

Serverové platformy pro aplikace IoT

Ondřej Šálek

Bakalářská práce

2023



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Ondřej Šálek
Osobní číslo: A19106
Studijní program: B3902 Inženýrská informatika
Studijní obor: Softwarové inženýrství
Forma studia: Prezenční
Téma práce: Serverové platformy pro aplikace IoT
Téma práce anglicky: Server Platforms for IoT Applications

Zásady pro vypracování

1. Prostudujte bezplatné nebo open source platformy pro realizaci serverové části IoT aplikací.
2. Prozkoumejte nabídku senzorů a měřidel pro použití v bytových domech.
3. Pro realizaci práce vyberte vhodný typ senzoru nebo měřidla a některou ze serverových platform.
4. S vybranými technologiemi realizujte demonstrátor IoT aplikace pro bytový dům.
5. Popište možnosti trvalé instalace serverové platformy u vybraného poskytovatele.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. YARALI, Abdulrahman. IoT: Platforms, Connectivity, Applications and Services. New York: Nova, 2018. ISBN 978-1-53613-401-8.
2. KERAMIDAS, Georgios a Michael HÜBNER. Components and Services for IoT Platforms: Paving the Way for IoT Standards. Cham: Springer, 2017. ISBN 978-3-319-42304-3.
3. WAN, Jiafu, Iztok HUMAR a Daqiang ZHANG. Industrial IoT Technologies and Applications. Cham: Springer, 2016. ISBN 978-3-030-71061-3.
4. MOHANTY, Sachi Nandan, Jyotir Moy CHATTERJEE a Suneeta SATPATHY. Internet of Things and Its Applications. Cham: Springer, 2021. ISBN 978-3-030-77527-8.
5. SHOVIC, John C. Raspberry Pi IoT Projects: Prototyping Experiments for Makers. New York: Apress, 2016. ISBN 9781484213773.

Vedoucí bakalářské práce: **Ing. Tomáš Dulík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **2. prosince 2022**

Termín odevzdání bakalářské práce: **26. května 2023**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 7. prosince 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnaní případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 23.05.2023

Šálek Ondřej
podpis studenta

ABSTRAKT

Cílem této práce je prostudování bezplatných a open-source platform pro realizaci serverové části aplikací IoT. Zkoumá se nabídka zařízení a senzorů pro použití v bytových domech. Praktická část práce se věnuje výběru bezplatné serverové části, volbě dostupného hardwaru umožňujícího připojení na serverovou platformu a vhodného senzoru pro použití v bytových domech. Současně se vybírá vhodná komunikační technologie. V práci jsou rozebrány programy jednotlivých zařízení a jejich propojení pomocí komunikační sítě a připojení do serverové platformy. V závěru je popsána možnost trvalé instalace v bytovém domě.

Klíčová slova: IoT, LoRaWAN, Pygate, Wio-E5 mini, měření teploty, The Things Network, AllThingsTalk

ABSTRAKT

The aim of this thesis is to study free and open-source platforms for implementing the server side of IoT applications. The range of devices and sensors for use in residential homes is examined. The practical part of the thesis is devoted to the selection of the free server part, the choice of available hardware that allows connection to the server platform and the appropriate sensor for use in apartment buildings. At the same time, a suitable communication technology is selected. The thesis discusses the programs of the individual devices and their interconnection using a communication network and connection to the server platform. Finally, the possibility of permanent installation in a residential building is described.

Keywords: IoT, LoRaWAN, Pygate, Wio-E5 mini, temperature measurement, The Things Network, AllThingsTalk

Chtěl bych poděkovat vedoucímu své práce Ing. Tomáši Dulíkovi, Ph.D. za odborné vedení a konzultace, které mi poskytl v průběhu zpracování bakalářské práce. Také za podporu při problémech s nedostatkem hardwaru na trhu. Dále bych rád poděkoval Ing. Petru Vítkovi za konzultaci a Ing. Tomáši Juřenovi za poskytnutí návodu a další konzultaci.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	12
I TEORETICKÁ ČÁST	13
1 INTERNET OF THINGS	14
1.1 OBLASTI VYUŽITÍ IoT	14
1.1.1 Průmysl	15
1.1.2 Město.....	15
1.1.3 Domácnost.....	16
1.2 KOMUNIKAČNÍ PROTOKOLY IoT	16
1.2.1 MQTT	16
1.2.2 HTTP.....	17
1.2.3 CoAP	17
1.2.4 AMQP	17
1.3 KOMUNIKAČNÍ SÍTĚ IoT	18
1.3.1 Sigfox	18
1.3.2 LoRaWAN	18
1.3.3 NB-LTE.....	19
1.3.4 Bluetooth LE 4	19
1.3.5 Bluetooth LE 5	20
1.3.6 Wi-Fi	20
1.3.7 GSM	20
1.3.8 LTE	21
1.3.9 LTE-M.....	21
2 HARDWAROVÉ PLATFORMY PRO IOT	22

2.1	RASPBERRY PI.....	22
2.2	ARDUINO.....	22
2.3	ADAFRUIT	23
2.4	PARTICLE	24
2.5	ESPRESSIF.....	25
2.6	PYCOM	26
2.7	SEED STUDIO	27
2.8	STMICROELECTRONICS.....	28
3	PORTÁLOVÉ SYSTÉMY PRO IOT.....	30
3.1	KAA	30
3.2	THINGSPEAK	31
3.3	THINGSBOARD	31
3.4	APACHE STREAMPIPES.....	32
3.5	GOOGLE CLOUD IOT CORE	32
3.6	AMAZON WEB SERVICES IOT CORE.....	33
3.7	IBM WATSON IOT	33
3.8	MICROSOFT AZURE IOT SUITE	34
3.9	THE THINGS NETWORK	34
3.10	ALLTHINGSTALK MAKER	35
4	SENZORY A MĚŘIDLA PRO MĚŘENÍ TEPLoty V BYTOVÝCH DOMECH.....	36

4.1	DHT11.....	36
4.2	DHT20.....	37
4.3	DHT21.....	38
4.4	BME280	39
4.5	LM35DZ.....	40
4.6	DS18B20	41
4.7	EM300-TH-868M	42
II PRAKTICKÁ ČÁST		44
5	NÁVRH DEMONSTRÁTORU	45
5.1	POŽADAVKY NA MĚŘENÍ	45
5.1.1	Hrozba bakterií.....	45
5.2	VÝBĚR HARDWAROVÝCH A SOFTWAREVÝCH ČÁSTÍ IOT APLIKACE.....	45
5.2.1	Volba komunikační technologie	45
5.2.2	Výběr vhodného senzoru.....	46
5.2.3	Výběr vhodné hardwarové platformy	46
5.2.3.1	Hardware pro měřicí zařízení	46
5.2.3.2	Hardware pro komunikaci se zařízeními	47
5.2.4	Výběr vhodné softwarové platformy	49
6	REALIZACE DEMONSTRÁTORU.....	50
6.1	POUŽITÉ TECHNOLOGIE	50
6.1.1	Jazyk Python	50
6.1.2	Jazyk C.....	50
6.1.3	STM32CubeMX.....	51
6.1.4	STM32CubeProgrammer	51
6.1.5	Vývojové prostředí.....	52

6.1.5.1	Visual Studio Code	52
6.1.5.2	STM32CubeIDE	52
6.2	LoRAWAN BRÁNA	52
6.2.1	Aktualizace firmware	53
6.2.2	Registrace zařízení	54
6.2.3	Rozšíření editoru VS Code.....	55
6.2.4	Program	56
6.2.4.1	Soubor boot.py	56
6.2.4.2	Soubor main.py	56
6.2.4.3	Soubor gateway.py.....	57
6.2.4.4	Soubor config.py.....	60
6.2.4.5	Soubor global_conf.json	60
6.3	LoRAWAN KONCOVÉ ZAŘÍZENÍ	65
6.3.1	Registrace zařízení	65
6.3.2	Program	68
6.3.2.1	Otevření projektu	68
6.3.2.2	Nastavení výstupu projektu	69
6.3.2.3	Konfigurační nastavení programu	70
6.3.2.4	Odesílaná data	73
6.3.3	Sestavení programu	74
6.3.4	Nahrání programu do zařízení.....	75
6.4	NASTAVENÍ SERVEROVÝCH PLATFORM	78
6.4.1	Nastavení aplikace v platformě The Things Network.....	78
6.4.2	Nastavení platformy AllThingsTalk Maker	80

7 TRVALÁ INSTALACE PLATFORMY	85
ZÁVĚR	87
SEZNAM POUŽITÉ LITERATURY.....	89
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	96
SEZNAM OBRÁZKŮ	100
SEZNAM TABULEK.....	102
SEZNAM PŘÍLOH.....	103

ÚVOD

V dnešním světě internetu se internet věcí (IoT) stal důležitou součástí v různých odvětvích. Počet zařízení IoT neustále roste a tato zařízení se stávají nedílnou součástí průmyslu, měst a domácností. Z tohoto důvodu je při vývoji aplikací IoT potřeba myslet na výběr správné serverové platformy. Důležitý je i výběr vhodného zařízení IoT, a tedy i komunikační technologie.

Mezi serverovými platformami není snadné se zorientovat. Výběr serverové platformy závisí na několika faktorech. Jedním důležitým faktorem je umístění platformy. Ta může být umístěna v lokálním nebo v cloudovém serveru. Druhým důležitým faktorem je cena. Cena se odvíjí podle nabízených služeb platformy. Platforma může být zcela bezplatná nebo jen částečně bezplatná. Dále se může platit za některé služby nebo za celou platformu.

Výběr vhodného zařízení závisí na určitých požadavcích, kterými může být přístup k elektrické síti a správná funkčnost zařízení v různých teplotních podmínkách. Dále je nutné zvolit vyhovující komunikační technologii podle dosahu signálu.

Cílem této práce je prostudování bezplatných a open-source platform pro realizaci serverové části aplikací IoT. Výběr dostupného hardwaru umožňujícího připojení na serverovou platformu a zvolení vhodné komunikační technologie. Pro měření teploty je nutné zvolit vyhovující senzor.

Návrh demonstrátoru a jeho realizace je určena pro měření teploty vody v potrubí v bytových domech. Důvodem je nárůst cen energií a nebezpečí ohrožení lidských životů. Za energie lze ušetřit při přípravě teplé vody v domech. Je potřeba sledovat minimální teplotu, aby se nedostala pod určitou hranici. Pokud by teplota vody byla pod minimální hranicí, mohlo by dojít k tvorbě nebezpečných bakterií a ohrožení zdraví.

Vytvořený demonstrátor v této práci je možné využít pro kontrolování teploty vody z webového rozhraní serverové platformy, bez nutnosti fyzické kontroly hodnoty teploměru.

I. TEORETICKÁ ČÁST

1 INTERNET OF THINGS

Internet of Things (IoT, Internet věcí) je termín používající se k popisu prostředí, v němž se nachází miliony zařízení (věcí), které jsou připojeny k internetu a samostatně komunikují. S tak velkým množstvím objektů propojených v IoT se prostředí, ve kterém jsou umístěny, stává chytřejším [1].

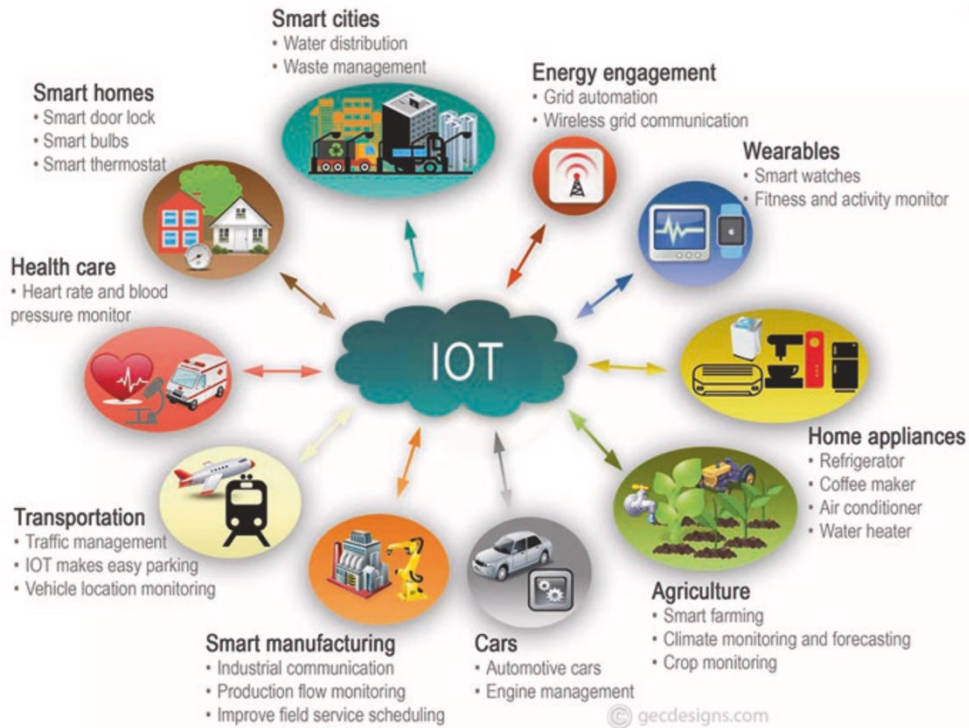
Koncept IoT předpokládá vytvoření distribuované sítě sestávající se z mnoha fyzických zařízení vybavených softwarem, senzory a možnostmi připojení, které shromažďují a sdílejí data mezi sebou a s centrální platformou prostřednictvím internetu [2].

Všechny možné definice IoT jsou si podobné. Jedna z nich zní: „Internet věcí je připojení zařízení k internetu a k dalším připojeným zařízením. Všechna tato zařízení sdílejí data o způsobu svého používání a o svém okolním prostředí. Většinou je připojení k síti bezdrátové [3].“

Připojená zařízení (věci) jsou podle IoT objekty, hardware, které obsahují software a senzory. Software každého zařízení se zabývá jeho logikou a funkcí. Senzory mají funkci snímání okolí, snímají například teplotu, vlhkost, pohyb atd. Informace získané ze sensorů se dále přenášejí po síti do úložiště, aby se následně tato data mohla zanalyzovat. Úložištěm je většinou cloud – vzdálený server. K uloženým datům lze přistupovat pomocí webové či mobilní aplikace. Dále lze s daty provádět různé operace, například vytvoření různých grafů a tabulek [4].

1.1 Oblasti využití IoT

IoT se využívá v několika oblastech, v nichž technologie lidem usnadňují a zabezpečují život, zlepšují kvalitu životního prostředí, zvyšují úsporu peněz a času. Na následujícím obrázku (Obrázek 1) lze vidět mnoho oblastí využití, které lze shrnout do tří oblastí: průmysl, město a domácnost [1].



Obrázek 1. Application of IoT in modern age [1]

1.1.1 Průmysl

První oblastí je průmysl, tzv. Industry 4.0. Cílem je nahrazení lidské síly. Robotizace a automatizace zajistí zvýšení rychlosti a efektivity při výrobě přesnějších a levnějších produktů. Zajistí se také efektivní využití materiálů a jejich ekologičtější zpracování. Hlavní smyslem robotizace je zlepšení dopadu průmyslu na životní prostředí [5].

1.1.2 Město

Další oblastí je město, tzv. Smart City (Chytré město). Z dokumentu OSN z roku 1987 známého jako „Brundtlandova zpráva“ vznikla nová definice: „Chytrá města jsou taková, která uspokojují potřeby současnosti, aniž by ohrožovala schopnost budoucích generací uspokojovat své vlastní potřeby“. Chytré město spojuje technologie, vládu a společnost s cílem inteligentní ekonomiky, kultury, životního prostředí a života lidí [6].

V chytrém městě se využívá mnoho technologií. Těmito technologiemi jsou například senzory zabudované v silnicích snímající zejména plynulost dopravy, ale také propadání a opotřebení vozovky. Dále senzory v interiérech budov řídí použití osvětlení, vytápění a klimatizace a další systémy za účelem omezení výdajů za energie v návaznosti počtu přítomných lidí v místnostech. Dnes jsou důležité i senzory u vchodů, které umožňují rozpoznávání obličejů pro lepší zabezpečení bytových domů, komerčních budov, vládních

úřadů a dalších. Mezi Smart City spadá i chytrá doprava, kde technologie řídí dopravu na křižovatkách a dávají vědět o dopravní situaci na frekventovaných úsecích a také napoví, kde jsou na parkovištích volná parkovací místa [6].

1.1.3 Domácnost

Poslední oblastí s využitím IoT je domácnost, tzv. Smart home (Chytrá domácnost). Domácnost lze označit za chytrou v případě, pokud je vybavena několika chytrými zařízeními, které lze ovládat na dálku a nastavit pro automatizaci údržby domu. Lze je sjednotit do jedné sítě, aby si zařízení mohla mezi sebou sdílet data. Pokud jsou zařízení připojena k internetové síti, může s nimi komunikovat i telefon a další zařízení které nejsou v blízkosti domácí sítě. Například přijíždíte autem k domu, tak telefon odešle informaci, že se blížíte, dům tuto informaci zpracuje, otevře garáž, rozsvítí v ní světla a dům se začne vytápět. Tomuto se už říká chytrá domácnost. Cílem chytré domácnosti je šetření energie a životního prostředí, zaručení lepší bezpečnosti domu a komfortnějšího bydlení [7].

1.2 Komunikační protokoly IoT

Pro zařízení IoT je připojení k internetu, popř. lokální síti, tak trochu podmínkou. Připojení k internetu umožňuje zařízením spolupracovat mezi sebou a s backendovými službami. Základním síťovým protokolem internetu je TCP/IP. Standardem pro komunikaci v rámci internetu věcí se stal protokol MQTT (Message Queue Telemetry Transport), který je postaven nad zásobníkem TCP/IP. Protokol MQTT může běžet také na protokolu SSL/TLS, což je zabezpečený protokol postavený na protokolu TCP/IP, který zajišťuje, že veškerá datová komunikace mezi zařízeními je šifrovaná a bezpečná. Dalšími protokoly jsou HTTP, CoAP a AMQP [8].

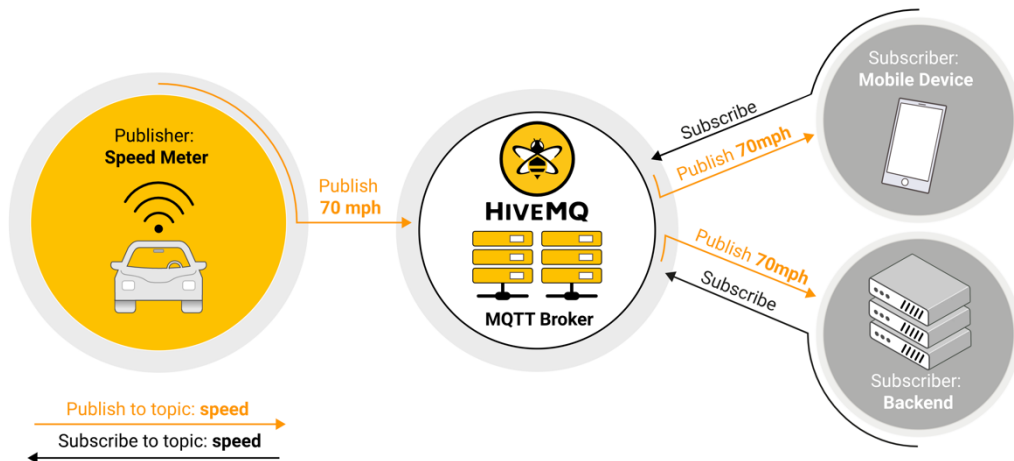
1.2.1 MQTT

Protokol MQTT (Message Queue Telemetry Transport) byl vynalezen a vyvinut společností IBM na konci 90. let. Původně se používal k propojení senzorů na ropovodech se satelity. Jak název napovídá, jedná se o protokol pro zasílání zpráv, který podporuje asynchronní komunikaci mezi stranami. Asynchronní protokol zpráv odděluje odesílatele a příjemce zprávy v prostoru i čase, a proto je škálovatelný v nespolehlivých síťových prostředích [1].

Hlavními výhodami protokolu je minimalizování zdrojů potřebných pro klienta a šířku pásma. Protokol je lehký a efektivní. Umožňuje obousměrnou komunikaci mezi zařízeními a servery, a také vysílání skupinám zařízení. Podporuje trvalé relace mezi zařízeními a

serverem, které zkracují dobu potřebnou pro opětovné připojení v nespolehlivých sítích. Zprávy mohou být šifrovány pomocí TLS a podporují protokoly ověřování klienta [9].

Schéma obousměrné komunikace protokolu MQTT je znázorněno na obrázku (Obrázek 2) [9].



Obrázek 2. MQTT Publish/Subscribe Architecture [9]

1.2.2 HTTP

HTTP (Hypertext Transfer Protocol) je to základ datové komunikace pro World Wide Web. Stávající struktura internetu je použitelná jakýmkoli zařízením IoT, ale pro většinu aplikací IoT je příliš energeticky náročná. To vedlo k tomu, že je HTTP odmítán jako protokol nevhodný pro IoT [10].

1.2.3 CoAP

Protokol CoAP (Constrained Application Protocol) je navržen pro potřeby systémů internetu věcí založených na protokolu HTTP. CoAp energetické omezení vyřešil převedením modelu HTTP na použití v omezených zařízeních a síťových prostředích. Má nízké náklady, je snadno použitelný a má možnost umožnit podporu vícesměrového vysílání [10].

Je ideální pro použití v zařízeních s omezenými zdroji, jako jsou mikrokontroléry IoT. Tradičně se používá v aplikacích zahrnujících inteligentní energetiku a automatizaci budov [10].

1.2.4 AMQP

Společnost JPMorgan Chase&Co založila protokol AMQP (Advanced Message Queuing Protocol) v roce 2003. Nabízí výměnu dat v rámci sítě a jedná se o otevřený protokol TCP/IP [10].

Protokol AMQP se zaměřuje na analytické prostředí založené na serverech, především na bankovní systémy. Kvůli své náročnosti není vhodný pro senzorová zařízení s omezenou pamětí [10].

1.3 Komunikační sítě IoT

Komunikace hraje klíčovou roli v IoT, umožňuje zpracování dat, předpovídání nebo interakci v reálném čase. Bezdrátové komunikační systémy se využívají k zajištění výměny dat. Typy bezdrátových komunikačních systémů se odlišují rozsahem pokrytí/kapacity. Poskytují několik možností, aby bylo možné zajistit různé typy připojení v závislosti na konfiguraci, umístění a topologických požadavcích sítě. Lze poskytovat různé služby, jako jsou logistika, údržba, lokalizace, sledování nebo diagnostika prostřednictvím bezdrátových osobních sítí (Bluetooth, NFC, RFID atd.), bezdrátových senzorových sítí (ZigBee, Sigfox, LoRa/LoRaWAN) nebo mobilních sítí (s důrazem na systémy 4G a 5G, včetně specifických sítí pro komunikaci na bázi IoT, jako je NB-IoT) [5].

1.3.1 Sigfox

Sigfox je francouzská společnost založená v roce 2010 s vizí propojit každý předmět v našem fyzickém světě. Společnost vytvořila stejnojmennou bezdrátovou technologii určenou ke spojení nízkopříkonových zařízení, jako jsou například chytré hodinky, měřidla atd., které mají být neustále spuštěné, a přitom vysílají minimální množství dat. Zařízení jsou schopna komunikovat na velkou vzdálenost (10–50 kilometrů), záleží na typu krajiny [11].

1.3.2 LoRaWAN

LoRaWAN (Long Range Wide Area Network) neboli zkráceně LoRa, je síťový protokol LPWAN (Low Power Wide Area Network) určený k bezdrátovému připojení zařízení napájených z baterií k internetu v regionálních, národních nebo globálních sítích. Je zaměřený na klíčové požadavky IoT, jako je obousměrná komunikace, zabezpečení end-to-end, mobilita a lokalizační služby. Architektura LoRaWAN obsahuje brány (Gateways), které předávají zprávy mezi zařízeními a centrálním síťovým serverem. Brány jsou připojeny k síťovému serveru prostřednictvím standardních IP připojení a fungují jako transportní most [12].

V různých zemích se pro bezdrátovou komunikaci LoRaWAN používají různá frekvenční pásma, která jsou nastavena v závislosti na předpisech a normách o bezdrátové komunikaci pro danou zemi. V Evropě se používá LoRaWAN 868 MHz s frekvenčním rozsahem od 863 MHz do 870 MHz. V Severní Americe se používá pásmo 915 MHz s rozsahem od 902 MHz do 928 MHz. V Asii se používá pásmo 923 MHz. V Austrálii se používá pásmo 915 MHz, podobně jako v Severní Americe. V Indii se používají dvě frekvenční pásma, rozmezí jsou od 865 MHz do 867 MHz a od 923 MHz do 925 MHz. V Jižní Americe se používá pásmo v rozmezí od 915 MHz do 928 MHz [12].

1.3.3 NB-LTE

NB-LTE neboli NB-IoT (Narrowband IoT, Úzkopásmový IoT) je rádiová technologie nasazená v mobilních sítích. Má velké pokrytí, nízké náklady, dlouhou výdrž baterie a rozmanitost. NB-IoT využívá šířku pásma 200 kHz, což znamená, že má více kanálů než LTE nebo Wi-Fi. Technologie využívá všech bezpečnostních prvků a prvků ochrany soukromí mobilních sítí. Například podporu důvěrnosti identity uživatele, ověřování entit, důvěrnosti, integrity dat a identifikace mobilního zařízení. NB-IoT je technologie založená na standardech LPWAN, která byla vyvinuta s cílem umožnit širokou škálu nových zařízení a služeb internetu věcí [13].

1.3.4 Bluetooth LE 4

Bluetooth je obsažen v mnoha zařízeních. Původní verze Bluetooth byly energeticky náročné, z tohoto důvodu byla potřeba zlepšit energetickou účinnost připojení Bluetooth a bezdrátových zařízení. V roce 2010 byla vydána verze Bluetooth 4.0. Rychlost zůstala stejná jako u předchozí verze, tedy 3 Mb/s, zdvojnásobil se však dosah na 60 metrů. Vývoj Bluetooth se rozdělil do dvou větví, klasická verze a Bluetooth Low Energy. Low Energy je přizpůsoben pro tzv. dávkovanou komunikaci, ta je vhodná pro zařízení IoT, protože aplikace IoT potřebují pravidelně posílat pouze malé bity dat [14].

Díky nejnovější verzi Bluetooth 4.2, je vylepšené automatické řízení spotřeby energie na základě energetických plánů, což zvýšilo účinnost a výdrž baterie bezdrátových zařízení [15].

Jedná se o spolehlivý a bezpečný bezdrátový standard. BLE Secure Connection zahrnuje šifrování schválené Federálními standardy pro zpracování informací (FIPS), které poskytuje možnost šifrovat spojení mezi zařízeními podle kryptografie s eliptickou křivkou veřejného klíče P-256. Bluetooth 4.2 využívá profil IPSP (Internet Protocol Support Profile) k připojení

snímačů Bluetooth k internetu za účelem odesílání a přijímání přenosů přes brány prostřednictvím protokolu IPv6 [15].

1.3.5 Bluetooth LE 5

Verze Bluetooth 5 je vydaná v roce 2016. Je to vylepšení předchozích technologií Bluetooth Low Energy. Poskytuje dvojnásobnou rychlost oproti technologii Bluetooth 4.0 Low Energy a zvýšený dosah až 240 metrů. S vyšší rychlostí a větším dosahem to vypadá, že Bluetooth 5 bude vyžadovat více energie. Díky chytrým změnám ve způsobu modulace signálů a lepšímu využití frekvenčního spektra však Bluetooth 5 ve skutečnosti spotřebovává méně energie, v některých případech až 2,5krát méně. Může se také pochlubit osminásobnou kapacitou zpráv, což dále zvyšuje jeho účinnost [14].

Zásadní výhoda oproti jakékoli jiné technologii vyplývá ze skutečnosti, že Bluetooth zužitkovává informace z předchozích verzí protokolu Bluetooth, a hlavně těží z obliby na trhu. Proto jsou i v dnešní době rozhraní Bluetooth Low Energy integrovány do jakéhokoli mobilního telefonu, tabletu, nositelného zařízení atd. a nabízejí tak obrovské přednosti v ovládnutí relativních trhů [15].

1.3.6 Wi-Fi

Wi-Fi podporuje mnoho aplikací a zařízení. Wi-Fi Alliance vlastní a řídí certifikaci "Wi-Fi Certified", která certifikuje všechny produkty založené na standardu 802.11. Nabízí výrazně vyšší rychlost přenosu dat než jakýkoli jiný bezdrátový standard, ale jeho vysoká spotřeba energie představuje notoricky známý problém pro životnost baterií v mobilních zařízeních. Wi-Fi Alliance představila v lednu 2016 Wi-Fi HaLow jako rozšíření standardu 802.11ah, který je konkurenceschopný s nízkoenergetickým Bluetooth, ale s větším dosahem. Wi-Fi HaLow bezdrátový protokol pracuje v pásmu 900 MHz, ale zachovává podporu stávajících přístupových bodů v pásmech 2,5 a 5 GHz [15][16].

1.3.7 GSM

GSM je zkratka pro Global System for Mobile communication. Jedná se o standard mobilní komunikace, který se používá k přenosu mobilních hlasových a datových služeb. Byl vyvinut jako náhrada analogových sítí první generace (1G) a připravil půdu pro síť druhé generace (2G). GSM je hodně používaná technologie v IoT pro svou jednoduchost, cenovou dostupnost a přístupnost. Standardy GSM byly navrženy s ohledem na mobilní telefony, nikoli na IoT. Dnes se na mobilní síť spoléhají miliardy dalších zařízení, jako jsou parkovací automaty, průmyslová zařízení, zábavní systémy v automobilech, bezpečnostní systémy a

využívají je jiným způsobem než telefony. V důsledku toho vznikly specializované sítě, které se zabývají moderním prostředím mobilního připojení [17].

1.3.8 LTE

LTE (Long Term Evolution) je internetová komunikační technologie pro mobilní sítě. Jedná se o nástupce GSM. Oproti starším 3G a 2G sítím má vysokou rychlost jak stahování, tak nahrávání. LTE není to stejné jako 4G, je to mezistupeň mezi 3G a 4G. Rychlost LTE je kolem 100 Mbit/s při stahování a 50 Mbit/s při nahrávání, to je desetkrát méně, než je u 4G [18].

1.3.9 LTE-M

LTE-M (Long Term Evolution Machine Type Communication) je speciálně navržena pro IoT. Jedná se o rozsáhlou síť s nízkou spotřebou energie (LPWAN), specializovanou kategorií bezdrátových úzkopásmových sítí, která je navržena tak, aby maximalizovala pokrytí, snížila náklady a spotřebu energie pro zařízení internetu věcí. V porovnání s NB-IoT jsou datové rychlosti LTE-M více než desetkrát vyšší. Jeho latence je přibližně 10 až 100krát nižší a má mnohem větší pokrytí, protože využívá stávající infrastrukturu LTE [19].

2 HARDWAROVÉ PLATFORMY PRO IOT

2.1 Raspberry Pi

Raspberry Pi vyrábí jednodeskové počítače, desky a čipy s mikrokontrolery a stavebnice osobních počítačů. Výrobky Raspberry Pi, které jsou známé a ceněné pro svou přístupnost, jsou oblíbené mezi mladšími uživateli, ale hodí se i pro vývojové projekty IoT. Existuje spousta aktivních online komunit, kde se vývojáři mohou dělit o své úspěchy, klást otázky a spolupracovat na projektech, což napomáhá rychlému růstu vývoje nových řešení IoT [20].

Desky Raspberry Pi jsou vybaveny několika možnostmi vstupu dat a vestavěnou pamětí. Produkty společnosti mají dobrou pověst díky své spolehlivosti. V závislosti na produktu se nabízí mnoho možností ukládání a zpracování dat, přičemž je možné přidat další. K deskám se lze připojit pomocí USB nebo ethernetového kabelu a většina z desek umožňuje také bezdrátové připojení pomocí Wi-Fi nebo Bluetooth. Mobilní připojení lze doplnit pomocí mobilního adaptéru Raspberry Pi a mobilního modemu. Počítače Raspberry Pi postrádají funkce správy napájení, jako je režim spánku nebo pozastavení. To je nevýhodou pro bateriová zařízení IoT [21].

Příkladem kompaktního zařízení je deska Raspberry Pi Zero 2, která je znázorněna na obrázku (Obrázek 3) [22].



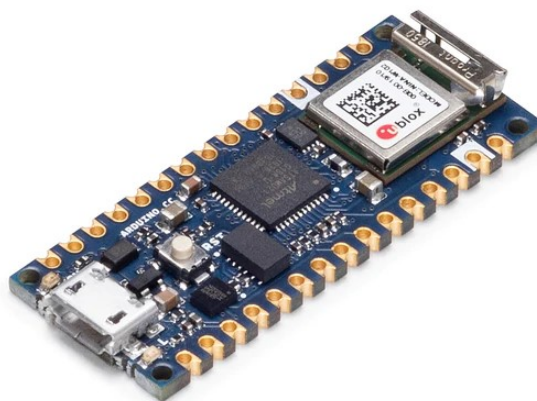
Obrázek 3. Raspberry Pi Zero 2 [22]

2.2 Arduino

Společnost Arduino je známá jako tvůrce prototypovacích produktů. Jelikož má Arduino velkou komunitu, je k dispozici mnoho zdrojů podpory spolu s komunitní podporou

podobnou té, kterou lze nalézt u produktů Raspberry Pi. Arduino také navrhuje vlastní open-source software, který lze s jeho produkty používat [21][23].

Arduino obvykle ukládá data na SD kartu nebo je odesílá do připojeného počítače. Pomocí Wi-Fi lze odesílat data do cloudu. Většina kusů má vestavěnou paměť RAM, která se využívá k dočasnému ukládání dat. Produkty Arduino umožňují mnoho možností bezdrátového připojení, včetně Wi-Fi a Bluetooth. Některé desky, jako například desky MKR, obsahují podporu pro NB-IoT, LoRaWAN a SigFox. Jen málo produktů Arduino má možnost režimu spánku pro úsporu energie, včetně pohotovostního režimu, režimu nečinnosti a úsporného režimu [21]. Příkladem desky s možností režimu spánku je deska Arduino Nano 33 IoT, která je zobrazena na obrázku (Obrázek 4) [23].



Obrázek 4. Arduino Nano 33 IoT [23]

2.3 Adafruit

Hardware společnosti Adafruit umožňuje snadný vstup do oblasti DIY (Do It Yourself, v překladu Udělej si sám) elektroniky. Společnost prodává hardwary pro IoT od jiných společností spolu s vlastní řadou produktů Adafruit Feather. Feather umožňuje vytvářet projekty IoT všeho druhu, včetně vzdáleného monitorování a ovládání, sledování domácích zvířat a detekce kvality ovzduší. Deska je ideální pro vývoj embedded systémů (embedded, v překladu vestavěný) a jedná se o nízkoenergetický počítač, který lze pro úsporu energie přepnout do režimu spánku [21][24].

Vývojové desky řady Feather umožňují snadné zaznamenávání dat a lze k nim přidávat různé senzory pro sběr všech typů dat. V závislosti na modelu obsahují desky různá integrovaná úložiště a některá mají místa pro přidání SD karet k dalším typům ukládání. Desky mají možnost bezdrátového připojení pomocí Wi-Fi, Bluetooth, LoRaWAN a rádiového

připojení. V minulosti zákazníkům nabízeli možnost mobilního připojení u modelu Adafruit Feather 32u4 FONA, ale výroba již skončila. Tato deska je na obrázku (Obrázek 5). Napájení desek je možné pomocí kabelu nebo pomocí lithiové baterie. Hardware lze nastavit tak, aby přešel do úsporného režimu, zejména při připojení k Wi-Fi, aby se zabránilo vyšší spotřebě energie [21].



Obrázek 5. Adafruit Feather 32u4 FONA [25]

2.4 Particle

Nabídka hardwaru společnosti Particle zahrnuje průmyslové moduly pro připojení a vývojové sady vytvořené pro vývojáře IoT. Společnost má také vlastní platformu pro vytváření a správu zařízení IoT. Jejich startovací sada obsahuje vývojovou desku Argon s podporou Wi-Fi a několik modulů, které shromažďují různé typy informací, například světelný a ultrazvukový dosah. Desky obsahují vestavěné paměti pro ukládání dat a lze k nim snadno přidat SD karty, díky kterým je snadné posílat data do různých cloudů. Particle podporují bezdrátové připojení Wi-Fi, mobilních sítí (2G/3G/LTE) a Bluetooth LE. Na zařízení lze nastavit režim spánku a probuzení podle potřeby [21].

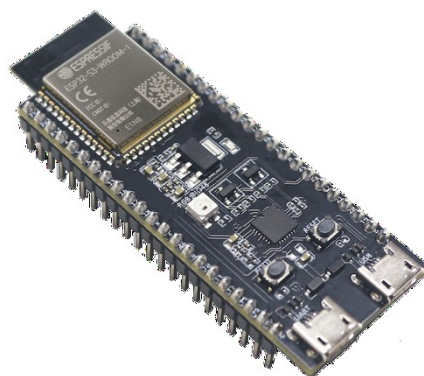
Deska Argon je zobrazena na obrázku (Obrázek 6) [26].



Obrázek 6. Particle Argon
IoT [26]

2.5 Espressif

Společnost Espressif vyrábí mikrokontrolery, které mají velkou oblibu mezi vývojáři IoT. Přístup all-in-one umožňuje pořídit si jeden produkt a získat řadu funkcí, bez nutnosti kupovat tolik doplňků, jak to vyžadují desky jiných výrobců. Například základní vývojová deska Espressif ESP32-S3-DevKitC-1 obsahuje vestavěný modul pro připojení Wi-Fi a Bluetooth. Tato deska je na obrázku (Obrázek 7). Společnost vyrábí čipy s vysokou úrovní integrace, které pomáhají navrhovat zařízení IoT, jež se snadno vyrábějí a nasazují [21][27].



Obrázek 7. Espressif ESP32-S3-
DevKitC-1 [27]

Desky Espressif jsou určeny k získávání dat ze senzorů a jejich přenosu na přídatnou SD kartu nebo do cloudu prostřednictvím USB kabelu nebo bezdrátového připojení. Desky podporují pouze bezdrátová připojení Wi-Fi a Bluetooth. Espressif klade důraz na

hardwarová řešení IoT s nízkou spotřebou energie a většina hardwaru obsahuje několik možností režimu spánku (lehký spánek, spánek modemu a hluboký spánek) [21].

2.6 Pycom

Společnost Pycom vytvořila ekosystém produktů, jehož cílem je podpora vývojářů IoT realizovat jejich nápady a dostat je rychle na trh. Nabídka hardwaru zahrnuje vývojové desky, moduly s nízkou spotřebou energie a rozšiřující desky. Společnost vytvořila mnoho softwaru pro práci s jejich hardwarem. Podporuje řadu síťových možností a nabízí velkou komunitní podporu [21][28].

Produkty Pycom jsou navrženy pro IoT, a proto existuje mnoho možností sběru a řízení dat, včetně nástroje pro sběr dat integrovaného s jejich softwarem. Rozšiřující desky obsahují senzory, které detekují světlo, vlhkost, teplotu, barometrický tlak a další. Data se obvykle ukládají na SD karty nebo do připojeného počítače. Desky podporují řadu možností připojení, včetně Wi-Fi, Bluetooth, LoRaWAN, Sigfox, LTE a NB-IoT. Deska FiPy podporuje všechna síťová připojení, je zobrazena na obrázku (Obrázek 8) [28]. Část produktů Pycom může fungovat v úsporném režimu, který umožňuje zvolit, jak často se bude zařízení připojovat k síti a jak dlouho zůstane připojené, tím se omezí spotřeba energie a pomáhá tak regulovat tok dat [21].



Obrázek 8. PYCOM FiPy [28]

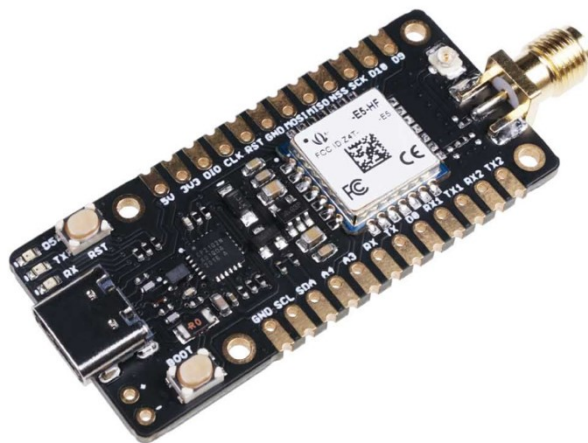
Společnost Pycom Ltd. je od 16. září 2022 pod správou společnosti Season Group, která má sídlo v Hong Kongu a pobočky v Číně, Malajsii, Mexiku a ve Spojeném království. Season Group je společnost, jež mezinárodně poskytuje služby v oblastech designu a výroby elektroniky [29].

2.7 Seeed studio

Společnost Seeed Studio je předním poskytovatelem hardwaru pro trh. Díky rozsáhlému dodavatelskému řetězci v Šen-Čenu v Číně nabízí společnost služby přizpůsobení výroby, od jednoho až po více než desetitisíce kusů. Společnost Seeed sídlí v Šen-Čenu, z kterého řídí globální trh. Má pobočky v USA a Japonsku. Seeed nabízí řešení all-in-one, které vývojářům umožňuje získat jediný produkt s řadou funkcí bez nutnosti dalšího příslušenství. Například systém Grove je standardizovaná sada konektorů a senzorů, která urychluje a zjednodušuje sestavování zařízení IoT [30][31].

Produkty, jako je vývojová deska Wio LTE, jsou vybaveny integrovaným připojením Wi-Fi a Bluetooth, což usnadňuje připojení k internetu. Seeeduino je další hardwarový produkt, který je vybaven řadou vestavěných senzorů a komponentů. Seeed Studio také nabízí brány LoRaWAN, které umožňují zařízením IoT připojení k internetu prostřednictvím sítě LoRaWAN [31].

Příkladem zařízení, které se připojuje do sítě LoRaWAN, je deska Wio-E5 mini. Je zobrazena na obrázku (Obrázek 9) [32].



Obrázek 9. Wio-E5 mini Dev Board [32]

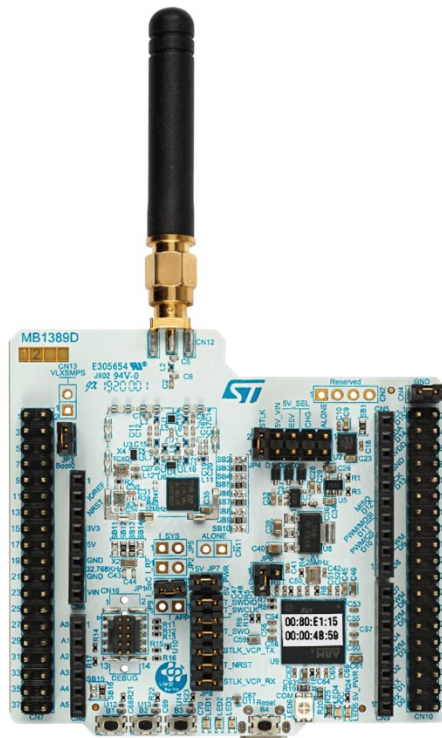
Řada vývojových desek LinkIt Smart je navržena speciálně pro vytváření aplikací IoT a nabízí vestavěné připojení Wi-Fi a Bluetooth. BeagleBone Green je výkonná vývojová deska, která je vybavena vestavěným konektorem Grove a je kompatibilní s deskou BeagleBone Black [31].

2.8 STMicroelectronics

STMicroelectronics je globální nezávislá společnost a lídr ve vývoji a dodávkách polovodičových technologií napříč různými aplikacemi v mikroelektronice. Díky bezkonkurenční kombinaci odborných znalostí v oblasti křemíku, síle výroby a pozici strategických partnerů, stojí společnost STMicroelectronics v čele technologie System on Chip a její produkty hrají klíčovou roli při realizaci nejnovějších projektů. Produktová řada desek STM32 Nucleo umožňuje vývojářům vyzkoušet nové nápady a rychle vytvořit prototypy s jakýmkoliv MCU STM32 [33][34].

Desky STM32 Nucleo sdílejí stejné konektory a lze je snadno rozšířit o další příslušenství. Desky integrují ladící programátor STLINK, čímž odpadá potřeba použití dalšího hardwaru pro ladění. Jednou z nabídek je vývojová deska NUCLEO-WL55JC1 [34].

NUCLEO-WL55JC1 je kompaktní a levná vývojová deska, která je navržena tak, aby pomohla vytvářet prototypy a vyhodnocovat bezdrátové aplikace IoT založené na technologii LoRaWAN. Deska má mikrokontroler s velmi nízkou spotřebou a vysílač LoRaWAN. Deska je vybavena rozšiřujícími hlavičkami Arduino™ Uno V3 a ST morpho, které umožňují snadné připojení k široké škále externích senzorů. Obsahuje integrovaný ladící programátor STLINK, který umožňuje snadné ladění ve vývojovém prostředí STM32CubeIDE, bezplatného integrovaného vývojového prostředí pro mikrokontrolery STM32. Tato deska je zobrazena na obrázku (Obrázek 10) [34][35].



Obrázek 10. STM32 Nucleo-64 development board [35]

3 PORTÁLOVÉ SYSTÉMY PRO IOT

IoT platformy fungují jako prostředník (Middleware – software umožňující jeden nebo více druhů komunikace nebo připojení mezi dvěma nebo více aplikacemi v síti [36]) mezi světy fyzických zařízení a světem použitelných informací. Poskytují připojení a správu hardware zařízení a dat. IoT platformy jsou navrženy tak, aby zařídily zabezpečenou komunikaci mezi zařízeními a aplikacemi. Urychlují a usnadňují vývoj a nasazení aplikací IoT. Díky využití standardů a dobře zdokumentovaných rozhraní API a SDK platformy umožňují integrování zařízení, aplikací a služeb do systému IoT a poskytují tak flexibilitu při přizpůsobování a nahrazování komponent [4].

Platformy IoT mají mnoho vlastností, funkcí, které jsou různé a liší se podle oblastí použití dané platformy. Význam každé z vlastností závisí na požadavcích. Například u chytré domácnosti je méně důležitá analýza historických dat ze senzorů. Důležitá je analýza dat získaných v reálném čase a nutná okamžitá reakce na tyto data. Klíčovými vlastnostmi platformy jsou správa zařízení, protokoly datové komunikace, datové úložiště, pravidla a analytika, rychlý vývoj a nasazení aplikací, integrace a bezpečnost a náklady na vývoj a údržbu [4].

K dispozici jsou stovky platformy IoT od různých dodavatelů. Některé platformy jsou úzce specializované, jen průmyslový IoT nebo jen IoT v domácnosti. Jiné platformy se zaměřují pouze na podmnožiny funkcí, které mohou být pro systém IoT potřebné, tyto platformy se například zaměřují výhradně na analýzu. Další platformy jsou univerzální, lze je použít pro vývoj aplikací IoT v jakékoliv oblasti. Obsahují funkce pro analýzu dat starších i získaných v reálném čase, správu zařízení, ověřování a autorizaci, bezpečné ukládání dat atd. [15].

3.1 Kaa

Kaa byla v původní verzi bezplatnou open-source IoT platformou, ve verzi Enterprise je bezplatná pouze v případě na vyzkoušení po dobu 14 dnů. Kaa je zveřejněna pod licencí Apache 2.0, což umožňuje její vlastní hostování. Obsahuje rozhraní REST API a SDK pro jazyky Java, C++ a C s funkcemi pro správu zařízení, sběr dat, oznámení, vyrovnávání zátěže a analýzu dat. Je vhodná pro aplikace IoT podnikové úrovně i pro domácí projekty a experimenty IoT [37].

Kaa platformu lze aplikovat na jakýkoliv IoT projekt. Poskytuje řadu funkcí umožňující vývojářům vytvářet pokročilé aplikace pro chytré produkty. V cloudu Kaa lze jednoduše spravovat připojená zařízení, analyzovat data, vytvářet diagramy a mnoho dalšího. Kaa jako

celek je založena na několika mikro službách, to znamená, že lze přizpůsobit každou funkci samostatně: přidání, odebrání a nahrazení kterékoli služby. Využívá služeb třetích stran např. Kibana pro zobrazení analyzovaných dat nebo MinIO k ukládání dat [38].

Po zaregistrování a spuštění cloudu jsou k dispozici všechny funkce IoT, které jsou potřebné pro typickou aplikaci IoT – sběr dat, analýza, správa zařízení, řídicí panely. Lze použít i rozhraní API k integraci Kaa do vlastních modulů a aplikací. Bezplatná verze je limitována pro pět zařízení a pouze na 14denní vyzkoušení [38].

3.2 ThingSpeak

ThingSpeak je bezplatná open-source IoT platforma zaměřená na analýzu. Ze zařízení lze odesílat data přímo do ThingSpeak™ cloudu. Umožňuje vytvářet okamžité vizualizace a analýzy reálných dat a odesílat upozornění emailem nebo pomocí webových služeb jako jsou Twitter a Twilio [39].

ThingSpeak využívá aplikaci MATLAB® pro analýzu a vizualizaci umožňující zkoumat a prohlížet data z kanálů. V ThingSpeak lze psát a spouštět MATLAB kód. ThingSpeak umožňuje interakci se sociálními médii, webovými službami a zařízeními. Výhodou je, že nabízí vytvoření IoT systémů bez nastavování serverů a vývoje webového software [39][40].

ThingSpeak je pouze cloudová IoT platforma. Využívá kanály k ukládání dat odesílaných z aplikací nebo zařízení. Jeden kanál je primárně určen pro jedno zařízení. Po vytvoření nového kanálu lze do něj a z něj odesílat a načítat data. Samostatné kanály lze zveřejnit a sdílet tak data. Pomocí rozhraní REST API, jako jsou GET, POST, PUT a DELETE, je možné vytvořit kanál a aktualizovat existující kanál, vymazat kanál a odstranit kanál. Lze také použít metodu MQTT pro aktualizaci kanálu a pro příjem zpráv [40].

3.3 ThingsBoard

ThingsBoard je open-source IoT platforma, která umožňuje rychlý vývoj, správu a rozšiřování projektů IoT. Cílem je poskytnutí cloudového nebo lokálního řešení IoT, které umožňuje infrastrukturu na straně serveru aplikace IoT. S ThingsBoard lze shromažďovat a vizualizovat data ze zařízení, analyzovat příchozí telemetrii a spouštět alarmy pomocí komplexního zpracování událostí, ovládat zařízení pomocí volání vzdálených procedur (Remote procedure calls – RPC, to je technika komunikace, která může jeden program použít k vyžádání služby od programu v jiném zařízení [42]). ThingsBoard dále umožňuje

navrhování dynamických a responzivních ovládacích panelů a posílání dat o zařízení do jiných systémů. Platforma využívá mikro služby i mikro služby třetích stran [41].

ThingsBoard nabízí lokální i cloudové řešení. Cloudové je součástí Professional Edition a není tedy k dostání bezplatně [43].

3.4 Apache StreamPipes

StreamPipes platforma je určena pro průmyslovou analýzu. Platforma je vytvořena tak, aby ji mohli používat techničtí i netechničtí uživatelé, pomáhá shromažďovat a analyzovat data. StreamPipes využívá algoritmy strojového učení k provádění pokročilé analýzy. Hlavní výhodou pro netechnické uživatele je jednoduché a intuitivní webové rozhraní a grafický editor [44].

Platforma využívá i služby třetích stran. Pro nabídky těchto služeb má vytvořený modul nazvaný StreamPipes Connect. StreamPipes Connect nabízí různé adaptéry a pro běžné komunikační protokoly a některé specifické senzory. Kromě shromažďování dat nejen archivovaných, ale i získaných v reálném čase, nabízí i způsoby předzpracování dat, integruje inteligentní náhledy a v neposlední řadě nabízí nástroj pro odhad náležitého schématu, který sleduje příchozí data a doporučuje schémata [45].

3.5 Google Cloud IoT Core

Google IoT Core je platforma, která umožňuje snadné a bezpečné připojení, správu a příjem dat z mnoha zařízení. IoT Core v kombinaci s dalšími službami Google Cloud poskytuje kompletní řešení pro sběr, zpracování, analýzu a vizualizaci dat IoT v reálném čase, které podporuje zvýšení efektivity a předvídání problémů díky nástrojům strojového učení. IoT Core pracuje na bez serverové infrastruktuře společnosti Google, která se automaticky rozšiřuje v reakci na změny v reálném čase a dodržuje přísné bezpečnostní protokoly podle průmyslových standardů, které chrání data [46].

Pomocí IoT Core Správce zařízení lze bezpečně spravovat a konfigurovat zařízení. Tento správce určuje identitu zařízení a poskytuje mechanismus pro ověřování zařízení při připojení. Udržuje také logickou konfiguraci každého zařízení a lze jej použít ke vzdálenému ovládání zařízení z cloudu [46].

IoT Core je bezplatný do chvíle, dokud objem přenesených dat nebude vyšší než 250 MB za měsíc. Poté se připlácí za každý přenesený MB [47].

3.6 Amazon Web Services IoT Core

AWS (Amazon Web Services) IoT je vysoce škálovatelná platforma IoT. Obsahuje SDK, autentizaci a autorizaci, registr zařízení, bránu zařízení, která komunikuje se zařízeními pomocí MQTT, WebSockets nebo HTTP, a tzv. Rules Engine [4].

Rules Engine je program, který na datech provádí pravidla, a pokud některá podmínka vyhovuje, provede odpovídající akce. Rules Engine se integruje se stávajícími službami AWS, jako je DynamoDB [48].

Zajímavou funkcí AWS IoT jsou "stíny zařízení" (Device Shadow). Device Shadow je služba, která přidává stíny k IoT zařízení. Umožňují tak zpřístupnit stav zařízení aplikacím a dalším službám bez ohledu na to, zda je zařízení připojeno k AWS IoT, nebo ne. Zařízení IoT mohou mít více stínů, takže řešení IoT má více možností připojení zařízení k jiným aplikacím a službám. Device Shadow je tedy trvalá virtuální verze každého zařízení včetně posledního známého stavu zařízení [49].

Bezplatná verze AWS je platná pouze po dobu 12 měsíců od založení uživatelského profilu. Po uplynutí bezplatné doby, nebo pokud se přesáhne hranice určená pro bezplatné využívání, se začne počítat klasická cena AWS IoT Core. Cena se počítá v řádu jednoho dolaru za jeden milion zpráv, v Evropě za 0.08 dolaru. Například za jedno zařízení v regionu Evropy, které odešle zprávu každou minutu, se za rok zaplatí 0.042 dolaru.

Rovnice, kterou se vypočítá roční cena za zařízení vypadá následovně:

$$v = \frac{p}{1000000} * t$$

v	roční cena (\$)	
p	cena za milion zpráv (\$)	0.08 \$
1000000	představuje 1 milion zpráv	
t	počet zpráv za rok	(365 dní * 24 hodin * 60 minut = 525600)

[50].

3.7 IBM Watson IoT

IBM Watson IoT je vyspělá, pro vývojáře přívětivá platforma IoT postavená na platformě IBM Cloud, která se vyznačuje silnými stránkami v oblasti analýzy v reálném čase a kognitivních výpočtů. Platforma také podporuje správu zařízení, autentizaci a autorizaci,

bezpečné ukládání dat a komunikaci, standardní komunikační protokoly, jako jsou MQTT a HTTPS, rozhraní REST API a sady SDK [4].

Watson IoT je cloudová platforma, která je navržena pro zjednodušení práce se zařízeními IoT. Nabízí webový řídicí panel s jednoduchým uživatelským rozhraním, pro přidávání a správu zařízení a monitoring používání. Pomocí správy zařízení lze provádět různé akce zařízení, jako například restarty či aktualizace firmwaru, získat diagnostiku a metadata ze zařízení, nebo provádět hromadné přidávání a odebírání zařízení [51].

K připojování zařízení a aplikací se používá protokol standardu MQTT. MQTT je navržen pro efektivní výměnu dat do zařízení a ze zařízení v reálném čase. MQTT s TLS se používá pro zabezpečenou komunikaci mezi zařízeními IoT a službami IBM [51].

3.8 Microsoft Azure IoT Suite

Microsoft Azure IoT Suite je další komplexní platformou IoT, která zahrnuje podporu správy zařízení a Device Twins (podobné jako Device Shadow u AWS IoT Core) pomocí Azure IoT Hub, standardní komunikační protokoly včetně MQTT, AMQP a HTTP, zabezpečené úložiště a analytické nástroje podporující prediktivní analýzu a vizualizaci dat [4].

Azure IoT je určeno pro různé průmyslové potřeby. Lze jej použít od výroby přes dopravu až po maloobchod. Poskytuje řešení pro vzdálené monitorování, prediktivní údržbu, chytré prostory a propojené produkty [52].

V cloudovém systému Microsoftu lze získat přístup k vytvořeným několika stovkám řešení problémů. Aplikacím zaměřeným na fungování s daty v reálném čase poskytuje automatické operace řešené pomocí AI. Vše lze ovládat a nastavovat z webového prostředí Azure Microsoft [53].

3.9 The Things Network

Platforma The Things Network (TTN) poskytuje sadu otevřených nástrojů a globální otevřenou síť pro vytvoření aplikací IoT. TTN je založen na práci s komunikační sítí LoRaWAN. Umožňuje uživatelům vytvářející IoT aplikace, aby nemuseli pořizovat LoRaWAN brány, ale mohou se tak připojit na již registrované TTN LoRaWAN brány. Pokud si vývojář zakoupí a nastaví vlastní LoRaWAN bránu, lze ji zaregistrovat do sítě TTN, kde může zvolit, zda bude brána veřejně dostupná nebo soukromá. Pokud je veřejná, mohou ji používat i jiní uživatelé, ale pokud je soukromá, tak ji mohou využívat pouze uživatelé s API klíčem [54].

The Things Stack (TTS) je síťový server LoRaWAN, který umožňuje připojení, správu a monitorování zařízení, bran a aplikací. Účelem je zajištění bezpečnosti a spolehlivosti v rámci komunikace, přeposílání dat mezi branami, zařízeními a aplikacemi. TTS je open-source projekt spravovaný společností The Things Industries (TTI), která stojí i za TTN. Rozdílem mezi TTN a TTI je komunita a komerčnost [54][55].

TTS nabízí i cloudové úložiště, ale v komunitní verzi je omezen pouze na dobu 24 hodin. Nabízí integrace s AWS IoT, Azure IoT, dále pomocí webhooks a MQTT [55].

Společnosti TTI a Microchip vyvinuly bezpečnostní řešení pro LoRaWAN, které umožňuje bezpečné poskytování klíčů a bezpečné kryptografické operace pomocí zabezpečených prvků. TTS podporuje bezpečnou komunikaci. Využívá hardwarový zabezpečený prvek společnosti Microchip (ATECC608B-TNGLORA) a server The Things Join Server k bezpečnému ukládání kořenových klíčů zařízení LoRaWAN. Vzhledem k tomu, že server The Things Join Server je síťově agnostický, mohou koncoví uživatelé používat jakoukoli distribuci TTS nebo jakýkoli síťový server LoRaWAN kompatibilní se standardy [56].

3.10 AllThingsTalk Maker

AllThingsTalk Maker je platforma IoT určená pro vývojáře. Jedná se o cloudovou platformu, která poskytuje sadu nástrojů a služeb pro připojení a správu zařízení IoT, sběr a analýzu dat a vytváření vlastních aplikací. Nabízí sadu rozhraní API a sad SDK, které mohou vývojáři využít k rychlému a snadnému vytváření vlastních IoT aplikací a integrací [57].

Jednou z klíčových funkcí platformy jsou možnosti správy zařízení. Platforma umožňuje uživatelům registrovat a spravovat zařízení IoT, konfigurovat jejich nastavení a sledovat jejich výkon v reálném čase. Uživatelé mohou vytvářet vlastní ovládací panely a vizualizace pro zobrazení a analýzu svých dat a nastavovat upozornění a oznámení, aby byli informováni o stavu zařízení a událostech. Platforma také poskytuje řadu možností připojení, včetně podpory populárních protokolů IoT, jako jsou MQTT a HTTP. AllThingsTalk Maker lze integrovat s populárními hardwarovými platformami IoT, jako jsou Arduino a Raspberry Pi. Vývojáři tak mohou snadno připojit svá zařízení a ihned začít shromažďovat data [57].

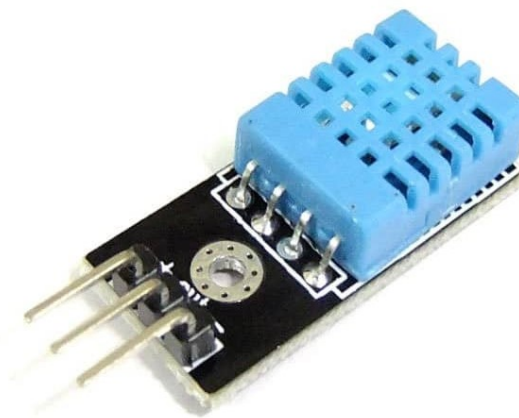
4 SENZORY A MĚŘIDLA PRO MĚŘENÍ TEPLoty V BYTOVÝCH DOMECH

Teplou vodu v bytových domech zajišťují převážně centrální teplárny. Z centrální teplárny vede potrubím ohřátá kapalina do tepelného výměníku, který se nachází ve sklepeních domů. Ohřátá kapalina z teplárny je voda, jejíž teplota se v závislosti na ročním období pohybuje mezi 70 až 150 °C. Tepelný výměník je využíván k oddělení užitkové vody od topné kapaliny, aby nedošlo ke styku těchto kapalin, tento proces se nazývá nepřímý ohřev. Teplota užitkové vody vycházející z výměníku se průměrně pohybuje kolem 52 °C - [58][59][60].

4.1 DHT11

Senzor DHT11 je vybaven čidlem teploty a vlhkosti s kalibrovaným digitálním výstupním signálem. Pomocí digitálního snímání signálu a technologie snímání teploty a vlhkosti zajišťuje vysokou spolehlivost, odolnost a dlouhodobou stabilitu. Tento snímač obsahuje odporový typ měření vlhkosti a součástku pro měření teploty typu NTC (Negative Temperature Coefficient – termistor, jehož odpor klesá se zvyšující se teplotou [62]) a připojuje se k vysokorychlostní síti 8bitovým mikrokontrolérem, který nabízí rychlou odezvu, ochranu proti rušení a nízkou cenu. Každý senzor je pečlivě kalibrován v laboratoři [61].

Senzor má čtyři konektory, první je napájecí konektor, používá se stejnosměrné napětí o velikosti 3 V–5.5 V. Druhým konektorem se přenáší data. Třetí konektor je nulový a čtvrtý je uzemňovací. Senzor je na obrázku (Obrázek 11) [61].



Obrázek 11. Senzor teploty a vlhkosti – DHT11 [61]

Rozsah a přesnost měření teploty a vlhkosti tohoto zařízení je v tabulce (Tabulka 1) [61].

Tabulka 1. Rozsah a přesnost měření – DHT11 [61]

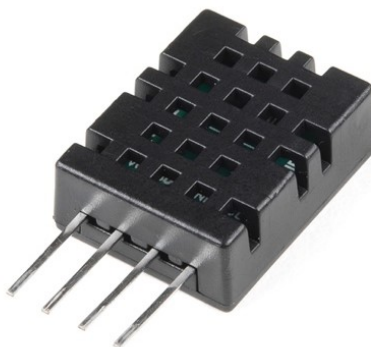
	Teplota		Vlhkost	
Rozsah měření	Min	Max	Min	Max
	0 °C	+50 °C	20 %	95 %
Přesnost	±2 °C		±5 %	

4.2 DHT20

Senzor DHT20 je vylepšený produkt DHT11, měřící vlhkost a teplotu, ale je vybaven výkonným polovodičovým senzorem vlhkosti na bázi křemíku. Zvýšila se tak jeho výkonnost a spolehlivost oproti předchozí verzi DHT11. Senzor DHT20 je stabilnější v prostředí s vysokou teplotou a vlhkostí, zároveň je přesnější, má rychlejší dobu odezvy a rozsah měření. Každý senzor je také pečlivě kalibrován a testován v laboratořích [63].

Senzor lze použít ve vzduchotechnice, odvlhčovačích, spotřebních výrobcích, automobilech, meteorologických stanicích, lékařství a dalších aplikacích, které potřebují detekovat a kontrolovat teplotu a vlhkost [63].

Senzor má čtyři konektory, první je napájecí konektor, používá se stejnosměrné napětí o velikosti 2.5 V–5.5 V. Druhý konektor SDA (Serial Data) se používá pro vstup a výstup dat ze senzoru. Třetí konektor je uzemňovací. Čtvrtý je SCL (Serial Clock), sloužící pro synchronizaci komunikace mezi mikroprocesorem a DHT20. Senzor je na obrázku (Obrázek 12) [63].



Obrázek 12. Senzor teploty a vlhkosti – DHT20 [63]

Rozsah a přesnost měření teploty a vlhkosti tohoto zařízení je v tabulce (Tabulka 2) [63].

Tabulka 2. Rozsah a přesnost měření – DHT20 [63]

Rozsah měření	Teplota		Vlhkost	
	Min	Max	Min	Max
	-40 °C	+80 °C	0 %	100 %
Přesnost	±0.5 °C		±3 % (při 25 °C)	

4.3 DHT21

Senzor DHT21 (jiným označením jako AM2301) je velmi podobný senzoru DHT20 a je používán k měření teploty a vlhkosti. Má lepší přesnost měření a vylepšenou technologii pro delší životnost [64].

Senzor má čtyři konektory, první je napájecí, používá se stejnosměrné napětí o velikosti 3.3 V–5.2 V. Druhý konektor SDA (Serial Data) se používá pro vstup a výstup dat ze senzoru. Třetí konektor je uzemňovací a čtvrtý je prázdný [64].



Obrázek 13. Senzor teploty a vlhkosti – DHT21 [64]

Rozsah a přesnost měření teploty a vlhkosti tohoto zařízení je v tabulce (Tabulka 3) [64].

Tabulka 3. Rozsah a přesnost měření – DHT21 [64]

Rozsah měření	Teplota		Vlhkost	
	Min	Max	Min	Max
	-40 °C	+80 °C	0 %	99.9 %
Přesnost	±0.3 °C		±3 % (při 25 °C)	

4.4 BME280

Senzor BME280 je vytvořen společností Bosch. Senzor je v prostředí využíván k měření teploty, barometrického tlaku a vlhkosti. Je řešením pro měření vlhkosti s přesností ±3 %, barometrického tlaku s absolutní přesností ±1 hPa a teploty s přesností ±1 °C. Protože se tlak mění s nadmořskou výškou, lze senzor použít také jako výškoměr s přesností ±1 metr [65].

V senzoru je zabudován regulátor napětí. Dále je možné senzor bez obav použít s logickým mikrokontrolérem v rozsahu 3 V–5 V. Senzor je na obrázku (Obrázek 14) [65].



Obrázek 14. Senzor teploty, vlhkosti a tlaku – BME280 [65]

Rozsah a přesnost měření teploty, vlhkosti a tlaku tohoto zařízení je v tabulce (Tabulka 4) [65].

Tabulka 4. Rozsah a přesnost měření – BME280 [65]

Rozsah měření	Teplota		Vlhkost		Tlak	
	Min	Max	Min	Max	Min	Max
	-40 °C	+65 °C	20 %	80 %	300 hPa	1100 hPa
Přesnost	±1 °C		±3 %		±1 hPa	

4.5 LM35DZ

LM35DZ je analogový teplotní senzor s integrovaným obvodem, jehož výstupní napětí je lineárně úměrné teplotě v Celsíích. Senzor má tedy výhodu oproti snímačům teploty v Kelvinech, protože uživatel nemusí od výstupu odečítat velké konstantní napětí. Senzor nevyžaduje žádnou externí kalibraci. LM35DZ má tři konektory. Jeden napájecí, druhý uzemňovací a třetím konektorem se přenášejí data. Senzor je na obrázku (Obrázek 15) [66].



Obrázek 16. Senzor teploty – DS18B20 [67]

Rozsah a přesnost měření teploty tohoto zařízení je v tabulce (Tabulka 6) [67].

Tabulka 6. Rozsah a přesnost měření – DS18B20 [67]

	Teplota	
Rozsah měření	Min	Max
	-55 °C	125 °C
Přesnost	±0.5 °C	

4.7 EM300-TH-868M

EM300-TH-868M je senzor pro měření teploty a vlhkosti. Soustava má zabudované NFC rozhraní pro konfiguraci a dále technologii LoRaWAN pro bezdrátovou komunikaci. Pouzdro je voděodolné s ochranou IP67 a senzor přes kryt nepoškodí ani UV záření. Měření teploty je přizpůsobeno i venkovnímu použití, lze tedy měřit i hluboko pod nulou. Při teplotách nad nulou do 70 °C má senzor přesnost měření na tři desetiny stupně Celsia. Zařízení se připevňuje na stěnu pomocí šroubů nebo samolepky. Baterie s 4000 mAh má v zařízení životnost až pět let, je možné dosadit baterii až s 8000 mAh. Měřidlo je na obrázku (Obrázek 17) [68].



Obrázek 17. Senzor teploty a vlhkosti – EM300-TH-868M [68]

Rozsah a přesnost měření teploty a vlhkosti tohoto zařízení je v tabulce (Tabulka 7) [68].

Tabulka 7. Rozsah a přesnost měření – EM300-TH-868M [68]

Rozsah měření	Teplota		Vlhkost	
	Min	Max	Min	Max
	-30 °C	+70 °C	0 %	100 %
Přesnost	±0.3 °C		±3 % (mezi 10 % až 90 %)	

II. PRAKTICKÁ ČÁST

5 NÁVRH DEMONSTRÁTORU

Pro návrh demonstrátoru je důležité zanalyzovat požadavky a následně očekávané výsledky. Je nutné zvolit vhodný hardware pro sestavení měřidel a také je důležitý správný výběr softwaru pro serverovou část IoT aplikace.

5.1 Požadavky na měření

Měření teploty bude probíhat na výměníku teplé vody a ve stoupačkách bytového domu, kde senzory teploty budou pevně připevněny na trubkách teplé vody. Naměřené hodnoty budou posílány pomocí komunikační sítě na cloudovou platformu, kde se data budou zpracovávat a následně z nich budou vytvářeny grafy.

Software část IoT platformy by měla fungovat v cloudu a umožnit bezplatnou správu pro alespoň pět zařízení. Dále by měla umět pracovat se získanými daty, porovnávat je, vytvářet grafy a tabulky.

5.1.1 Hrozba bakterií

Mezi další požadavky měření by měla být funkce sledování teplot v jednotlivých částech oběhového okruhu teplé vody v bytovém domě. Funkce by kontrolovala získaná data se známými hodnotami vhodných podmínek pro vznik škodlivých bakterií.

Nejrozšířenější bakterií v těchto systémech je legionella. Legionella je pojem, pod kterým je zahrnuto přibližně 60 typů různých bakterií, ale minimálně 20 z nich jsou zdraví nebezpečné. Tato bakterie se vyskytuje ve vlhku, převážně ve vodovodním potrubí s teplou vodou, zásobnících teplé vody a klimatizacích. K nákaze dochází nejčastěji vypitím kontaminované vody a postihuje dýchací cesty. Je nutné udržovat teplotu vody nad 45 stupňů Celsia [69].

5.2 Výběr hardwarových a softwarových částí IoT aplikace

5.2.1 Volba komunikační technologie

Jelikož jsou bytové domy postaveny převážně ze železobetonu, bude potřeba použít komunikační technologii s výrazným prostupem v zastavěných oblastech. Hlavním faktorem výběru je také nízká spotřeba s nižší přenosovou rychlostí. Nejvhodnější je použití komunikační technologie s LPWAN. Z technologií v kapitole 1.3 jsou vhodné tyto tři technologie – LoRaWAN, NB-LTE, LTE-M, protože splňují všechny požadavky pro komunikaci v zastavěných oblastech. Pro realizaci byla zvolena komunikační technologie LoRaWAN, která nejlépe vyhovuje daným podmínkám pro realizaci v bytovém domě.

Hlavním důvodem výběru této technologie bylo, že je oproti jiným komunikačním technologiím novější a byla méně známá. Nyní se dostává mezi nejpoužívanější LPWAN komunikační technologie v prostředí IoT.

5.2.2 Výběr vhodného senzoru

Na internetu lze nalézt mnoho různých senzorů a měřidel pro měření teploty. K měření teploty v bytových domech lze použít jak senzory určené do interiéru, tak i senzory určené do exteriéru. V případě měření teploty na výměníku a ve stoupačkách na trubce s teplou vodou, kde je teplota v průměru kolem 52 °C, jak bylo zmíněno v kapitole 4, je potřeba vybírat senzor vyhovující této teplotě, ale lze využít i senzory s vyšším rozsahem. Dále je důležité vybrat senzor s nejlepší přesností. Na trhu se objevují senzory s přesností v řádu od 1 °C do 0.3 °C. Pro měření v daných podmínkách jsou nejvhodnější tyto dva senzory: DHT21 a DS18B20.

Senzor DHT21, jehož rozsah měření od -40 °C do 80 °C s přesností na 0.3 °C, je více než dostačující. Umožňuje i měření vlhkosti, které lze využít k možnému doplnění aplikace.

Senzor DS18B20 má rozsah měření od -55 °C do 125 °C s přesností 0.5 °C. Tento typ byl zvolen pro realizaci demonstrátoru z důvodu lepší dostupnosti na trhu.

Další senzory DHT11, DHT20, BME280, LM35DZ a měřidlo EM300-TH-868M jsou pro potřeby měření teploty v bytových domech nevhodné.

5.2.3 Výběr vhodné hardwarové platformy

Pro použití zařízení v místech s nedostupným připojením k elektrické síti je potřeba zvolit hardware, který má možnost bezdrátového napájení – LiPo baterie nebo powerbanka. Dále by měl hardware mít vlastnost nízké spotřeby energie neboli schopnost šetření baterie – režim spánku. Z důvodu vybrané komunikační technologie LoRaWAN by zařízení mělo mít možnost komunikovat v této síti. Lze vybírat ze zařízení, které již obsahují LoRaWAN vysílač nebo je možnost LoRa modem na zařízení nainstalovat.

5.2.3.1 Hardware pro měřící zařízení

Mikročip ESP32 od výrobce Espressif je vhodný, jelikož má vlastnost snížení spotřeby a dokáže spolehlivě pracovat v extrémních podmínkách s teplotou od -40 °C do +125 °C. S mikročipem ESP32 se vyvíjela deska FiPy od společnosti PYCOM. Tato deska může fungovat v podmínkách o teplotě -20 °C do +85 °C, obsahuje LoRaWAN vysílač. Pokud by bylo potřeba zvýšit dosah signálu, je možnost pořídit rozšiřující LoRaWAN anténu.

Desku lze využít i s dalšími komunikačními technologiemi (Bluetooth LE, LTE, SigFox a Wi-Fi) pro případné rozšíření dalších funkcí zařízení. K desce lze pořídit i rozšiřující desku, ke které je možnost připojení LiPo baterie, nebo i powerbanky s MicroUSB konektorem pro napájení [70]. Ve druhé polovině roku 2022 se z důvodu nedostatku čipů přestaly vyvíjet desky FiPy. Touto skutečností bylo zvoleno pro realizaci demonstrátoru zařízení Wio-E5 mini.

Vývojová deska Wio-E5 mini je cenově dostupná. Deska je určená pro rychlé prototypování a testování malých zařízení IoT s možností přenosu na velké vzdálenosti. Dále je kompaktních rozměrů a je osazena modulem Wio-E5 STM32WLE5JC, který kombinuje RF LoRa a MCU čip do jednoho malého čipu. Deska poskytuje snadno použitelnou vývojovou platformu, která obsahuje všechny GPIO Wio-E5, včetně UART, ADC, SPI, IIC a dalších. Obsahuje dvě tlačítka pro resetování a boot. Díky podpoře protokolu LoRaWAN je možné dosáhnout přenosu na velmi dlouhé vzdálenosti a velmi nízké spotřeby energie, přičemž přenos dosahuje až 10 km. Deska je navržena tak, aby pracovala v teplotách od $-40\text{ }^{\circ}\text{C}$ do $85\text{ }^{\circ}\text{C}$ [71].

Deska s připojenou anténou je na obrázku (Obrázek 18).



Obrázek 18. Wio-E5 mini (zdroj vlastní)

5.2.3.2 *Hardware pro komunikaci se zařízeními*

Pro komunikaci s koncovými zařízeními je potřeba sestavit LoRaWAN bránu. Od společnosti PYCOM lze pořídit hardware PYGATE868 nebo PYGATE915, které umožňují

komunikaci mezi více zařízeními LoRaWAN. Číselné označení v názvu desky určuje, v jakém frekvenčním pásmu technologie LoRaWAN komunikuje. Frekvence jsou určeny pro různé země světa jinak, skrze různá frekvenční pásma v závislosti na předpisech a normách o bezdrátové komunikaci (viz. kapitola 1.3.2). V Evropě je rozsah pro bezdrátovou komunikaci LoRaWAN ve frekvenčním pásmu od 863 MHz do 870 MHz. Z tohoto důvodu se pro návrh demonstrátoru zvolila varianta PYGATE868 [72].

Desku lze napájet pomocí Ethernet s PoE, LiPo baterie nebo z adaptéru a powerbanky pomocí USB kabelu. Na desku Pygate je potřeba připojit vývojovou desku FiPy nebo WiPy, se kterým je kompatibilní. Deska WiPy je podobná desce FiPy, ale pouze s komunikační technologií Wi-Fi. Takto sestavené zařízení dokáže komunikovat s více koncovými zařízeními pomocí technologie LoRaWAN [72].

Zařízení Pygate je na obrázku (Obrázek 19).



Obrázek 19. PYGATE868 (zdroj vlastní)

5.2.4 Výběr vhodné softwarové platformy

Platforma pro serverovou část aplikace musí splňovat určité požadavky. Musí být v cloudu, bezplatná alespoň pro pět zařízení. Dále musí umět pracovat s daty, porovnávat je a vytvářet tabulky a grafy.

Pro snadné ovládání, připojení zařízení a zabezpečení dat lze vybírat z následujících platform: Kaa, ThingSpeak, ThingsBoard, Apache StreamPipes, Google Cloud IoT Core, AWS IoT Core, IBM Watson IoT, Microsoft Azure IoT, The Thing Network a AllThingsTalk Maker (viz. kapitola 3).

Z výše uvedených platform nabízejí bezplatné připojení pro alespoň pět zařízení tyto platformy: ThingSpeak a AllThingsTalk Maker.

Platforma ThingSpeak je přívětivá pro nekomerční uživatele, jelikož nabízí bezplatnou licenci, ve které lze užívat čtyř kanálů. Maximálně lze odesílat tři miliony zpráv ročně s omezením jedné zprávy za 15 sekund. Primárně je jeden kanál využíván jedním zařízením. Pro použití jednoho kanálu více zařízeními je možné posílat z každého zařízení jeho index a takovým způsobem rozlišit jednotlivé zprávy od jednotlivých zařízení v kanálu. Může zde docházet ke kolizi, kdy dojde ke zpoždění v průběhu připojování nebo komunikace, a potřebná data by v tomto případě nedorazila. Toto je důvodem, proč není pro demonstrátor vhodnou platformou.

Platforma AllThingsTalk Maker nabízí bezplatnou licenci, která umožňuje připojení až deseti zařízení. Platforma je cloudová a nabízí uložení dat po dobu jednoho měsíce, na konci období lze stáhnout uložená data.

Pro vytvoření demonstrátoru se nabízí jako nevhodnější kombinací platform AllThingsTalk Maker a The Things Network. LoRaWAN bránu a LoRaWAN koncová zařízení lze zaregistrovat a připojit do platformy The Things Network, která je velmi zabezpečená veřejná síť a používá LoRaWAN síťový server The Things Stack, který umožňuje připojení, správu a monitorování zařízení, bran a aplikací pro koncové uživatele.

6 REALIZACE DEMONSTRÁTORU

Pro realizaci bylo potřeba vytvořit dva programy. První z programů byl pro řízení LoRaWAN brány a druhý pro řízení koncových zařízení LoRaWAN.

Poté bylo nutné naprogramovaná zařízení připojit do sítě LoRaWAN v platformě The Things Network. První se zaregistrovala LoRaWAN brána. Dále se musela vytvořit aplikace pro koncová zařízení, kde se jednotlivá koncová zařízení zaregistrovala. Poté se v aplikaci vytvořil webhook pro platformu AllThingsTalk Maker. V této platformě se vytvořily nová zařízení, na která přicházejí data z jednotlivých zařízení v aplikaci v platformě The Things Network. Zde je možné data zpracovat a graficky zobrazit.

6.1 Použité technologie

6.1.1 Jazyk Python

Jazyk Python je vysokoúrovňový objektově orientovaný programovací jazyk, který se používá například pro vývoj počítačových aplikací, vývoj webových stránek, analýzu dat, strojové učení, vývoj umělé inteligence a vědeckých výpočtů. Od vydání v roce 1991 (autorem jazyka je programátor Guido van Rossum) se Python stal jedním z nejoblíbenějších programovacích jazyků na světě [73].

Jednou z klíčových vlastností jazyka Python je jeho jednoduchost. Má jasnou a stručnou syntaxi, která usnadňuje čtení a učení se jazyka, je tedy vhodný pro začátečníky v programování. Python umožňuje jednoduché a interaktivní ladění a testování. Je možné jazyk Python integrovat s dalšími programovacími jazyky, včetně jazyků Java, C a C++. Python má také velkou a aktivní komunitu vývojářů, kteří přispívají do otevřených knihoven a nástrojů [73].

MicroPython je nenáročná a účinná verze programovacího jazyka Python, která obsahuje část standardní knihovny a je optimalizovaná pro práci na mikrokontrolerech a v omezeném režimu [74].

6.1.2 Jazyk C

Jazyk C je strukturovaný programovací jazyk střední úrovně, který je rovněž univerzální. Jazyk C je kompilovaný jazyk, což znamená, že kód je nejprve napsán v textovém editoru nebo IDE a poté zkompilován do strojového kódu, který může být spuštěn počítačem. Je známý svou rychlostí, a také schopností přímého propojení s hardwarem. Lze jej použít pro

systemové programování, vestavěné systémy a další aplikace, kde je důležitá efektivita a kontrola [73].

Jazyk C má poměrně snadno čitelnou a srozumitelnou syntaxi, ale k jeho efektivnímu používání je zapotřebí určitá úroveň programátorských dovedností. Je to výkonný jazyk, který lze použít k psaní operačních systémů, ovladačů zařízení a dalšího softwaru, který vyžaduje nízkou úrovně přístup k hardwaru. Jazyk byl navržen v letech 1969-1973 současně s počátečním vývojem operačního systému UNIX. Vycházel z jazyka CPL (Combined Programming Language), který byl následně upraven počítačovým vědcem Dennisem M. Ritchiem a vytvořil jazyk C, ve kterém přepsal operační systém UNIX [73][75].

Jazyk C měl významný vliv na vývoj počítačového programování a ovlivnil mnoho dalších jazyků, například C++, Javu a Python. Dodnes zůstává důležitým jazykem, zejména v oblasti vestavěných systémů a programování v reálném čase [73][75].

6.1.3 STM32CubeMX

STM32CubeMX je grafický softwarový konfigurační nástroj poskytovaný společností STMicroelectronics pro mikrokontrolery STM32. Umožňuje uživatelům generovat inicializační kód pro různé periferie, jako jsou časovače, UART, SPI, I2C a další, pomocí jednoduchého a intuitivního rozhraní. STM32CubeMX také poskytuje informace o rozvržení pinů, konfiguraci hodin a analýzu spotřeby energie [76].

Jednou z hlavních výhod STM32CubeMX je jeho schopnost zefektivnit proces vývoje generováním kódu, který lze snadno integrovat do projektu. Tím se snižuje čas a úsilí potřebné ke konfiguraci periférií a vývojáři se mohou soustředit na úkoly vyšší úrovně, jako je vývoj aplikací a ladění [76].

6.1.4 STM32CubeProgrammer

STM32CubeProgrammer je další softwarový nástroj poskytovaný společností STMicroelectronics pro programování a ladění mikrokontrolerů STM32. Nabízí komplexní a snadno použitelné rozhraní pro programování mikrokontrolerů STM32 s různými programovacími rozhraními, jako jsou STLINK, JTAG a SWD. Podporuje také programování externích pamětí, jako jsou paměti NOR a NAND flash [77].

STM32CubeProgrammer je navržen tak, aby bez problémů spolupracoval s dalšími vývojovými nástroji STM32, jako jsou STM32CubeIDE a STM32CubeMX. Lze jej

používat s různými operačními systémy, jako jsou Windows, Linux a MacOS, a podporuje programování široké škály mikrokontrolerů STM32 [77].

6.1.5 Vývojové prostředí

6.1.5.1 *Visual Studio Code*

Visual Studio Code je populární textový editor vyvinutý společností Microsoft, který je bezplatný, open-source a dostupný pro systémy Windows, macOS a Linux. Mezi vývojáři je oblíbený díky snadnému použití, přizpůsobitelnému rozhraní a široké škále rozšíření, díky kterým se z textového editoru stává vývojářské prostředí [78].

Jednou z klíčových rozšíření editoru VS Code je funkce Intellisense, která poskytuje inteligentní doplňování kódu, návrhy a zvýrazňování syntaxe během psaní. Díky tomu je kódování rychlejší a přesnější, protože vývojáři mohou psát kód efektivněji a s menším počtem chyb [78].

6.1.5.2 *STM32CubeIDE*

STM32CubeIDE je integrované vývojové prostředí (IDE) vyvinuté společností STMicroelectronics pro programování mikrokontrolerů STM32. Jedná se o bezplatný open-source nástroj, který podporuje různé operační systémy, například Windows, Linux a MacOS. STM32CubeIDE vychází z prostředí Eclipse IDE a obsahuje řadu nástrojů a funkcí, které vývojářům usnadňují programování mikrokontrolerů STM32. Poskytuje jednotnou platformu pro konfiguraci, vývoj a ladění vestavných aplikací založených na STM32 [79].

IDE obsahuje řadu vývojových nástrojů, včetně editoru kódu, kompilátoru a ladicího programu. Editor kódu poskytuje zvýrazňování syntaxe, doplňování kódu a další funkce, které usnadňují a urychlují psaní kódu. Kompilátor a ladicí program umožňují vývojářům sestavovat a ladit kód přímo v prostředí IDE, což usnadňuje identifikaci a opravu chyb. Hlavní vlastností STM32CubeIDE je jeho integrace s dalšími nástroji a softwarem STM32, jako je STM32CubeMX. Společně tyto nástroje poskytují kompletní vývojové prostředí pro embedded aplikace založené na STM32 [79].

6.2 LoRaWAN brána

Program pro zařízení Pygate je poměrně jednoduchý. Před samotným programováním jsou důležité tyto body:

- Zkontrolování verze firmware a případná aktualizace
- Registrace zařízení v platformě The Things Network

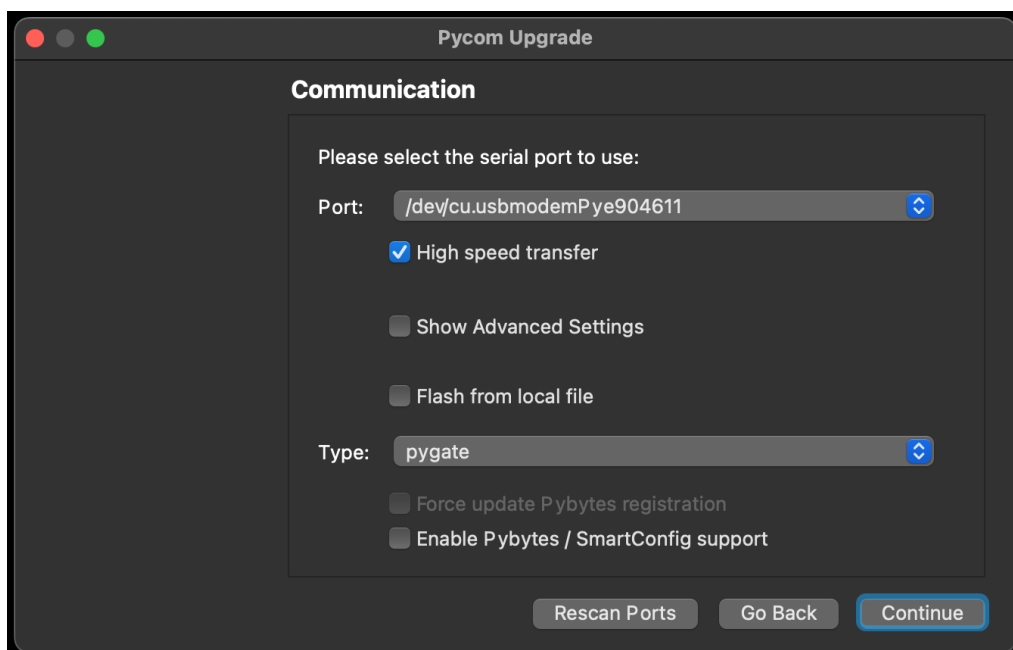
- Přidání rozšíření Pymakr ve VS Code
- Vytvoření programu

6.2.1 Aktualizace firmware

Potřebná kontrola verze firmware zařízení Pygate. Verze byla zastaralá, proto bylo nutné aktualizovat verzi. Na webových stránkách firmy Pycom je možné nalézt staré i nové verze firmware pro všechna zařízení, lze ale použít aplikaci *Pycom firmware update*, která je kompatibilní s operačními systémy Windows, macOS i Linux. Tuto aplikaci lze stáhnout z webové adresy <https://docs.pycom.io/updatefirmware/device/>. Pro aktualizaci firmware zařízení se postupuje podle pokynů v aplikaci. Pokyny jsou následující:

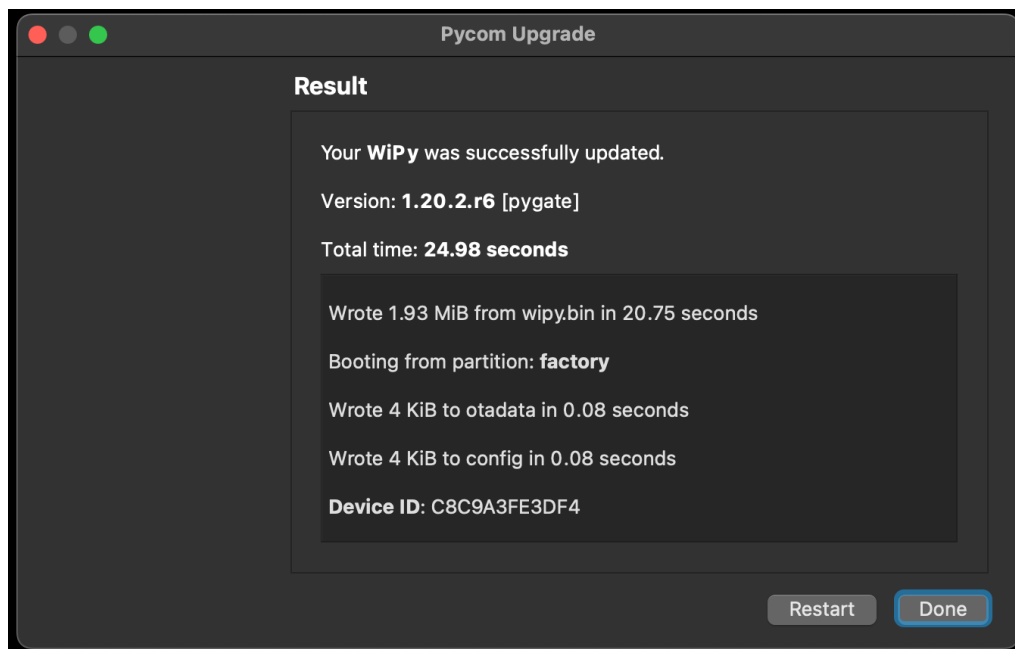
1. **Výběr portu** – pro automatický výběr portu, na kterém je zařízení v počítači připojeno, je možné použít tlačítko *Rescan Ports*, pokud je v počítači připojeno pouze zařízení Pygate, port může vypadat následovně: `/dev/cu.usbmodemPye904611`
2. **Výběr typu zařízení** – zde se zvolí typ zařízení, které potřebuje aktualizovat firmware, v tomto případě to bude typ *pygate*
3. **Odškrtnutí Pybytes / SmartConfig support** – Pybytes lze povolit v případě, pokud je potřeba využívat webovou aplikaci pro správu zařízení, v tomto případě to bude *neoznačeno*

Okno aplikace by mělo vypadat jako na obrázku (Obrázek 20).



Obrázek 20. Nastavení Pycom firmware Upgrade (zdroj vlastní)

Po kliknutí na tlačítko *Pokračovat* se spustí aktualizace. Po úspěšné aktualizaci okno aplikace vypadá jako na obrázku (Obrázek 21).



Obrázek 21. Úspěšná aktualizace firmware (zdroj vlastní)

6.2.2 Registrace zařízení

Zařízení Pygate je potřeba zaregistrovat v The Things Network na webové adrese <https://eu1.cloud.thethings.network/console/gateways>, kde je nutné zadat identifikační číslo (ID) zařízení. To lze najít na obrázku níže (Obrázek 21) pod označením *Device ID*: *C8C9A3FE3DF4*. Potom je nutné přidat za první tři bajty „FFFF“, výsledné ID je *C8C9A3FFFFE3DF4*.

Dalším krokem při registraci je nutné zvolit si *Gateway ID* v platformě TTN např. *gw-eui-c8c9a3ffffe3df4* a nastavit frekvenční rozsah *Europe 863-870 MHz (SF9 for RX2 - recommended)*. Nastavení bude vypadat jako na obrázku (Obrázek 22).

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.

Learn more in our guide on [Adding Gateways](#).

Gateway EUI [?](#)

Gateway ID [?](#) *

Gateway name [?](#)

Frequency plan [?](#) *

 | [v](#)

Require authenticated connection [?](#)

Choose this option eg. if your gateway is powered by [LoRa Basic Station](#)

Share gateway information

Select which information can be seen by other network participants, including [Packet Broker](#)

Share status within network [?](#)

Share location within network [?](#)

Obrázek 22. Registrace LoRa brány v TTN (zdroj vlastní)

6.2.3 Rozšíření editoru VS Code

V editoru Visual Studio Code je potřeba nainstalovat rozšíření Pymakr (viz. <https://marketplace.visualstudio.com/items?itemName=pycom.Pymakr>), které umožní nahrávání programu do zařízení, a dále ladění a testování zařízení. Zařízení Pygate využívá programovací jazyk Python, přesněji MicroPython, který je určen pro mikrokontrolery.

6.2.4 Program

Pomocí rozšíření Pymakr je možné vytvořit nový projekt, lze si vybrat název projektu, který bude i názvem složky. Dále je možné vygenerovat jeden ze dvou vzorových projektů *empty* nebo *led-example*. Vygenerují se tři soubory: *pymakr.conf*, *boot.py* a *main.py*.

- *pymakr.conf* – je konfigurační soubor pro rozšíření Pymakr, určuje název projektu
- *boot.py* – soubor se spustí hned po zavedení systému zařízení
- *main.py* – k přidávání vlastního kódu programu

Ze zaregistrovaného zařízení v platformě TTN je potřeba stáhnout konfigurační soubor sítě LoRaWAN *global_conf.json* a vložit jej do projektu.

V projektu je potřeba vytvořit ještě jeden konfigurační soubor *config.py*, ve kterém se vygeneruje identifikační číslo zařízení, nastaví barvy pro signalizaci zařízení a nastaví údaje SSID a heslo pro připojení do sítě Wi-Fi, pokud bude využívána.

Dále je vhodné vytvořit ještě jeden soubor *gateway.py* pro třídu **Gateway**, která se bude starat o připojení zařízení k internetové síti a síti LoRaWAN.

Kompletní zdrojový kód je součástí přílohy.

6.2.4.1 Soubor *boot.py*

Tento soubor, neboli skript, se spustí při zapnutí zařízení Pygate. Na začátku vypne LED diodu srdečního tepu v zařízení a vypne funkci Pybytes, která se používá k připojení do IoT platformy od společnosti Pycom.

Funkce **machine.main** přijímá argument, kterým je název souboru obsahující hlavní kód aplikace, jenž má být spuštěna.

```
import pycom
import machine
pycom.heartbeat(False)
pycom.pybytes_on_boot(False)
machine.main('main.py')
```

6.2.4.2 Soubor *main.py*

Tento skript importuje třídu **Gateway**, vytvoří instanci této třídy a pak ji spustí voláním metody **start**.


```
from gateway import Gateway
gateway = Gateway()
gateway.start()
```

6.2.4.3 Soubor *gateway.py*

V souboru je definována třída **Gateway**, která obsahuje několik metod.

Metoda **__init__** nastavuje funkci „callback“ pro události Pygate, která změní barvu LED podle typu události.

Metoda **init_gateway** inicializuje koncentrátor LoRaWAN otevřením a načtením konfiguračního souboru *global_conf.json* a jeho následným zpracováním funkcí **machine.pygate_init**.

Metoda **init_rtc** synchronizuje hodiny reálného času (RTC) zařízení se serverem síťového času (NTP).

Metoda **init_network** se pokouší navázat internetové spojení pomocí ethernetu nebo pomocí Wi-Fi. Pokud se povede připojit, vrátí objekt *Ethernet* nebo *Wi-Fi*. Pokud se žádné z možností připojení nezdaří, vrátí se *None*.

Funkce **eth_conn** a **wifi_conn** se volají v metodě **init_network** a čekají na připojení k internetové síti.

Metoda **start** je volána při spuštění zařízení mimo třídu *Gateway*. Volá výše uvedené metody, aby inicializovala síť, synchronizovala RTC a spustila koncentrátor LoRaWAN.

```
import config
import machine
from machine import RTC
from network import ETH
from network import WLAN
import pycom
import time

class Gateway:
    def machine_callback(self, arg):
        evt = machine.events()
        if (evt & machine.PYGATE_START_EVT):
            pycom.rgbled(config.RGB_GATEWAY_OK)
        elif (evt & machine.PYGATE_ERROR_EVT):
            pycom.rgbled(config.RGB_GATEWAY_ERROR)
```

```

elif (evt & machine.PYGATE_STOP_EVT):
    pycom.rgbled(config.RGB_GATEWAY_STOP)
def __init__(self):
    print("Init: Initialization of Gateway class...")
    machine.callback(trigger = (
        machine.PYGATE_START_EVT |
        machine.PYGATE_STOP_EVT |
        machine.PYGATE_ERROR_EVT
    ), handler=self.machine_callback)

    self.rtc = RTC()
    self.gateway_config_file = None

def init_gateway(self):
    print("Starting LoRaWAN Concentrator...")
    try:
        self.gateway_config_file = open('/flash/global_conf.json',
'r').read()
    except Exception as e:
        print("Error opening Gateway Config: {}".format(e))
        return False
    else:
        machine.pygate_init(self.gateway_config_file)
        print("Init GW: LoRaWAN Concentrator UP!")
        return True

def init_rtc(self):
    print("Syncing RTC...")
    try:
        self.rtc.ntp_sync(server="pool.ntp.org")
    except Exception as e:
        print("Exception RTC: {}".format(e))
        return False
    else:
        print("RTC: Synced!")
        return True

def init_network(self):
    print('\nconnecting to network...')
    eth = self.eth_conn()

```

```

    if eth is not None:
        return eth
    wifi = self.wifi_conn()
    if wifi is not None:
        return wifi
    return None

def eth_conn(self):
    print('\nconnecting to ethernet...')
    eth = ETH()
    timer = 0
    while not eth.isconnected():
        pycom.rgbled(config.RGB_CONNECTING)
        print('.', end='')
        time.sleep(1)
        pycom.rgbled(config.RGB_OFF)
        time.sleep(1)
        timer += 1
        if timer > 10:
            return None
    if eth.isconnected():
        print('\nethernet config:', eth.ifconfig())
    return eth

def wifi_conn(self):
    if config.WIFI_SSID is None:
        print('\nNo SSID provided')
        return None
    print('\nconnecting to wifi... ' + config.WIFI_SSID)
    wifi = WLAN(mode=WLAN.STA)
    timer = 0
    while not wifi.isconnected():
        pycom.rgbled(config.RGB_CONNECTING)
        print('.', end='')
        wifi.connect(config.WIFI_SSID, auth=(WLAN.WPA2,
config.WIFI_PASS))
        time.sleep(1)
        pycom.rgbled(config.RGB_OFF)
        time.sleep(1)
        timer += 1

```

```

        if timer > 10:
            return None

    print('\nnetwork config:', wifi.ifconfig)
    print('connected to wifi: ' + wifi.ssid())
    return wifi

def start(self):
    print("Start Up: Now starting up Gateway...")
    self.init_network()
    self.init_rtc()
    self.init_gateway()

```

6.2.4.4 Soubor *config.py*

Tento konfigurační soubor vygeneruje identifikační číslo zařízení, nastaví barvy pro signalizaci stavu zařízení a nastaví údaje SSID a heslo pro připojení do sítě Wi-Fi, pokud bude využívána.

```

import machine
import ubinascii

WIFI_MAC = ubinascii.hexlify(machine.unique_id()).upper()
GATEWAY_ID = WIFI_MAC[:6] + "FFFF" + WIFI_MAC[6:12]

WIFI_SSID = None
WIFI_PASS = None

RGB_OFF = (0x000000)
RGB_CONNECTING = (0x001033)
RGB_RTC_IS_SYNCING = (0x330033)
RGB_GATEWAY_OK = (0x003300)
RGB_GATEWAY_STOP = (0x333300)
RGB_GATEWAY_ERROR = (0x330000)

```

6.2.4.5 Soubor *global_conf.json*

Soubor obsahuje důležitou konfiguraci pro připojení do sítě LoRaWAN. Tato konfigurace se vygeneruje na portále v zaregistrovaném zařízení LoRaWAN brány. Údaje v konfiguraci jsou stejné, jaké se zadávaly při registraci zařízení.

```
{
  "SX1301_conf": {
    "lorawan_public": true,
    "clksrc": 1,
    "antenna_gain": 0,
    "radio_0": {
      "enable": true,
      "type": "SX1257",
      "freq": 867500000,
      "rssi_offset": -166,
      "tx_enable": true,
      "tx_freq_min": 863000000,
      "tx_freq_max": 870000000
    },
    "radio_1": {
      "enable": true,
      "type": "SX1257",
      "freq": 868500000,
      "rssi_offset": -166,
      "tx_enable": false
    },
    "chan_multiSF_0": {
      "enable": true,
      "radio": 1,
      "if": -400000
    },
    "chan_multiSF_1": {
      "enable": true,
      "radio": 1,
      "if": -200000
    },
    "chan_multiSF_2": {
      "enable": true,
      "radio": 1,
      "if": 0
    },
    "chan_multiSF_3": {
      "enable": true,
      "radio": 0,
      "if": -400000
    }
  }
}
```

```
},  
"chan_multiSF_4": {  
  "enable": true,  
  "radio": 0,  
  "if": -200000  
},  
"chan_multiSF_5": {  
  "enable": true,  
  "radio": 0,  
  "if": 0  
},  
"chan_multiSF_6": {  
  "enable": true,  
  "radio": 0,  
  "if": 200000  
},  
"chan_multiSF_7": {  
  "enable": true,  
  "radio": 0,  
  "if": 400000  
},  
"chan_Lora_std": {  
  "enable": true,  
  "radio": 1,  
  "if": -200000,  
  "bandwidth": 250000,  
  "spread_factor": 7  
},  
"chan_FSK": {  
  "enable": true,  
  "radio": 1,  
  "if": 300000,  
  "bandwidth": 125000,  
  "datarate": 50000  
},  
"tx_lut_0": {  
  "pa_gain": 0,  
  "mix_gain": 8,  
  "rf_power": -6,  
  "dig_gain": 0
```

```
},  
"tx_lut_1": {  
  "pa_gain": 0,  
  "mix_gain": 10,  
  "rf_power": -3,  
  "dig_gain": 0  
},  
"tx_lut_2": {  
  "pa_gain": 0,  
  "mix_gain": 12,  
  "rf_power": 0,  
  "dig_gain": 0  
},  
"tx_lut_3": {  
  "pa_gain": 1,  
  "mix_gain": 8,  
  "rf_power": 3,  
  "dig_gain": 0  
},  
"tx_lut_4": {  
  "pa_gain": 1,  
  "mix_gain": 10,  
  "rf_power": 6,  
  "dig_gain": 0  
},  
"tx_lut_5": {  
  "pa_gain": 1,  
  "mix_gain": 12,  
  "rf_power": 10,  
  "dig_gain": 0  
},  
"tx_lut_6": {  
  "pa_gain": 1,  
  "mix_gain": 13,  
  "rf_power": 11,  
  "dig_gain": 0  
},  
"tx_lut_7": {  
  "pa_gain": 2,  
  "mix_gain": 9,
```

```
"rf_power": 12,  
  "dig_gain": 0  
},  
"tx_lut_8": {  
  "pa_gain": 1,  
  "mix_gain": 15,  
  "rf_power": 13,  
  "dig_gain": 0  
},  
"tx_lut_9": {  
  "pa_gain": 2,  
  "mix_gain": 10,  
  "rf_power": 14,  
  "dig_gain": 0  
},  
"tx_lut_10": {  
  "pa_gain": 2,  
  "mix_gain": 11,  
  "rf_power": 16,  
  "dig_gain": 0  
},  
"tx_lut_11": {  
  "pa_gain": 3,  
  "mix_gain": 9,  
  "rf_power": 20,  
  "dig_gain": 0  
},  
"tx_lut_12": {  
  "pa_gain": 3,  
  "mix_gain": 10,  
  "rf_power": 23,  
  "dig_gain": 0  
},  
"tx_lut_13": {  
  "pa_gain": 3,  
  "mix_gain": 11,  
  "rf_power": 25,  
  "dig_gain": 0  
},  
"tx_lut_14": {
```



```

    "pa_gain": 3,
    "mix_gain": 12,
    "rf_power": 26,
    "dig_gain": 0
  },
  "tx_lut_15": {
    "pa_gain": 3,
    "mix_gain": 14,
    "rf_power": 27,
    "dig_gain": 0
  }
},
"gateway_conf": {
  "gateway_ID": "C8C9A3FFFFFFE3DF4",
  "server_address": "eu1.cloud.thethings.network",
  "serv_port_up": 1700,
  "serv_port_down": 1700,
  "servers": [
    {
      "gateway_ID": "C8C9A3FFFFFFE3DF4",
      "server_address": "eu1.cloud.thethings.network",
      "serv_port_up": 1700,
      "serv_port_down": 1700,
      "serv_enabled": true
    }
  ]
}
}

```

6.3 LoRaWAN koncové zařízení

6.3.1 Registrace zařízení

Stejně jako se registrovala LoRaWAN brána v platformě TTN, tak je nutné zaregistrovat koncová zařízení LoRaWAN.

První se musí vytvořit aplikace, kde je potřeba zadat unikátní ID aplikace: *footboards-pipes-meter*. Dále lze přidat jméno aplikace: *Footboards pipes meter*. Lze přidat i popis aplikace. Zadané hodnoty budou jako na obrázku (Obrázek 23).

Create application

Within applications, you can register and manage end devices and their network data. After setting up your device fleet, use one of our many integration options to pass relevant data to your external services.

Learn more in our guide on [Adding Applications](#).

Application ID *

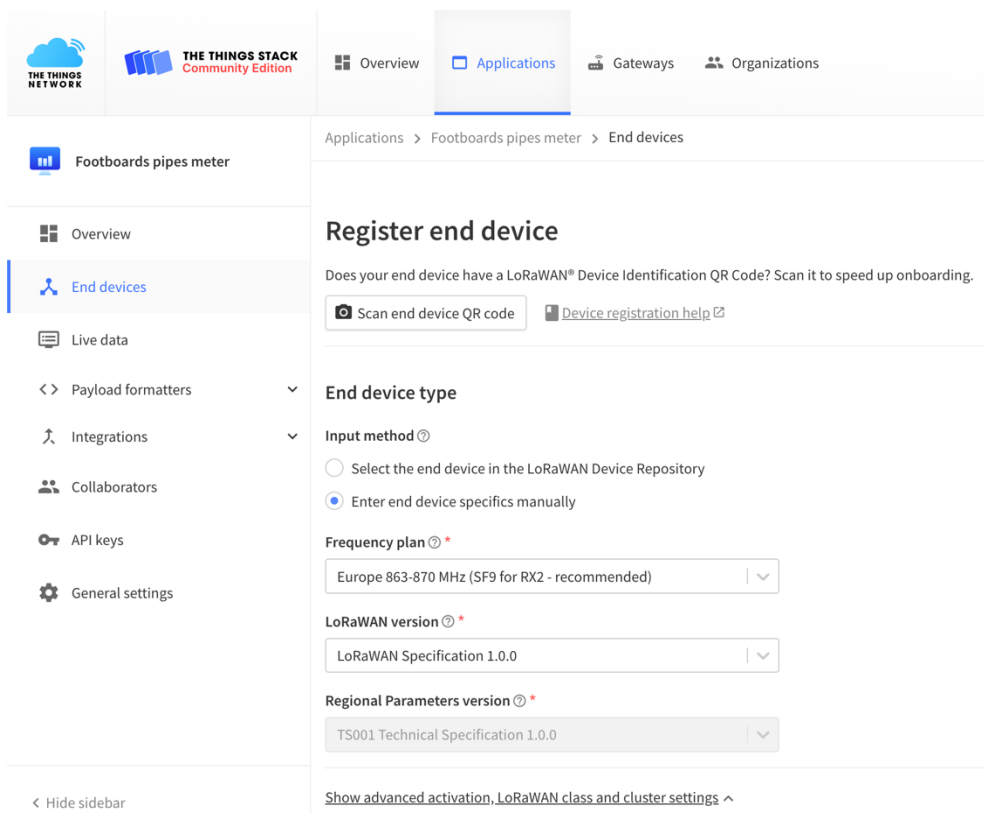
Application name

Description

Optional application description; can also be used to save notes about the application

Obrázek 23. Vytvoření aplikace v TTN (zdroj vlastní)

Dalším krokem je přidání zařízení do vytvořené aplikace v TTN. Informace o zařízení je potřeba zadat ručně, jelikož informace o zařízení doposud nebyly přidány do repositáře *The LoRaWAN Device Repository*. Zvolení frekvence závisí na regionu a samozřejmě i frekvenci LoRaWAN brány, na kterou se budou zařízení připojovat. Frekvenční rozsah je nastaven na *Europe 863-870 MHz (SF9 for RX2 – recommended)*. Verze LoRaWAN je nastavena na *LoRaWAN Specification 1.0.0*. Zvolené hodnoty registrovaného zařízení jsou na obrázku (Obrázek 24).



Obrázek 24. Registrace zařízení – základní nastavení (zdroj vlastní)

Dále je potřeba otevřít rozšířené možnosti nastavení, kde se vygenerují čtyři položky: DevEUI, Device address, AppSKey a NwkSKey. Tyto identifikační čísla a aplikační klíče bude potřeba nastavit v programu zařízení. Zvolené hodnoty registrovaného zařízení jsou na obrázku (Obrázek 25).

Footboards pipes meter

Show advanced activation, LoRaWAN class and cluster settings ▾

Overview

End devices

Live data

Payload formatters ▾

Integrations ▾

Collaborators

API keys

General settings

Provisioning information

DevEUI ⓘ

70 B3 D5 7E D0 05 D7 BA 5/50 used

Device address ⓘ *

26 0B 78 67

AppSKey ⓘ *

E5 AB FB 11 2C 8A BD 4F 4E EF 23 28 73 B6 B6 B1

NwkSKey ⓘ *

86 65 03 32 A7 5F 51 CD 9D 34 0A D0 42 96 93 B5

End device ID ⓘ *

e5-mini-stm32wle5jc-1

This value is automatically prefilled using the DevEUI

After registration

View registered end device

Register another end device of this type

< Hide sidebar

Obrázek 25. Registrace zařízení – rozšířené nastavení (zdroj vlastní)

Po vytvoření jednoho zařízení lze stejným způsobem přidávat další zařízení.

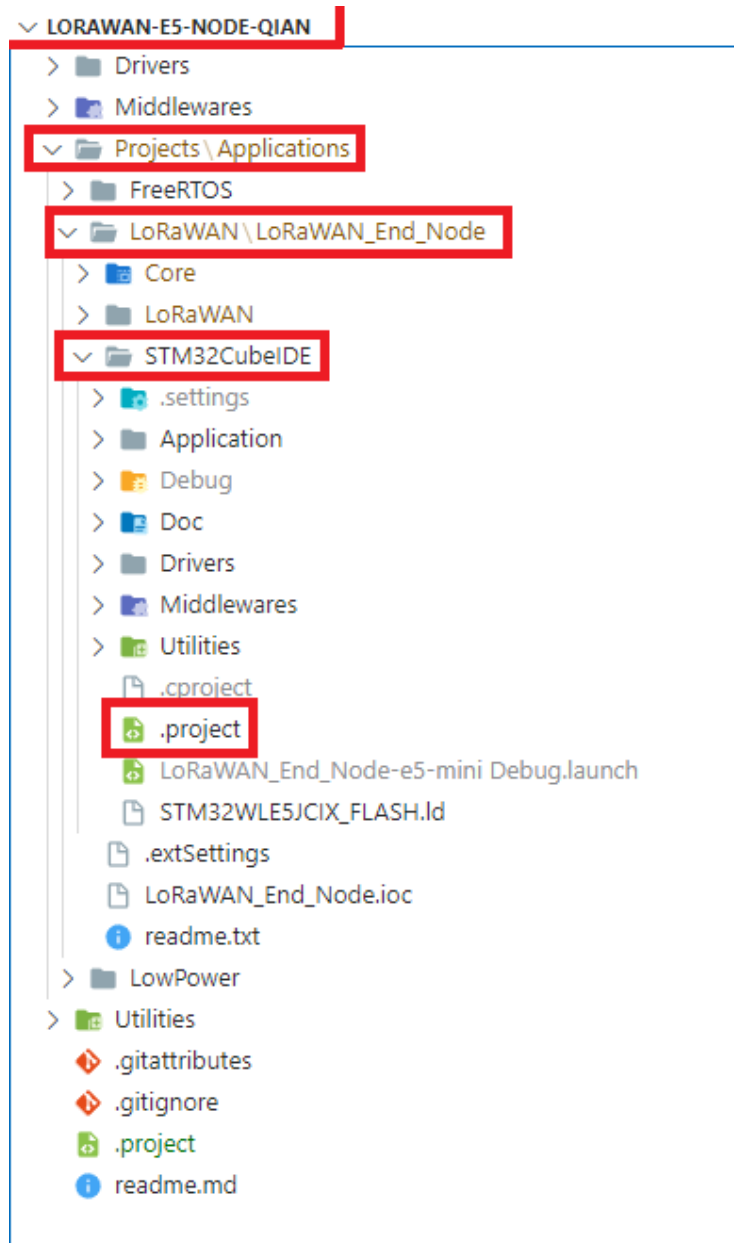
6.3.2 Program

V dokumentaci k zařízení Wio-E5 mini lze nalézt webový odkaz na repositář výrobce zařízení, kde se nachází open-source kód (<https://github.com/Seeed-Studio/LoRaWan-E5-Node/tree/qian>), který lze stáhnout a použít k programování zařízení a připojení k síti LoRaWAN.

Kompletní zdrojový kód je součástí přílohy.

6.3.2.1 Otevření projektu

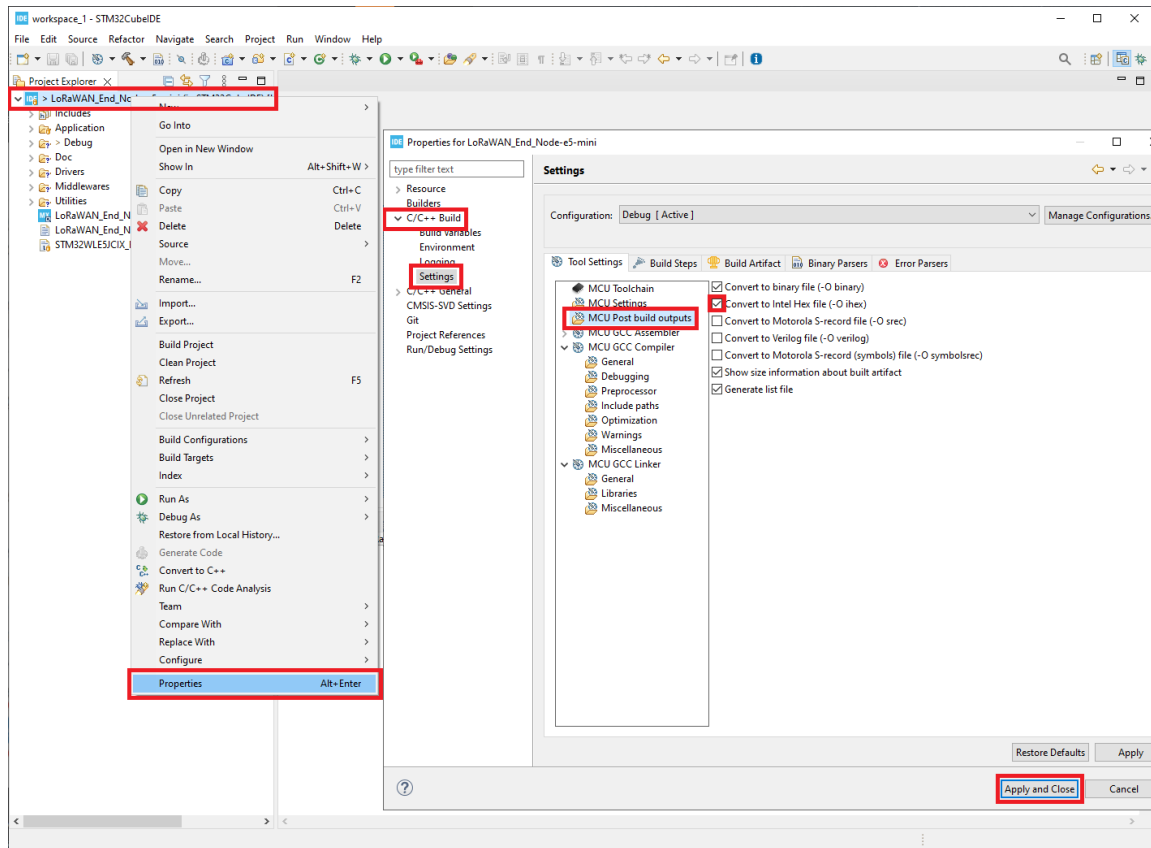
Po stažení projektu je potřeba najít spustitelný soubor *.project*, ten se nachází v podsložkách projektu: *Projects > Applications > LoRaWAN > LoRaWAN_End_Node > STM32CubeIDE*. Cesta k souboru je na obrázku (Obrázek 26).



Obrázek 26. Cesta k souboru „.project“ (zdroj vlastní)

6.3.2.2 Nastavení výstupu projektu

Po rozkliknutí souboru *.project* se otevře vývojové prostředí STM32CubeIDE, zde je potřeba nastavit výstup projektu po sestavení (tzv. build), aby byl výstupem soubor v hexadecimálním formátu. Soubor v hexadecimálním formátu lze nahrát do zařízení. Toto nastavení se provede podle obrázku níže (Obrázek 27).



Obrázek 27. Nastavení výstupu po sestavení (zdroj vlastní)

6.3.2.3 Konfigurační nastavení programu

V programu je nutné změnit několik hodnot, aby po nahrání programu do zařízení a spuštění, bylo koncové zařízení schopné připojit se do sítě LoRaWAN. Změny se provedou v souborech *lora_app.h* a *se-identity.h*.

6.3.2.3.1 Soubor *lora_app.h*

V tomto souboru je potřeba zkontrolovat nastavení LoRaWAN regionu, toto nastavení se nachází pod definicí makro konstanty *ACTIVE_REGION*. V regionu Evropy je zde nastavena hodnota *LORAMAC_REGION_EU868*, jak je vidět na obrázku (Obrázek 28).

V tomto souboru lze nastavit i cyklus přenosu dat v síti LoRaWAN. Nachází se pod makro konstantou *APP_TX_DUTYCYCLE* a hodnota se zde udává v milisekundách. V případě demonstrátoru může být hodnota nastavena na 30000ms, což je 30 sekund. Takto upravená hodnota je na obrázku (Obrázek 28).

```

lora_app.h x
+1
42 /* LoRaWAN application configuration (Mw is configured by lorawan_conf.h) */
43 #define ACTIVE_REGION LORAMAC_REGION_EU868
44
45 /*!
46 * CAYENNE_LPP is myDevices Application server.
47 */
48 /*#define CAYENNE_LPP*/
49
50 /*!
51 * Defines the application data transmission duty cycle. 10s, value in [ms] 10000.
52 */
53 #define APP_TX_DUTYCYCLE 30000
54
55 /*!
56 * LoRaWAN User application port

```

Obrázek 28. Nastavení regionu a cyklu přenosu dat (zdroj vlastní)

6.3.2.3.2 Soubor se-identity.h

V tomto souboru se nastavují identifikační čísla zařízení a aplikační a síťové klíče.

Po zaregistrování koncového zařízení do aplikace v TTN se získají hodnoty DevEUI, Device address, AppSKey a NwkSKey. Tyto hodnoty je potřeba vložit k příslušné definici makro konstanty.

Hodnota DevEUI se nastavuje v definici makro konstanty *LORAWAN_DEVICE_EUI*. Předtím je nutné nastavit hodnotu konstanty *STATIC_DEVICE_EUI* na „1“, aby se negenerovala náhodná hodnota DevEUI automaticky. V TTN v zaregistrovaném zařízení lze hodnotu zkopírovat ve dvou hexadecimálních formátech. První formát je posloupnost hodnot: 70B3D57ED005D7BA. Druhý formát má hexadecimální zápis, to znamená, že je před každou hodnotou bajtu předpona „0x“: 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0xD7, 0xBA. Do souboru se hodnota vkládá ve druhém tvaru do složených závorek, jak je zobrazeno na obrázku (Obrázek 29).

Hodnota Device address se nastavuje v definici makro konstanty *LORAWAN_DEVICE_ADDRESS*. Zde je také nutné nastavit konstantu *STATIC_DEVICE_ADDRESS* na hodnotu „1“, aby se negenerovala náhodná hodnota Device address. V TTN v zaregistrovaném zařízení lze hodnotu zkopírovat ve dvou hexadecimálních formátech. První formát má tvar čtyř bajtové posloupnosti: 260B7867. Druhý formát má před každou hodnotou bajtu předponu „0x“: 0x26, 0x0B, 0x78, 0x67. Do souboru se vkládá hodnota v prvním tvaru a před ní je označení datového typu *uint32_t* a předpona „0x“, jako je zobrazeno na obrázku (Obrázek 29).

```

se-identity.h x
88
89 /*!
90 * When set to 1 DevEui is LORAWAN_DEVICE_EUI
91 * When set to 0 DevEui is automatically set with a value provided by MCU platform
92 */
93 #define STATIC_DEVICE_EUI 1
94
95 /*!
96 * end-device IEEE EUI (big endian)
97 */
98 #define LORAWAN_DEVICE_EUI { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0xD7, 0xBA }
99
100 /*!
101 * App/Join server IEEE EUI (big endian)
102 */
103 #define LORAWAN_JOIN_EUI { 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01 }
104
105 /*!
106 * When set to 1 DevAddr is LORAWAN_DEVICE_ADDRESS
107 * When set to 0 DevAddr is automatically set with a value provided by a pseudo
108 * random generator seeded with a value provided by the MCU platform
109 */
110 #define STATIC_DEVICE_ADDRESS 1
111
112 /*!
113 * Device address on the network (big endian)
114 */
115 #define LORAWAN_DEVICE_ADDRESS ( uint32_t )0x260B7867
116
117 /*!
118 * Application root key

```

Obrázek 29. Nastavení hodnot „DevEUI“ a „Device address“ (zdroj vlastní)

Hodnoty AppSKey a NwkSKey se každá nastaví ve dvou makro konstantách. Obě hodnoty lze z TTN zkopírovat ve dvou formátech. První formát obou hodnot má tvar hexadecimální posloupnosti.

- AppSKey je: E5ABFB112C8ABD4F4EEF232873B6B6B1.
- NwkSKey je: 86650332A75F51CD9D340AD0429693B5.

Druhý formát má před každou hodnotou bajtu předponu „0x“.

- AppSKey je: 0xE5, 0xAB, 0xFB, 0x11, 0x2C, 0x8A, 0xBD, 0x4F, 0x4E, 0xEF, 0x23, 0x28, 0x73, 0xB6, 0xB6, 0xB1.
- NwkSKey je: 0x86, 0x65, 0x03, 0x32, 0xA7, 0x5F, 0x51, 0xCD, 0x9D, 0x34, 0x0A, 0xD0, 0x42, 0x96, 0x93, 0xB5.

Do souboru se vkládají hodnoty v prvním tvaru, ale každý bajt neboli každé dvojčíslí je odděleno čárkou podobně, jako na obrázku (Obrázek 30).


```

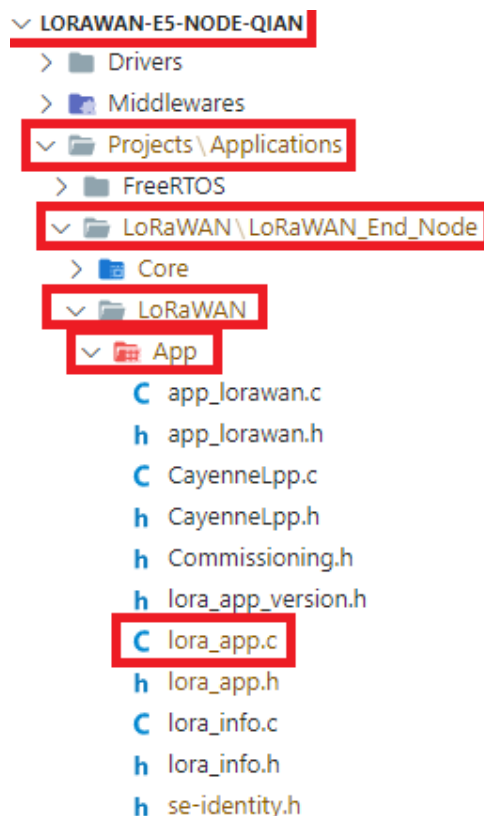
se-identity.h ×
112@ /*!
113 * Device address on the network (big endian)
114 */
115 #define LORAWAN_DEVICE_ADDRESS                ( uint32_t )0x260B7867
116
117@ /*!
118 * Application root key
119 */
120 #define LORAWAN_APP_KEY                        E5,AB,FB,11,2C,8A,BD,4F,4E,EF,23,28,73,B6,B6,B1
121
122@ /*!
123 * Network root key
124 */
125 #define LORAWAN_NWK_KEY                        86,65,03,32,A7,5F,51,CD,9D,34,0A,D0,42,96,93,B5
126
127@ /*!
128 * Forwarding Network session key
129 */
130 #define LORAWAN_NWK_S_KEY                      86,65,03,32,A7,5F,51,CD,9D,34,0A,D0,42,96,93,B5
131
132@ /*!
133 * Application session key
134 */
135 #define LORAWAN_APP_S_KEY                      E5,AB,FB,11,2C,8A,BD,4F,4E,EF,23,28,73,B6,B6,B1
136

```

Obrázek 30. Nastavení hodnot „AppSKey“ a „NwkSKey“ (zdroj vlastní)

6.3.2.4 Odesílaná data

V souboru *lora_app.c*, je potřeba upravit funkci **SendTxData**. Tento soubor se nachází v podadresáři projektu: *Projects > Applications > LoRaWAN > LoRaWAN_End_Node > LoRaWAN > App*, cesta je vidět na obrázku (Obrázek 31).



Obrázek 31. Cesta k souboru „lora_app.c“ (zdroj vlastní)

Ve funkci je potřeba definovat proměnné, které se budou ze zařízení odesílat. V tomto případě je to hodnota teploty a stav baterie.

Stav baterie se získá pomocí funkce **GetBatteryLevel**, vrácená hodnota je celé číslo od 1 do 254, kdy hodnota 1 znamená, že stav baterie je nejnižší a hodnota 254 označuje plné nabití, popř. připojení v elektrické síti. Hodnota stavu baterie bude uložena v 8bitové celočíselné proměnné (`uint8_t`) s názvem *batteryLevel*.

Hodnota teploty se získá z proměnné *sensor_data.temperature* a uloží hodnotu ze senzoru ve stupních Celsia. Tato hodnota se následně převede z reálného čísla (`float`) na celočíselnou hodnotu tak, že je vynásobena číslem deset a převedena na 16bitové celé číslo (`int16_t` / `uint16_t`) bez znaménka. Výsledná proměnná s hodnotou teploty se nazývá *temperature*.

AppData.Buffer[] je zásobník, do kterého se ukládají hodnoty z proměnných *batteryLevel* a *temperature*. Tyto data (tzv. payload) se posílají přes síť LoRaWAN.

Hodnota stavu baterie se do zásobníku vkládá jako jeden bajt vyjádřený v hexadecimálním formátu.

Teplota se do zásobníku vkládá tak, že na jeden bajt se uloží horních 8 bitů hodnoty a na druhý bajt uloží dolních 8 bitů hodnoty teploty. Důvodem je zachování desetinných hodnot.

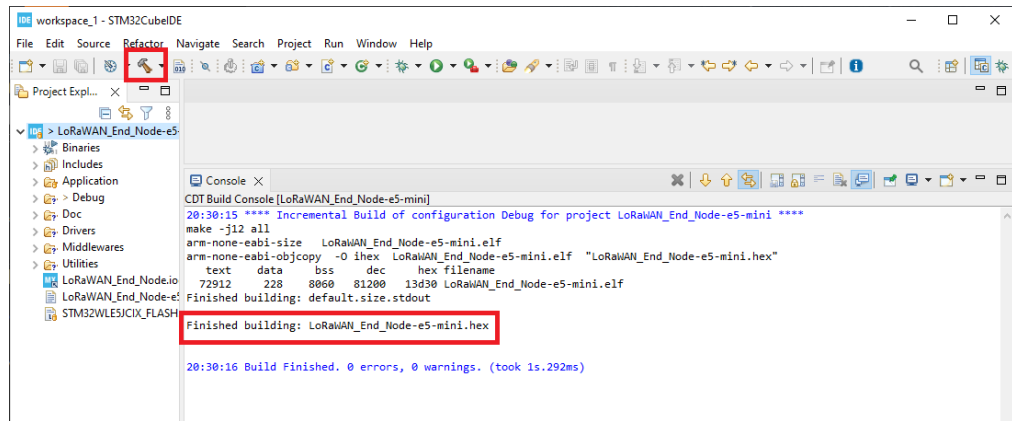
Kód ve funkci bude následovný.

```
uint8_t batteryLevel = GetBatteryLevel();
int16_t temperature = (uint16_t)(sensor_data.temperature * 10);
AppData.Buffer[i++] = (uint8_t)((temperature >> 8) & 0xFF);
AppData.Buffer[i++] = (uint8_t)(temperature & 0xFF);
AppData.Buffer[i++] = batteryLevel;
```

Kompletní kód je součástí přílohy.

6.3.3 Sestavení programu

Jakmile jsou všechny soubory upraveny, je možné program sestavit a vytvořit soubor v hexadecimálním formátu. Pro sestavení stačí kliknout na tlačítko s kladivem, pokud nebudou v kódu žádné chyby, které kompilátor odhalí, vše bude v pořádku jako na obrázku (Obrázek 32). Takto sestavený zdrojový kód lze nahrát do zařízení.

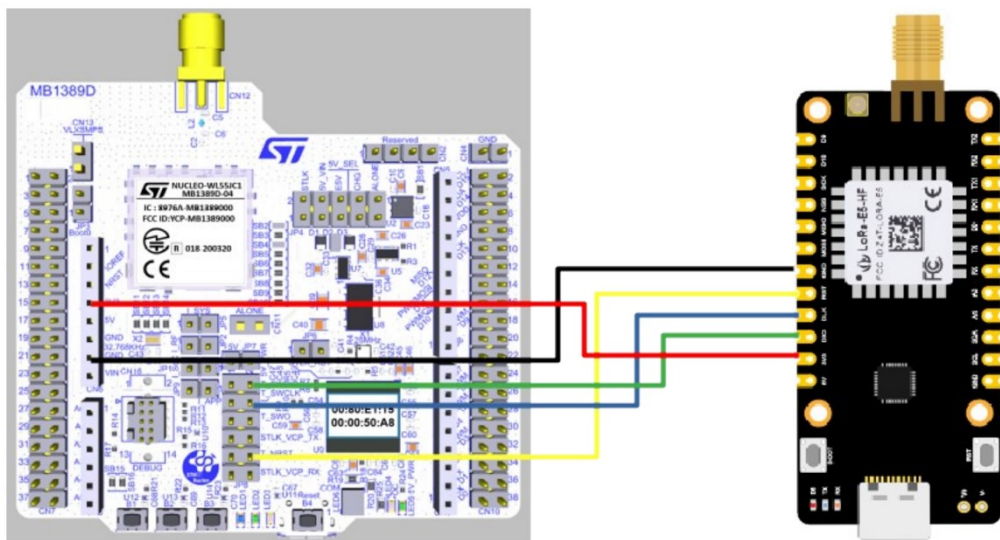


Obrázek 32. Sestavení programu (zdroj vlastní)

6.3.4 Nahrání programu do zařízení

K programování zařízení je potřeba ladící hardware ST-LINK od společnosti STMicroelectronics. K dispozici bylo zařízení NUCLEO-WL55JC1, které má integrovaný ladící a programovací modul ST-LINK.

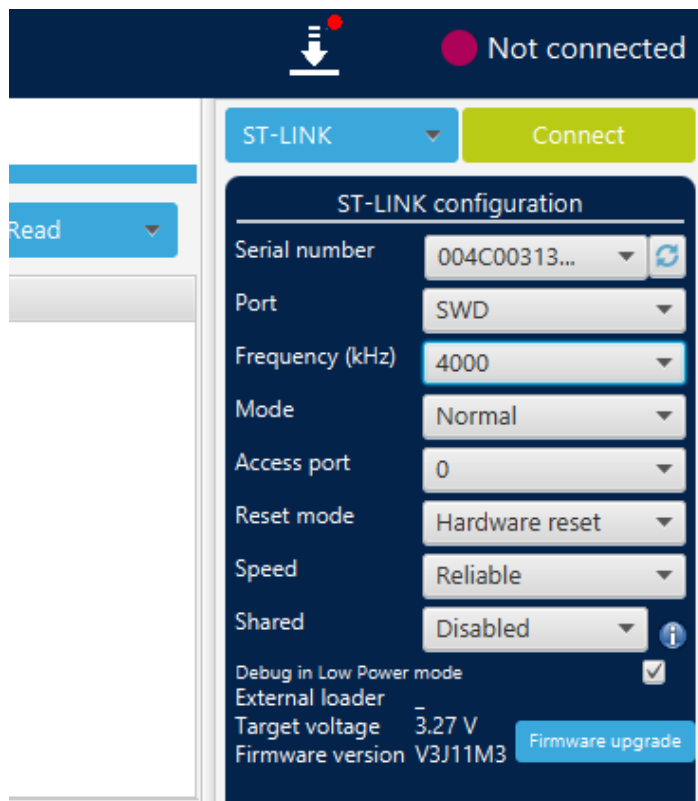
Desku Wio-E5 mini je potřeba propojit s deskou NUCLEO-WL55JC1, přesně podle postupu v pdf souboru, který je součástí přílohy na CD. Propojení je znázorněno na obrázku (Obrázek 33).



Obrázek 33. Propojení desek Wio-E5 mini a NUCLEO-WL55JC1 (zdroj pdf soubor v příloze)

Pokud jsou desky propojeny přesně podle postupu, lze připojit zařízení NUCLEO-WL55JC1 do počítače.

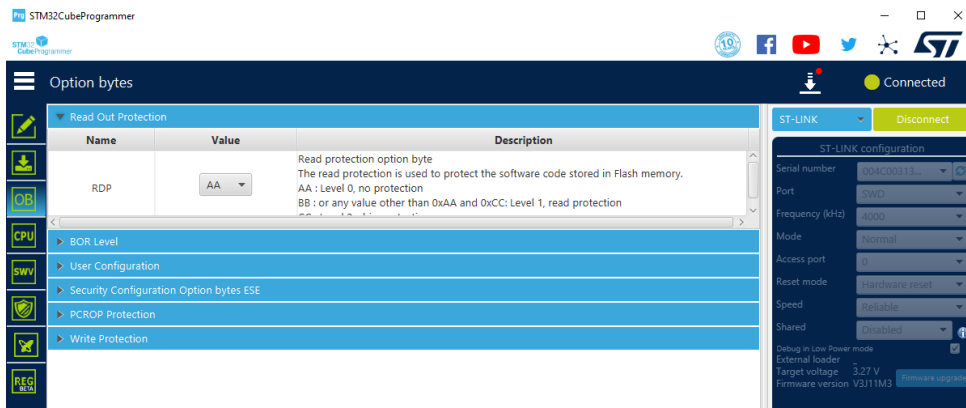
Prvním krokem je otevření aplikace STM32CubeProgrammer a v pravé části je potřeba nastavit hodnoty, jako je na obrázku (Obrázek 34). Všechny hodnoty by měly být nastaveny po připojení zařízení, kromě frekvence, ta se nastaví na 4000 kHz. Poté se lze v aplikaci připojit k zařízení.



Obrázek 34. Nastavení hodnot pro připojení zařízení
(zdroj vlastní)

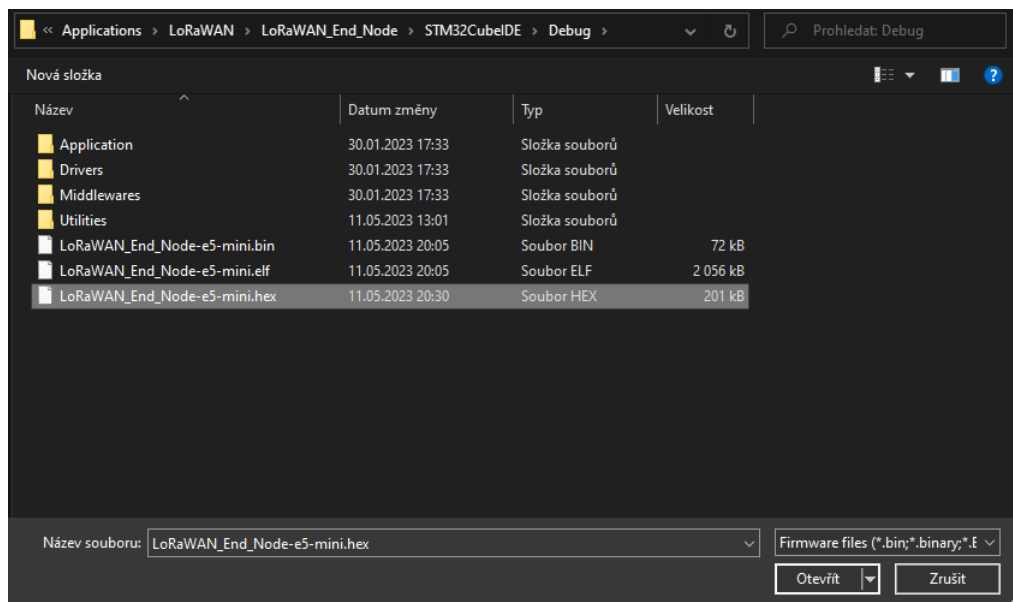
Po úspěšném připojení zařízení je možné pokračovat dále.

Ve druhém kroku je potřeba zkontrolovat nastavení ochrany kódu zařízení a popřípadě změnit hodnotu na „AA“ tak, jak je to na obrázku (Obrázek 35).



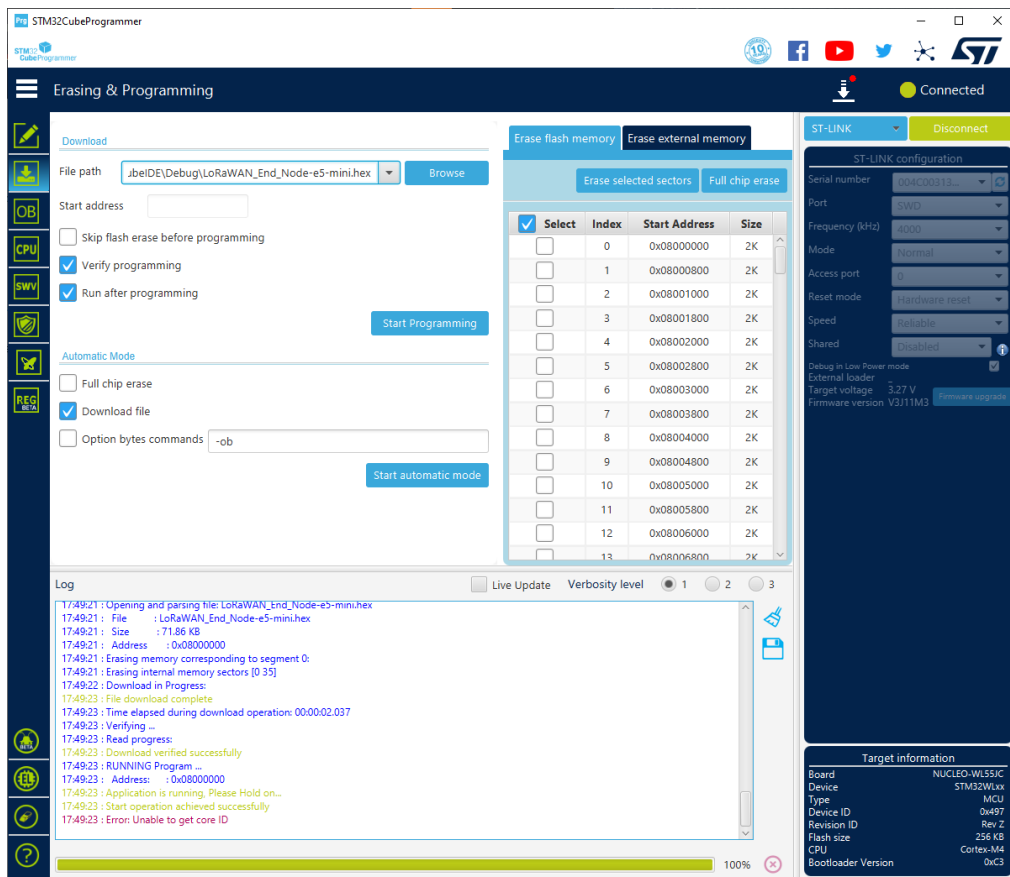
Obrázek 35. Nastavení ochrany (zdroj vlastní)

V posledním kroku lze nahrát program v hexadecimálním formátu do zařízení. Nejprve se musí zadat cesta k hexadecimálnímu souboru. Ten se nachází v adresáři *Debug*, který je ve stejné složce jako soubor *.project*. Cesta je vidět na obrázcích (Obrázek 26 a Obrázek 36).



Obrázek 36. Cesta k souboru s programem (zdroj vlastní)

Jakmile se přidá cesta k souboru do aplikace, lze začít soubor nahrávat do zařízení. To se spustí po stlačení tlačítka „Start programming“. Pokud nahrání proběhne v pořádku, tak ve výpisu událostí bude vypsáno, že se vše podařilo a zobrazí se pouze jedna chybová hláška, která říká, že aplikace nemůže získat ID jádra zařízení. Tato hláška se zobrazuje z důvodu, že po nahrání programu do Wio-E5 mini se zařízení odpojí, i když ukazuje, že je stále připojeno. Úspěšné naprogramování zařízení je zobrazeno na obrázku (Obrázek 37).



Obrázek 37. Nahrání programu do zařízení (zdroj vlastní)

6.4 Nastavení serverových platform

6.4.1 Nastavení aplikace v platformě The Things Network

Pokud jsou zařízení nakonfigurována správně podle dat v platformě TTN, tak brána LoRaWAN bude přijímat data v reálném čase podobně, jako je na obrázku (Obrázek 38).

Time	Type	Data preview
22:34:22	Receive gateway status	Metrics: { txok: 0, rxin: 2, rxok: 2, rxfw: 2, ackr: 0, txin: 1 } Versions: { ttn-lw-gateway-server:
22:34:11	Send downlink message	Tx Power: 29.15 Data rate: SF12BW125
22:34:11	Send downlink message	Tx Power: 16.15 Data rate: SF7BW125
22:34:10	Receive uplink message	DevAddr: 26 0B 78 67 <> FCnt: 27 FPort: 2 Data rate: SF7BW125 SNR: 9.5 RSSI: -4
22:34:10	Receive uplink message	DevAddr: 26 0B 1C 66 <> FCnt: 27 FPort: 2 Data rate: SF7BW125 SNR: 9.5 RSSI: -19
22:34:01	Console: Stream reconnected	The stream connection has been re-established
22:33:52	Receive gateway status	Metrics: { ackr: 0, txin: 2, txok: 0, rxin: 2, rxok: 2, rxfw: 2 } Versions: { ttn-lw-gateway-server:
22:33:41	Send downlink message	Tx Power: 29.15 Data rate: SF12BW125
22:33:41	Send downlink message	Tx Power: 16.15 Data rate: SF7BW125
22:33:40	Receive uplink message	DevAddr: 26 0B 78 67 <> FCnt: 26 FPort: 2 Data rate: SF7BW125 SNR: 10 RSSI: -5
22:33:40	Receive uplink message	DevAddr: 26 0B 1C 66 <> FCnt: 26 FPort: 2 Data rate: SF7BW125 SNR: 7.5 RSSI: -19
22:33:22	Receive gateway status	Metrics: { ackr: 0, txin: 0, txok: 0, rxin: 1, rxok: 0, rxfw: 0 } Versions: { ttn-lw-gateway-server:
22:33:15	Connect gateway	

Obrázek 38. Reálná data v TTN v LoRaWAN bráně (zdroj vlastní)

V aplikaci je potřeba nastavit dekodér příchozích dat. Jelikož zařízení posílají data v hexadecimálním formát, kde první bajt je informace o stavu baterie a druhý je teplota, tak se použije jednoduchá JS (JavaScript) funkce. Funkce se jmenuje **decodeUplink** a přijímá parametr *input*. Ve funkci se inicializuje prázdný objekt *data*. Objekt se naplní klíčem *temperature* a hodnotou teploty, která se získá spojením prvních dvou bajtů a vynásobí se číslem 10. Výsledkem bude hodnota ve stupních Celsia s jednou desetinnou čárkou. Dále se objekt naplní klíčem *battery* a hodnotou ze třetího bajtu pole *input.bytes*.

Funkce pak vrací objekt *data*.

```
function decodeUplink(input) {
  var data = {};
  data.temperature = (input.bytes[0] << 8 | input.bytes[1]) / 10;
  data.battery = input.bytes[2];

  return {
    data:data,
  };
}
```

V aplikaci i v jednotlivých zařízeních se bude zobrazovat posloupnost bajtů v hexadecimálním formátu (tzv. payload) a také dekodovaná data o stavu baterie a teplotě tak, jak je to vidět na obrázku (Obrázek 39).

Applications > Footboards pipes meter > Live data

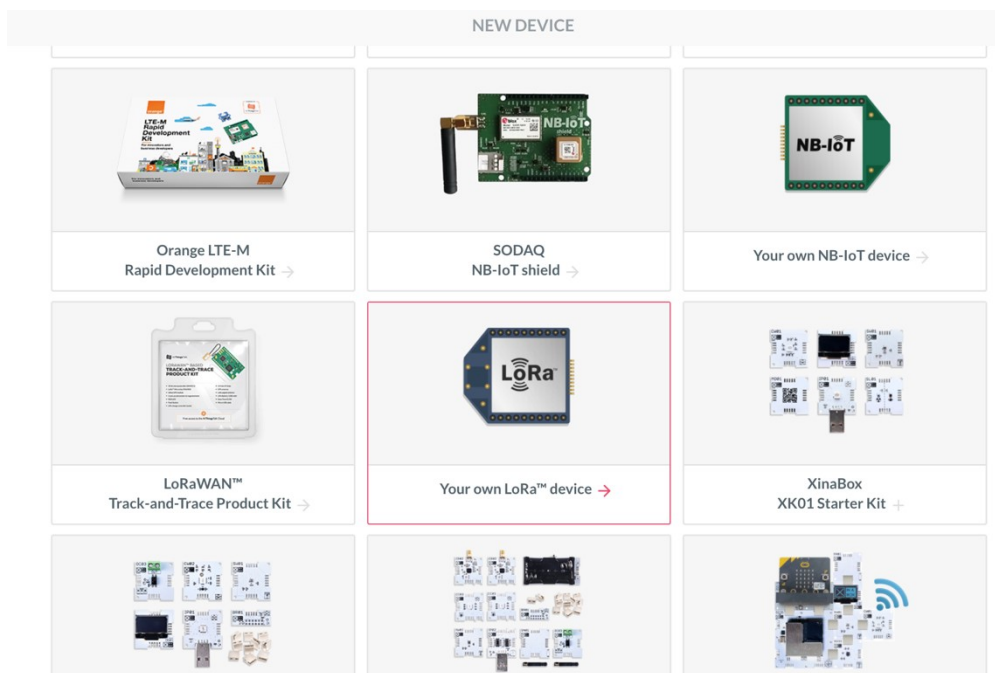
Time	Entity ID	Type	Data preview	Verbose stream	Export as JSON	Pause	Clear
↑ 22:35:40	e5-mini-stm32wle...	Forward uplink data message	DevAddr: 26 0B 78 67 <> Payload: { battery: 254, temperature: 52.2 }	<input type="checkbox"/>			
↑ 22:35:40	e5-mini-stm32wle...	Forward uplink data message	DevAddr: 26 0B 1C 66 <> Payload: { battery: 254, temperature: 48.5 }	<input type="checkbox"/>			
↑ 22:34:10	e5-mini-stm32wle...	Forward uplink data message	DevAddr: 26 0B 1C 66 <> Payload: { battery: 254, temperature: 48.5 }	<input type="checkbox"/>			
↑ 22:34:10	e5-mini-stm32wle...	Forward uplink data message	DevAddr: 26 0B 78 67 <> Payload: { battery: 254, temperature: 52.2 }	<input type="checkbox"/>			
ⓘ 22:34:04		Console: Stream reconnected	The stream connection has been re-established				
↑ 22:33:40	e5-mini-stm32wle...	Forward uplink data message	DevAddr: 26 0B 78 67 <> Payload: { battery: 254, temperature: 52.2 }	<input type="checkbox"/>			
↑ 22:33:40	e5-mini-stm32wle...	Forward uplink data message	DevAddr: 26 0B 1C 66 <> Payload: { battery: 254, temperature: 48.5 }	<input type="checkbox"/>			

Obrázek 39. Reálná data v TTN v aplikaci zařízení (zdroj vlastní)

6.4.2 Nastavení platformy AllThingsTalk Maker

Pro integraci dat s platformou AllThingsTalk Maker je nutné vytvořit nový tzv. ground, ve kterém se přidávají jednotlivá zařízení z TTN aplikace.

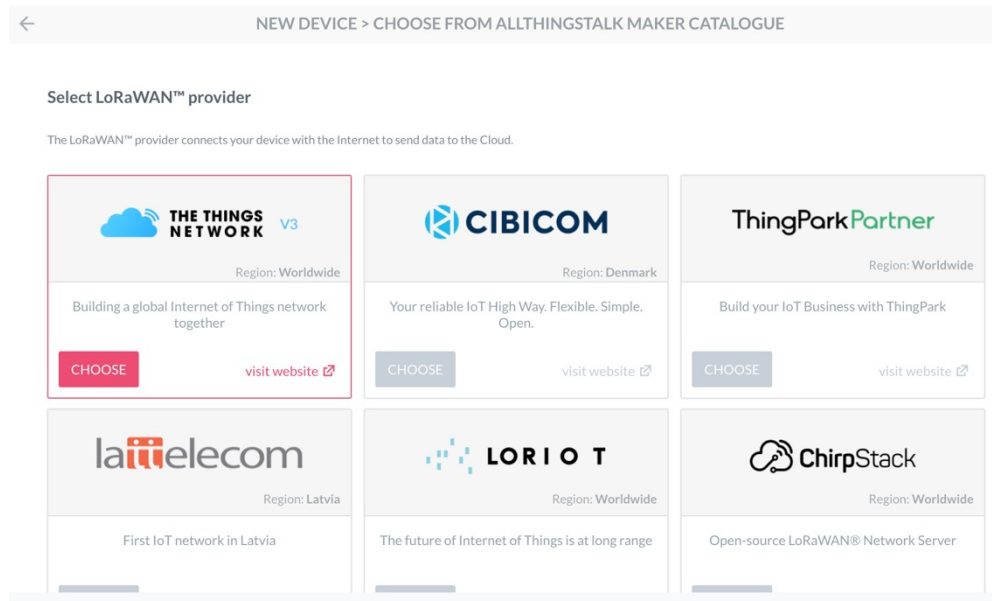
Po kliknutí na tlačítko *New Device* se vybere zařízení LoRa podle obrázku níže (Obrázek 40).



Obrázek 40. Přidání zařízení – výběr typu LoRa (zdroj vlastní)

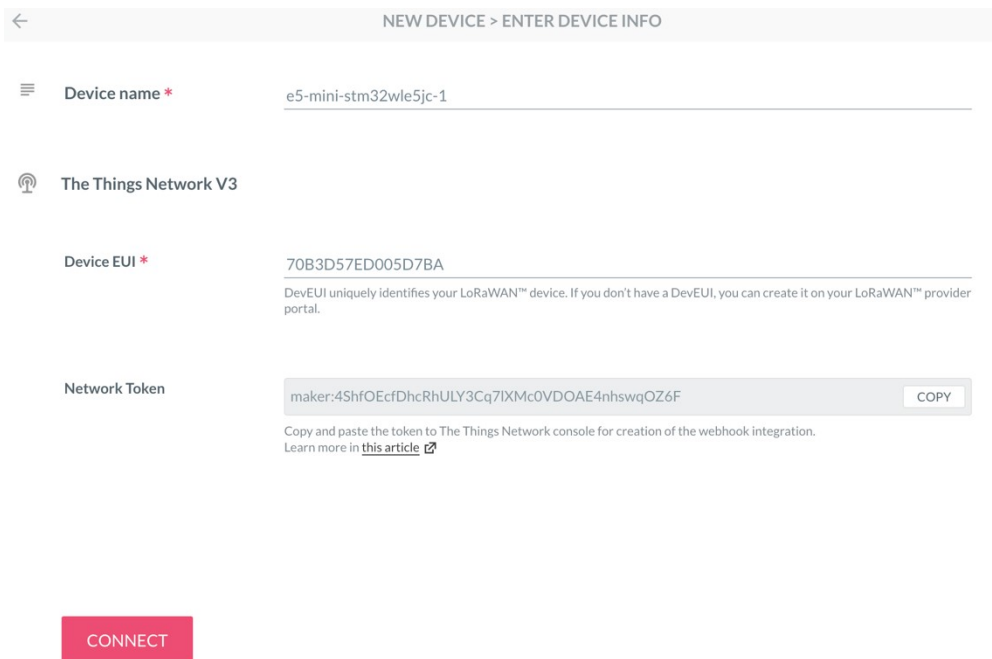
V dalším kroku se vybere platforma The Things Network, jako je na obrázku (Obrázek 41).

UTB ve Zlíně, Fakulta aplikované informatiky 81



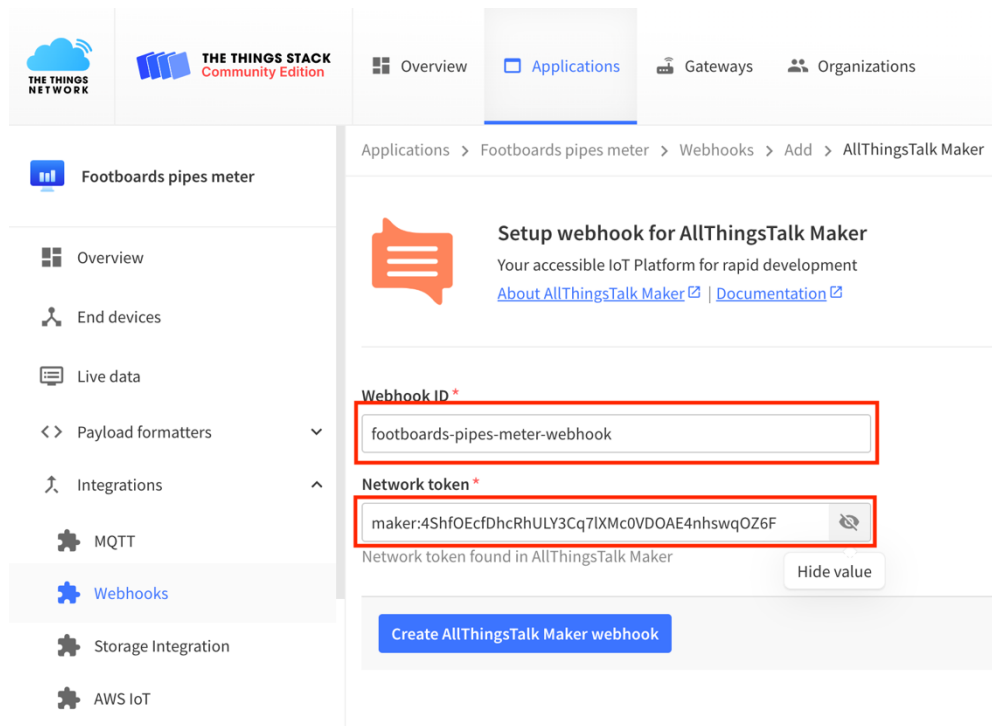
Obrázek 41. Přidání zařízení – výběr TTN (zdroj vlastní)

V posledním kroku se nastaví unikátní název zařízení a EUI zařízení, které se zkopíruje z platformy TTN u daného zařízení. Zde je potřeba zkopírovat síťový token, který se následně přidá do TTN aplikace. Zadané hodnoty budou vypadat jako na Obrázek 42.



Obrázek 42. Přidání zařízení – název a EUI (zdroj vlastní)

Pro integraci je potřeba vytvořit webhook v aplikaci TTN, kde se nastaví webhook pro platformu AllThingsTalk Maker, zvolí se unikátní ID a přidá se síťový token, který se zkopíroval u předchozího kroku. Výsledek je na následujícím obrázku (Obrázek 43).



Obrázek 43. Vytvoření integrace v TTN (zdroj vlastní)

Po vytvoření zařízení je nutné nastavit přijímání dat, proto se v zařízení nastaví konvertér příchozích dat (payload), vybere se konvertování pomocí JavaScriptu. Vytvořená funkce **converter** je podobná funkci **decodeUplink** z TTN aplikace. Je zde několik úprav, jelikož se pomocí webhooku posílají data v binárním formátu.

Proto je nejprve nutné převést binární data do hexadecimálního formátu, rozdělit hexadecimální data na části po dvou znacích a poté každou část převést na desítkové číslo. Na závěr se čísla namapují na odpovídající klíče v objektu JSON. Funkce vrací JSON objekt s názvem *data*.

```
function converter(code) {
  code = code.replace(/\\s/g, "");
  code = code.match(/.{1,2}/g);
  code = code.map(function (x) {
    return parseInt(x, 16);
  });
  var data = {
    "temperature": {
      "value": (code[0] << 8 | code[1])/10,
    },
    "battery": {
      "value": code[2],
    }
  }
}
```

```

    },
  };
  return JSON.stringify(data);
}

```

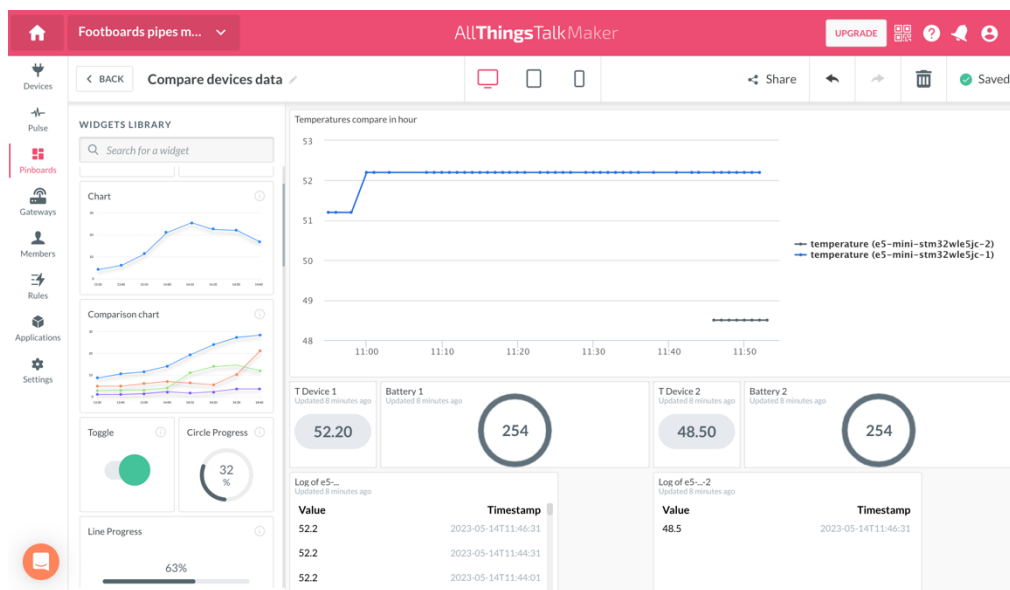
Dalším krokem je vytvoření prostředků zařízení, těmi jsou Teplota a Baterie.

Teplota se udělá podle obrázku níže (Obrázek 44). Obdobně se vytvoří i prostředek Baterie.

Obrázek 44. Vytvoření prostředku Teplota (zdroj vlastní)

V zařízení lze nastavit hlídacího psa, který zašle oznámení na e-mail nebo v platformě na webu, popř. mobilní aplikaci. Hlídací pes kontroluje status zařízení, zda je online a posílá data nebo ne.

Posledním krokem je vytvoření tabule, ve které lze generovat grafy v závislosti na teplotě a času jednoho a více zařízení. Dále je zde možnost vytvoření různých widgetů, jako je na obrázku (Obrázek 45).



Obrázek 45. Sestavení tabule s widgety (zdroj vlastní)

Tabuli je možné sdílet veřejně pomocí webového odkazu. Tabule demonstrátoru je zveřejněna na adrese <https://maker.allthingstalk.com/ep/641c530c62f7e300012ac731/maker:40nQi0Rj3pqjOWhJhwcCutOolXpWq5ESTx2ktdH2>.

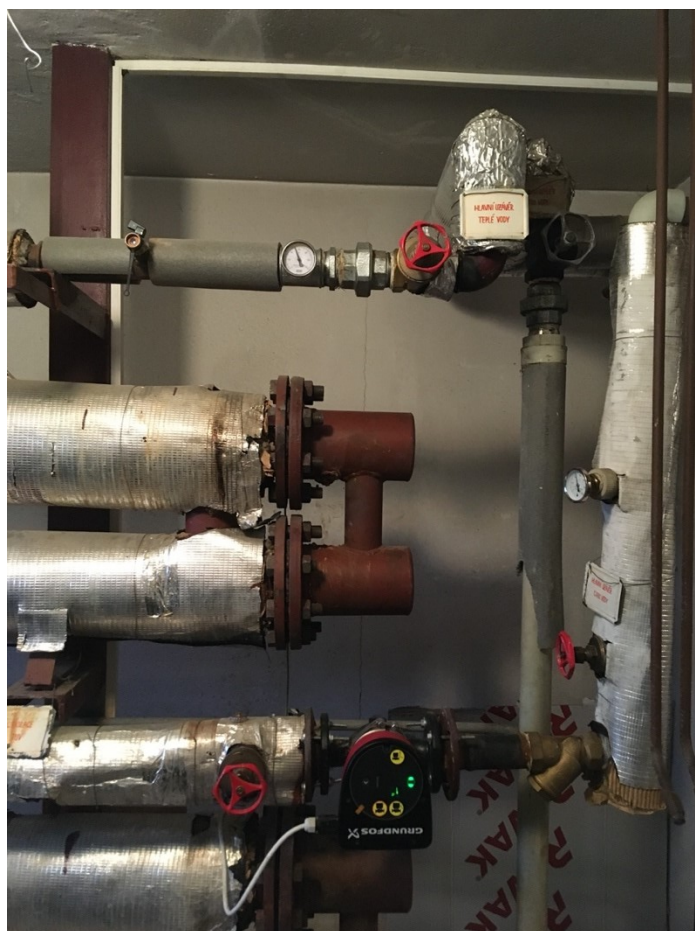
7 TRVALÁ INSTALACE PLATFORMY

Instalace platformy byla vyzkoušena v bytovém domě v Bystřici pod Hostýnem.

Po domluvě s SVJ v tomto bytovém domě byla platforma nainstalována ve sklepení bytového domu. Dále bylo umožněno připojení zařízení LoRaWAN brány k Wi-Fi síti jednoho z vlastníků. Pro zařízení LoRaWAN brány bylo dohodnuto využití elektrické sítě pro napájení.

Zařízení LoRaWAN brány bylo zprovozněno v místnosti s výměníkem tepla.

Koncová zařízení LoRaWAN byla připevněna na výměníku tepla. Jeden měřák byl připevněn na trubku přívodního okruhu teplé vody a druhý na trubku oběhového okruhu teplé vody. Měření teploty probíhalo na části výměníku na obrázku níže (Obrázek 46).



Obrázek 46. Výměník teplé vody – část pro měření
(zdroj vlastní)

Na těchto trubkách jsou namontovány ručičkové teploměry s jímkou, které zobrazují aktuální přesnou teplotu kapaliny v trubkách. V těchto místech jsou přichycena koncová

zařízení LoRaWAN, která jsou spolu napájena z powerbanky. Tento způsob napájení je zvolen z důvodu omezení elektrických zásuvek ve společných prostorech a v této místnosti. Z obou zařízení vedou senzory pod izolaci na trubkách. Senzory zde měří pouze teplotu ohřáté kovové trubky, proto zde dochází k nepatrné odchylce a je potřeba senzor zaizolovat a připevnit přímo na trubku.

Měření teploty vody je potřeba kontrolovat na více místech v bytovém domě. Dalším místem měření jsou stoupačky, jako na obrázku (Obrázek 47). V těchto prostorech se přichytí koncové zařízení a senzor měřící teplotu se připevní na trubku teplé vody pod izolaci. Je potřeba zamezit úniku tepla, a zaizolovat senzor na kovové trubce. Pokud bude trubka z plastu, měření nebude přesné.



Obrázek 47. Stoupačky – část pro měření (zdroj vlastní)

Naprogramování těchto zařízení a připojení do sítě LoRaWAN a serverových platform je popsáno v kapitole 6.

ZÁVĚR

Výsledkem této práce je navržení funkčního systému pro dálkové měření teploty vody v potrubí v bytových domech.

Hlavním bodem je zvolení bezplatné cloudové platformy, na kterou se připojují sestavená bezdrátová zařízení.

Zprvu bylo potřebné shromáždit informace o dostupných technologiích vhodných k vytvoření systému pro aplikace IoT. Následně se vybíral vhodný hardware pro stoprocentní funkčnost v teplém prostředí a senzor pro nejpřesnější možné měření teploty.

Dalším krokem byl výběr bezplatného portálového systému pro aplikace IoT. Portálový systém by měl být dostačující pro menší projekty. Platformy AWS IoT Core, Microsoft Azure IoT, IBM Watson IoT a Google Cloud IoT Core jsou velmi komplexní, ale pro menší projekty nevhodné, z důvodu nevyužití jejich potenciálu.

Systém se může využít k hlídání teplé vody po trase v celém domě, aby teplota neklesla pod stanovené stupně Celsia, které jsou bezpečné proti legionelle.

Legionella je bakterie, které se velmi dobře daří přežívat ve vlhku nebo v teplé vodě mezi 20-45 stupni Celsia. Přežívá ve vodovodním potrubí, zásobnících teplé vody nebo klimatizacích. Nemoci způsobené touto bakterií jsou smrtelné. K nákaze dochází po vypití kontaminované vody. Nejčastěji postihuje dýchací cesty.

Nákazy mohou být vyšší právě v dnešní době, kdy jsou ceny energií vysoké a lidé se pokoušejí ušetřit, kde se dá. Lidé méně topí a snižují ohřev zásobníků teplé vody, to může být nejčastějším důvodem vzniku nákazy touto bakterií.

Z důvodu hustě zastavěné oblasti byla pro demonstrátor zvolena komunikační technologie LoRaWAN. Dále bylo nutné vybrat zařízení, která dokážou komunikovat v této síti.

Pro vybranou komunikační technologii LoRaWAN je potřeba mít v blízkosti koncových zařízení připojenou LoRaWAN bránu, která s těmito zařízeními komunikuje. Z důvodu slabého pokrytí veřejnými LoRaWAN bránami bylo potřeba vytvořit vlastní LoRaWAN bránu. Využilo se zařízení PYGATE868, které bylo dostupné a vhodné pro tuto práci.

Při výběru zařízení byla problémem čipová krize, která ovlivnila celosvětový nedostatek čipů pro výrobu hardwarů, a tedy i dostupnost zařízení pro IoT. Původně se pro demonstrátor vybralo zařízení FiPy, ale z důvodu čipové krize byla společnost Pycom na pokraji bankrotu a upustili od výroby tohoto zařízení.

Pro demonstrátor bylo zvoleno dostupné zařízení Wio-E5 mini. Nevýhodou při programování tohoto zařízení je nutnost programovacího zařízení ST-Link, to však nahradila vývojová deska NUCLEO-WL55JC1.

Vybraný senzor DS18B20 je vhodný k připevnění na trubku, ale může zde dojít k nepřesnému měření, jelikož se neměří teplota vody, ale jen teplota ohřáté trubky. Nejlepší by bylo použití teploměru s jímkou, který by přímo snímal teplotu teplé vody.

Jakmile byla vybraná komunikační technologie, zařízení LoRaWAN brány a typ koncových zařízení, bylo nutné tato zařízení naprogramovat a nakonfigurovat. Po zprovoznění těchto zařízení a připojení s platformou The Things Network se nastavila integrace s platformou AllThingsTalk Maker, která umožňuje grafické zobrazení příchozích dat.

V platformě AllThingsTalk Maker lze přizpůsobit grafické zobrazení podle potřeb a zobrazit grafy příchozích dat v rámci hodiny, dne, týdne a měsíce. Vedle grafů je možné zobrazení nejaktuálnějších dat jednotlivých zařízení.

V této platformě lze nastavit upozornění, které sleduje aktivitu příchozích dat ze zařízení. Pokud je zařízení delší dobu neaktivní, upozornění se odešle e-mailem.

Jestliže budou data potřebná i po delší době, než je jeden měsíc, je možné z jednotlivých zařízení stáhnout všechna data ve formátu CSV na konci měsíce.

Při trvalé instalaci platformy je potřeba domluvit se s vlastníky SVJ nebo správcem domu a sepsat smlouvu o zapojení zařízení v prostoru výměníku teplé vody a stoupaček u jednotlivých bytů, aby byla možná správa těchto zařízení. Dále je potřeba domluvit přístup k internetové síti pro zařízení LoRaWAN brány.

Dosažení reálných dat o stavu teplé vody v bytovém domě nebylo na poslední chvíli umožněno z důvodu nedohodnutí se ze strany s SVJ. Nesvolili k připojení LoRaWAN brány k internetové síti.

SEZNAM POUŽITÉ LITERATURY

- [1] MOHANTY, Sachi Nandan, Jyotir Moy CHATTERJEE a Suneeta SATPATHY. *Internet of Things and Its Applications*. Cham: Springer, 2021. ISBN 978-3-030-77527-8.
- [2] 10 Best IoT Platforms for 2022. *Sam Solutions* [online]. 2022 [cit. 2022-05-12]. Dostupné z: <https://www.sam-solutions.com/blog/top-iot-platforms/>
- [3] IoT. *IBM Developer* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://developer.ibm.com/technologies/iot/>
- [4] Streamlining the development of your IoT applications by using an IoT platform. *IBM Developer* [online]. 2020 [cit. 2022-04-10]. Dostupné z: <https://developer.ibm.com/learningpaths/iot-getting-started-iot-development/iot-platforms/>
- [5] WAN, Jiafu, Iztok HUMAR a Daqiang ZHANG. *Industrial IoT Technologies and Applications*. Cham: Springer, 2016. ISBN 978-3-030-71061-3.
- [6] What Is a Smart City?. *IoT For All* [online]. 2018 [cit. 2022-05-12]. Dostupné z: <https://www.iotforall.com/what-is-smart-city>
- [7] Internet of Things In Smart Home. *Scand* [online]. 2019 [cit. 2022-05-12]. Dostupné z: <https://scand.com/company/blog/internet-of-things-in-smart-home/>
- [8] Getting to know MQTT. *IBM Developer* [online]. 2021 [cit. 2022-05-12]. Dostupné z: <https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/>
- [9] Getting Started with MQTT. *HiveMQ* [online]. 2020 [cit. 2022-05-12]. Dostupné z: <https://www.hivemq.com/blog/how-to-get-started-with-mqtt/>
- [10] A Complete Guide to IoT Protocols & Standards In 2021. *Nabto* [online]. 2020 [cit. 2022-05-12]. Dostupné z: <https://www.nabto.com/guide-iot-protocols-standards/>
- [11] Sigfox. *IoT portál* [online]. 2016 [cit. 2022-05-12]. Dostupné z: <https://www.iot-portal.cz/2016/02/26/sigfox/>
- [12] What is LoRaWAN® Specification. *LoRa Alliance* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://lora-alliance.org/about-lorawan/>
- [13] What is Narrowband IoT?. *Narrowband* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://www.narrowband.com/what-is-narrowband-iot>

- [14] Bluetooth Classic vs Bluetooth LE (5). *Appuals* [online]. 2021 [cit. 2022-05-12].
Dostupné z: <https://appuals.com/bluetooth-classic-vs-bluetooth-le-5/>
- [15] KERAMIDAS, Georgios a Michael HÜBNER. *Components and Services for IoT Platforms: Paving the Way for IoT Standards*. Cham: Springer, 2017. ISBN 978-3-319-42304-3.
- [16] Wi-Fi CERTIFIED HaLow. *Wi-Fi Alliance* [online]. c 2022 [cit. 2022-05-12].
Dostupné z: <https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-halow>
- [17] What is GSM? A simple overview of the GSM Communication Standard. *Technobyte* [online]. © 2016 - 2022 [cit. 2022-05-12]. Dostupné z: <https://technobyte.org/gsm-communication-standard/>
- [18] LTE for the IoT. *IOTEVOLUTIONWORLD* [online]. 2016 [cit. 2022-05-12].
Dostupné z: <https://www.iotevolutionworld.com/iot/articles/420026-lte-the-iot.htm>
- [19] LTE-M: What's in it for Your IoT Solution?. *IoT For All* [online]. 2021 [cit. 2022-05-12]. Dostupné z: <https://www.iotforall.com/lte-m-whats-in-it-for-your-iot-solution>
- [20] About us. *Raspberry Pi* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://www.raspberrypi.com/about/>
- [21] 8 best IoT hardware platforms (2022 edition). *Hologram* [online]. 2021 [cit. 2022-05-12]. Dostupné z: <https://www.hologram.io/blog/iot-hardware>
- [22] Raspberry Pi Zero 2 W. *Raspberry Pi* [online]. c 2022 [cit. 2022-05-12].
Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/>
- [23] Arduino Hardware. *Arduino* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://www.arduino.cc/en/hardware>
- [24] Feather. *Adafruit* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://learn.adafruit.com/category/feather>
- [25] Adafruit Feather 32u4 FONA. *Adafruit* [online]. c 2022 [cit. 2022-05-12].
Dostupné z: <https://www.adafruit.com/product/3027>
- [26] Argon, IoT Starter Kit. *Particle* [online]. c 2020 [cit. 2022-05-12]. Dostupné z: https://store.particle.io/products/argon-kit?_pos=1&_sid=5d62765c4&_ss=r
- [27] Hardware. *Espressif* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://www.espressif.com/en/products/hardware>

- [28] Hardware. *Pycom* [online]. c 2020 [cit. 2022-05-12]. Dostupné z:
<https://pycom.io/products/hardware/>
- [29] Season Group's acquisition of Pycom Ltd-a global IoT technology startup with a vision. *Season Group* [online]. 2022 [cit. 2023-04-24]. Dostupné z:
<https://www.seasongroup.com/news/season-groups-acquisition-of-pycom-ltd-a-global-iot-technology-startup-with-a-vision/>
- [30] About Seeed. *Seeed Studio* [online]. 2023 [cit. 2023-04-24]. Dostupné z:
https://www.seeedstudio.com/about_seed
- [31] Wiki Seeed Studio. *Seeed Studio* [online]. 2023 [cit. 2023-04-24]. Dostupné z:
https://wiki.seeedstudio.com/Getting_Started
- [32] LoRa-LoRaWAN Modules. *Seeed Studio* [online]. 2023 [cit. 2023-04-24].
Dostupné z: <https://www.seeedstudio.com/LoRa-LoRaWAN-Modules-c-1950.html>
- [33] Who we are - STMicroelectronic. *STMicroelectronics* [online]. 2023 [cit. 2023-04-24]. Dostupné z:
https://www.st.com/content/st_com/en/about/st_company_information/who-we-are.html
- [34] STM32 Nucleo Boards. *STMicroelectronics* [online]. 2023 [cit. 2023-04-24].
Dostupné z: <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>
- [35] STM32 Nucleo-64 development board with STM32WL55JC MCU, SMPS, supports Arduino and morpho connectivity. *STMicroelectronics* [online]. 2023 [cit. 2023-04-24]. Dostupné z: <https://www.st.com/en/evaluation-tools/nucleo-wl55jc.html>
- [36] Middleware. *IBM* [online]. 2021 [cit. 2022-05-12]. Dostupné z:
<https://www.ibm.com/cloud/learn/middleware>
- [37] Kaa open-source. *Kaa IoT* [online]. 2023 [cit. 2023-04-30]. Dostupné z:
<https://www.kaaiot.com/kaa-open-source>
- [38] Kaa IoT Platform Features for Enterprise IoT Projects. *KaaIoT* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://www.kaaiot.com/products/overview>
- [39] ThingSpeak. *MathWorks* [online]. c 1994-2022 [cit. 2022-05-12]. Dostupné z:
<https://se.mathworks.com/help/thingspeak/>
- [40] API Reference. *MathWorks* [online]. c 1994-2022 [cit. 2022-05-12]. Dostupné z:
<https://se.mathworks.com/help/thingspeak/channels-and-charts-api.html>

- [41] What is ThingsBoard?. *ThingsBoard* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://thingsboard.io/docs/getting-started-guides/what-is-thingsboard/>
- [42] Remote Procedure Call (RPC). *Tech Target* [online]. 2021 [cit. 2022-05-12]. Dostupné z: <https://www.techtarget.com/searchapparchitecture/definition/Remote-Procedure-Call-RPC>
- [43] ThingsBoard Pricing. *ThingsBoard* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://thingsboard.io/pricing/>
- [44] Features. *Apache StreamPipes* [online]. c 2019 [cit. 2022-05-12]. Dostupné z: <https://streampipes.apache.org/feature-overview.html>
- [45] StreamPipes Connect. *Apache StreamPipes* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://streampipes.apache.org/docs/docs/use-connect.html>
- [46] IoT Core. *Google Cloud* [online]. c 2007-2022 [cit. 2022-05-12]. Dostupné z: <https://cloud.google.com/iot-core/>
- [47] Google Cloud IoT Core pricing. *Google Cloud* [online]. c 2007-2022 [cit. 2022-05-12]. Dostupné z: <https://cloud.google.com/iot/pricing>
- [48] What is Rule-Engine?. *Medium* [online]. 2019 [cit. 2022-05-12]. Dostupné z: <https://medium.com/@er.rameshkatiyar/what-is-rule-engine-86ea759ad97d>
- [49] AWS IoT Device Shadow service. *Amazon Web Services* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html>
- [50] AWS IoT Core Pricing. *AWS Amazon* [online]. 2023 [cit. 2023-04-30]. Dostupné z: https://aws.amazon.com/iot-core/pricing/?loc=ft#AWS_Free_Tier_with_AWS_IoT_Core
- [51] IBM Watson IoT Platform. *IBM Watson IoT Platform* [online]. c 2021 [cit. 2022-05-12]. Dostupné z: <https://internetofthings.ibmcloud.com/>
- [52] 10 Best IoT Platforms To Watch Out In 2022. *Software Testing Help* [online]. 2022 [cit. 2022-05-12]. Dostupné z: <https://www.softwaretestinghelp.com/best-iot-platforms/>
- [53] Azure IoT. *Microsoft* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://azure.microsoft.com/en-us/overview/iot/>
- [54] The workbench to get you started with IoT. *The Things Network* [online]. c 2022 [cit. 2022-05-12]. Dostupné z: <https://www.thethingsnetwork.org/>

- [55] The Things Stack integration. *The Things Industries* [online]. 2023 [cit. 2023-04-30]. Dostupné z: <https://www.thethingsindustries.com/docs/integrations/>
- [56] Secure elements. *The Things Industries* [online]. 2023 [cit. 2023-04-30]. Dostupné z: <https://www.thethingsindustries.com/secure-elements/>
- [57] AllThingsTalk Docs. *AllThingsTalk* [online]. 2023 [cit. 2023-04-25]. Dostupné z: <https://docs.allthingstalk.com/>
- [58] Tepelný výměník na ohřev vody. *Energie pro lidi* [online]. [cit. 2023-04-25]. Dostupné z: <https://www.energiepro lidi.cz/tepelny-vymenik-na-ohrev-vody/>
- [59] Dálkové zásobování teplem: Jak funguje a z čeho se skládá výměňková stanice. *ESTAV* [online]. 2021 [cit. 2023-04-25]. Dostupné z: <https://www.estav.cz/cz/9607.dalkove-zasobovani-teplem-jak-funguje-a-z-ceho-se-sklada-vymenikova-stanice>
- [60] Průběh spotřeby teplé vody v bytových domech. *ASB Portal* [online]. 2014, 10. listopadu 2014 [cit. 2023-04-25]. Dostupné z: <https://www.asb-portal.cz/stavebnictvi/technicka-zarizeni-budov/vytapeni/prubeh-spotreby-teple-vody-v-bytovych-domech>
- [61] *DHT11 Humidity & Temperature Sensor* [online]. 2018 [cit. 2022-05-12]. Dostupné z: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- [62] NTC Thermistor. *EEPower* [online]. 2021 [cit. 2022-05-12]. Dostupné z: <https://eepower.com/resistor-guide/resistor-types/ntc-thermistor/#>
- [63] *DHT20-2498062.pdf* [online]. 2021 [cit. 2022-05-12]. Dostupné z: <https://cz.mouser.com/datasheet/2/813/DHT20-2498062.pdf>
- [64] *AM2301.pdf* [online]. 2013 [cit. 2022-05-12]. Dostupné z: <https://hobbytronics.com.pk/Files/Datasheets/AM2301.pdf>
- [65] Adafruit BME280 Humidity + Barometric Pressure + Temperature Sensor Breakout. *Adafruit* [online]. 2022 [cit. 2022-05-12]. Dostupné z: <https://learn.adafruit.com/adafruit-bme280-humidity-barometric-pressure-temperature-sensor-breakout>
- [66] *LM35 Precision Centigrade Temperature Sensors datasheet (Rev. H)* [online]. 2022 [cit. 2022-05-12]. Dostupné z: <https://www.ti.com/lit/ds/symlink/lm35.pdf?HQS=dis-mous-null-mousermode-dsf->

pf-null-

www&ts=1647592616894&ref_url=https%253A%252F%252Ffeu.mouser.com%252F

- [67] *DS18B20 - Programmable Resolution 1-Wire Digital Thermometer* [online]. 2019 [cit. 2022-05-12]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [68] EM300-TH-868M. *ESCAD Trade* [online]. c 2020 [cit. 2022-05-12]. Dostupné z: <https://www.escadtrade.cz/milesight-em300-th-868m-lorawan-senzor-pro-mereni-teploty-a-vlhkosti/>
- [69] Legionella. *Legionella* [online]. 2023 [cit. 2023-05-15]. Dostupné z: <https://legionella.cz/>
- [70] *Pycom_002_Specsheets_FiPy_v2.pdf* [online]. In: . 2022 [cit. 2023-04-28]. Dostupné z: https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_FiPy_v2.pdf
- [71] Wiki Seeed Studio LoRa E5 mini. *Seeed Studio* [online]. 2023 [cit. 2023-04-24]. Dostupné z: https://wiki.seeedstudio.com/LoRa_E5_mini
- [72] *Pycom_002_Specsheets_Pygate_v1.pdf* [online]. In: . 2022 [cit. 2023-04-28]. Dostupné z: https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_Pygate_v1.pdf
- [73] Python vs C. *Udemy.com* [online]. 2023 [cit. 2023-05-07]. Dostupné z: <https://blog.udemy.com/python-vs-c/>
- [74] MicroPython. *MicroPython* [online]. 2023 [cit. 2023-05-09]. Dostupné z: <https://micropython.org/>
- [75] C programming language. *Britannica* [online]. 2022 [cit. 2023-05-07]. Dostupné z: <https://www.britannica.com/technology/C-computer-programming-language>
- [76] STM32CubeMX. *STMicroelectronics* [online]. 2023 [cit. 2023-05-07]. Dostupné z: <https://www.st.com/en/development-tools/stm32cubemx.html>
- [77] STM32CubeProgrammer. *STMicroelectronics* [online]. 2023 [cit. 2023-05-07]. Dostupné z: <https://www.st.com/en/development-tools/stm32cubeprog.html>

[78] VS Code. *Visual Studio* [online]. 2023 [cit. 2023-05-07]. Dostupné z:
<https://code.visualstudio.com/>

[79] STM32CubeIDE. *STMicroelectronics* [online]. 2023 [cit. 2023-05-07]. Dostupné
z: <https://www.st.com/en/development-tools/stm32cubeide.html>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AMQP	Advanced Message Queuing Protocol – pokročilý protokol fronty zpráv
API	An application programming interface – aplikační programovací rozhraní
AppKey	Application Key
AppSKey	Application Session Key
AWS	Amazon Web Services
C	Celsius
CoAP	Constrained Application Protocol – omezený aplikační protokol
CPL	Combined Programming Language
CSV	Comma-separated values
DevEUI	Device EUI
DIY	Do It Yourself – udělej si sám
EUI	Extended Unique Identifier
FIPS	Federal Information Processing Standards – Federální standardy zpracování informací
GHz	Gigahertz
GSM	Global System for Mobile communication
hPa	Hektopascal
HTTP	Hypertext Transfer Protocol – hypertextový protokol pro přenos dokumentů
HTTPS	Hypertext Transfer Protocol Secure – zabezpečený hypertextový protokol pro přenos dokumentů
IBM	International Business Machines Corporation

I2C	Inter-Integrated Circuit
ID	Identification
IDE	Integrated Development Environment – vývojové prostředí
IP	Internet Protocol – internetový protokol
int16_t	Datový typ 16bitového celého čísla
IPSP	Internet Protocol Support Profile – profil podpory internetových protokolů
JS	JavaScript
kHz	Kilohertz
LE	Low Energy
LED	Light-Emitting Diode
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
LTE	Long Term Evolution
LTE-M	Long Term Evolution Machine Type Communication
mAh	Miliampér hodin
MHz	Megahertz
MQTT	Message Queue Telemetry Transport
ms	Milisekunda
NB	Narrowband
NFC	Near Field Communication

NTC	Negative Temperature Coefficient
NTP	Netwo Time Protocol
NwkKey	Network Key
NwkSKey	Network Session Key
OSN	Organizace spojených národů
OTAA	Over the Air Activation
PoE	Power over Ethernet
RAM	Random-access memory
REST	Representational state transfer
RFID	Radio Frequency Identification
RTC	Real Time Clock
SCL	Serial Clock
SD	Secure Digital
SDK	Software Development Kit
SDA	Serial Data
SPI	Serial Peripheral Interface
SSID	Service Set Identifier
SSL	Secure Sockets Layer
ST	SGS-Thomson
STM	SGS-Thomson Microcontroller
STM32	SGS-Thomson Microcontroller 32-bit microcontrollers

SVJ	Společenství vlastníků jednotek
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTN	The Things Network
TTI	The Things Industries
TTS	The Things Stack
UART	Universal Asynchronous Receiver-Transmitter
uint16_t	Datový typ 16bitového celého čísla bez znaménka
uint32_t	Datový typ 32bitového celého čísla bez znaménka
uint8_t	Datový typ 8bitového celého čísla bez znaménka
V	Volt
VS	Visual Studio
Wi-Fi	Wireless Fidelity

SEZNAM OBRÁZKŮ

Obrázek 1. Application of IoT in modern age [1]	15
Obrázek 2. MQTT Publish/Subscribe Architecture [9]	17
Obrázek 3. Raspberry Pi Zero 2 [22]	22
Obrázek 4. Arduino Nano 33 IoT [23]	23
Obrázek 5. Adafruit Feather 32u4 FONA [25]	24
Obrázek 6. Particle Argon IoT [26]	25
Obrázek 7. Espressif ESP32-S3-DevKitC-1 [27]	25
Obrázek 8. PYCOM FiPy [28]	26
Obrázek 9. Wio-E5 mini Dev Board [32]	27
Obrázek 10. STM32 Nucleo-64 development board [35]	29
Obrázek 11. Senzor teploty a vlhkosti – DHT11 [61]	36
Obrázek 12. Senzor teploty a vlhkosti – DHT20 [63]	38
Obrázek 13. Senzor teploty a vlhkosti – DHT21 [64]	39
Obrázek 14. Senzor teploty, vlhkosti a tlaku – BME280 [65]	40
Obrázek 15. Senzor teploty – LM35DZ [66]	41
Obrázek 16. Senzor teploty – DS18B20 [67]	42
Obrázek 17. Senzor teploty a vlhkosti – EM300-TH-868M [68]	43
Obrázek 18. Wio-E5 mini (zdroj vlastní)	47
Obrázek 19. PYGATE868 (zdroj vlastní)	48
Obrázek 20. Nastavení Pycom firmware Upgrade (zdroj vlastní)	53
Obrázek 21. Úspěšná aktualizace firmware (zdroj vlastní)	54
Obrázek 22. Registrace LoRa brány v TTN (zdroj vlastní)	55
Obrázek 23. Vytvoření aplikace v TTN (zdroj vlastní)	66
Obrázek 24. Registrace zařízení – základní nastavení (zdroj vlastní)	67
Obrázek 25. Registrace zařízení – rozšířené nastavení (zdroj vlastní)	68

Obrázek 26. Cesta k souboru „project“ (zdroj vlastní).....	69
Obrázek 27. Nastavení výstupu po sestavení (zdroj vlastní).....	70
Obrázek 28. Nastavení regionu a cyklu přenosu dat (zdroj vlastní).....	71
Obrázek 29. Nastavení hodnot „DevEUI“ a „Device address“ (zdroj vlastní).....	72
Obrázek 30. Nastavení hodnot „AppSKey“ a „NwkSKey“ (zdroj vlastní).....	73
Obrázek 31. Cesta k souboru „lora_app.c“ (zdroj vlastní).....	73
Obrázek 32. Sestavení programu (zdroj vlastní).....	75
Obrázek 33. Propojení desek Wio-E5 mini a NUCLEO-WL55JC1 (zdroj pdf soubor v příloze).....	75
Obrázek 34. Nastavení hodnot pro připojení zařízení (zdroj vlastní).....	76
Obrázek 35. Nastavení ochrany (zdroj vlastní).....	77
Obrázek 36. Cesta k souboru s programem (zdroj vlastní).....	77
Obrázek 37. Nahrání programu do zařízení (zdroj vlastní).....	78
Obrázek 38. Reálná data v TTN v LoRaWAN bráně (zdroj vlastní).....	79
Obrázek 39. Reálná data v TTN v aplikaci zařízení (zdroj vlastní).....	80
Obrázek 40. Přidání zařízení – výběr typu LoRa (zdroj vlastní).....	80
Obrázek 41. Přidání zařízení – výběr TTN (zdroj vlastní).....	81
Obrázek 42. Přidání zařízení – název a EUI (zdroj vlastní).....	81
Obrázek 43. Vytvoření integrace v TTN (zdroj vlastní).....	82
Obrázek 44. Vytvoření prostředku Teplota (zdroj vlastní).....	83
Obrázek 45. Sestavení tabule s widgety (zdroj vlastní).....	84
Obrázek 46. Výměník teplé vody – část pro měření (zdroj vlastní).....	85
Obrázek 47. Stoupačky – část pro měření (zdroj vlastní).....	86

SEZNAM TABULEK

Tabulka 1. Rozsah a přesnost měření – DHT11 [61]	37
Tabulka 2. Rozsah a přesnost měření – DHT20 [63]	38
Tabulka 3. Rozsah a přesnost měření – DHT21 [64]	39
Tabulka 4. Rozsah a přesnost měření – BME280 [65]	40
Tabulka 5. Rozsah a přesnost měření – LM35DZ [66]	41
Tabulka 6. Rozsah a přesnost měření – DS18B20 [67]	42
Tabulka 7. Rozsah a přesnost měření – EM300-TH-868M [68]	43

SEZNAM PŘÍLOH

P I CD

PŘÍLOHA P I: CD

- Bakalářská práce v elektronické podobě
- Postup propojení Nucleo WL55JC1 a Wio-E5 mini
- Zdrojové kódy programů zařízení LoRa brány a koncových zařízení