalším krokem při implementaci bylo nastavení oData služby a propojení s aplikací v deskriptoru aplikace. Poslední věcí, kterou autor nastavil byla v prostředí kokpitu SCP, kde autor nastavil destinace pro přístup aplikace k datům a službě IoT. Schéma aplikace vytvořené pomocí stavového diagramu je přiložené v příloze: Příloha 56.

4.3.8 Princip komunikace aplikace se službou IoT

Komunikace aplikace s IoT službou je zprostředkována pomocí dodávané knihovny od společnosti SAP. Autor této diplomové práce nastavil soubor neo-app.json, ve kterém stanovil navigaci na IoT služby. Nastavení této služby je důležité, z důvodu zabránění chyb CORS (Cross-Origin Resource Sharing) u webových aplikací. Nastavení v aplikaci vypadá následovně:

```
{
  "path": "/iotmms/",
  "target": {
    "type": "destination",
    "name": "iotmms",
    "entryPath": "/http"
  },
  "description": "iotmms"
},
```

Dalším krokem bylo nastavení této destinace v prostředí SCP. Pro komunikaci s IoT službou je v aplikaci definován typ zprávy „message type“. Tato hodnota je zadána v setModels funkci ve třídě master.Controller.js. V budoucím rozvoji aplikace, bude tato hodnota vyčítána z databáze pro konkrétní zařízení. Pro odeslání do IoT služby je potřeba znát údaje Message type a Id zařízení. Id zařízení se přebírá z hlavní obrazovky z atributu G_DEVICE a je nezbytný pro směrování pro konkrétní typ dronu. Pro správnou komunikaci s IoT službou je potřeba využívat přesné hodnoty z nastavené této služby.

4.3.9 Zdrojový kód aplikace

Zdrojový kód aplikace je psán jazyky XML a JavaScript. Pro deskriptor aplikace byl využit soubor json. Celá aplikace je přiložená v CD příloze dodané s diplomovou prací. Aplikace byla psaná v prostředí SCP ve webovém editoru SAP WebIde.

4.3.10 Potenciální rozvoj aplikace

Autor této práce aplikaci nadále rozšiřuje a upravuje dle vlastních potřeb pro přenos telemetrie. Potenciálním rozvojem aplikace bude dodělán funkčnosti pro Google mapu, která bude zobrazovat aktuální dráhu letu dronu. Další funkčností bude rozšíření sady příkazů do dronu, kde bude kladen důraz na let dle zadaných souřadnic vybraných z mapy. Další rozvoj je plánován u obrazovky DB tabulka, kde bude přidána možnost zvolit

si sloupce. Dále bude rozšířena stávající sada filtrů pro tabulku. Autor této práce s těmito možnostmi počítal a tvořil aplikaci pro možné rozšíření.

5 Výsledky a diskuse

V této diplomové práci autor splnil hlavní a dílčí cíle. V následujících odstavcích jsou popsány výsledky této práce s přílohami pro zobrazení výsledků.

5.1 Shrnutí vícekriteriální analýzy variant

Výsledkem vícekriteriální analýzy variant, jak již bylo popsáno, je výherní platforma od společnosti SAP. Tato platforma se ukázala jako nejvhodnější prostředek pro tvorbu přenosu telemetrie. Platforma v bezplatném developerském účtu poskytuje databázi, což byl velký přínos a usnadnění práce. Dále se v platformě nachází důmyslný systém rolí a oprávnění, který by mohl mít potenciál v komerčním využití. Autor s výsledkem vícekriteriálních analýz souhlasí a implementoval na jejich základě aplikaci a komunikační modul.

5.2 Shrnutí vytvořené aplikace

Pro zobrazení telemetrie byla vytvořena responsivní aplikace otestovaná na nejtypičtějším zařízeních (mobil, tablet a desktop). Aplikace funguje bez chyb a na všech těchto zařízeních. Aplikace je psaná v návrhovém vzoru MVC s modelem pro jazykovou sadu i18n. Základní jazyk v aplikaci je čeština, aplikaci lze pak jednoduše rozšířit o další jazykové mutace přidáním souboru do složky i18n. Aplikace není vázána přímo na platformu SCP, jednoduchou úpravou by šla využít i u dalších poskytovatelů představených v analýze. Skrz aplikaci lze tedy zobrazit data z dronu a odeslat jednoduché příkazy do dronu. U aplikace je počítáno s postupným rozšiřováním a autor této práce na ní stále pracuje a rozvíjí ji.

5.3 Shrnutí přenesené obousměrné telemetrie

Pomocí vytvořené aplikace a komunikačního modulu pro přenos telemetrie autor přenesl obousměrně data. Pomocí programu Pycharm autor naprogramoval komunikační modul. V přílohách: Příloha 57 a Příloha 58 je zobrazena úspěšná komunikace s IoT službou. V následujících přílohách: Příloha 59 a Příloha 60 je zobrazena komunikace aplikace s IoT službou. Autor úspěšně přenesl obousměrně telemetrii v terénu i testovacím prostředí.

5.4 Vytvořené prototypy pro přenos telemetrie

Pro přenos telemetrie byly vytvořeny dva prototypy dronu s rámem HEXA a QUAD. Na těchto dronech byly implementovány počítače Raspberry Pi. Na dronech byl nainstalován software CleanFlight pro základní ovládání dronu. Telemetrie byla úspěšně přenesena za letu dronu a bylo ověřeno paralelní odesílání a přijímání dat z obou zařízení. Ukázky dronů jsou přiloženy v přílohách: Příloha 61, Příloha 62, Příloha 64 a Příloha 63.

5.5 Přílohy na CD disku

Součástí této diplomové práce je přiložený CD disk s přílohami. Na kompaktním disku se nachází Microsoft excel soubor s provedenými analýzami. Dále se zde nachází vyexportovaná aplikace ve formátu zip a paket s databází také ve formátu zip. Poslední přílohou na CD disku je modul pro přenos telemetrie z dronu v programovacím jazyce Python.

6 Závěr

Hlavním cílem diplomové práce bylo přenést telemetrii z dronu přes Internet of Things (IoT). V teoretické části byly charakterizovány vybrané pojmy a problematika, ve které se využívá IoT a s ním spojená komunikace. Dále byly popsány nezbytné metody a principy potřebné k přenosu telemetrie přes IoT.

Na tvorbu rešeršní části práce byly použity odborné a vědecké zdroje v elektronické a tištěné podobě. V této práci, z důvodu rychle se rozvíjejícího se odvětví, převažují zejména zahraniční a internetové zdroje. Při dohledávání kvalitních zdrojů a titulů se autorovi nejvíce osvědčily známé databáze Ebrary, ScienceDirect a Google Scholar. Z internetových zdrojů pak autor využil především oficiální stránky výrobců a technologií použitých v této diplomové práci.

Jedním z dílčích cílů diplomové práce byla analýza IoT rozhraní včetně porovnání s nejběžnějšími poskytovateli IoT. V této části autor porovnal dvacet poskytovatelů IoT a vybral nejvhodnějšího poskytovatele pro realizaci přenosu přes IoT. Pro výběr poskytovatele využil autor dvojího srovnání pomocí vícekritériální analýzy variant, a to zejména metodu Bodovací a Saatyho metodu vah kritérií. Pro metodu vah kritérií autor stanovil šestnáct srovnávacích kritérií. V obou případech zvítězila platforma SAP Cloud Platform s modulem IoT Leonardo, skrze kterou pak autor uskutečnil přenos telemetrie.

Druhým dílčím cílem byla tvorba responsivní aplikace pro komunikaci s telemetrickým zařízením prostřednictvím IoT. Aplikace byla autorem vytvořena prostřednictvím technologie Fiori s využitím frameworků SAPUI5 a integrací Google map knihoven. Do aplikace autor integroval knihovnu od vítězné platformy SAP Cloud Platform pro komunikaci s IoT službou. Aplikace byla otestována na třech typech zařízení: mobilní telefon, osobní počítač a tablet. Na všech zařízeních aplikace pracovala bez problémů. S využitím aplikace autor odeslal příkazy do dronu a zobrazil data z dronu. V aplikaci autor zobrazil pomocí Google map trasu dronu a data. Autor prostřednictvím odeslaných příkazů zapnul na dronu alarm a vytvořil fotografii pomocí přídavné kamery pro počítač Raspberry Pi. Aplikaci vytvořil autor pomocí jazyků JavaScript a XML. Jako datovou základnu pro aplikaci využil autor databázi od poskytovatele SAP Cloud Platform. Dále autor vytvořil xsodata službu pro propojení aplikace s databází a obslužné metody pro základní CRUD operace.

Posledním dílčím cílem byla implementace komunikace s prototypem dronu. Autor pro tento cíl nastavil databázi a založil v ní potřebné tabulky a další nastavení pro správný běh databáze. Pro komunikaci s IoT připravil službu poskytovatele, ve které nastavil propojení do databáze, typy zpráv pro odesílání telemetrie, typy zařízení a zaregistroval koncová zařízení. Při uskutečnění přenosu autor využil dva navržené prototypy dronu, vytvořené konkrétně pro účely této práce. Drony byly popsány detailněji v praktické části této práce. Pro komunikaci s IoT službou autor dále vytvořil vlastní knihovnu v jazyce Python, kterou použil v počítačích Raspberry Pi. Autor vytvořil spouštěcí skripty pro systém Linux, které zabezpečují propojení skrze Internet a zapnutí programu po spuštění počítače.

Autor tedy vytvořil dva prototypy dronů, provedl analýzu IoT poskytovatelů, vytvořil aplikaci a přenesl obousměrnou telemetrii z dronů. Segment IoT a drony se stále vyvíjí a autor nadále využívá poznatky z této práce a rozšiřuje je. Vytvořená aplikace a moduly pro přenos telemetrie lze s mírnou úpravou přenést i na ostatní poskytovatele IoT. Tato diplomová práce má budoucí potenciál pro autonomní ovládání dronů jako je například monitoring zadaného prostoru dle souřadnic a výšky, doručování zásilek na zadané souřadnice a sbírání nejrůznějších telemetrických dat s následným odesláním pomocí počítače Raspberry Pi.

7 Seznam použitých zdrojů

1. **Pavel, Burian.** *Internet inteligentních aktivit.* Praha : Grada, 2014. 978-80-247-5137-5.
2. **Shancang Li, Li Da Xu.** *Securing the internet of things.* Cambridge : MA: Elsevier, 2017. 9780128044582.
3. **Pohanka, Pavel.** Internet věcí. *Pavel Pohanka.* [Online] 2017. <http://i2ot.eu/internet-of-things/>.
4. **Sklenák, Vilém.** *Data, informace, znalosti a Internet.* Praha : C.H. Beck, 2001. 9788071794097.
5. **Arora, Ritu.** *Conquering Big Data with High Performance Computing.* Berlín : Springer, 2016. 9783319337425.
6. **Zhou, Honbo.** *The Internet of Things in the Cloud: A Middleware Perspective.* Boca Raton : CRC Press, 2015. 9781498760010.
7. **Sarangi, Pallavi Sethi and Smruti R.** <https://www.hindawi.com>. *Internet of Things: Architectures, Protocols, and Applications.* [Online] 26. 1 2017. <https://www.hindawi.com/journals/jece/2017/9324035/>.
8. **Bessis Nik, Dobre Ciprian.** *Big data and internet of things: a roadmap for smart environments.* Cham : Springer, 2014. 978-3-319-05029-4.
9. **Xavier, Franch.** *2014 International Conference on Future Internet of Things and Cloud (FiCloud 2014).* Barcelona : Piscataway, NJ: IEEE., 2014. 9781479943562.
10. **Hwaiyu, Geng.** *The internet of things and data analytics handbook.* New Jersey : John Wiley, 2017. 9781119173649.
11. **Technologie, MQTT.** *IoT portál.* [Online] 24. květen 2016. <https://www.iot-portal.cz/2016/05/24/mqtt/>.
12. **redakce. Technologie MiWi.** *IoT portál.* [Online] 5. květen 2016. <https://www.iot-portal.cz/2016/05/06/miwi/>.
13. **Redakce. NarrowBand IoT.** *IoT portál.* [Online] 30. duben 2016. <https://www.iot-portal.cz/2016/04/30/narrowband-iot/>.
14. **Wi-Fi HaLow.** *IoT portál.* [Online] 29. únor 2016. <https://www.iot-portal.cz/2016/02/29/wi-fi-halow/>.
15. **Sigfox.** *IoT Portál.* [Online] 26. únor 2016. <https://www.iot-portal.cz/2016/02/26/sigfox/>.
16. **LoRaWan.** *IoT portál.* [Online] 29. únor 2016. <https://www.iot-portal.cz/2016/02/29/lorawan/>.
17. **Z-Wave.** *IoT portál.* [Online] 26. únor 2016. <https://www.iot-portal.cz/2016/02/26/z-wave/>.
18. **Bluetooth Smart.** *IoT portál.* [Online] 26. únor 2016. <https://www.iot-portal.cz/2016/02/26/bluetooth-smart/>.
19. **ZigBee.** *IoT portál.* [Online] 24. únor 2016. <https://www.iot-portal.cz/2016/02/24/zigbee/>.
20. **IoT, redakce. Internet of Things (2.část) – Technologie pro bezdrátový přenos dat.** *IoT portál.* [Online] 25. 12 2017. <https://www.iot-portal.cz/2017/12/25/internet-of-things-2-cast-technologie-pro-bezdratovy-prenos-dat/>.
21. **2013 27th International Conference on Advanced Information Networking and Applications workshops (WAINA 2013).** Barcelona : Piscataway, 2013. 978-1-4673-6239-9.

- 22. IoT, redakce. Internet of Things (3.část) – Technologie pro bezdrátový přenos dat.** *IoT portál*. [Online] 27. 12 2017. <https://www.iot-portal.cz/2017/12/27/internet-of-things-3-cast-technologie-pro-bezdratovy-prenos-dat/>.
- 23. Redakce. Mapa pokrytí IoT.** *IoT portál*. [Online] 2017. <https://www.iot-portal.cz/mapa-pokryti/>.
- 24. IoT, redakce. Komerčních projektů na síti Sigfox přibývá.** *IoT portál*. [Online] 8. říjen 2017. <https://www.iot-portal.cz/2017/10/08/komercnich-projektu-na-siti-sigfox-pribyva/>.
- 25. Prevence je lepší než chemický postřik.** *IoT portál*. [Online] 3. leden 2018. <https://www.iot-portal.cz/2018/01/03/prevence-je-lepsi-nez-chemicky-postrik/>.
- 26. Hydroponie je věda, rostliny pěstuje chytrý software.** *IoT portál*. [Online] 17. říjen 2017. <https://www.iot-portal.cz/2017/08/17/hydroponie-je-veda-rostliny-pestuje-chytry-software/>.
- 27. Inteligentní systém zavlažování Airdrop – vítězný projekt Czech IoT Summer Jam 2017.** *IoT portál*. [Online] 30. 11 2017. <https://www.iot-portal.cz/2017/11/30/inteligentni-system-zavlazovani-airdrop-vitezny-projekt-czech-iot-summer-jam-2017/>.
- 28. Vodafone zajistí připojení nového vozu ŠKODA Kodiaq.** *IoT portál*. [Online] 9. 6 2016. <https://www.iot-portal.cz/2016/09/06/vodafone-zajisti-pripojeni-noveho-vozu-skoda-kodiaq/>.
- 29. Innogy nabízí domácnostem kontrolu spotřeby plynu přes Sigfox.** *IoT portál*. [Online] 25. 10 2017. <https://www.iot-portal.cz/2017/10/25/innogy-nabizi-domacnostem-kontrolu-spotreby-plynu-pres-sigfox/>.
- 30. What is Arduino.** *Arduino*. [Online] <https://www.arduino.cc/en/Guide/Introduction#>.
- 31. HobbyKing, redakce. Multiwii and Megapirate AIO Flight Controller w/FTDI (ATmega 2560) V2.0.** *HobbyKing*. [Online] https://hobbyking.com/en_us/multiwii-and-megapirate-aio-flight-controller-w-ftdi-atmega-2560-v2-0.html?__store=en_us.
- 32. Kadamatt, V. Best flight controller for quadcopter [2017] : How do I choose?** *droneybee*. [Online] 1. 8 2017. <http://www.droneybee.com/best-flight-controller-quadcopter/>.
- 33. CleanFlight, redakce. WHAT'S CLEANFLIGHT?** *cleanflight*. [Online] <http://cleanflight.com/>.
- 34. Keller, Michael. Introduction. Welcome to the BetaFlight Wiki!** [Online] 15. 1 2018. <https://github.com/betaflight/betaflight/wiki>.
- 35. PI, redakce Raspberry.** *raspberrypi*. *Getting started with the Raspberry Pi*. [Online] <https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started>.
- 36. Meet the Raspberry Pi.** *raspberrypi*. [Online] <https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started/3>.
- 37. Upton Eben, Halfacree Gareth. Raspberry Pi: uživatelská příručka. 2., aktualizované vydání.** Brno : Computer Press, 2016. 978-80-251-4819-8.
- 38. RASPBERRY PI ZERO.** *raspberrypi*. [Online] <https://www.raspberrypi.org/products/raspberry-pi-zero/>.
- 39. RASPBERRY PI ZERO W.** *raspberrypi*. [Online] <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>.
- 40. RASPBERRY PI 2 MODEL B.** *raspberrypi*. [Online] <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.
- 41. RASPBERRY PI 3 MODEL B.** *raspberrypi*. [Online] <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.

- 42. What is Raspbmc?. Mapt beta.** [Online]
https://www.packtpub.com/mapt/book/hardware_and_creative/9781782163022/1/ch011v11sec10/what-is-raspbmc.
- 43. Gordon, Whitson. Whitson Gordno.** *lifehacker*. [Online] 20. 11 2010.
<https://lifehacker.com/5851924/openelec-is-a-hassle-free-xbmc-distribution-for-home-theater-pcs>.
- 44. Larabel, Michael. XBMC-Focused OpenELEC 1.0 Released.** *phoronix*. [Online] 20. 10 2011. https://www.phoronix.com/scan.php?page=news_item&px=MTAwMzM.
- 45. Klosowski, Thorin. Pidora Is a Fedora-Based Operating System for the Raspberry Pi.** *lifehacker*. [Online] 24. 5 2013. <https://lifehacker.com/pidora-is-an-alternative-operating-system-for-the-raspb-509687690>.
- 46. archWiki,** redakce. Arch Linux. *archlinux*. [Online] 13. 1 2018.
https://wiki.archlinux.org/index.php/Arch_Linux.
- 47. NOOBS.** *raspberrypi*. [Online] <https://www.raspberrypi.org/downloads/noobs/>.
- 48. LAFAY, Mark.** *Drones for Dummies*. Philadelphia : John Wiley & Sons, 2015. 9781119049784.
- 49. CCSDS REPORT CONCERNING TELEMETRY: SUMMARY OF CONCEPT AND RATIONALE.** Meeting, CCSDS Plenary. Washington : CCSDS Secretariat, 1987. CCSDS 100.0-G-1.
- 50. CARDEN, Frank., Russell P. JEDLICKA a Robert HENRY.** *Telemetry systems engineering*. Artech House telecommunications library : Artech House, 2002. 9781580532570 .
- 51. go.sap.** [Online] <http://go.sap.com/corporate/en/company.html>.
- 52. Mathew, Bince.** *Beginning Sap Fiori*. New York : NY: Springer Science+Business Media, 2015. 978-1-4842-1336-0.
- 53. Coplien, Trygve Reenskaug and James O.** The DCI Architecture: A New Vision of Object-Oriented Programming. *artima*. [Online] 20. 3 2009.
https://www.artima.com/articles/dci_vision.html.
- 54. Dr. Ing. Tomáš Šubrt, RNDr. Helena Brožová, CSc., Ing. Ludmila Dömeová, CSc., RNDr. Petr Kučera.** *Ekonomicko matematické metody II: aplikace a cvičení. Vyd. 2*. Praha : Česká zemědělská univerzita, 2001. ISBN 978-80-213-0721-6.
- 55. Fowler, Martin.** *Destilované UML*. Praha : Grada, 2009. 9788024720623.
- 56. Sam Palmer.** 10 Best Internet of Things (IoT) Cloud Platforms. *DevTeamSpace*. [Online] <https://www.devteam.space/blog/10-best-internet-of-things-iot-cloud-platforms/>.
- 57. Amazon.** AWS IoT Core. *aws.amazon*. [Online] <https://aws.amazon.com/iot-core/>.
- 58. Microsoft.** Windows 10 IoT Core. *developer.microsoft*. [Online]
<https://developer.microsoft.com/cs-cz/windows/iot>.
- 59. Centrum IoT.** *azure.microsoft*. [Online] <https://azure.microsoft.com/cs-cz/services/iot-hub/>.
- 60. IBM.** Watson Internet of Things. *ibm*. [Online] <https://www.ibm.com/internet-of-things/spotlight/watson-iot-platform>.
- 61. Google.** GOOGLE CLOUD IOT. *cloud.google*. [Online]
<https://cloud.google.com/solutions/iot/>.
- 62. ORACLE.** Oracle Internet of Things. *oracle*. [Online]
<https://www.oracle.com/solutions/internet-of-things/>.
- 63. Thunder (Salesforce Thunder).** *searchsalesforce*. [Online] 2 2016.
<http://searchsalesforce.techtarget.com/definition/Thunder-Salesforce-Thunder>.

- 64. Salesforce.** Salesforce IoT. *salesforce*. [Online]
<https://www.salesforce.com/products/salesforce-iot/overview/>.
- 65. BOSCH.** Software Innovations · Bosch IoT Suite. *bosch-si*. [Online]
<https://www.bosch-si.com/iot-platform/bosch-iot-suite/homepage-bosch-iot-suite.html>.
- 66. apps.cisco.** *Cloud and Systems Management*. [Online] 9. 2 2018.
<https://apps.cisco.com/ccw/cpc/guest/concept/268439477>.
- 67. Predix.** The Industrial IoT Platform. *Predix Developer Nerwork*. [Online]
<https://www.predix.io/>.
- 68. SAP.** SAP Leonardo . *SAP*. [Online] <https://www.sap.com/products/leonardo.html>.
- 69. Cloud, SAMSUNG Artik.** Pricing Plans. *SAMSUNG ARTIK*. [Online]
<https://artikel.cloud/pricing/>.
- 70. Make Connections,** Not Silos. *SAMSUNG Artik*. [Online] <https://artikel.cloud/>.
- 71. ThingWorx.** Deliver the Value of Industrial IoT. *ptc*. [Online]
<https://www.ptc.com/en/products/iot>.
- 72. Gitex.** Kaa IoT Development Platform overview. *KAA*. [Online] 2017.
- 73. macchina.io.** PRODUCT. *macchina.io*. [Online] 2018. <https://macchina.io/>.
- 74. FEATURES & PRICING.** *macchina.io*. [Online] 2018.
<https://macchina.io/pricing.html>.
- 75. SiteWhere.** The Open Platform for the Internet of Things. *SiteWhere*. [Online]
<http://www.sitewhere.org/>.
- 76. ThingSpeak.** Learn More About ThingSpeak. *ThingSpeak*. [Online] 2018.
https://thingspeak.com/pages/learn_more.
- 77. License Options.** *ThingSpeak*. [Online] <https://thingspeak.com/prices>.
- 78. Carriots.** What is Carriots. *Carriots*. [Online] 2018. <https://www.carriots.com/what-is-carriots>.
- 79. ubidots.** INTERNET OF THINGS EXPERTISE. *ubidots*. [Online] 2018.
<https://ubidots.com/about>.
- 80. Pricing.** *Ubidots*. [Online] 2018. <https://ubidots.com/pricing>.
- 81. myDevice.** myDevice HOME. *myDevice* . [Online] 207.
<https://mydevices.com/cayenne/marketplace/>.
- 82. Temboo.** Tools for Digital Transformation. *Temboo*. [Online] 2018.
<https://temboo.com/>.
- 83. KERNIGHAN, Brian W. a Rob. PIKE.** *The UNIX programming environment*.
 Englewood Cliffs : N.J.: Prentice-Hall, 1984. 0-13-937699-2.

8 Přílohy

Tabulka s udělováním bodů pro kritérium Web v Bodovací metodě

Příloha 1 bodování pro kritérium Web

Poskyvatel	Responsibilita	Přehlednost	Čeština	Vícejazyčnost	Online podpora	Bez reklam	Suma
Amazon web services IoT platform (AWS)	20	20	0	10	0	20	70
Microsoft Azure IoT hub	20	20	10	10	0	20	80
IBM Watson IoT Platform	20	20	10	10	20	20	100
Google Cloud IoT	20	20	5	10	0	20	75
ORACLE IoT	20	20	10	10	20	20	100
Salesforce IoT Cloud	10	20	0	10	20	20	80
Bosch IoT Cloud	20	20	0	10	0	10	60
Cisco Jasper Control Center	10	20	0	10	0	20	60
GE Predix	20	20	0	0	0	20	60
SAP IOT LEONARDO	20	20	10	10	20	20	100
Samsung ARTIK Cloud	15	20	0	10	0	20	65
ThingWorx	20	20	0	10	0	20	70
Kaa Platform	20	20	0	0	0	20	60
Macchina Platform	20	20	0	0	0	20	60
SiteWhere Platform	20	20	0	0	0	20	60
ThingSpeak Platform	20	20	0	0	0	20	60
Carriots Platform	20	20	0	10	0	20	70
Ubidots	20	20	0	0	20	20	80
MyDevices Cayenne	20	20	0	0	0	20	60
Temboo Platform	10	20	0	0	20	20	70

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Licence v Bodovací metodě

Příloha 2 bodování pro kritérium Licence

Poskyvatel	Trial	Developerská	Komerční	Free	Suma
Amazon web services IoT platform (AWS)	25	0	25	0	50
Microsoft Azure IoT hub	25	0	25	0	50
IBM Watson IoT Platform	25	0	25	0	50
Google Cloud IoT	25	0	25	0	50
ORACLE IoT	25	0	25	0	50
Salesforce IoT Cloud	25	0	25	0	50
Bosch IoT Cloud	25	0	25	0	50
Cisco Jasper Control Center	25	0	25	0	50
GE Predix	0	25	25	0	50
SAP IOT LEONARDO	25	25	25	0	75
Samsung ARTIK Cloud	0	25	25	25	75
ThingWorx	25	0	25	0	50
Kaa Platform	0	25	0	25	50
Macchina Platform	0	25	25	25	75
SiteWhere Platform	0	25	0	25	50
ThingSpeak Platform	0	25	25	25	75
Carriots Platform	0	25	25	25	75
Ubidots	0	25	25	25	75
MyDevices Cayenne	25	0	25	0	50
Temboo Platform	25	0	25	0	50

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Protokoly v Bodovací metodě

Příloha 3 bodování pro kritérium Protokoly

Poskytovatel	AMQP	CoAP	MQTT	LWM2M	IPv4	IPv6	TCP	HTTP(s)	Ostatní	UDP	Suma
Amazon web services IoT platform (AWS)	0	0	10	0	0	0	0	10	10	0	30
Microsoft Azure IoT hub	10	0	10	0	0	0	0	10	0	0	30
IBM Watson IoT Platform	0	0	10	0	0	0	0	0	0	0	10
Google Cloud IoT	0	0	10	0	0	0	0	10	0	0	20
ORACLE IoT	0	0	0	0	0	0	0	0	10	0	10
Salesforce IoT Cloud	0	0	0	0	0	0	0	0	10	0	10
Bosch IoT Cloud	0	0	0	0	10	10	0	0	10	0	30
Cisco Jasper Control Center	0	0	0	0	0	0	0	0	10	0	10
GE Predix	10	0	0	0	0	0	0	10	0	0	20
SAP IOT LEONARDO	10	10	10	0	0	0	10	10	10	10	70
Samsung ARTIK Cloud	0	10	10	10	0	10	0	0	10	0	50
ThingWorx	0	0	10	0	0	0	0	0	0	0	10
Kaa Platform	0	0	0	0	0	0	10	10	0	0	20
Macchina Platform	0	0	10	0	0	0	0	10	10	0	30
SiteWhere Platform	10	0	10	0	0	0	0	0	10	0	30
ThingSpeak Platform	0	0	10	0	0	0	0	10	0	0	20
Carriots Platform	0	0	10	0	0	0	0	10	0	0	20
Ubidots	0	0	10	0	0	0	10	10	0	10	40
MyDevices Cayenne	0	0	10	0	0	0	0	0	0	0	10
Temboo Platform	0	10	10	0	0	0	0	10	0	0	30

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Cena v Bodovací metodě

Příloha 4 bodování pro kritérium Cena

Poskytovatel	Poznámky	Free účet	Jednotky	Kategorie cen	Neomezený přenos	Hodnocení
Amazon web services IoT platform (AWS)	Dle regionu za 1 mil minut	NE	1 mil minut	ANO	NE	75
Microsoft Azure IoT hub	8000 zpráv denně	ANO	vy na 1 jed / za	ANO	NE	75
IBM Watson IoT Platform	100MB free for month	ANO	per MB	ANO	ANO	100
Google Cloud IoT	do 250MB free	ANO	per MB	ANO	ANO	100
ORACLE IoT	nabízí unlimited za 2500\$	NE	OCPU per hour	ANO	ANO	75
Salesforce IoT Cloud	individuální přístup, od 6\$ za měsíc	NE	individuální	?	?	25
Bosch IoT Cloud	neuvádí přesné ceny	NE	startuje na 2,5€	?	?	25
Cisco Jasper Control Center	kategorie, ale nejdou najít ceny	NE	?	ANO	?	25
GE Predix	komplet balíček za rok	NE	služba za rok	ANO	ANO	75
SAP IOT LEONARDO	drahá licence, 2 možnosti	ANO	služba za měsíc	ANO	ANO	100
Samsung ARTIK Cloud	licence na míru dle potřeb	ANO	per 1 Mil msg	ANO	ANO	100
ThingWorx	neuvádí ceny !! , pouze po registraci	?	?	?	?	0
Kaa Platform	platí se jen za služby a podporu	ANO	lužby, podpor	?	?	50
Macchina Platform	individuální přístup, pro komerce	ANO	?	?	?	25
SiteWhere Platform	free řešení ale omezené	ANO	NE	NE	NE	25
ThingSpeak Platform	mají kalkulačku, home a student acc	ANO	device, collect	ANO	NE	75
Carriots Platform	FREE bez kreditky	ANO	per month	ANO	ANO	100
Ubidots	podobně drahý jak SAP, pro education	ANO	devices, dots	ANO	NE	75
MyDevices Cayenne	specializovaná platforma	NE	měsíc	ANO	NE	50
Temboo Platform	neuvádí ceny !! , pouze po registraci	NE	?	?	?	0

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Online školení v Bodovací metodě

Příloha 5 bodování pro kritérium Online školení

Poskytovatel	Poznámky	Školení	Certifikace	Kurz zdarma	Suma
Amazon web services IoT platform (AWS)		25	25	0	50
Microsoft Azure IoT hub		25	25	50	100
IBM Watson IoT Platform		25	25	50	100
Google Cloud IoT	neposkytuje	0	0	0	0
ORACLE IoT		25	25	0	50
Salesforce IoT Cloud		25	25	50	100
Bosch IoT Cloud		25	25	0	50
Cisco Jasper Control Center		25	25	0	50
GE Predix		25	25	50	100
SAP IOT LEONARDO		25	25	50	100
Samsung ARTIK Cloud	neposkytuje	0	0	0	0
ThingWorx		25	25	0	50
Kaa Platform	neposkytuje	0	0	0	0
Macchina Platform	neposkytuje	0	0	0	0
SiteWhere Platform		25	25	0	50
ThingSpeak Platform	neposkytuje	0	0	0	0
Carriots Platform		25	0	0	25
Ubidots		25	0	0	25
MyDevices Cayenne	neposkytuje	0	0	0	0
Temboo Platform	neposkytuje	0	0	0	0

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Vlastní sada aplikací v Bodovací metodě

Příloha 6 bodování pro kritérium Vlastní sada aplikací

Poskytovatel	Poznámky	Vlastní aplikace	Suma
Amazon web services IoT platform (AWS)		100	100
Microsoft Azure IoT hub		100	100
IBM Watson IoT Platform		100	100
Google Cloud IoT		100	100
ORACLE IoT		100	100
Salesforce IoT Cloud		100	100
Bosch IoT Cloud	pouze poskytovatel	0	0
Cisco Jasper Control Center	centrum pro management	100	100
GE Predix		100	100
SAP IOT LEONARDO		100	100
Samsung ARTIK Cloud		100	100
ThingWorx		100	100
Kaa Platform	pouze poskytovatel	0	0
Macchina Platform		100	100
SiteWhere Platform	pouze poskytovatel	0	0
ThingSpeak Platform		100	100
Carriots Platform	pouze poskytovatel	0	0
Ubidots	pouze poskytovatel	0	0
MyDevices Cayenne		100	100
Temboo Platform	pouze poskytovatel	0	0

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Demo v Bodovací metodě

Příloha 7 bodování pro kritérium Demo

Poskytovatel	Poznámky	Existence Dema	Suma
Amazon web services IoT platform (AWS)	nemá	0	0
Microsoft Azure IoT hub		100	100
IBM Watson IoT Platform		100	100
Google Cloud IoT		100	100
ORACLE IoT		100	100
Salesforce IoT Cloud		100	100
Bosch IoT Cloud	nemá	0	0
Cisco Jasper Control Center		100	100
GE Predix		100	100
SAP IOT LEONARDO		100	100
Samsung ARTIK Cloud		100	100
ThingWorx		100	100
Kaa Platform		100	100
Macchina Platform		100	100
SiteWhere Platform	nemá	0	0
ThingSpeak Platform	nemá	0	0
Carriots Platform		100	100
Ubidots		100	100
MyDevices Cayenne	nemá	0	0
Temboo Platform		100	100

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Git v Bodovací metodě

Příloha 8 bodování pro kritérium Git

Poskytovatel	Poznámky	Git- napojeni	Git - Demo rep.	Suma
Amazon web services IoT platform (AWS)		50	0	50
Microsoft Azure IoT hub		50	50	100
IBM Watson IoT Platform		50	50	100
Google Cloud IoT		50	50	100
ORACLE IoT		0	50	50
Salesforce IoT Cloud		0	50	50
Bosch IoT Cloud		50	50	100
Cisco Jasper Control Center	neexistuje přímo od Ciso	0	0	0
GE Predix		50	50	100
SAP IOT LEONARDO		50	50	100
Samsung ARTIK Cloud		50	50	100
ThingWorx	neexistuje přímo	0	0	0
Kaa Platform		0	50	50
Macchina Platform		0	50	50
SiteWhere Platform		0	50	50
ThingSpeak Platform		50	50	100
Carriots Platform		0	50	50
Ubidots		0	50	50
MyDevices Cayenne		0	50	50
Temboo Platform		0	50	50

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium API dokumentace v Bodovací metodě

Příloha 9 bodování pro kritérium API dokumentace

Poskytovatel	Dokumentace	Suma
Amazon web services IoT platform (AWS)	100	100
Microsoft Azure IoT hub	100	100
IBM Watson IoT Platform	100	100
Google Cloud IoT	100	100
ORACLE IoT	100	100
Salesforce IoT Cloud	100	100
Bosch IoT Cloud	100	100
Cisco Jasper Control Center	100	100
GE Predix	100	100
SAP IOT LEONARDO	100	100
Samsung ARTIK Cloud	100	100
ThingWorx	100	100
Kaa Platform	100	100
Macchina Platform	100	100
SiteWhere Platform	100	100
ThingSpeak Platform	100	100
Carriots Platform	100	100
Ubidots	100	100
MyDevices Cayenne	100	100
Temboo Platform	100	100

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Vlastní knihovny v Bodovací metodě

Příloha 10 bodování pro kritérium Vlastní knihovny

Poskytovatel	Poznámky	Existence knihoven	Suma
Amazon web services IoT platform (AWS)		100	100
Microsoft Azure IoT hub		100	100
IBM Watson IoT Platform		100	100
Google Cloud IoT		100	100
ORACLE IoT		100	100
Salesforce IoT Cloud		100	100
Bosch IoT Cloud		100	100
Cisco Jasper Control Center	nemá volně zobrazitelné	0	0
GE Predix		100	100
SAP IOT LEONARDO		100	100
Samsung ARTIK Cloud		100	100
ThingWorx		100	100
Kaa Platform		100	100
Macchina Platform		100	100
SiteWhere Platform		100	100
ThingSpeak Platform		100	100
Carriots Platform		100	100
Ubidots		100	100
MyDevices Cayenne		100	100
Temboo Platform		100	100

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Programovací jazyky v Bodovací metodě

Příloha 11 bodování pro kritérium Programovací jazyky

Poskytovatel	C	C++	C#	Python	Node.js	Java	Android	PHP	JavaScript	Ostatní	Suma
Amazon web services IoT platform (AWS)	0	0	0	10	10	10	0	10	0	10	50
Microsoft Azure IoT hub	10	0	0	10	10	10	0	0	0	10	50
IBM Watson IoT Platform	10	10	10	10	10	10	0	0	0	0	60
Google Cloud IoT	0	0	0	10	10	10	0	10	0	10	50
ORACLE IoT	0	0	0	10	10	10	0	0	10	0	40
Salesforce IoT Cloud	10	10	10	10	10	10	10	10	10	10	100
Bosch IoT Cloud	0	0	0	0	0	10	0	0	0	0	10
Cisco Jasper Control Center	10	10	10	10	10	10	10	10	10	10	100
GE Predix	0	0	0	10	10	10	0	10	0	10	50
SAP IOT LEONARDO	0	0	0	10	10	0	0	0	10	10	40
Samsung ARTIK Cloud	10	10	10	10	0	10	10	10	10	10	90
ThingWorx	10	0	0	0	0	10	10	0	0	10	40
Kaa Platform	10	10	10	0	0	10	0	0	0	0	40
Macchina Platform	10	10	0	0	0	0	0	0	0	10	30
SiteWhere Platform	0	0	0	10	0	10	0	0	10	10	40
ThingSpeak Platform	0	0	10	0	0	0	0	0	10	0	20
Carriots Platform	0	0	0	10	0	10	0	0	0	10	30
Ubidots	10	0	10	10	10	10	10	10	0	10	80
MyDevices Cayenne	10	10	0	10	0	0	0	0	0	10	40
Temboo Platform	10	10	0	0	10	10	10	10	10	10	80

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Průmyslové odvětví v Bodovací metodě

Příloha 12 bodování pro kritérium Průmyslové odvětví

Poskytovatel	Poznámky	Normální použití	Specializace	Suma
Amazon web services IoT platform (AWS)		50	50	100
Microsoft Azure IoT hub		50	50	100
IBM Watson IoT Platform		50	50	100
Google Cloud IoT	Manufacturing	50	50	100
ORACLE IoT	Industrial	50	50	100
Salesforce IoT Cloud		50	0	50
Bosch IoT Cloud	Agriculture, Healthcare, manufacturing ...	50	50	100
Cisco Jasper Control Center	Agriculture, Healthcare, manufacturing ...	50	50	100
GE Predix	Industrial	0	50	50
SAP IOT LEONARDO	Manufacturing, Company	50	50	100
Samsung ARTIK Cloud	Manufacturing, Company	50	50	100
ThingWorx	Enterprise, Industrial	0	50	50
Kaa Platform		50	0	50
Macchina Platform	Industrial	50	50	100
SiteWhere Platform		50	0	50
ThingSpeak Platform		50	0	50
Carriots Platform	Smart things.	50	50	100
Ubidots	Agriculture, Healthcare, manufacturing...	50	50	100
MyDevices Cayenne	Restaurant and foo	0	50	50
Temboo Platform	Manufacturing, Company	50	50	100

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Hardware v Bodovací metodě

Příloha 13 bodování pro kritérium Hardware

Poskytovatel	Poznámky	Lora Ready	Raspberry Pi	BeagleBone	Arduíno	Ostatní	Neomezeno	Suma
Amazon web services IoT platform (AWS)	neomezeny	10	10	10	10	10	50	100
Microsoft Azure IoT hub	široká škála	0	10	10	10	10	0	40
IBM Watson IoT Platform	neomezeny	10	10	10	10	10	50	100
Google Cloud IoT	neomezeny	10	10	10	10	10	50	100
ORACLE IoT	neudává	0	0	0	0	10	0	10
Salesforce IoT Cloud	neomezeny	10	10	10	10	10	50	100
Bosch IoT Cloud	neudává	0	0	0	0	10	0	10
Cisco Jasper Control Center	neomezeny	10	10	10	10	10	50	100
GE Predix	neomezeny	10	10	10	10	10	50	100
SAP IOT LEONARDO	neomezeny	10	10	10	10	10	50	100
Samsung ARTIK Cloud	neomezeny	10	10	10	10	10	50	100
ThingWorx	neudává	0	0	0	0	10	0	10
Kaa Platform		0	10	10	0	10	0	30
Macchina Platform	neomezeno	10	10	10	10	10	50	100
SiteWhere Platform	neomezeno	10	10	10	10	10	50	100
ThingSpeak Platform		0	10	10	10	10	0	40
Carriots Platform		0	10	10	10	10	0	40
Ubidots		10	10	10	10	10	0	50
MyDevices Cayenne		10	0	0	10	10	0	30
Temboo Platform	neomezeny	10	10	10	10	10	50	100

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Konektivita v Bodovací metodě

Příloha 14 bodování pro kritérium Konektivita

Poskytovatel	Wi-Fi	Cellular (mobilní)	Sigfo10	LoRaWAN	LTE-M	NB-IOT	Ethernet	Bluetooth	Sigfox	Ostatní	Suma
Amazon web services IoT platform (AWS)	0	0	0	0	0	0	10	0	0	10	20
Microsoft Azure IoT hub	0	0	0	0	0	0	10	0	0	10	20
IBM Watson IoT Platform	0	0	0	0	0	0	10	0	0	10	20
Google Cloud IoT	0	0	0	0	0	0	10	0	0	10	20
ORACLE IoT	0	0	0	0	0	0	10	10	0	10	30
Salesforce IoT Cloud	0	0	0	0	0	0	0	0	0	10	10
Bosch IoT Cloud	0	0	0	0	0	0	10	0	0	10	20
Cisco Jasper Control Center	0	10	0	0	10	10	0	0	0	10	40
GE Predix	0	10	0	0	0	0	10	0	0	10	30
SAP IOT LEONARDO	0	10	0	0	0	0	10	0	0	10	30
Samsung ARTIK Cloud	0	0	0	0	0	0	10	10	0	10	30
ThingWorx	0	0	0	0	0	0	10	0	0	10	20
Kaa Platform	0	10	0	0	0	0	10	0	0	10	30
Macchina Platform	0	0	0	10	0	0	10	0	10	10	40
SiteWhere Platform	0	0	0	0	0	0	10	0	0	10	20
ThingSpeak Platform	0	0	0	0	0	0	0	0	0	10	10
Carriots Platform	0	0	0	0	0	0	10	0	0	10	20
Ubidots	10	10	10	10	10	10	0	0	0	10	70
MyDevices Cayenne	10	0	0	0	0	0	10	0	0	10	30
Temboo Platform	0	10	0	0	0	0	10	10	0	0	30

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Databáze v Bodovací metodě

Příloha 15 bodování pro kritérium Databáze

Poskytovatel	Poznámky	SQLite	Redis	MongoDB	pache	HBase	InfluxDB	stigeSC	MATLAB	Cassandra	zon	Dynam	HANA	Data	zure	storage	Oracle	idant	NoSQL	Rigtable	Počet DB	SUMA	
Amazon web services IoT platform (AWS)		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	100	
Microsoft Azure IoT hub		0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	100	
IBM Watson IoT Platform		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	100	
Google Cloud IoT		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	100	
ORACLE IoT		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	100	
Salesforce IoT Cloud	neuvádí typ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bosch IoT Cloud		0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	100	
Cisco Jasper Control Center	neuvádí typ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GE Predix		0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	100	
SAP IOT LEONARDO		0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	100	
Samsung ARTIK Cloud	neuvádí typ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ThingWorx	neuvádí typ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kaa Platform		1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	100
Macchina Platform		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	100	
SiteWhere Platform		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	100	
ThingSpeak Platform		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	100	
Carriots Platform	neuvádí typ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubidots	neuvádí typ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MyDevices Cayenne	neuvádí typ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Temboo Platform	neuvádí typ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Zdroj: vlastní práce autora

Tabulka s udělováním bodů pro kritérium Vlastní hodnocení v Bodovací metodě

Příloha 16 bodování pro kritérium Vlastní hodnocení

Poskytovatel	Poznámky	Subjektivní názor	Suma
Amazon web services IoT platform (AWS)	drahé, komplexní, nepřehledná api	60	60
Microsoft Azure IoT hub	W10 IoT core, dobrá podpora, příjemná dema	80	80
IBM Watson IoT Platform	málo přehledné stránky, nepřehledná API	70	70
Google Cloud IoT	pěkná api a podpora	80	80
ORACLE IoT	nabídka trialu, aplikací a dokumentace	70	70
Salesforce IoT Cloud	oslovili mě stránky, zajímavý koncept	75	75
Bosch IoT Cloud	nepřehledná api, málo informací	60	60
Cisco Jasper Control Center	špatně dostupné informace, zajímavá platforma	65	65
GE Predix	specializace, málo informací	50	50
SAP IOT LEONARDO	školení, velká podpora, vlastní zkušenost	90	90
Samsung ARTIK Cloud	pěkné řešení a podpora	80	80
ThingWorx	hodně omezená platforma	50	50
Kaa Platform	příjemná platforma, málo aplikací	65	65
Macchina Platform	hodně možností v open licenci	70	70
SiteWhere Platform	nepřehledný web a dokumentace	65	65
ThingSpeak Platform	licence pro studenty a home použití	80	80
Carriots Platform	menší podpora	75	75
Ubidots	velmi pěkná platforma, ale omezený trial účet	65	65
MyDevices Cayenne	hodně omezená platforma	50	50
Temboo Platform	podpora hodně jazyků, chybí mi lepší management cloudu, 14 dní trial, bez kreditky	75	75

Zdroj: vlastní práce autora

Tabulka preferencí pro určení vah kritérií v Saatyho metodě, využitá v této práci

Příloha 17 tabulka preferencí pro Saatyho metodu

Preference	Dominance	Submisivita
Rovnocennost	1	1,0000
Slabá preference	3	0,3333
Silná preference	5	0,2000
Velmi silná preference	7	0,1429
Absolutní preference	9	0,1111

Zdroj: vlastní práce autora

Výsledná tabulka vah kritérií pro Bodovací metodu získanou Saatyho metodou

Příloha 18 výsledná tabulka vah kritérií

Pořadí	Kritérium	Váha
1	D) Cena	0,180421
2	B) Licence	0,147655
3	H) Propojení na Github	0,108668
4	P) Databáze	0,10638
5	N) Konektivita	0,075792
6	M) Hardware podpora	0,06966
7	C) Protokoly	0,058828
8	J) Vlastní knihovny	0,057941
9	I) API dokumentace	0,043526
10	F) Vlastní sada aplikací	0,035387
11	O) Vlastní hodnocení	0,033418
12	K) Jazyky	0,028814
13	E) Online školení	0,020514
14	G) Demo	0,013525
15	A) Web	0,01083
16	L) Průmyslové odvětví	0,00864

Zdroj: vlastní práce autora

Vstupní data pro Saatyho metodu, rozlišené dominantní a submisivní hodnoty, tabulka v lepším rozlišení je přiložená v Excel souboru na přiloženém CD disku

Priloha 19 Tabulka vstupnich dat pro Saatyho metodu

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Geometricky prumer	Vahy	Poradi
A) Web	1	0,2000	0,1429	0,1429	0,3333	0,3333	0,2000	0,1429	0,1111	0,2000	0,2000	0,3333	0,1429	0,1429	3	0,1429	0,244943219	0,010830281	15
B) Licence	5	1	3	1,0000	3	5	7	5	5	3	3	3	5	5	3	3	3,339434488	0,147654688	2
C) Protokoly	7	0,3333	1	0,1429	5	5	3	5	3	3	3	9	0,1429	0,3333	0,3333	0,2000	1,33049381	0,058828418	7
D) Cena	7	1	7	1	5	7	9	3	3	5	7	9	3	3	5	3	4,080487874	0,180420716	1
E) Online skoleni	3	0,3333	0,2000	0,2000	1	0,1429	3	0,2000	0,2000	0,2000	0,2000	7	0,2000	0,2000	3	0,2000	0,463948238	0,020513692	13
F) Vlastni sada aplikaci	3	0,2000	0,2000	0,1429	7	1	5	0,2000	3	3	5	9	0,1429	0,1429	0,2000	0,1429	0,800337671	0,035387312	10
G) Demo	5	0,1429	0,3333	0,1111	0,3333	0,2000	1	0,2000	0,1429	0,1429	0,2000	7	0,1429	0,1429	0,2000	0,1429	0,305893095	0,013525209	14
H) Propojeni na Github	7	0,2000	0,2000	0,3333	5	5	5	1	3	3	5	9	5	3	5	5	2,4576889803	0,108667926	3
I) API dokumentace	9	0,2000	0,3333	0,3333	5	0,3333	7	0,3333	1	0,3333	3	9	0,3333	0,3333	3	0,3333	0,984415562	0,043526403	9
J) Vlastni knihovny	5	0,3333	0,3333	0,2000	5	0,3333	7	0,3333	3	1	0,3333	7	5	5	0,2000	5	1,31042997	0,057941286	8
K) Jazyky	5	0,3333	0,3333	0,1429	5	0,2000	5	0,2000	0,3333	3	1	5	0,2000	0,2000	0,3333	0,2000	0,651681679	0,028814416	12
L) Prirucny skve odvetvi	3	0,3333	0,1111	0,1111	0,1429	0,1111	5	0,1111	0,1111	0,1429	0,2000	1	0,1111	0,1111	0,3333	0,1111	0,195393667	0,008639557	16
M) Hardware podpora	7	0,2000	7	0,3333	5	7	7	0,2000	3	0,2000	5	9	1	0,3333	7	0,1429	1,575465739	0,069659969	6
N) Konektivita	7	0,2000	3	0,3333	5	7	7	0,2000	3	0,2000	5	9	3	1	7	0,1429	1,714156899	0,075792264	5
O) Vlastni hodnoceni	0,3333	0,3333	3	0,2000	0,3333	5	5	0,3333	0,3333	5	3	3	0,1429	0,2000	1	0,1429	0,755802307	0,0334418159	11
P) Databaze	7	0,3333	5	0,3333	5	7	7	0,2000	3	0,2000	5	9	5	7	7	1	2,405938193	0,106379704	4
																	SUMA	22,616151522	1

Zdroj: vlastni prace autora

Vstupní data pro Bodovací metodu s určením pořadí, tabulka je přiložená v lepším formátu na disku CD

Příloha 20 Výsledná tabulka pro Bodovací metodu

Bodovací metoda	Web	Licence	Protokoly	Cena	Online školení	Vlastní sada aplikací	Demo na Github	Propojení	API dokumentace	Vlastní knihovny	Jazyky	Plnýsústo ve odvětví	Hardware podpora	Konektivita	Databáze	Vlastní hodnocení	SUMA	Pořadí	
Poskytovatel																			
Amazon web services IoT platform (AWS)	70	50	30	75	50	100	0	50	100	100	50	100	100	20	100	60	1055	9	
Microsoft Azure IoT hub	80	50	30	75	100	100	100	100	100	100	50	100	40	20	100	80	1225	3	
IBM Watson IoT Platform	100	50	10	100	100	100	100	100	100	100	60	100	100	20	100	70	1310	2	
Google Cloud IoT	75	50	20	100	0	100	100	100	100	100	50	100	100	20	100	80	1195	4	
ORACLE IoT	100	50	10	75	50	100	100	50	100	100	40	100	10	30	100	70	1085	7	
Salesforce IoT Cloud	80	50	10	25	100	100	100	50	100	100	100	100	10	10	0	75	1050	10	
Bosch IoT Cloud	60	50	30	25	50	0	0	100	100	100	10	100	10	20	100	60	815	18	
Cisco Jasper Control Center	60	50	10	25	50	100	100	0	100	0	100	100	100	40	0	65	900	14	
GE Predix	60	50	20	75	100	100	100	100	100	100	50	50	100	30	100	50	1185	6	
SAP IOT LEONARDO	100	75	70	100	100	100	100	100	100	100	40	100	100	30	100	90	1405	1	
Samsung ARTIK Cloud	65	75	50	100	0	100	100	100	100	100	90	100	100	30	0	80	1190	5	
ThingWorx	70	50	10	0	50	100	100	0	100	100	40	50	10	20	0	50	750	19	
Kaa Platform	60	50	20	50	0	100	100	50	100	100	40	50	30	30	100	65	845	16	
Machina Platform	60	75	30	25	0	100	100	50	100	100	30	100	100	40	100	70	1080	8	
SiteWhere Platform	60	50	30	25	50	0	0	50	100	100	40	50	100	20	100	65	840	17	
ThingSpeak Platform	60	75	20	75	0	100	0	100	100	100	20	50	40	10	100	80	930	12	
Carriots Platform	70	75	20	100	25	0	100	50	100	100	30	100	40	20	0	75	905	13	
Ubidots	80	50	40	75	25	0	100	50	100	100	80	100	50	30	0	65	1010	11	
MyDevices Cayenne	60	50	10	50	0	100	0	50	100	100	40	50	30	30	0	50	720	20	
Temboo Platform	70	50	30	0	0	0	100	50	100	100	80	100	100	30	0	75	885	15	
MAX	100	75	70	100	100	100	100	100	100	100	100	100	100	70	100	90	1405	20	
MIN	60	50	10	0	0	0	0	0	0	0	10	50	10	10	0	50	720	1	

Zdroj: vlastní práce autora

Výsledná tabulka pěti nejlepších poskytovatelů získaná Bodovací metodou, zeleně je označen nejlepší poskytovatel

Příloha 21 tabulka pět nejlepších poskytovatelů Bodovací metodou

Prvních 5	Poskytovatel	Body
1	SAP IOT LEONARDO	1405
2	IBM Watson IoT Platform	1310
3	Microsoft Azure IoT hub	1225
4	Google Cloud IoT	1195
5	Samsung ARTIK Cloud	1190

Zdroj: vlastní práce autora

Výsledná tabulka pěti nejhorsích poskytovatelů získaná Bodovací metodou, červeně je označen nejhorsí poskytovatel

Příloha 22 tabulka pět nejhorsích poskytovatelů Bodovací metodou

Njhorsích 5	Poskytovatel	Body
20	MyDevices Cayenne	720
19	ThingWorx	750
18	Bosch IoT Cloud	815
17	SiteWhere Platform	840
16	Kaa Platform	845

Zdroj: vlastní práce autora

Výsledná tabulka pěti nejlepších poskytovatelů vyhodnocená Bodovací metodou odhodnocenou váhami ze Saatyho metody

Příloha 23 tabulka pět nejlepších poskytovatelů Bodovací metodou s váhami

5 Nejlepších	Poskytovatel	Body
1	SAP IOT LEONARDO	87,17527521
2	IBM Watson IoT Platform	79,10420544
3	Google Cloud IoT	77,41640078
4	GE Predix	74,11819792
5	Samsung ARTIK Cloud	74,03684658

Zdroj: vlastní práce autora

Výsledná tabulka pěti nejhorších poskytovatelů vyhodnocená Bodovací metodou odhodnocenou váhami ze Saatyho metody

Příloha 24 tabulka pět nejlepších poskytovatelů Bodovací metodou s váhami

5 Nejhorších	Poskytovatel	Body
20	ThingWorx	30,26075124
19	Cisco Jasper Control Center	39,31619542
18	Temboo Platform	41,7536284
17	MyDevices Cayenne	44,37979704
16	Salesforce IoT Cloud	49,41444636

Zdroj: vlastní práce autora

Skript pro nastavení modemu v programu wvdial na systému Linux.

Příloha 26 Skript pro nastavení modemu přes wvdial

```
[Dialer Defaults]
Modem = /dev/gsmmodem
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0
Init3 = AT+CGDCONT=1,"IP","internet"
Stupid Mode = 1
ISDN = 0
Dial Command = ATD
Modem Type = Analog Modem
New PPPD = yes
Phone = *99***1#
Stupid mode = yes
Username = "blank"
Password = "blank"
```

Zdroj: vlastní práce autora

Nastavení typu zprávy v prostředí IoT modulu SAP Cloud Platform Leonardo, využitě zprávy jsou CMD_TO_DRONE, DATA_FROM_DRONE a PHOTO_FROM_DRONE

Příloha 27 nastavení Message Types

Position	Name	Type	Optional Settings
1	timestamp	date	
2	photoBase64	string	Max. Length 1000000
3	deviceId	integer	

Zdroj: vlastní práce autora

JSON struktura pro přenos telemetrie pro zprávu DATA_FROM_DRONE

Příloha 28 IoT zpráva DATA_FROM_DRONE

```
{
  'mode': 'sync',
  'messageType': self.getFromMetadata('msgTypeTelemetry'),
  'messages': [{
    'log': datetime.datetime.now().isoformat(),
    'deviceID': self.getFromMetadata('modelID'),
    'roll': '0',
    'pitch': '0',
    'throttle': '0',
    'yaw': '0',
    'aux1': '0',
    'aux2': '0',
    'aux3': '0',
    'aux4': '0',
    'fix': 'false',
    'numsat': '0',
    'longitude': '0',
    'latitude': '0',
    'speed': '0',
    'altitude': '0',
    'distanceHome': '0',
    'headingHome': '0',
    'amperage': '0',
    'rssi': '0',
    'vbat': '0',
    'powermetersum': '0',
    'motor1': '0',
    'motor2': '0',
    'motor3': '0',
    'motor4': '0',
    'motor5': '0',
    'motor6': '0',
    'motor7': '0',
    'motor8': '0',
    'version': '0',
    'type': '0',
    'angx': '0',
    'angy': '0',
    'heading': '0',
  ]
}
```

Zdroj: vlastní práce autora

JSON struktura pro odesílání příkazů z aplikace do dronu zpráva CMD_TO_DRONE

Příloha 29 IoT zpráva CMD_TO_DRONE

```
{
  "timestamp": new Date(),
  "deviceID": 123,
  "commandID": 1
}
```

Zdroj: vlastní práce autora

JSON struktura pro odesílání fotografií z dronu zpráva PHOTO_FROM_DRONE

Příloha 30 IoT zpráva PHOTO_FROM_DRONE

```
{
  'mode': 'sync',
  'messageType': self.getFromMetadata('msgTypePhoto'),
  'messages':[{
    'timestamp': datetime.datetime.now().isoformat(),
    'photoBase64': "",
    'deviceID': self.getFromMetadata('modelID'),
  }]
}
```

Zdroj: vlastní práce autora

Metoda pro obsluhu oData služby s příkazem CREATE pro založení záznamu do databáze

Příloha 31 funkce na uložení příkazu odeslaného do dronu

```
function createNewRequest(param) {
  $.trace.error("ERROR:: " + param);
  var after = param.afterTableName;
  var pStmt = param.connection.prepareStatement("select * from \"" + after +
  "\"");
  var Data = SESSIONINFO.recordSetToJSON(pStmt.executeQuery(),
  Details");
  pStmt.close();
  var G_DEVICE = Data.Details[0].G_DEVICE;
  var ALARM = Data.Details[0].ALARM;
  var FOTO = Data.Details[0].FOTO;
  var LED = Data.Details[0].LED;
  var GPS_HOLD = Data.Details[0].GPS_HOLD;
  var GPS_HOME = Data.Details[0].GPS_HOME;
  var CREATE = new Date(Data.Details[0].CREATE);
  var STATE = Data.Details[0].STATE;
  var connection = $.hdb.getConnection();
  connection.executeUpdate(
  'INSERT INTO "DRONMASTER"."DRONCOMMAND"
  G_DEVICE,ALARM,FOTO,LED,GPS_HOLD,GPS_HOME,CREATE,STATE ) VALUES
  (?, ?, ?, ?, ?, ?, ?, ?)', G_DEVICE, ALARM,
  FOTO, LED, GPS_HOLD, GPS_HOME, CREATE, STATE);
  connection.commit();
}
```

Zdroj: vlastní práce autora

Metoda pro obsluhu oData služby s příkazem DELETE pro smazání záznamu z databáze

Příloha 32 funkce na mazání ze seznamu dronů

```
function deleteDroneList(param) {
  $.trace.error("ERROR:: " + param);
  var after = param.beforeTableName;
  var pStmt = param.connection.prepareStatement("select * from \"" +
  after + "\"");
  var Data = SESSIONINFO.recordSetToJSON(pStmt.executeQuery(),
  "Details");
  pStmt.close();
  var id = Data.Details[0].DRONID;
  var connection = $.hdb.getConnection();
  connection.executeUpdate('DELETE FROM "DRONMASTER"."DRONLIST" WHERE
  DRONID = ?', id);
  connection.commit();
  $.trace.debug("Entered create test function...");
}
```

Zdroj: vlastní práce autora

Metoda pro obsluhu oData služby s příkazem UPDATE pro aktualizaci záznamu v databázi

Příloha 33 funkce na aktualizaci záznamu o dronu

```
function updateDroneData(param) {
    $.trace.error("ERROR:: " + param);
    var after = param.afterTableName;
    var pStmt = param.connection.prepareStatement("select * from \"" +
        after + "\"");
    var Data = SESSIONINFO.recordSetToJSON(pStmt.executeQuery(),
        "Details");
    pStmt.close();
    var id = Data.Details[0].DRONID;
    var OWNER = Data.Details[0].OWNER;
    var DRONENAME = Data.Details[0].DRONENAME;
    var COUNTFLY = Data.Details[0].COUNTFLY;
    var COUNTFLYFAR = Data.Details[0].COUNTFLYFAR;
    var FRAME = Data.Details[0].FRAME;
    var LASTFLYFAR = Data.Details[0].LASTFLYFAR;
    var LASTFLYTIME = Data.Details[0].LASTFLYTIME;
    var LASTUSE = Data.Details[0].LASTUSE;
    var connection = $.hdb.getConnection();
    connection.executeUpdate('UPDATE "DRONMASTER"."DRONLIST" SET OWNER=?,
        FRAME=? , DRONENAME=? , COUNTFLY=? , COUNTFLYFAR=? , LASTFLYFAR=?,
        LASTFLYTIME=?, LASTUSE=? WHERE DRONID =?', OWNER, FRAME, DRONENAME,
        COUNTFLY, COUNTFLYFAR, LASTFLYFAR, LASTFLYTIME, LASTUSE, id);
    connection.commit();
    $.trace.debug("Entered create test function...");
}
}
```

Zdroj: vlastní práce autora

Příklad xsodata služby pro Fiori aplikaci pro základní CRUD operace

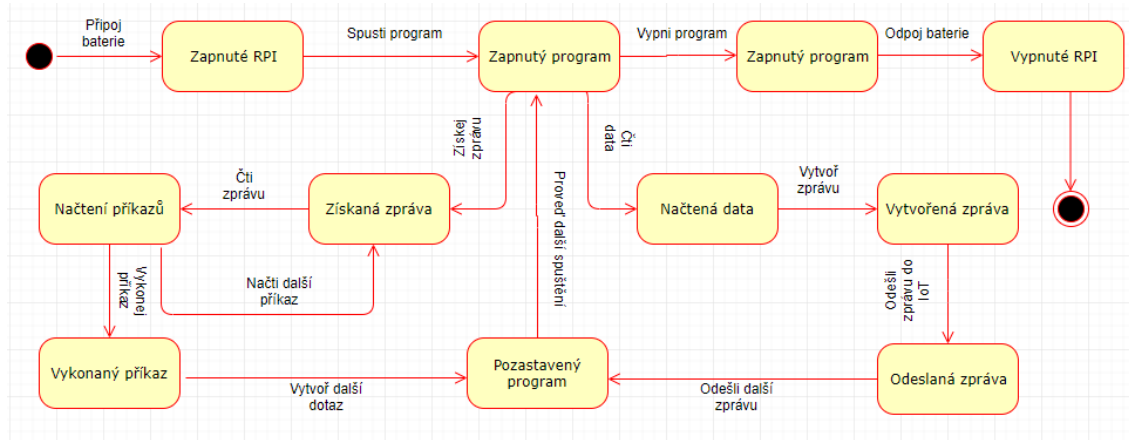
Příloha 34 iotscenario.xsodata služba pro Fiori aplikaci

```
service {
    "DRONMASTER"."T_IOT_BB68C2DC4F0D33C616BC" as "Photo_Data"
        create forbidden
        update forbidden
        delete using
            "droneiot.util.DELETE:deleteMethods.xsjslib::deletePhotoData";
    "DRONMASTER"."DRONCOMMAND" as "Req_Data"
        create using
            "droneiot.util.CREATE:createMethods.xsjslib::createNewRequest"
        update forbidden
        delete using
            "droneiot.util.DELETE:deleteMethods.xsjslib::deleteDronCommand";
    "DRONMASTER"."DRONLIST" as "Main_Data"
        create using
            "droneiot.util.CREATE:createMethods.xsjslib::createDroneData"
        update using
            "droneiot.util.UPDATE:updateMethod.xsjslib::updateDroneData"
        delete using
            "droneiot.util.DELETE:deleteMethods.xsjslib::deleteDroneList";
    "DRONMASTER"."UNS_DEVICES" as "UNASSIGNED_DEVICES" keys generate local
        "ID"
        create forbidden
        update forbidden
        delete forbidden;
    "DRONMASTER"."T_IOT_408EB93C73C389B430CB" as "MSP_Data"
        create forbidden
        update forbidden
        delete using
            "droneiot.util.DELETE:deleteMethods.xsjslib::deleteLogData";
}
}
```

Zdroj: vlastní práce autora

Stavový diagram vytvoření metodikou UML pro zobrazení propojení dronu s IoT službou

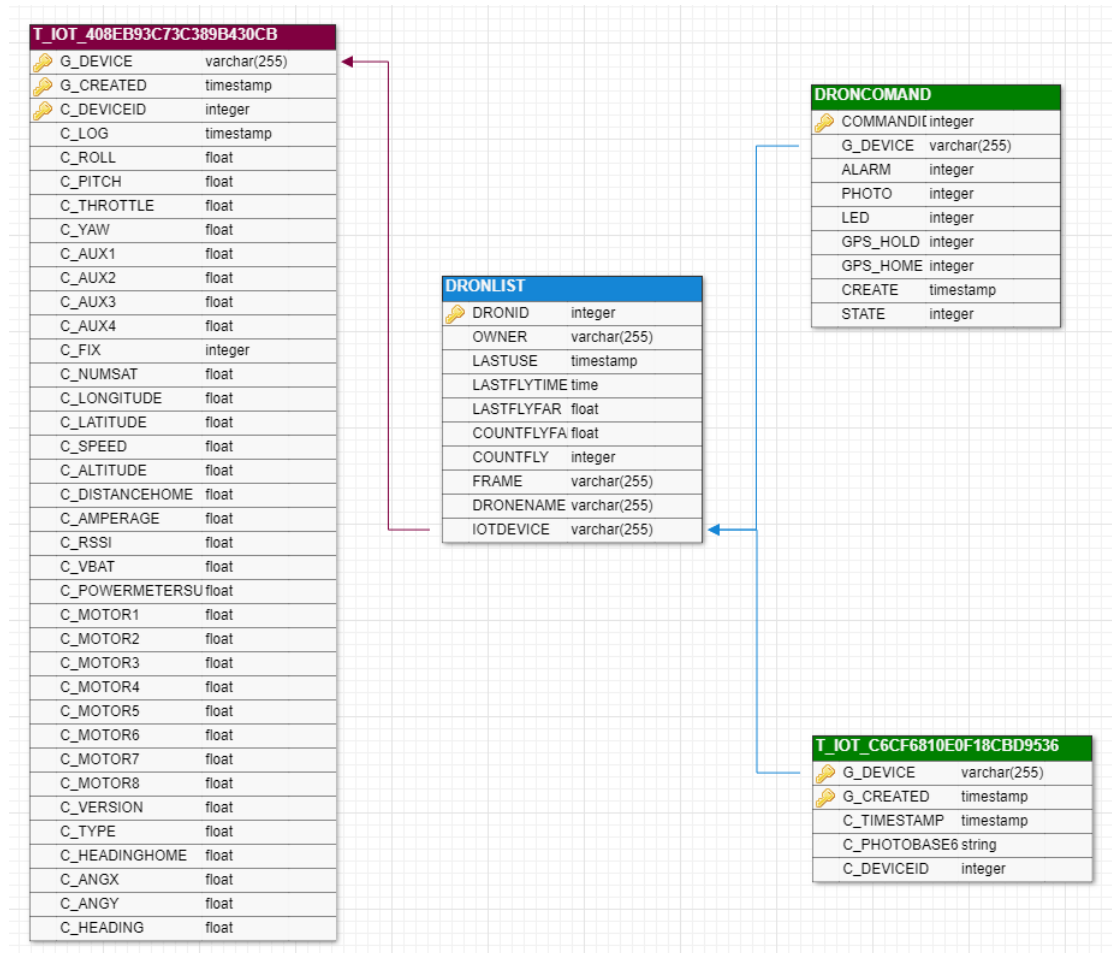
Príloha 35 Stavový diagram propojení dronu s IoT



Zdroj: vlastní práce autora

Schéma databáze DRONMASTER využité pro přenos telemetrie přes IoT

Príloha 36 schéma databáze pro IoT



Zdroj: vlastní práce autora

Třída Config.py pro nastavení odesílání a přijímání telemetrie v dronu vytvořená autorem této práce

Příloha 37 Třída Config.py

```
#!/usr/bin/env python
#Configurace zprav a zarizeni
metadata = {
    'deviceID': '46b877a1-1639-4e79-b86d-1eb8a5e92421',
    'modelID': 1,
    'baudRate': 115200,
    'timeSleep': 5,
    'serailPort': '/dev/ttyACM0',
    'msgTypePhoto': 'c6cf6810e0f18cbd9536',
    'msgTypeTelemetry': '408eb93c73c389b430cb',
    'deviceToken': 'e52f9c4dce05355799246f915da50', #HEXA TOKEN
    'url' : '/com.sap.iotservices.mms/v1/api/http/data/',
    'host': 'iotmms0015766703trial.hanatrial.ondemand.com'
}
import datetime
"""Get data script"""
class Config (object):
    __VERSION__ = "0.0.1"
    __AUTHOR__ = "Filip Maly XMALF002"
    __FACULTY__ = "PEF CZU INFONK"
# OBSLUZNE METODY PRO ODESILANI A PRIJIMANI ZPRAV
    def getDeviceId(self):
        id = metadata['deviceID']
        return id

    def getSerialPort(self):
        return metadata['serailPort']

    def getBaudRate(self):
        return metadata['baudRate']

    def getFromMetadata(self , attribute):
        return metadata[attribute]

    def getBodyPhoto(self):
        bodyPhoto = {
            'mode': 'sync',
            'messageType': self.getFromMetadata('msgTypePhoto'),
            'messages': [{
                'timestamp': datetime.datetime.now().isoformat(),
                'photoBase64': "",
                'deviceID': self.getFromMetadata('modelID'),
            }]
        }
        return bodyPhoto

    def getDeviceUrl(self):
        url = self.getFromMetadata('url') +
            self.getFromMetadata('deviceID')
        return url

    def getHost(self):
        host = self.getFromMetadata('host')
        return host

    def getTelemetryBody(self):
        body = {
            'mode': 'sync',
            'messageType': self.getFromMetadata('msgTypeTelemetry'),
            'messages': [{
                'log': datetime.datetime.now().isoformat(),
                'deviceID': self.getFromMetadata('modelID'),
                'roll': '0',
            }]
        }
```

```

        'pitch': '0',
        'throttle': '0',
        'yaw': '0',
        'aux1': '0',
        'aux2': '0',
        'aux3': '0',
        'aux4': '0',
        'fix': 'false',
        'numsat': '0',
        'longitude': '0',
        'latitude': '0',
        'speed': '0',
        'altitude': '0',
        'distanceHome': '0',
        'headingHome': '0',
        'amperage': '0',
        'rssi': '0',
        'vbat': '0',
        'powermetersum': '0',
        'motor1': '0',
        'motor2': '0',
        'motor3': '0',
        'motor4': '0',
        'motor5': '0',
        'motor6': '0',
        'motor7': '0',
        'motor8': '0',
        'version': '0',
        'type': '0',
        'angx': '0',
        'angy': '0',
        'heading': '0',
    }}
}
return body

def getRequestHeader(self):
    requestHeader = {
        'Authorization': 'Bearer '+
            self.getFromMetadata('deviceToken'),
        'Content-Type' : 'application/json;charset=utf-8'
    }
    return requestHeader

def getPhotoHeader(self):
    photoHeader = {
        'Authorization': 'Bearer '+
            self.getFromMetadata('deviceToken'),
        'Content-Type' : 'application/json;charset=utf-8'
    }
    return photoHeader

def getTelemetryHeader(self):
    telemetryHeader = {
        'Authorization': 'Bearer '+
            self.getFromMetadata('deviceToken'),
        'Content-Type' : 'application/json;charset=utf-8'
    }
    return telemetryHeader

```

Zdroj: vlastní práce autora

Třída Main.py pro odesílání a přijímání telemetrie z dronu hlavní jádro programu

Příloha 38 Třída Main.py

```
#!/usr/bin/env python
"""Get data script"""
__author__ = "Filip Maly"
__copyright__ = "Copyright 2018 Filip Maly"
__license__ = "GPL"
__version__ = "1.2"
__maintainer__ = "Filip Maly"
__email__ = "filipmaly92@gmail.com"
__status__ = "Development"
#from MultiWii import MultiWii
from Multiwii2 import MultiWii
from sys import stdout
from Config import Config

import httpplib, urllib
import json
import datetime
import time
import pygame
import requests
import json
import picamera
import base64

class Telemetry:
    def __init__(self):
        self.config = Config()
        self.board = MultiWii(self.config.getSerialPort())
        self.initializeCamera()
        self.connection = httpplib.HTTPSConnection(self.config.getHost())
        self.photoConn = httpplib.HTTPSConnection(self.config.getHost())
        print 'Nastaven port: ' +
            str(self.config.getFromMetadata('serailPort')) + ', a baudRate: ' +
            str(self.config.getFromMetadata('baudRate'))
        pygame.init()
        pygame.mixer.music.load("sound.mp3")
        pygame.mixer.music.set_volume(1)
        self.startProgram()

    def initializeCamera(self):
        try:
            self.camera = picamera.PiCamera()
            self.camera.capture('image.jpg')
            self.camera.hflip = True
            self.camera.vflip = True
        except Exception, error:
            print "Kamera neni inicializovana: " + str(error)

    def startProgram(self):
        counter = 1
        try:
            while True:
                self.sendTelemetry(self.getTelemetryData(),
                    self.config.getDeviceUrl(),
                    self.config.getTelemetryHeader())
                self.getResponse()
                time.sleep(self.config.getFromMetadata('timeSleep'))
                print counter
                counter = counter + 1
        except Exception, error:
            print "Chyba v cyklu programu " + str(error)
```

```

def getResponse(self):
    response = requests.get('https://' + self.config.getHost() +
        self.config.getDeviceUrl(), headers=self.config.getRequestHeader())
    #print 'test odpovedi:' + str(response.status_code)
    if response.status_code == 200:
        data = json.loads(response.text)
        for x in range(0, len(data)):
            print data[x]
            messageData = data[x]['messages']
            if (messageData[0]['commandID'] == 1):
                # ALARM ON
                print messageData[0]['commandID']
                pygame.mixer.music.play()
            elif (messageData[0]['commandID'] == 2):
                # ALARM OFF
                pygame.mixer.music.stop()
                print messageData[0]['commandID']
            elif (messageData[0]['commandID'] == 3):
                # TAKE PHOTO
                print messageData[0]['commandID']
            elif (messageData[0]['commandID'] == 4):
                # LED ON
                print messageData[0]['commandID']
            elif (messageData[0]['commandID'] == 5):
                # LED OFF
                print messageData[0]['commandID']
            elif (messageData[0]['commandID'] == 6):
                # GpS_HOLD ON
                print messageData[0]['commandID']
            elif (messageData[0]['commandID'] == 7):
                # GpS_HOLD OFF
                print messageData[0]['commandID']
            elif (messageData[0]['commandID'] == 8):
                #GPS_HOME ON
                print messageData[0]['commandID']
            elif (messageData[0]['commandID'] == 9):
                # GPS_HOME OFF
                print messageData[0]['commandID']

def takePhoto(self):
    try:
        self.camera.capture('image.jpg')
        image = 'image.jpg'
        image_64 = base64.encodestring(open(image, "rb").read())
        photoString = "data:image/png;base64," + image_64
        self.sendPhoto(photoString)
    except Exception, error:
        print "Error on Photo " + str(error)

def getTelemetryData(self):
    RC = self.board.getData(MultiWii.RC)
    GPS = self.board.getData(MultiWii.RAW_GPS)
    MOTORS = self.board.getData(MultiWii.MOTOR)
    ATTITUDE = self.board.getData(MultiWii.ALTITUDE)
    DISTANCEHOME = self.board.getData(MultiWii.COMP_GPS)
    body = self.config.getTelemetryBody()
    body['messages'][0]['roll'] = RC['roll']
    body['messages'][0]['throttle'] = RC['throttle']
    body['messages'][0]['yaw'] = RC['yaw']
    body['messages'][0]['pitch'] = RC['pitch']
    body['messages'][0]['motor1'] = MOTORS['m1'],
    body['messages'][0]['motor2'] = MOTORS['m2'],
    body['messages'][0]['motor3'] = MOTORS['m6'],
    body['messages'][0]['motor4'] = MOTORS['m4'],
    body['messages'][0]['motor5'] = MOTORS['m5'],
    body['messages'][0]['motor6'] = MOTORS['m6'],
    body['messages'][0]['motor7'] = MOTORS['m7'],
    body['messages'][0]['motor8'] = MOTORS['m8'],

```

```

        return body

def sendPhoto(self , photo):
    bodyPhoto = self.config.getBodyPhoto()
    bodyPhoto['messages'][0]['photoBase64'] = photo,
    self.photoConn.request("POST", self.config.getDeviceUrl(),
        json.dumps(bodyPhoto), self.config.getPhotoHeader())
    r2 = self.photoConn.getresponse()
    print r2.status, r2.reason
    self.photoConn.close()

def sendTelemetry(self , body, url, header):
    self.connection.request("POST", url, json.dumps(body), header)
    r1 = self.connection.getresponse()
    print r1.status, r1.reason
    print body
    self.connection.close()

# spusteni main a kodu

def main():
    classTelemetry = Telemetry()
    classTelemetry.__init__()

if __name__ == "__main__":
    main()

```

Zdroj: vlastní práce autora


Master view pro výběr dronu a zobrazení informací v needitačním módu.

Příloha 39 master view pro výběr dronu a zobrazení informací

Základní info +

Refresh DB

— 5 +

 Refresh dat za 5

Výběr dronu

HEXA v

Název Dronu

HEXA_CZU

Vlastník

Filip Maly

Poslední použití

15.3.2018

Délka posledního letu

2.4 km

Čas poslední cesty

20 minut

Počet nálétaných KM


45 km

Počet letů

15

Frame dronu

HEXA

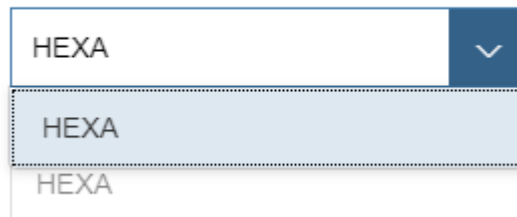


Zdroj: vlastní práce autora

Prvek v Master view pro výběr dronu v needitačním módu.

Príloha 40 prvek pro výběr dronu

Výběr dronu



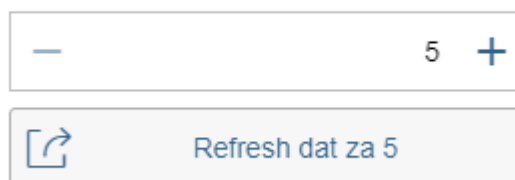
The image shows a UI element titled "Výběr dronu". It consists of a dropdown menu. The top part is a white box with a dark blue border containing the text "HEXA" and a dark blue arrow pointing downwards. Below this is a list of options, with the first one "HEXA" highlighted in a light blue background. A second "HEXA" option is visible below it.

Zdroj: vlastní práce autora

Prvek v Master view pro aktualizaci dat z databáze.

Príloha 41 prvek na aktualizaci dat z databáze

Refresh DB



The image shows a UI element titled "Refresh DB". It features a horizontal control with a minus sign on the left, the number "5" in the center, and a plus sign on the right. Below this is a button with a refresh icon (a square with a curved arrow) and the text "Refresh dat za 5".

Zdroj: vlastní práce autora


Master view pro editaci informací o dronu.

Příloha 42 master view pro editaci informací o dronu

Základní info +

Refresh DB

— 5 +

 Refresh dat za 5

Výběr dronu

HEXA v

Název Dronu

HEXA_CZU

Vlastník

Filip Maly

Poslední použití

15.3.2018

Délka posledního letu

2.4 km

Čas poslední cesty

20 minut

Počet nálétaných KM

45 km

Počet letů

15

Frame dronu

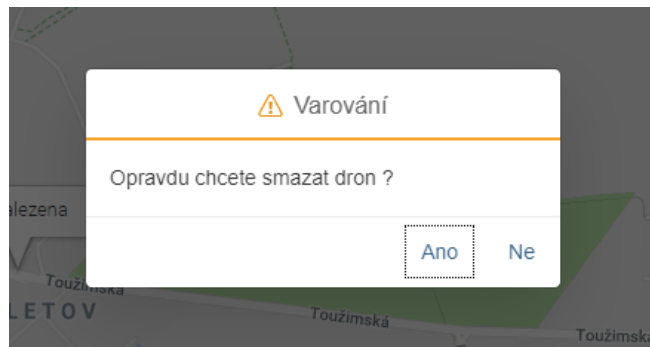
HEXA



Zdroj: vlastní práce autora

Master view, dialog pro potvrzení volby mazání dronu.

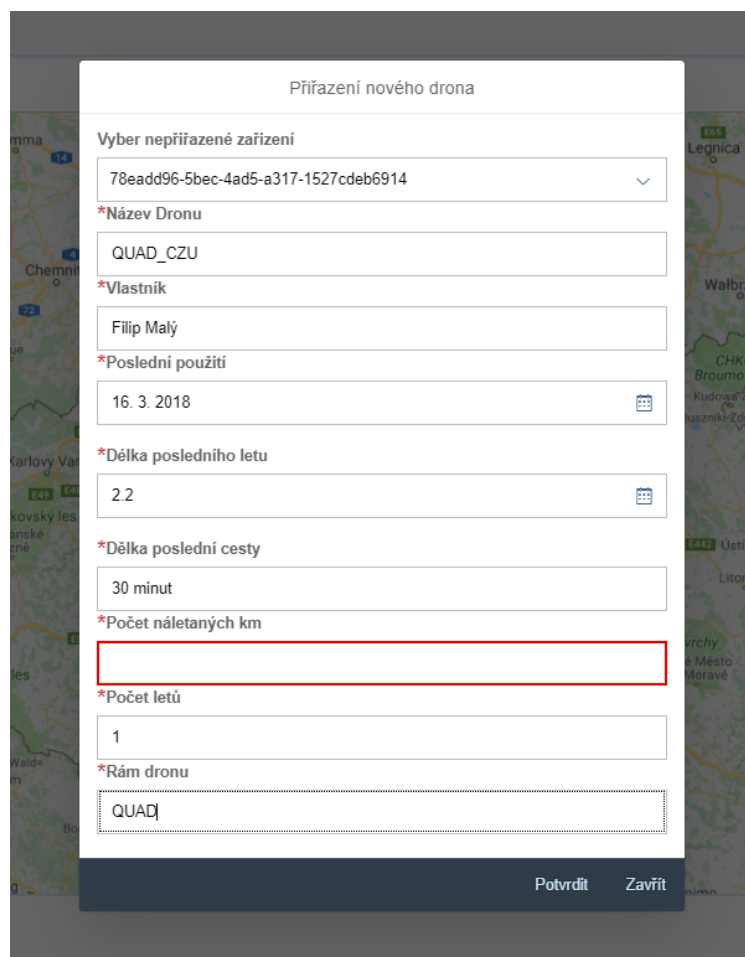
Priloha 43 potvrzovací dialog pro mazání dronu



Zdroj: vlastní práce autora

Master view, dialog pro přidání nového dronu, spuštěného tlačítkem „+“ v levé části aplikace. Červeně je označené nevyplněné pole.

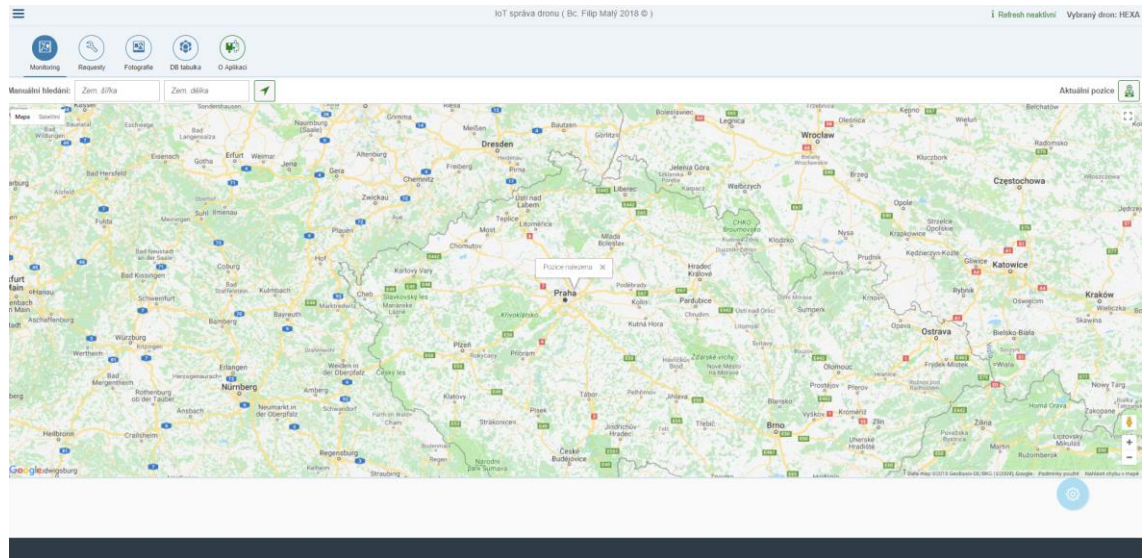
Priloha 44 dialog přidání nového dronu



Zdroj: vlastní práce autora

Detailní obrazovka Monitoring, ve výchozím stavu.

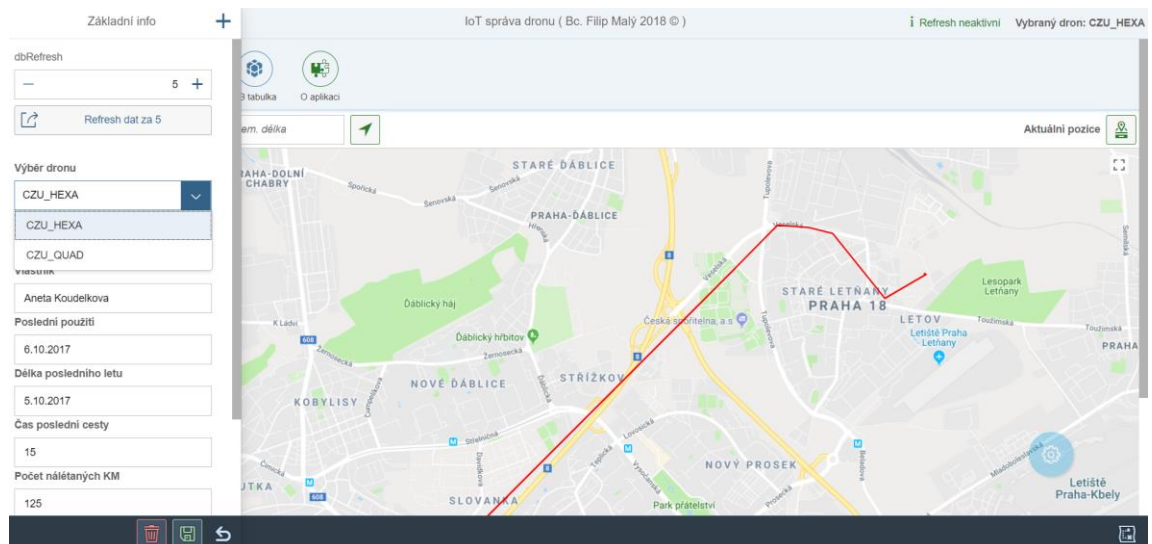
Příloha 45 detailní obrazovka Monitoring ve výchozím stavu



Zdroj: vlastní práce autora

Detailní obrazovka Monitoring, se zobrazením dráhy letu vybranou v obrazovce DB tabulka a levým panelem pro editaci záznamu. Dráha letu je zobrazena červeně.

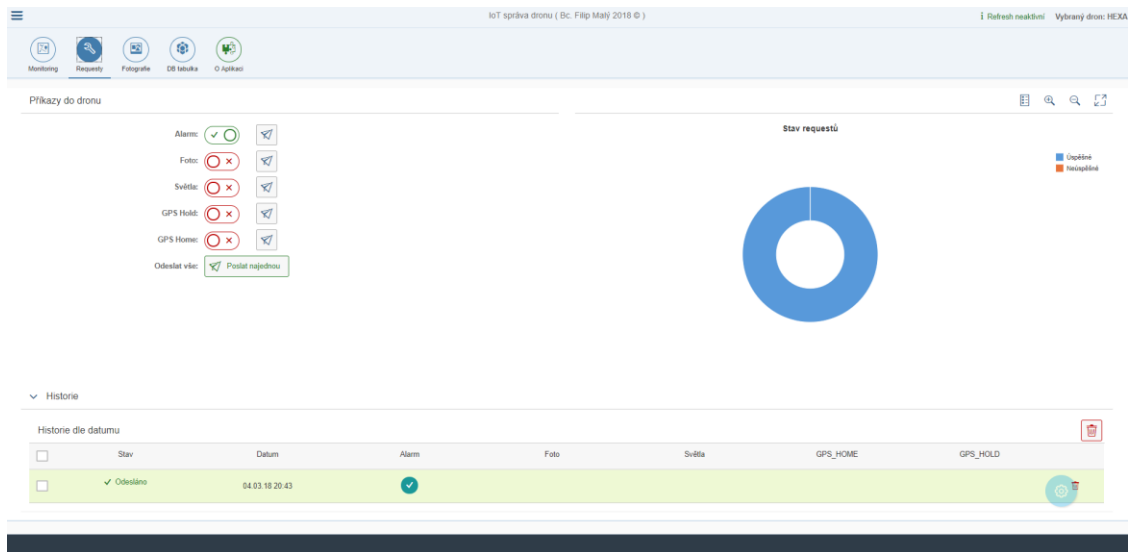
Příloha 46 detailní obrazovka Monitoring s vyznačenou dráhou letu



Zdroj: vlastní práce autora

Detailní obrazovka Requesty, ve výchozím stavu.

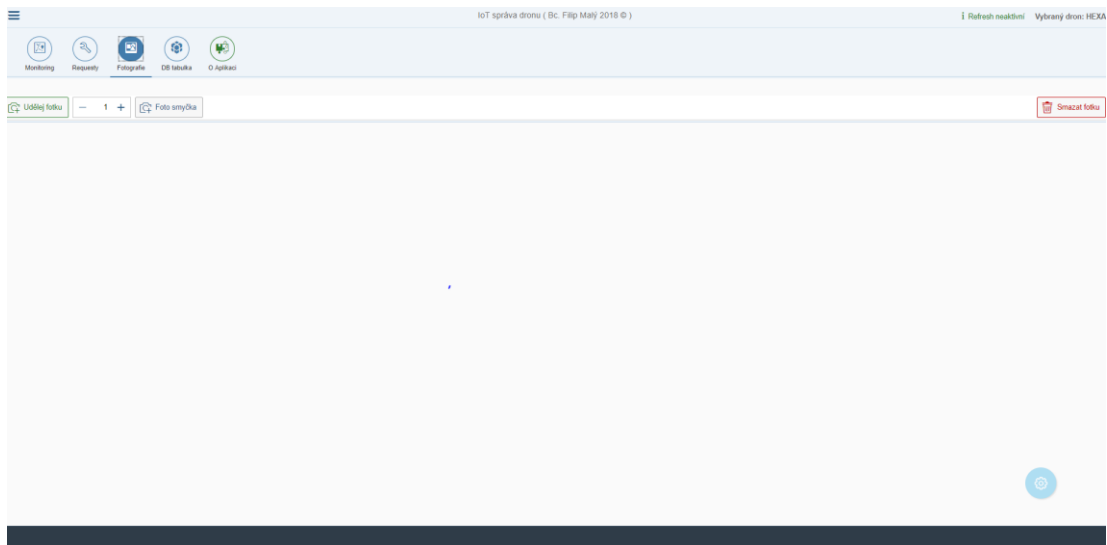
Příloha 47 detailní obrazovka Requesty



Zdroj: vlastní práce autora

Detailní obrazovka Fotografie, ve výchozím stavu.

Příloha 48 detailní obrazovka Fotografie



Zdroj: vlastní práce autora

Detailní obrazovka DB tabulka, ve výchozím stavu s filtrem na aktuální den.

Příloha 49 detailní obrazovka DB tabulka

IoT správa dronu (Bc. Filip Malý 2018 ©) | Refresh neakční | Vybraný dron: HEXA

Monitoring | Requesty | Fotografie | DB tabulka | O aplikaci

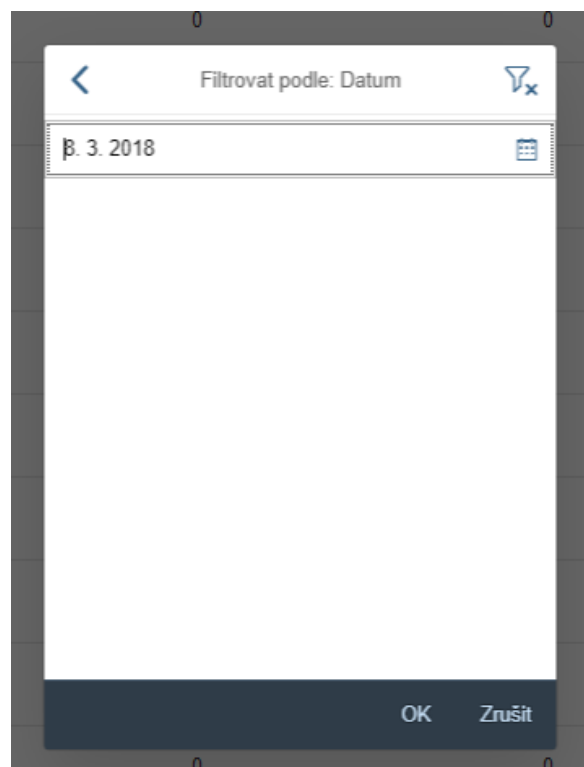
Záznamy z letu

<input type="checkbox"/>	Datum	Model	Napětí	Proud mA	Satelity	Zem. sílka	Zem. délka	
<input type="checkbox"/>	8.3.2018 14:14:45	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:14:34	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:14:23	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:14:13	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:14:03	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:13:53	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:13:43	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:13:33	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:13:23	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:13:13	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:13:03	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:12:53	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:12:43	1	0	0	0	0	0	
<input type="checkbox"/>	8.3.2018 14:12:33	1	0	0	0	0	0	

Zdroj: vlastní práce autora

Dialog pro filtrování dle datumu, otevřený po stisknutí tlačítka filtru v obrazovce DB tabulka

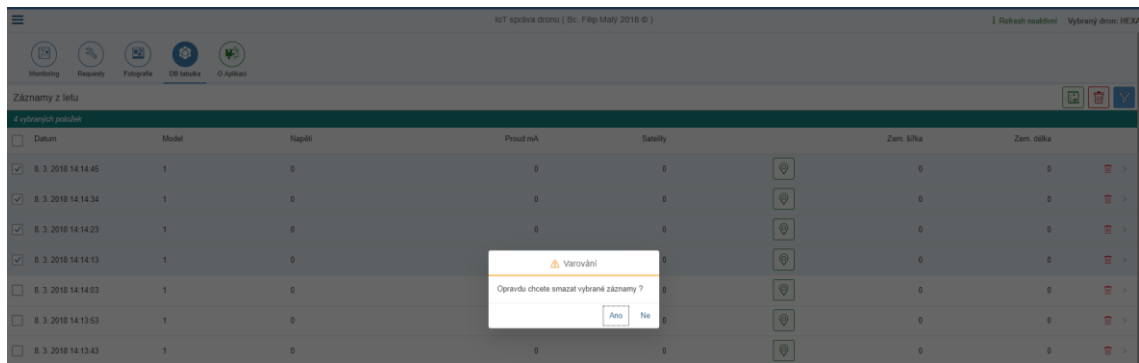
Příloha 50 dialog pro filtrování dle datumu



Zdroj: vlastní práce autora

Dialog pro potvrzení mazání zvolených záznamů s informací o počtu zaškrtnutých položek v pozadí aplikace na obrazovce DB tabulka.

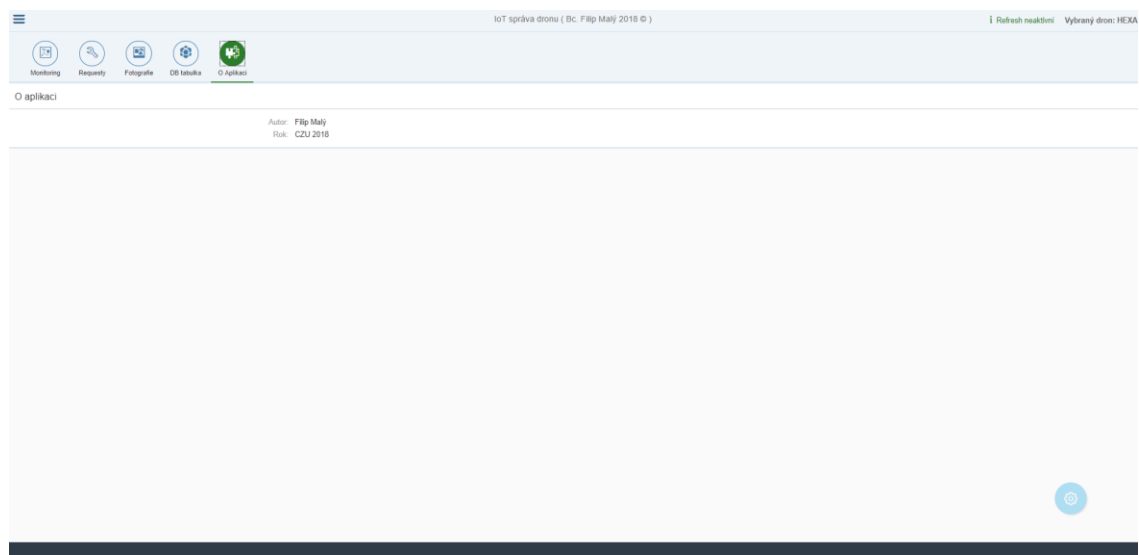
Příloha 51 mazání více záznamů z tabulky a informativní toolbar



Zdroj: vlastní práce autora

Detailní obrazovka O Aplikaci, ve výchozím stavu.

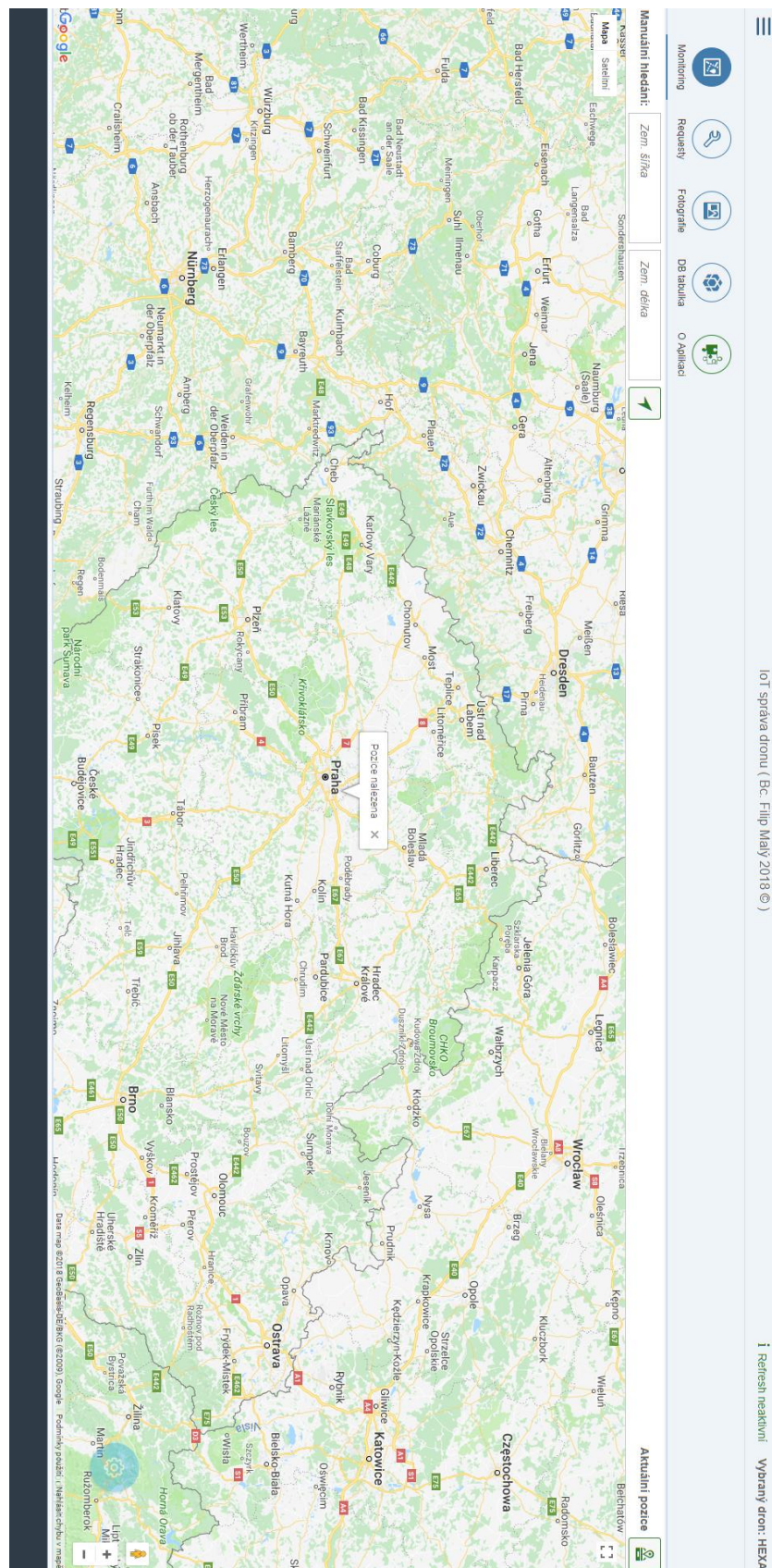
Příloha 52 detailní obrazovka O Aplikaci



Zdroj: vlastní práce autora

Zobrazení aplikace ve výchozím stavu na zařízení desktop.

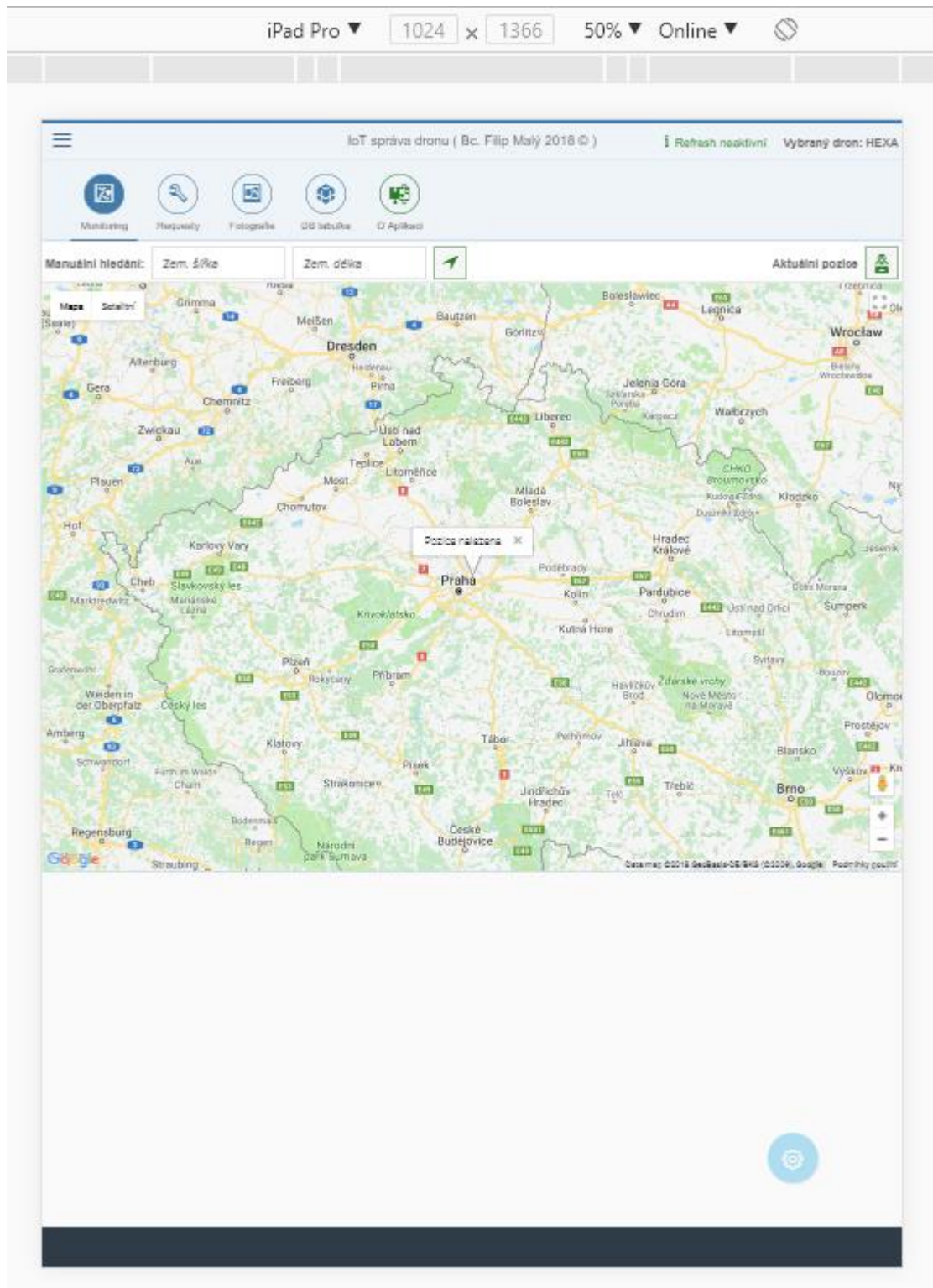
Priloha 53 aplikace ve výchozím stavu na desktopu



Zdroj: vlastní práce autora

Zobrazení aplikace ve výchozím stavu na zařízení tablet.

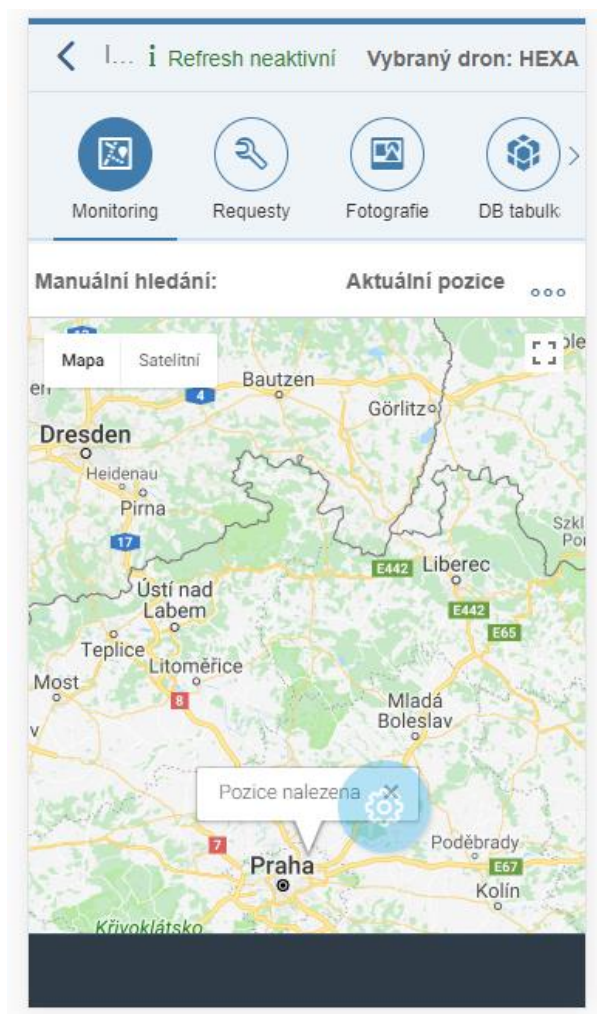
Příloha 54 zobrazení aplikace na zařízení tablet



Zdroj: vlastní práce autora

Zobrazení aplikace ve výchozím stavu na mobilním zařízení s přidanou ikonou pro navigaci mezi master a detail obrazovkou.

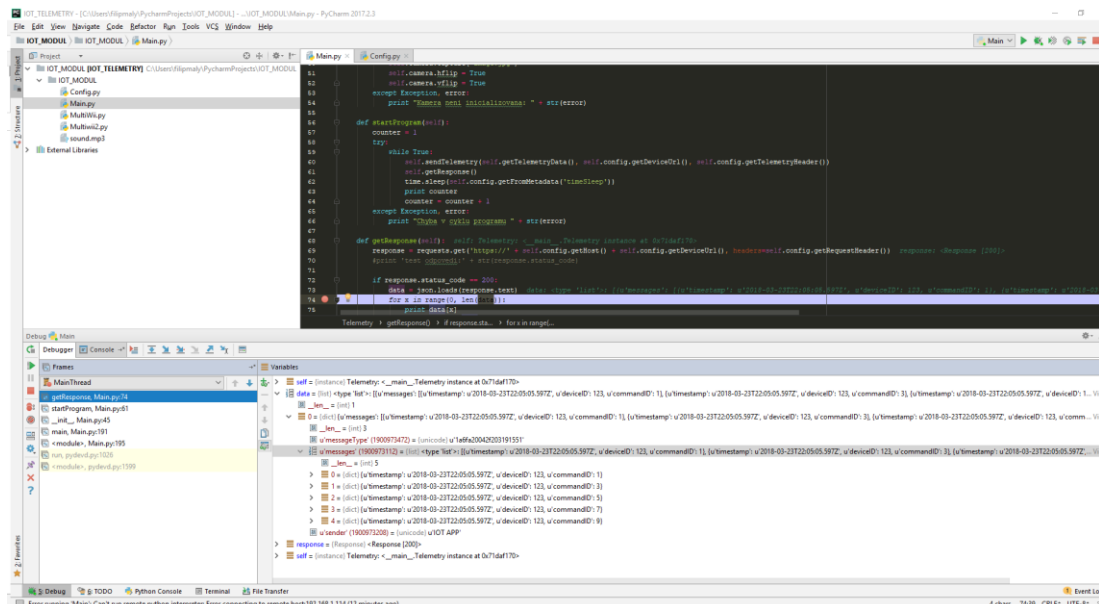
Příloha 55 zobrazení aplikace na mobilním zařízení



Zdroj: vlastní práce autora

Přijmutí telemetrie v dronu pomocí naprogramovaného IoT modulu. Fotografie obrazovky z debuggeru programu Pycharm. Na obrazovce jsou vidět příšle zprávy z aplikace.

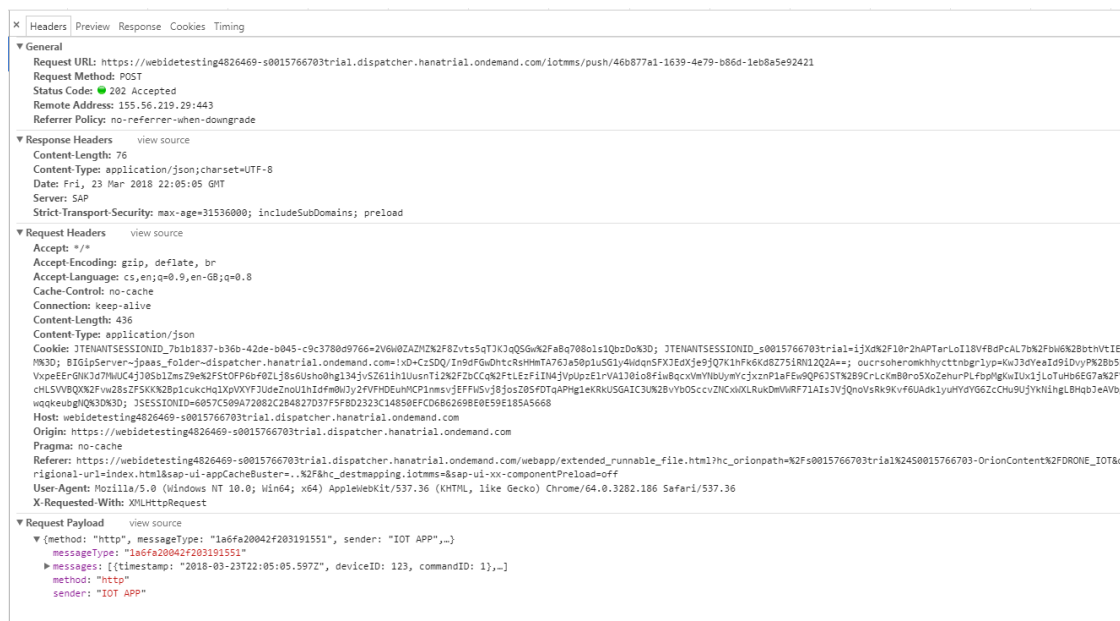
Příloha 58 přijmutí příkazu z aplikace v dronu



Zdroj: vlastní práce autora

Odeslání příkazu z aplikace. Na obrázku je vidět batch se zprávami pro odeslání do dronu přes IoT službu.

Příloha 59 odeslání příkazu do dronu



Zdroj: vlastní práce autora

Ukázka úspěšného odeslání příkazu do dronu záložka v Google Chrome Response.

Příloha 60 odeslání příkazu do dronu Response

```
× Headers Preview Response Cookies Timing
1 [{"msg": "Message(s) pushed to device [46b877a1-1639-4e79-b86d-1eb8a5e92421]"}]
```

Zdroj: vlastní práce autora

Dron typu QUAD s jednotkou RPI 2

Příloha 61 quad dron s jednotkou RPI 2



Zdroj: vlastní práce autora

Dron typu QUAD s jednotkou RPI 2 detailní pohled

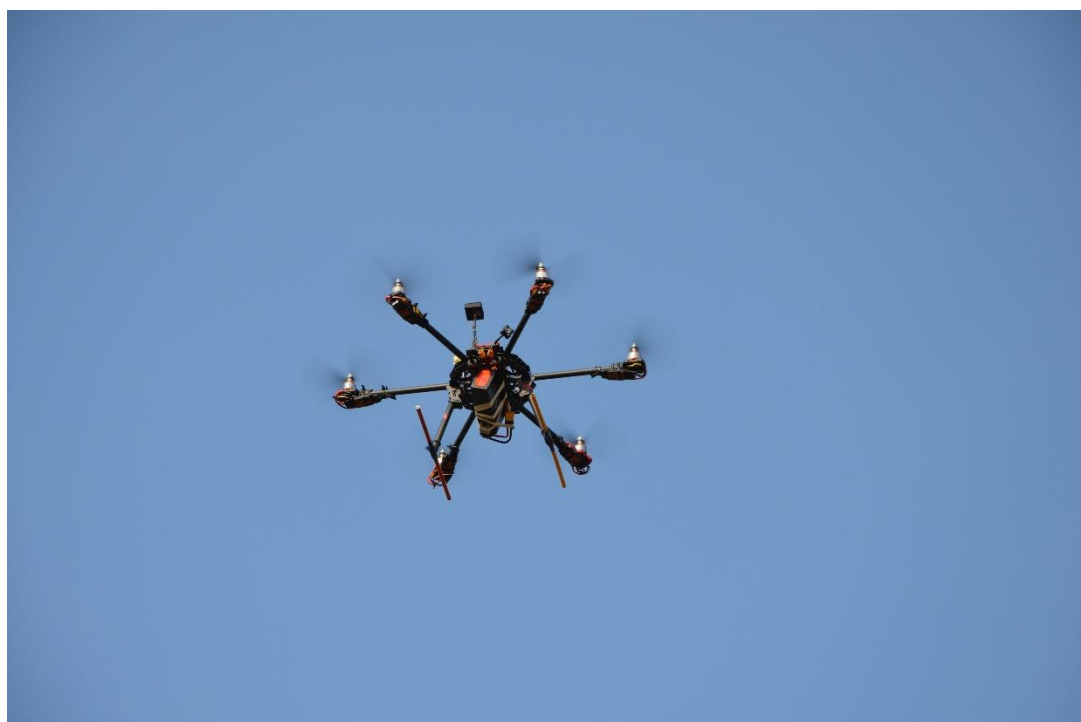
Příloha 62 dron s RPI 2 detail



Zdroj: vlastní práce autora

Dron typu HEXA s jednotkou RPI 3

Příloha 63 hexa dron s jednotkou RPI 3



Zdroj: vlastní práce autora

Dron typu HEXA s jednotkou RPI 3 detailní pohled

Příloha 64 dron s RPI 3 detail



Zdroj: vlastní práce autora