



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

**INTERNET OF THINGS ZAŘÍZENÍ S PODPOROU
THREAD A 6LOWPAN**
INTERNET OF THINGS DEVICE BASED ON THREAD AND 6LOWPAN

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DAVID PIŠKULA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PETR MUSIL

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Piškula David**

Obor: Informační technologie

Téma: **Internet of Things zařízení s podporou Thread a 6LoWPAN
Internet of Things Device Based on Thread and 6LoWPAN**

Kategorie: Vestavěné systémy

Pokyny:

1. Prostudujte dostupnou literaturu týkající se fenomenu Internetu věcí (Internet of Things - IoT).
2. Seznamte se s používanými komunikačními rozhraními a protokoly využitými v IoT. Zaměřte se na rozhraní Thread a protokol 6LoWPAN.
3. Navrhněte zařízení spadající do kategorie IoT.
4. Zařízení realizujte a otestujte.
5. Zhodnoťte výsledky práce a diskutujte případné pokračování nebo rozšíření práce.

Literatura:

- Dle pokynů vedoucího

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

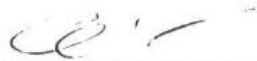
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Musil Petr, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta Informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Sochetůvova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Internet věcí patří mezi nejnovější fenomény informačních technologií. Tato práce popisuje vývoj a budoucnost internetu věcí a poskytuje přehled jeho současného využití. Dále se pak věnuje studiu komunikačních rozhraní a protokolů, vhodných pro jeho realizaci, zejména standardu IEEE 802.15.4, protokolu 6LoWPAN a technologie Thread. Vysvětluje také pojem Cloud Computing, a s ním spojené typy databází, a platformy, které jej zprostředkovávají. Součástí projektu je návrh, implementace a testování řešení založeného na Thread.

Abstract

Internet of Things is among the newest phenomena of information technologies. This work describes the evolution and future of the Internet of Things and provides an overview of its current applications, then summarize the communication interfaces and protocols usable for the creation of the Internet of Things. The main focus is put especially on the IEEE 802.15.4 standard, 6LoWPAN protocol and the Thread technology. It also explains the term of Cloud Computing, platforms that are providing it and related types of databases. The design, implementation and testing of a Thread based solution are part of this thesis.

Klíčová slova

Internet věcí, 6LoWPAN, Thread, Zigbee, IEEE 802.15.4, RFID, Bluetooth, Z-wave, Cloud, Zigbee PRO, Zigbee IP, SQL, NoSQL, chytré domácnosti, chytré obchody

Keywords

Internet of Things, 6LoWPAN, Thread, Zigbee, IEEE 802.15.4, RFID, Bluetooth, Z-wave, Cloud, Zigbee PRO, Zigbee IP, SQL, NoSQL, home automation, smart shops

Citace

Piškula David: Internet of Things zařízení s podporou Thread a 6LoWPAN, bakalářská práce, Brno, FIT VUT v Brně, 2017

Internet of Things zařízení s podporou Thread a 6LoWPAN

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petra Musila. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
David Piškula
12. května 2017

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Petru Musilovi za odbornou pomoc a cenné rady při zpracování mé bakalářské práce.

© David Piškula, 2017

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod.....	2
2	Internet věcí	3
2.1	Co je internet věcí.....	3
2.2	Vývoj a přehled technologií.....	4
2.3	Využití internetu věcí	5
2.4	Problémy.....	7
2.5	Budoucnost	8
2.6	Protokoly a komunikační rozhraní.....	8
2.7	Raspberry Pi 2 B.....	13
3	Cloudové služby pro IoT.....	17
3.1	Databázové jazyky SQL a NoSQL	17
3.2	Cloud	18
3.3	Cloudové platformy	19
4	Návrh.....	21
4.1	Specifikace zadání	21
4.2	Výběr HW.....	21
4.3	Výběr operačního systému	21
4.4	Výběr komunikačního protokolu	22
4.5	Problémy při návrhu sítě zařízení	22
4.6	Výběr Cloudové platformy	23
5	Implementace a testování.....	27
5.1	Koncové zařízení FRDM-KW41Z.....	27
5.2	USB Dongle USB-KW41Z.....	28
5.3	Raspberry Pi 2 B.....	28
5.4	Cloud	29
5.5	Testování	30
6	Závěr	32

1 Úvod

Internet je v dnešní době každodenní součástí životů většiny lidí ve vyspělých zemích světa. S příchodem nových technologií, jako jsou chytré telefony, tablety nebo chytré hodinky, se internet dostává do nových oblastí a jeho využití se tak neustále rozšiřuje. Díky nepřetržitému vývoji a zmenšování čipů, mikroprocesorů a jiných elektronických součástek, se k síti připojují i velmi malá zařízení. Společně s evolucí hardware se ale také zlepšuje software, umožňující tato malá zařízení využít co nejefektivněji a nejlevněji. Z těchto důvodů se objevují tendence připojit do internetu i věci, které lidé doteď využívali manuálně, a zautomatizovat tak co největší část světa. Vizionáři si představují svět, který je kompletně připojený k internetu, a kde počítače ovládají vše, včetně domácností, dopravy, průmyslu, zemědělství a zdravotnictví.

Cílem této práce je obecně i konkrétně popsat fenomén zvaný internet věcí, který v poslední době rychle nabývá popularity, a navrhnout a implementovat systém spadající do této kategorie informatiky. Práce nabízí přehled dostupných technologií, diskutuje je a implementuje příklad jejich využití.

V první sekci je popsán samotný internet věcí, jeho vznik, vývoj a budoucnost. Dále jsou pak podrobněji probrány komunikační technologie IEEE 802.15.4, 6LoWPAN, Zigbee PRO a Thread, což jsou jedny z nejpoužívanějších technologií týkajících se tohoto fenoménu. Následně je popsáno zařízení Raspberry Pi 2 B, které se často používá právě pro tvorbu sítí internetu věcí. V další sekci jsou rozebrány databázové technologie SQL a NoSQL a Cloud, který je neoddelitelnou součástí internetu věcí, a platformy, které umožňují jeho využití. Navazující část diskutuje dostupné technologie a znázorňuje návrh praktické tvorby komerčně využitelné sítě internetu věcí, a problémy, které se při této tvorbě vyskytly. V předposlední kapitole je ukázána finální implementace navržené sítě a její testování. V závěrečné sekci je celé řešení shrnuto, a jsou probrány další možné kroky k jeho rozšíření.

2 Internet věcí

V této kapitole je rozebrán vznik internetu věcí a internet věcí samotný. Jsou popsány technologie, které ho tvoří, jeho budoucnost a s ním spojené problémy. Nakonec jsou podrobněji popsány technologie 6LoWPAN a Thread a zařízení Raspberry Pi 2 B. Jelikož jsou zmiňované technologie velmi moderní a nejsou zatím dostatečně knižně publikovány, bylo potřeba jako zdroje informací využít i referenční manuály, internetové dokumenty a popisy programovacích knihoven.

2.1 Co je internet věcí

Internet věcí [1, 2], anglicky Internet of Things (IoT), je síť zařízení, které mezi sebou komunikují a spolupracují pomocí drátových i bezdrátových připojení. S využitím senzorů, elektroniky a správného software pak tyto věci vytváří autonomní prostředí, fungující bez lidského zásahu.

Název tomuto fenoménu dal v roce 1999 Kevin Ashton při prezentaci pro společnost P&G týkající se technologie RFID¹. Jeho cílem bylo zajistit díky této technologii, aby počítače měly přehled a informace o věcech bez toho, že by jim je poskytl člověk. Takto by lidé jednoduše získali přehled o daných věcech a přesně by věděli, kdy je potřeba je vyměnit nebo opravit. Vedlo by to ke snížení nákladů na údržbu a omezení odpadu a ztrát.

Záměrem internetu věcí je propojení věcí pomocí nějaké komunikační technologie a tím vytvoření entit, jejichž spolupráce vede k dosažení určitých cílů. Každá taková věc nebo zařízení musí být jednoznačně identifikovatelná a musí mít schopnost dorozumívat se s ostatními věcmi a pracovat nezávisle na člověku. Dále je často potřeba schopnosti snímat své okolní prostředí pomocí senzorů a zaznamenávat a odesílat takto získaná data. Na tyto data a informace pak reagují ostatní věci, ať už změnou svého chování, upozorněním uživatele na nějaký problém, předáním informací dále do nějakého úložiště a podobně.

Tohle autonomní prostředí má pak za úkol nejen usnadnit člověku život, ale také například snížit negativní dopad technologie na životní prostředí, zmenšit spotřebu energií nebo redukovat náklady na provozování domácnosti nebo jakékoliv jiné budovy či oblasti. Díky zabudovaným senzorům lze zefektivnit využití elektřiny zařízeními a spotřebiči v domácnostech i v průmyslu. Zařízení mohou také pravidelně předávat informace o svém stavu, své poloze a o potřebě údržby, čímž se dá předejít nehodám nebo zbytečným peněžním i materiálním ztrátám.

Díky postupnému vývoji technologií se původní vize expandovala, a dochází ke snaze propojit vše, včetně živých objektů, jako jsou rostliny, zvířata, nebo i lidi. Takhle rozšířená verze internetu věcí bývá často nazývána internetem všeho. Poskytuje například možnost sledovat zdraví dobytka a způsob jejich pohybu.

Úspěšnost internetu věcí by kvůli těmto vlastnostem měla velmi dobrý dopad na ekonomiku, životní prostředí, průmysl, zdravotnictví i celkové pohodlí jeho uživatelů.

¹ ASHTON, Kevin. That 'Internet of Things' Thing. *RFID Journal* [online]. [cit. 2015-12-27]. Dostupné z: <http://www.rfidjournal.com/articles/view?4986>

2.2 Vývoj a přehled technologií

Technologie umožňující vznik internetu věcí se posledních několik let rychle rozvíjí. Vývojové a vědecké týmy objevují nové způsoby komunikace mezi zařízeními, vznikají čím dál menší a efektivnější mikrokontroléry a mikroprocesory a přichází chytrá zařízení. Díky tomuto se rychle rozšiřuje množství věcí, které lze do internetu věcí zakomponovat.

RFID

Jedna z prvních technologií, která přibližovala vznik internetu věcí, byl Radiofrekvenční systém identifikace, anglicky Radio-Frequency Identification (RFID)². Tato technologie nabízí podobnou funkčnost jako čárové kódy. Stejně, jako čárový kód, se musí i RFID zařízení snímat, ale není potřeba, aby snímač byl v bezprostřední blízkosti daného zařízení. RFID funguje až na vzdálenost 6 metrů. Umožňuje tak uživateli rychle si načíst informace o věcech v jeho okolí bez potřeby, aby je někde ručně zadával nebo někde vyhledával.

Bluetooth

Další technologií, která už se na trhu pohybuje několik let, je Bluetooth³. Od svého vzniku v roce 1994 se již významně vyvinula, ale původně získala popularitu, protože umožňovala bezdrátový přenos zvuku do sluchátek a hands-free připojení do aut. V dnešní době existuje verze Bluetooth s velmi nízkou spotřebou energie, která umožňuje tvorbu malých senzorů komunikujících mezi sebou nebo s jinými zařízeními, což z ní dělá populární volbu pro vývoj sítě pro internet věcí.

Z-wave

Z-wave⁴ je nejpobulárnější technologií pro chytré domovy. Využívá mesh topologie a jednotlivé zařízení mezi sebou komunikují pomocí transceiverů na krátkou vzdálenost. Z-wave pracuje v pásmu 900MHz a je podporován ve většině světa. Mezi jeho hlavní výhody patří zpětná kompatibilita všech jeho verzí a možnost spolupráce zařízení různých výrobců.

Zigbee PRO

Zigbee PRO⁵ je další velmi oblíbenou technologií pro tvorbu internetu věcí. Zařízení používající Zigbee jsou zasazeny do mesh topologie a komunikují podle pravidel udávaných standardem IEEE 802.15.4. Globálně využívají pásmo 2.4GHz, regionálně pak pod 1GHz. Zigbee PRO dokáže podporovat více než 64000 uzlů zároveň, což znamená, že s ním lze vytvářet extrémně velké sítě. Dále také využívá technologii Green Power, která umožňuje využití senzorů a jiných nízko-spotřebních zařízení bez baterie. Místo baterií totiž pro napájení využívá malé zdroje energie jako světlo, vibrace nebo tření.

² What is RFID? *Technovelgy* [online]. [cit. 2015-12-28]. Dostupné z:

<http://www.technovelgy.com/ct/technology-article.asp>

³ Discover Bluetooth. *bluetooth* [online]. [cit. 2015-12-28]. Dostupné z: <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth>

⁴ Why Z-Wave. *Z-Wave* [online]. [cit. 2015-12-28]. Dostupné z: http://z-wave.sigmadesigns.com/about_z-wave

⁵ Zigbee PRO with Green Power. *zigbee alliance* [online]. [cit. 2015-12-28]. Dostupné z:

<http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbee-pro/>

Zigbee IP

Zigbee IP⁶ je technologií podporující síť založené na IPv6 a poskytuje možnost spojit desítky nízko-spotřebních a nízko-nákladových zařízení do jedné sítě, ovladatelné přes internetové připojení. Opět podporuje standard IEEE 802.15.4 a pracuje v globálně v pásmu 2.4GHz a regionálně v pásmech pod 1GHz. Dokáže využívat standardní protokoly, jako jsou TCP, UDP, 6LoWPAN apod.

6LoWPAN

6LoWPAN [3] je technologie pro nízko-spotřební zařízení v bezdrátových sítích. Umožňuje využívat IPv6 technologii a efektivně přenášet její packety v malých rámcích linkové vrstvy tak, jak je to definované standardem IEEE 802.15.4. Používá také pásma definována tímto standardem a mesh topologií, stejně jako některé z výše uvedených technologií.

Thread

Thread⁷ je technologie založená na protokolu 6LoWPAN a využívá standard IEEE 802.15.4. Zaměřuje se na zařízení pro chytré domácnosti a od počátku vývoje byla vytvářena pro velmi nízko spotřební zařízení. Jako topologii sítě používá mesh nebo hvězdu a v jedné síti dokáže udržovat přes 250 zařízení najednou. Technologie je podporována stovkami společností, včetně velkých firem jako jsou NXP, ARM, Qualcomm a další.

Cloud

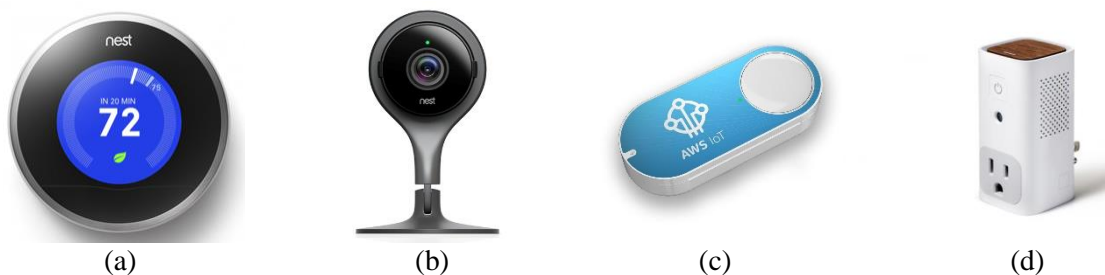
Cloud je název označující on-line internetovou službu, poskytující především vzdálené uložení pro data všeho druhu. Jeho popularita vzrůstá každým rokem a je ideálním řešením pro uchování informací získaných senzory a jinými zařízeními v síti internetu věcí. Uživatelé Cloudu se nemusí zatěžovat žádnou složitou údržbou a mohou si zakoupit přesně tolik místa, kolik potřebují, a v budoucnu jej rozšířit nebo zmenšit. Poskytovatelé Cloudu se pak starají o veškerý hardware, servery a údržbu potřebnou pro hladký běh systému.

2.3 Využití internetu věcí

Již dnes se internet věcí dá nalézt v reálném světě. Podle společnosti Cisco se internet věcí stal skutečností někdy mezi roky 2008 a 2009 [4], kdy získaly chytré telefony nesmírnou popularitu a od toho se odvíjel vznik tabletů a různých příslušenství, včetně chytrých hodinek.

⁶ Zigbee IP and 920IP. *zigbee alliance* [online]. [cit. 2015-12-28]. Dostupné z: <http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeeip/>

⁷ Our technology. *Threadgroup* [online]. [cit. 2017-03-20]. Dostupné z: <https://threadgroup.org/technology/ourtechnology>



Obr. 2.1-2.4: (a): Google Nest⁸, (b): Nest Cam⁹, (c): AWS IoT Button¹⁰, (d): Awair Glow¹¹

Chytré domácnosti

Postupně se také začínají objevovat chytré domácnosti. Jednou z hlavních výhod chytrých domácností je šetření energií. K tomu slouží například chytré termostaty, které sledují pohyb lidí uvnitř domu, venkovní teploty, předpověď počasí a další faktory ovlivňující potřebu vytápět domácnost a ovládají vyhřívání tak, aby bylo co nejefektivnější a ušetřilo uživatelům co nejvíce financí.

Dále existují videokamery, které snímají záznam vašeho domu, a v případě pokusu o krádež upozorní majitele nebo mohou sloužit pro hlídání batolat a domácích mazlíčků. Mezi další zařízení starající se o bezpečí domu patří také chytré detektory kouře.

Za zmínění stojí i zařízení s účelem zvýšení komfortu uživatelů, jako chytré žárovky a zásuvky, ovladatelné přes aplikace v mobilních zařízeních a počítačích, chytré přívěšky na klíče, které ulehčují ztracené klíče najít apod.

Zdraví

Pro vylepšení zdraví člověka se dá využít několik zařízení, například hlídač a plánovač užívání léků pro seniory, který je pomocí mobilní aplikace upozorní, v případě, že nějaký lék zapoměli nebo potřebují daný lék doplnit.

Dalšími zařízeními, sloužícími za tímto účelem, jsou snímače tepu pro sportovce, které dokáží i snímat pohyb a poskytnout tak mapu pohybu nebo typ cvičení. Podobnou funkci plní i senzory pro batolata, snímající jejich životní funkce, jako dýchání, tep nebo teplotu a umožňující rodičům nemocných dětí mít dokonalý a nepřetržitý přehled o jejich stavu.

Průmysl, zemědělství a hospodářství

Kromě spravování spotřeby energie a zabezpečení, stejným způsobem jako v případě chytrých domů, internet věcí pomáhá zlepšovat průmysl, zemědělství a hospodářství i v dalších oblastech. Jednou

⁸ What Google really gets out of buying Nest for \$3.2 billion. *Wired* [online]. [cit. 2017-05-11]. Dostupné z: https://www.wired.com/images_blogs/business/2014/01/Nest-therm-Small.jpg

⁹ Nest Cam: Connected camera. *Iotlist* [online]. [cit. 2017-05-11]. Dostupné z: http://iotlist-production.s3-us-west-2.amazonaws.com/uploads/production/post_attachment/979/file/nest-cam.jpg

¹⁰ AWS IoT Button: Programmable dash button. *Iotlist* [online]. [cit. 2017-05-11]. Dostupné z: http://iotlist-production.s3-us-west-2.amazonaws.com/uploads/production/post_attachment/1037/file/51Jq0GnSKWL_SL1000.jpg

¹¹ Awair Glow: Smart plug that tracks air quality. *Iotlist* [online]. [cit. 2017-05-11]. Dostupné z: http://iotlist-production.s3-us-west-2.amazonaws.com/uploads/production/post_attachment/1054/file/Awair-Glow.jpg

takovou oblastí je údržba strojů. Pomocí snímačů kontrolujících stav a stáří strojů aplikace odhadují čas, kdy bude potřeba provést údržbu, předpovídají chyby a problémy, které mohou nastat kvůli opotřebením a upozorňují o tom majitele nebo výrobce, nebo případně plánují a zařizují údržbu automaticky.

V zemědělství pak nabízí možnosti sledování vlhkosti půdy a předpovědi počasí a na základě toho ovládání ostřikovačů a zavlažovačů, kontrolu zdraví rostlin a hnojení. Celkově tak usnadňuje a zlepšuje práci farmářů.

V hospodářství se mohou sledovat životní funkce chovaných zvířat, kontrolovat a plánovat naplnění koryt, ovládat dveře od boxů atd.

Transport

Čím dál více se také začínají vyrábět auta se zabudovaným internetovým připojením, díky kterému nabízí služby jako wi-fi připojení pro pasažéry uvnitř vozidla, stahování informací o nehodách a ucpaných silnicích a různé aplikace komunikující s mobilními zařízeními.

Chytré obchody

Objevují se také zařízení jako iBeacon, které se rozmístí do různých částí obchodu a posílají do chytrých mobilních telefonů zákazníků odkazy, které jim zobrazí databázi uloženou na Cloudu daného obchodního řetězce. Podle pozice, ve které se zákazník nachází, se pak zobrazují určité nabídky a slevy, které se týkají zboží v jeho blízkosti.

2.4 Problémy

Kompatibilita a standardizace

Příliš velký počet standardů a typů technologií je jeden hlavních problémů, které brzdí vývoj internetu věcí. V dnešní době je velmi velké množství standardů pro různé vrstvy. Firmy jako IETF a W3C tvoří standardy pro aplikační vrstvy, oneM2M a další se zabývají standardy pro rámce a IEEE 802.15 a Bluetooth se zaměřují na optimalizace síťových přenosů¹². Důsledkem je spousta zařízení od různých výrobců, které vyžadují zapojení v jiných topologiích, využívají jiná pásma, používají zcela rozdílné způsoby přenosu dat a jsou navzájem nekompatibilní. Nové standardy se objevují každý rok, a ještě nějakou dobu potrvá, než se tato překážka vyřeší.

Bezpečnost a soukromí

S připojením věcí k internetu se začnou objevovat problémy s bezpečností a soukromím uživatelů. V dnešní době je spousta zkušených hackerů, jejichž cílem je nabourat se do všemožných zařízení, aplikací a internetových účtů a nějakým způsobem poškodit jejich majitele. Dokáží nejen omezit funkčnost nebo zneužít přístup k financím, ale také dostat se k soukromým věcem a informacím nebo zneužít připojených kamer.

Příkladem potřeby zvýšit bezpečnost je experiment z léta 2015, kdy Andy Greenberg, reportér pro magazín Wired, nechal dva hackery převzít kontrolu nad autem, ve kterém jel po dálnici. Bez

¹² NULL, Christopher. The state of IoT Standards. *TechBeacon* [online]. [cit. 2015-12-29]. Dostupné z:

<http://techbeacon.com/state-iot-standards-stand-big-shakeout>

jakékoliv úpravy na autě, s využitím pouze služby výrobce auta, která se připojuje k internetu, dokázali mimo jiné ovládat klimatizaci, rádio, zámky na dveřích, a dokonce i auto úplně zastavit¹³.

Dalším problémem by mohlo být již zmíněné zneužití kamer. S příchodem zabezpečení pomocí kamerových systémů totiž přichází i nebezpečí, že tyto kamery ovládne někdo vzdáleně, a tím získá přehled o všem, co dané kamery vidí. Kromě narušení soukromí tak může jednodušeji daný dům nebo budovu i vykrást.

Neméně problematická situace může nastat v případě, že se hacker dostane k zařízení sledujícímu nebo i ovládajícímu životní funkce člověka.

2.5 Budoucnost

Většina velkých firem zabývajících se elektronikou a zařízeními spojenými s internetem věcí se shoduje na tom, že v blízké budoucnosti se tento fenomén extrémně rychle rozšíří a stane se každodenní součástí života většiny lidí ve vyspělých částech světa. Předpoklady počtu zařízení připojených k internetu se liší, od 30 miliard až po 50 miliard [4] v roce 2020. Tyto výpočty berou v potaz aktuální trendy a vývoj množství připojených zařízení za posledních několik let. Avšak společnost Cisco IBSG připouští, že jejich 50 miliardový předpoklad nedokáže předpovědět náhlé rychlé vývoje a změny v internetu a technologiích zařízení, a tak se tahle čísla mohou i významně lišit.

Předpokládá se, že v budoucnosti se dokáže vytvořit nějaký jednotný standard pro všechna zařízení podílející se na tvorbě internetu věcí, a tím se umožní tzv. „plug-and-play“, což znamená, že nově zakoupené zařízení bude možné okamžitě a jednoduše zakomponovat do sítě, bez potřeby nějakých nastavení nebo převodů, a že zařízení všech výrobců budou navzájem kompatibilní.

Další ambiciózní vizí do budoucnosti jsou chytrá města. Taková města by měla širokou škálu zařízení, senzorů a kamer, které by informovaly lidi o stavu dopravy, volných parkovacích místech, zahuštěnosti některých částí města apod. Dále by se tento autonomní systém staral o přesměrování dopravy a okamžité zavolání záchranných služeb v případě nehody, upozornění policie v případě porušení zákonů atd. Lidé by mohli být automaticky připojováni na veřejné wi-fi připojení a dostávali by upozornění o zajímavých místech v jejich okolí [5].

2.6 Protokoly a komunikační rozhraní

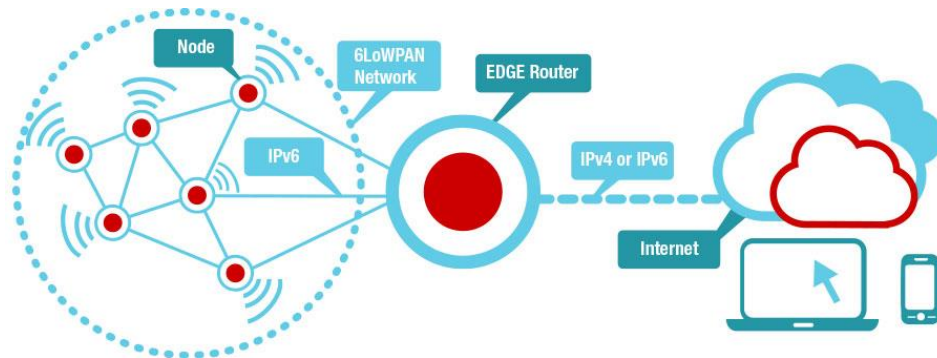
IEEE 802.15.4

IEEE 802.15.4 [6, 7] je standard pro bezdrátovou komunikaci vydaný společností IEEE (Institute for Electrical and Electronics Engineers). Standard udává, že globální komunikace musí probíhat na 5MHz kanálech, v pásmu od 2,4 do 2,4835GHz. V tomto pásmu pak specifikuje maximální rychlost přenosu dat jako 250 kb/s. Pro regionální komunikaci vyhrazuje pásma od 868 do 868.6MHz pro Evropu, kde existuje pouze jeden kanál a maximální rychlost přenosu určuje na 20 kb/s a pásma od 902 do 928MHz pro Severní Ameriku, s kanály na 2MHz a nejvyšší rychlostí přenosu 40kb/s. Dále taky určuje, že maximální jednotka přenosu je 127 bytů.

¹³ GREENBERG, Andy. Hackers remotely kill a jeep on the highway – with me in it. *Wired* [online]. 2015 [cit. 2015-12-29]. Dostupné z: <http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/>

6LoWPAN

6LoWPAN [3, 8] byl vytvořen jako adaptační vrstva pro přenos IPv6 packetů přes síť založenou na standardu IEEE 802.15.4. Je to otevřený standard definovaný společností IETF. Topologie sítě, využívající 6LoWPAN, je mesh. Tato síť obsahuje několik druhů zařízení. Tím nejdůležitějším, je tzv. okrajový směrovač, anglicky edge router, který má tři úkoly. Stará se o výměnu dat mezi internetem a zařízeními v 6LoWPAN síti, o výměnu dat mezi samotnými zařízeními uvnitř 6LoWPAN sítě a tvorbu a údržbu celé sítě. Díky tomu, že využívá IP, lze spojit 6LoWPAN síť s jinou sítí jednoduše pomocí směrovačů.



Obr. 2.5: Topologie sítě 6LoWPAN¹⁴

Dalšími typy zařízení jsou směrovač a hostitel. Směrovače se starají o přeposlání dat do uzlů v síti, zatímco hostitelé jsou koncová zařízení, která schopná směrování nejsou. Každé koncové zařízení v 6LoWPAN síti má svou vlastní IPv6 adresu, což znamená, že se k němu je možné dostat odkudkoliv z internetu, i ze sítě používajících pouze IPv4. O to, aby nedocházelo ke kolizím IP adres uvnitř 6LoWPAN sítě se opět stará okrajový směrovač.

Hlavní výhodou této technologie, je její schopnost komprese hlavičky protokolu IPv6. IPv6 byl vytvořen proto, že s rapidně narůstajícím počtem zařízení připojených k internetu přestával stačit počet možných IPv4 adres. Z tohoto důvodu je hlavička IPv6 protokolu mnohem větší než hlavička jeho předchůdce. Velikost hlavičky IPv4 je 20 bytů, ale velikost hlavičky IPv6 je 40 bytů. Tento počet je příliš vysoký pro využití na nízko-spotřebních zařízeních, které se používají v internetu věcí. 6LoWPAN uvedený problém ale dokáže vyřešit. Využívá toho, že v hlavičce je několik údajů, které jsou v každém packetu stejné. Verze je vždy šest, třída provozu a značka toku jsou vždy nula, délka dat může být získána z nižších vrstev, z hlavičky 6LoWPANu nebo hlavičky IEEE 802.15.4, maximální počet skoků je hodnota známá zdroji dat a adresy se dají vytvořit s pomocí lokálního prefixu nebo malé směrovací tabulky, platící pro celou síť, a ID rozhraní získaného z adres IEEE 802.15.4.

V nejlepším případě dokáže kompresí zmenšit hlavičku na 2 byty, ale toho lze docílit pouze pro vnitřní komunikaci zařízení patřících do dané 6LoWPAN sítě. Pro zařízení mimo 6LoWPAN síť, u kterých je známý prefix dané externí sítě, je možné snížit velikost IPv6 hlavičky na 12 bytů. V nejhorsím případě, pro zařízení mimo 6LoWPAN síť, u kterých nevíme prefix, lze hlavičku zmenšit pouze na 20 bytů. I to je ale pouhá polovina původní velikosti.

Další potřebnou vlastností je schopnost fragmentace. Pokud se všechna přenášená data vlezou do jednoho IEEE 802.15.4 rámce, fragmentace se neprovádí a není potřeba vytvářet fragmentační

¹⁴ 6LoWPAN Mesh. In: *Texas Instruments* [online]. [cit. 2015-12-29]. Dostupné z:

http://www.ti.com/lscs/media/images/wireless_connectivity/20979_6LowPan_mesh.jpg

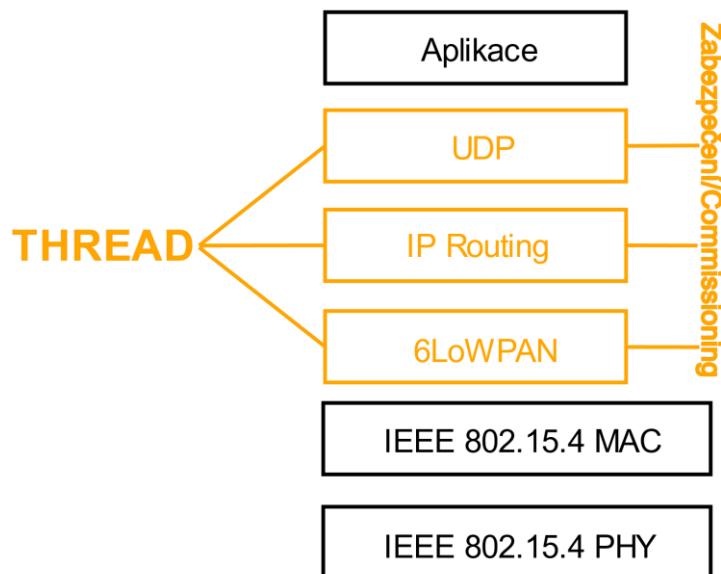
hlavičku. V případě, že je payload větší než jeden rámeček, využívá se fragmentační hlavička, obsahující informaci o celkové velikosti fragmentovaných dat, tag označující fragmenty patřící k sobě a offset. Pokud se pro přenos packetu musí vykonat více než jeden skok, je potřeba použít mesh adresující hlavičku, která obsahuje informace o maximálním počtu skoků, adresu zdroje dat a adresu cíle.

U 6LoWPAN existují dva druhy směrování, mesh-under a route-over. Mesh-under využívá adresy linkové vrstvy a celá síť se pak chová jako jedna podsíť a jediný IP směrovač v síti je okrajový směrovač. Pro kompatibilitu s IPv6 protokoly vyšších vrstev se používá pouze jedna broadcast adresa, a zprávy typu detekce duplikátních adres se posílají všem zařízením v síti. Proto je mesh-under vhodné pouze pro malé sítě. U route-over je na každém skoku směrovač, plní všechny funkce normálního směrovače, a každé zařízení má svou vlastní IP adresu, což z route-over dělá dobrou volbu pro rozsáhlé sítě.

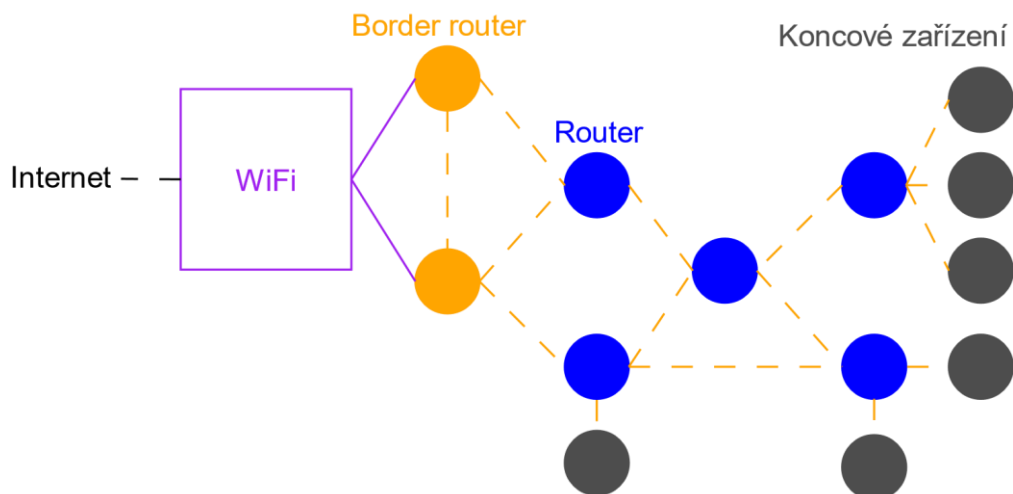
Thread

Thread [14] je komunikační protokol pro bezdrátové sítě vyvíjený organizací Thread Group. Jejím záměrem bylo vytvořit technologii specificky pro chytré domácnosti, která je spolehlivá, má velmi nízké energetické nároky, je jednoduše využitelná zákazníky a kombinuje již existující a ověřené technologie.

Zařízení využívající Thread komunikují přes vysílače vyrobené podle standardu IEEE 802.15.4, využívají nízko spotřební protokol 6LoWPAN pro podporu IPv6 a vytváří sítě v topologii mesh, případně hvězdice, pokud je využito málo zařízení a pouze jeden router. Thread umožňuje do sítě zapojit přes 250 zařízení najednou.



Obr. 2.6: Architektura technologie Thread



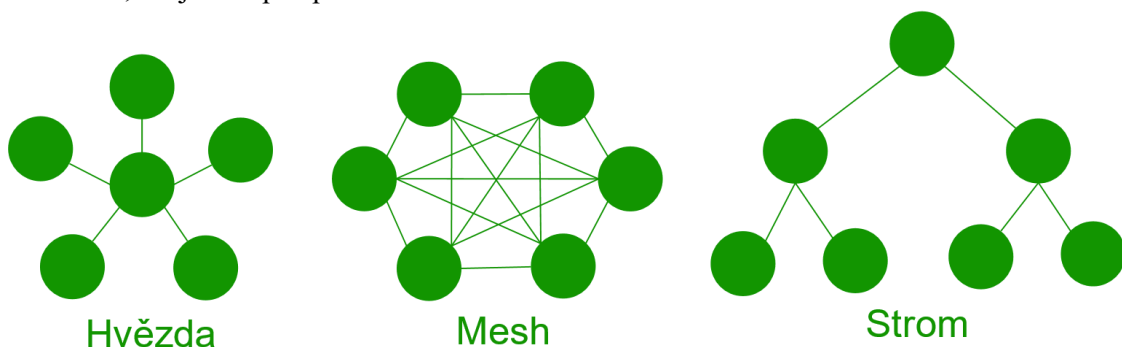
Obr. 2.7: Základní topologie sítě Thread

V síti Thread může být několik druhů zařízení. Těmito zařízeními jsou border router, router, router-eligible end device a sleepy end device. Border router poskytuje spojení mezi 802.15.4 sítí a ostatními sítěmi, jako jsou Wi-Fi a ethernet. Poskytují také služby zařízením uvnitř jejich sítě, jako směrování a operace mimo jejich síť. V jedné síti může být i více border routerů.

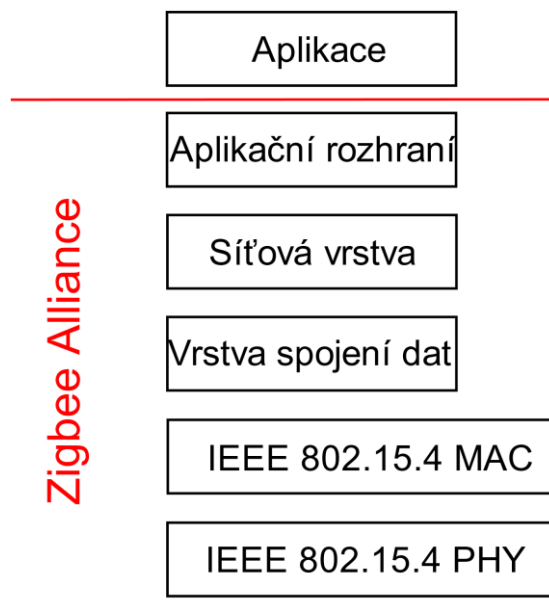
Router poskytuje směrování zařízením uvnitř sítě, stará se o připojování nových zařízení a o bezpečnost. Router-eligible end device je koncové zařízení, které je v případě potřeby, například při výpadku jiného routeru, schopné ze sebe udělat router. O jejich povýšení na router se stará síť samotná, ne uživatel. Sleepy end device je koncové zařízení, které komunikuje pouze přes svůj nadřazený router.

Zigbee PRO

Zigbee PRO [15] je standard udržovaný sdružením společností, zvaným Zigbee Alliance. Je založený na IEEE 802.15.4. Stará se o síťovou a aplikační vrstvu protokolu. Jako topologii sítě využívá především mesh, ale je schopné pracovat i v hvězdici a stromě.

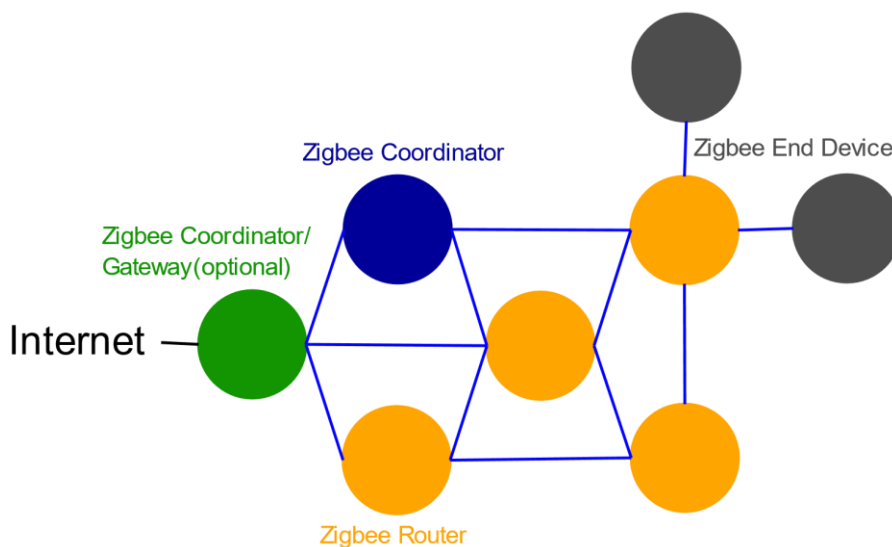


Obr. 2.8: Topologie sítí podporované Zigbee



Obr. 2.9: Architektura technologie Zigbee

V síti je potřeba tři, až čtyři zařízení¹⁵. Těmi jsou Zigbee router, což je směrovač, starající se o přeposílání zpráv a dá se využít na rozšíření sítě do větší plochy, Zigbee Coordinator, který se stará o sestavení sítě, určení její topologie a její zabezpečení, Zigbee End Device, koncové zařízení, a Zigbee Coordinator/Gateway, které není povinnou součástí sítě. Zigbee Coordinator/Gateway je kombinací koordinátoru a brány a slouží k připojení celé sítě k internetu.



Obr. 2.10: Topologie mesh sítě Zigbee PRO

Významnou vlastností mesh sítí Zigbee PRO je jejich schopnost sebe-léčení a sebe-sestavení. Sebe-sestavení znamená, že ihned po spuštění všech zařízení patřících do sítě, se celá síť sama automaticky vytvoří a sestaví. Sebe-léčící síť se dokáže vyrovnat s případy, kdy se některá cesta zahltí nebo některý ze směrovačů, náležitých cestě, po které je potřeba poslat data, přestane fungovat.

¹⁵ ZigBee PRO with Green Power. *Zigbee Alliance* [online]. [cit. 2015-12-28]. Dostupné z:

<http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbee-pro/>

V takových případech síť sama od sebe zvolí jinou vhodnou cestu pro přenos, bez pomoci vnějších zdrojů.

Koordinátory a směrovače si udržují směrovací tabulky, obsahující informace, které se využívají při vytváření spojení mezi dvěma zařízeními v síti. Vytváří si také dočasné tabulky pro objevení cesty, které se tvoří na požádání některým z jiných zařízení v síti. Tyto tabulky obsahují adresu zařízení, které vyvolalo požadavek, adresu zařízení, které požadavek předalo momentálnímu zařízení a ceny možných cest. Další obsaženou tabulkou je tabulka sousedů, která uchovává informace o dosažitelných zařízeních. Tato tabulka se obnovuje pokaždé, když dané zařízení přijme nový packet.

2.7 Raspberry Pi 2 B

Série Raspberry Pi jsou jednodeskové počítače o velikosti kreditní karty. Model 2 B je druhou generací těchto počítačů. Součástí Raspberry Pi 2 B¹⁶ je 900MHz čtyř jádrový procesor ARM Cortex-A7 s čipem Broadcom BCM2836 SoC, který umožňuje využití velkého množství Linuxových distribucí a Windows 10 IoT Core. Dále obsahuje 1 GB RAM, 4 USB 2.0 porty, 40 GPIO pinů, HDMI port plné velikosti, ethernet port, kombinovaný 3,5mm audio jack a výstup pro kompozitní video, rozhraní pro kameru a displej, slot pro mikro SD kartu a grafické jádro VideoCore IV 3D. Zařízení je potřeba napájet 5V přes mikro USB.

V roce 2006 vznikl nápad vytvořit malý a cenově dostupný počítač pro děti. Za tímto nápadem stáli Eben Upton, Rob Mullins, Jack Lang a Alan Mycroft, které znepokojoval každoroční úpadek počtu a schopností studentů hlásících se na studium počítačových technologií. Do roku 2008 potom byly procesory pro mobilní zařízení čím dál levnější a dostatečně silné pro multimédia, která by přitahovala i děti, které by se nezajímaly o čistě programovací zařízení. O tři roky později se začal prodávat Raspberry Pi model B, kterého se během dvou let prodalo přes 2 miliony kusů.

Procesor

ARM Cortex-A7 je jeden z nejvíce energeticky efektivních vícejádrových procesorů. Dokáže využívat 1 až 4 jádra a nabízí vysoký výkon s nízkou spotřebou, díky čemuž je často využíván v levných mobilních telefonech. Implementuje RISC architekturu ARMv7-A, zaměřenou pro zařízení s nízkou spotřebou. ARM architektury využívají standardní RISC prvky, jako velký jednotný registrový soubor, načítání a ukládání způsobem, kdy operace zpracovávající data operují pouze nad obsahem registrů, ne přímo nad obsahem paměti a jednoduché adresující režimy. Dále RISC obohacují o instrukce kombinující posuvy s aritmetickými nebo logickými operacemi, optimalizaci cyklů přes režimy auto-inkrementace a auto-dekrementace, načítání a ukládání několika instrukcí najednou pro zvýšení průchodu dat a podmíněné vykonávání velkého množství instrukcí. Verze ARMv7-A pak navíc přidává podporu architektury systému s virtuální pamětí (VMSA) a instrukční sady ARM a Thumb [9].

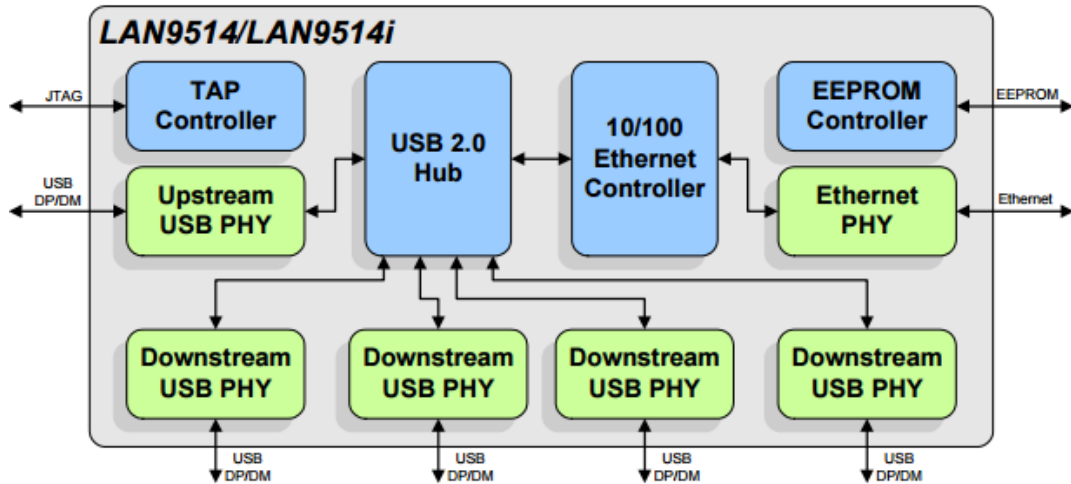
V Raspberry Pi 2 je použit čip BCM2836 od firmy Broadcom, vyvinutým přímo pro Raspberry.

¹⁶ Raspberry Pi 2 model B. *Raspberry Pi* [online]. [cit. 2015-12-29]. Dostupné z:

<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

USB

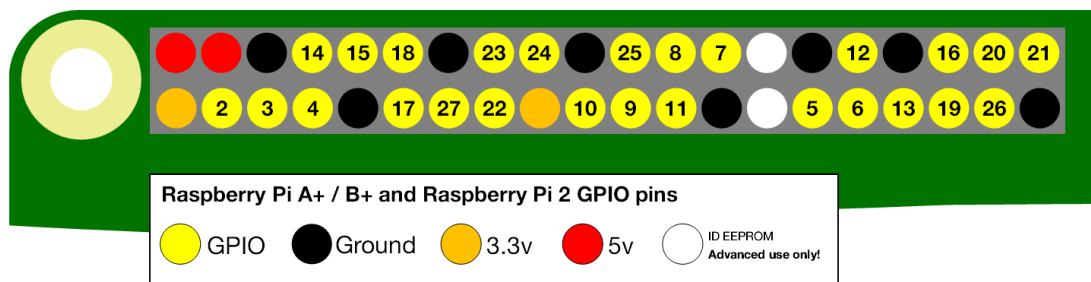
Do USB portů lze zapojit klávesnici, myš, přijímač bezdrátového připojení apod. Všechny čtyři USB porty a ethernet port jsou zapojeny do LAN9514, který se skládá z USB 2.0 hubu a ethernetového kontroléru 10/100. LAN9514 je pak napojeno přes jediný USB port přímo do čipu BCM2836.



Obrázek 2.11: Blokový diagram vnitřní struktury¹⁷

GPIO

Raspberry Pi 2 B nabízí 40 pinů, z toho 26 je čistě GPIO, 2 piny vydávají 3,3V napětí s maximálním proudem až 50 mA, 2 piny vydávají 5 V napětí s proudem zhruba 1,5A, 8 pinů slouží pro uzemnění a 2 jsou ID EEPROM piny. GPIO piny 2 a 3 jsou pro I²C a 7, 8, 9, 10 a 11 jsou pro SPI (platí pro GPIO číslování).

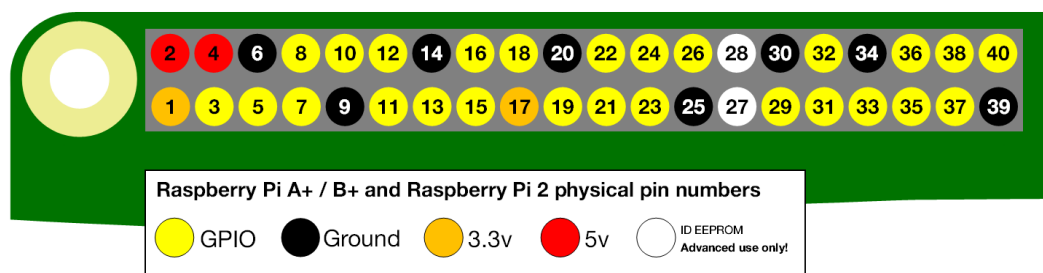


Obrázek 2.12: GPIO číslování pinů¹⁸

¹⁷ SMSC. *USB 2.0 Hub and 10/100 Ethernet Controller* [online]. Dostupné z:

<http://ww1.microchip.com/downloads/en/DeviceDoc/9514.pdf>

¹⁸ GPIO. *Raspberry Pi* [online]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>



Obrázek 2.13: Fyzické číslování pinů¹³

Software

Všechny modely zařízení Raspberry Pi využívají jako uložení pro operační systémy SD kartu. Nejjednodušší způsob instalace operačního systému je přes program NOOBs od výrobců Raspberry Pi. Tento program umožní výběr a instalaci zvoleného systému ze seznamu, který uživateli nabídne. Mezi operační systémy, které lze nainstalovat na Raspberry Pi 2 patří Raspbian, Ubuntu Mate, Snappy Ubuntu Core, Windows 10 IoT Core, OSMC, OpenELEC, PiNet, a RISC OS.

Raspbian je bezplatný operační systém založený na unixovém systému Debian a optimalizovaný přímo pro zařízení Raspberry Pi. Raspbian nabízí tisíce rozšiřujících balíčků, které lze jednoduše nainstalovat a obohatit tak základní systém o spoustu dalších funkcí.

Ubuntu Mate¹⁹ je spojení základu operačního systému Ubuntu, jedné z nejpobulárnějších unixových distribucí v dnešní době, a pracovního prostředí MATE Desktop. MATE Desktop má již dlouhou historii užívání a je pokračováním prostředí GNOME2. Nabízí správce souborů a archivů, textový editor, prohlížeč obrázků a dokumentů a další.

Snappy Ubuntu Core²⁰ je nové provedení systému Ubuntu s transakčními aktualizacemi. Nabízí stejné knihovny jako běžné Ubuntu, ale jednodušším způsobem. Tento nový přístup je rychlejší, spolehlivější a bezpečnější pro aplikace i uživatele. Díky velmi malé velikosti a transakčním aktualizacím, při kterých se buď provede instalace každé instalace nebo žádná, je Snappy Ubuntu Core vhodné pro prostředí, kde je potřebná naprostá spolehlivost a předvídatelnost chování systému, jako například u serverů pro Cloud.

Windows 10 IoT Core²¹ je verze Windows 10 optimalizovaná pro malá zařízení s i bez displeje. Nabízí například možnost využití plně nejnovější verze programu Visual Studio a umožňuje aplikace napsané na jednom zařízení využívat na všech ostatních zařízeních podporujících Windows.

RISC OS je operační systém, který byl v roce 1987 vyvinut pro procesory ARM. V dnešní době je vhodný pro použití na zařízeních jako Raspberry Pi, protože se celé jeho jádro i s několika aplikacemi vleze do 6 MB. Je také rychlý a nabízí základní desktopové operační prostředí.

OSMC a OpenELEC jsou systémy pro mediová centra, nabízející přehrávání filmů, seriálů, hudby, hraní her a další druhy zábavy. PiNet je systém pro správu tříd ve školách, kde se pracuje na projektech pro Raspberry Pi.

¹⁹ What is Ubuntu MATE? *Ubuntu MATE* [online]. [cit. 2015-12-30]. Dostupné z: <https://ubuntu-mate.org/what-is-ubuntu-mate/>

²⁰ Core. *Ubuntu* [online]. [cit. 2015-12-30]. Dostupné z: <https://developer.ubuntu.com/en/snappy/>

²¹ Why IoT Core? *Microsoft* [online]. [cit. 2015-12-30]. Dostupné z: <https://developer.microsoft.com/en-us/windows/iot/Explore/IoTCore>

Doporučeným programovacím jazykem pro vývoj aplikací na Raspberry Pi²² je objektově orientovaný Python. Zároveň lze ale použít jakýkoliv jazyk, který jde přeložit na procesorech ARMv6 a ARMv7 používaných v Raspberry Pi a Raspberry Pi 2. Proto je možnost využít C, C++, Java, Scratch nebo Ruby.

²² FAQs. *Raspberry Pi* [online]. [cit. 2015-12-30]. Dostupné z: <https://www.raspberrypi.org/help/faqs/>

3 Cloudové služby pro IoT

Uvnitř této kapitoly jsou popsány databázové technologie, důležité pro udržování dat o zařízeních v síti, technologie Cloud a platformy, které tuto technologii poskytují.

3.1 Databázové jazyky SQL a NoSQL

Tato podkapitola popisuje technologie SQL a NoSQL a jejich využitelnost pro internet věcí. Věnuje se jejich rozdílům a výběru správné technologie pro tuto bakalářskou práci.

SQL

SQL^{23 24 25} je zkratka pro Structured Query Language. Jazyk se stal standardem v roce 1986 a od té doby získal na popularitě a vzniklo velké množství jeho odnoží, například od firem Microsoft a Oracle nebo open-source verze, jako MySQL.

SQL zpracovává data jako skupiny, spíše než jednotky, automaticky k datům navádí a používá komplexní příkazy. Pro získání několika řádků z tabulky je pouze potřeba definovat podmínky využitě pro filtraci řádků a všechny řádky, které podmínkám odpovídají, budou předány uživateli najednou, ne jednotlivě. Uživatel se také nemusí zajímat o to kde a jak jsou data fyzicky uložena. Jazyk umožňuje provádět dotazy na databázi pro získání dat, vkládání, změny a odstraňování řádků a objektů v tabulkách, správu přístupu k databázi a jejím objektům a garantuje konzistenci a integritu. Konzistence znamená, že všechny transakce mohou změnit data v databázi jen povoleným způsobem a musí odpovídat všem definovaným pravidlům. Integrita znamená, že všechna data v databázi jsou kompletní.

Využívá se pro ovládání relačních databází skládajících se z tabulek s řádky a sloupci, které mezi sebou mají definované vztahy. Tabulky v relační databázi musí odpovídat integritním pravidlům. Každý řádek v tabulce musí být odlišitelný od ostatních, jinak by nebylo možné správně v tabulce vyhledávat. Dále hodnoty ve sloupcích nesmí být opakující se skupiny nebo pole, a sloupce s chybějící hodnotou jsou naplněny hodnotou null. Jeden nebo více sloupců je používáno pro jednoznačnou identifikaci řádků, těmto sloupcům nebo skupinám sloupců se říká primární klíč.

NoSQL

Pojem NoSQL [12, 13] vznikl v roce 2009, když jej Johan Oskarsson použil pro název jeho konference o open-source, distribuovaných, nerelačních databázích. V dnešní době se NoSQL používá jako akronym pro Not only SQL, což znamená nejen SQL, naznačující, že dotazy ve stylu SQL nejsou jádrem těchto uložišť.

²³ Introduction to Oracle SQL. *Oracle* [online]. [cit. 2016-11-19]. Dostupné z:

http://docs.oracle.com/cd/B10501_01/server.920/a96540/intro.htm

²⁴ Structured Query Language (SQL). *Microsoft* [online]. [cit. 2016-11-19]. Dostupné z:

<https://docs.microsoft.com/en-us/sql/odbc/reference/structured-query-language-sql>

²⁵ A Relational Database Overview. *Oracle* [online]. [cit. 2016-11-19]. Dostupné z:

<https://docs.oracle.com/javase/tutorial/jdbc/overview/database.html>

NoSQL získává na popularitě především kvůli zvětšujícím se objemům dat a nárokům na databáze, kterým relační databáze nemohou vyhovět. NoSQL databáze mají několik datových modelů, které se výrazně liší od relačního modulu u SQL databází.

Páry klíčů a hodnot je jeden z využívaných data modelů. Hodnoty jsou adresovány pouze jejich klíči a jsou úplně izolované a nezávislé na zbytku hodnot. Proto jakékoliv vztahy mezi nimi musí být ovládány aplikační logikou. Díky tomuto jednoduchému způsobu ukládání dat nemá tento typ uložště žádné schéma a nové hodnoty jakéhokoliv typu mohou být do tabulky přidávány i za běhu. Jediný způsob, jak dát tomuto systému nějakou strukturu je svázáním párů do kolekcí.

Dalším z data modelů je dokumentové uložště. V tomto případě jsou páry klíčů a hodnot zabalovány do JSON dokumentů, které jsou udržované v kolekcích. Každý dokument obsahuje jedinečný identifikační klíč, který jej rozlišuje od ostatních dokumentů v jejich kolekci. Na rozdíl od párů klíčů a hodnot, lze provádět dotazy i přímo na hodnoty, což umožňuje lepší ovládání komplexních datových struktur. Dále díky ukládání do interpretovatelných JSON dokumentů, mohou mít hodnoty určený typ. Datový model nemá žádné schéma, takže přidávání dokumentů s jakýmkoliv druhem nových atributů nebo přidávání nových atributů do existujících dokumentů nevznikají žádné problémy.

Využívaný je také model uložště sloupcových rodin. V tomto modelu může být v řádcích uložený libovolný počet párů klíčů a hodnot. Hodnoty nejsou interpretovatelné systémem, a proto vztahy mezi jinými datovými typy než řetězci, musí být obstarány aplikační logikou. Více verzí jedné hodnoty je uložených v chronologickém pořadí, což umožňuje verzování, lepší výkon a konzistenci. Sloupce je možné kombinovat do sloupcových rodin pro lepší organizaci dat. Sloupce a řádky mohou být přidávány libovolně i za běhu, ale sloupcové rodiny musí být často předdefinovány, takže je tento data model méně flexibilní než předchozí dva uvedené.

Posledním modelem je grafový model. Tento je specializovaný na efektivní správu silně propojených dat. Díky tomu je vhodný pro aplikace využívající data s velkým množstvím vztahů, protože umožňuje nahradit náročné rekurzivní spoje efektivními přechody.

Díky flexibilitě, schopnosti efektivně spravovat velké množství dat a velké rozšiřitelnosti se NoSQL databáze hodí pro projekty využívající miliony senzorů a zařízení lépe než relační databáze. Proto byly NoSQL databáze zvoleny i pro tuto bakalářskou práci, pro demonstraci praktického a rozšiřitelného systému.

3.2 Cloud

Cloud²⁶ [10, 11] je výpočetní model poskytující autonomní zasíťované hardware a software zdroje online a na vyžádání. Mezi tyto zdroje patří databáze, výpočetní síla, uložště, servery a velké množství aplikací. Zdroje na vyžádání znamenají, že zákazník nepotřebuje mít vlastní datacentra pro své projekty. Autonomnost Cloudu zajišťuje, že zákazník, který ho využívá, nepotřebuje mít přehled o využitých technologiích, fyzických umístění hardware, sítích a lidských zdrojích starajících se o běh služeb. Výhodami Cloudu jsou také předdefinovaná kvalita služeb, které poskytovatel nabízí, jednoduchost využívání těchto služeb a bezproblémové zvýšení nebo snížení množství zdrojů a jejich využití. Dále díky sdílení zdrojů velkým množstvím zákazníků může poskytovatel Cloudu lépe optimalizovat ceny a poskytnout tak levnější možnosti oproti individuální výstavbě datacenter pro jednotlivé zákazníky. Služby jsou typicky nabízeny buď jako předplatný model, kdy si zákazník jednou

²⁶ What is Cloud Computing? *Amazon Web Services* [online]. [cit. 2016-12-26]. Dostupné z:

<https://aws.amazon.com/what-is-cloud-computing/>

za měsíc zaplatí za předem zvolené množství zdrojů nebo jako model placení za běhu, kdy zákazník platí pouze za to, co využije.

Pro internet věci jsou Cloudové technologie velmi důležité. Senzory a zařízení mohou díky Cloudu ukládat svá data do internetových databází bez jakýchkoliv velikostních omezení. Důležitým aspektem internetu věcí je také schopnost reakce na získaná data, což umožňují aplikace a triggery napojené na tyto databáze. Díky správě Cloudu společnostmi, které jej poskytují, se firmy mohou zaměřit na vývoj svých vlastních aplikací nebo zařízení a nemusí tak spotřebovat čas a finanční prostředky na zajištění bezpečného přístupu do databází, trvalé dosažitelnosti serverů a správě a rozšiřování složitého hardware podle nároků zákazníků a množství využitých produktů.

Infrastructure-as-a-Service (IaaS)

Infrastruktura jako služba je základní složkou Cloudu. Patří zde hardware jako počítače, sítě a uložště dat. Poskytuje flexibilní kontrolu nad zdroji a zákazník se stará o software, jako operační systémy a aplikace, ale ne o samotnou infrastrukturu.

Platform-as-a-Service (PaaS)

Platforma jako služba dává možnost vytváření vlastního software a aplikací využívajících poskytovanou infrastrukturu s pomocí programovacích jazyků, knihoven, služeb a nástrojů nabízených poskytovatelem. Zákazník se nestará o síť, server, operační systém ani uložště, ale má kontrolu nad aplikacemi a konfiguračními zdroji.

Software-as-a-Service (SaaS)

Software jako služba poskytuje zákazníkovi aplikace využívající infrastrukturu Cloudu. Tyto aplikace jsou přístupné z různých klientských zařízení přes programy nebo internetový prohlížeč. Zákazník se nestará o infrastrukturu ani samotné aplikace, nanejvýš může provádět určité konfigurace využívaných aplikací.

3.3 Cloudové platformy

Amazon Web Services

Cloud platforma od firmy Amazon, nabízející výpočetní sílu, uložště, databáze, síťové a aplikační služby a další. AWS²⁷ bylo spuštěno v roce 2006 a od té doby se rozšířilo do 190 zemí a má datacentra v USA, Evropě, Brazílii, Singapuru, Japonsku a Austrálii. Všechny poskytované služby využívají model placení za běhu, kdy zákazník platí pouze za to, co využije. Amazon také nabízí možnost využít velkou část portfolia AWS zadarmo, s určitými omezeními, po dobu 12 měsíců. Pro internet věci jsou užitečné například služby jako relační databáze Amazon Relational Database Service, NoSQL databáze Amazon DynamoDB, vizualizační nástroje služby Amazon QuickSight a AWS Lambda, která umožňuje vytvořit událostně řízenou aplikaci, reagující na jiné AWS služby.

²⁷ About AWS. *Amazon Web Services* [online]. [cit. 2016-12-26]. Dostupné z: <https://aws.amazon.com/about-aws/>

Google Cloud

Cloud platforma od firmy Google²⁸, nabízející výpočetní sílu, uložení, databáze, síťové služby a strojové učení. Jednotlivé části Google Cloud byly postupně spouštěny od roku 2008, ale několik podstatných služeb bylo otevřeno veřejnosti až v letech 2012, 2014 a 2016. Za Google Cloud zákazníci platí způsobem platby za běhu a ceny se účtují za využití minut. Je také možnost využít platformu na 2 měsíce zadarmo s kreditem 300 dolarů. Část této platformy je v beta nebo alpha verzích, jako Cloud Functions pro událostně řízené aplikace, Cloud Datalab pro vizualizaci dat a některé služby pro strojové učení. Dále Google nabízí pro internet věcí užitečné a plně funkční služby jako relační MySQL databázi Cloud SQL a NoSQL databáze Cloud Bigtable a Cloud Datastore.

IBM Bluemix

Cloud platforma od firmy IBM²⁹, nabízející výpočetní sílu, uložení, databáze, síťové služby, službu Watson a další. Platforma byla spuštěna v roce 2014 s využitím open-source platformy Cloud Foundry. IBM umožňuje volbu mezi modely placení za běhu a měsíčně placeného odebrání služeb. Nabízí také 30 dní využívání platformy zdarma. Mezi služby zajímavé pro internet věcí patří SQL databáze, NoSQL databáze Cloudant NoSQL DB, vizualizace dat s pomocí služby Watson a OpenWhisk pro událostně řízené aplikace.

Microsoft Azure

Cloud platforma Microsoft Azure³⁰, nabízející výpočetní sílu, uložení, databáze, síťové služby a další. Platforma byla spuštěna v roce 2010 pod názvem Windows Azure a později v roce 2014 byla přejmenována na Microsoft Azure. Lze ji využívat s modelem placení za běhu a účtováním za využití minut. Microsoft umožňuje využít jejich platformu na měsíc zadarmo s kreditem 200 dolarů, a poté nabízí kredit 25 euro na měsíc pro uživatele Visual Studia. Pro internet věcí jsou užitečné například SQL databáze, NoSQL databáze DocumentDB a Table Storage, vizualizace dat s pomocí PowerBI a služba App Service na vytváření webových a mobilních aplikací spojených s ostatními službami v Microsoft Azure.

Thinger, ThingSpeak, Ubidots

Jedná se o Cloud platformy pro internet věcí. Byly vytvořeny přímo pro zařízení jako jsou Raspberry Pi nebo Arduino a umožňují komunikaci mezi sensory a jinými malými zařízeními a Cloudem. Nabízí také webovou konzoli pro přehled a analýzu dat a různá nastavení a konfigurace. Uživatel u nich nemá přímo přehled o databázích a nemá k dispozici tak široké spektrum služeb, jako u AWS nebo Microsoft Azure, ale jsou zadarmo a jejich ovládání a programování je jednoduché a zařízení lze připojit, a spustit komunikaci mezi nimi, velmi rychle. ThingSpeak navíc například umožňuje analýzu a vizualizaci dat přes MATLAB.

²⁸ Why Google Cloud Platform? *Google* [online]. [cit. 2016-12-26]. Dostupné z: <https://cloud.google.com/why-google/>

²⁹ What is Bluemix. *IBM* [online]. [cit. 2016-12-26]. Dostupné z: <https://www.ibm.com/cloud-computing/bluemix/what-is-bluemix>

³⁰ What is Azure? *Microsoft Azure* [online]. [cit. 2016-12-26]. Dostupné z: <https://azure.microsoft.com/en-us/overview/what-is-azure/>

4 Návrh

Kapitola specifikuje zadání bakalářské práce. Dále je zde popsán výběr hardware zařízení a software technologií vhodných pro tuto práci. Popisuje důvody, proč byly dané zařízení a technologie zvoleny a popisuje návrh jejich praktického využití, a tím i řešení bakalářské práce.

4.1 Specifikace zadání

Cílem bakalářské práce je propojení několika technologií za účelem vytvoření sítě zařízení spadající do kategorie internetu věcí. Zadání určuje využití standardu IEEE 802.15.4 a protokolu 6LoWPAN, především kvůli jejich zaměření na zařízení s malým výkonem a nízkou spotřebou. S pomocí těchto technologií bude komunikovat senzor se svým okolím, a především s internetem. Využitým senzorem bude teploměr měřící teplotu v lednici. Dále je potřebná Cloudová platforma pro ukládání naměřených dat, jejich zpracování, analýzu, vykreslení a následnou reakci na hodnoty neodpovídající stanoveným limitům.

4.2 Výběr HW

Při výběru hardware bylo potřeba zaměřit se na několik klíčových vlastností. Zařízení pro měření hodnot muselo být malé a nízko spotřební, napájení muselo být možné i z knoflíkové baterie. Dále bylo důležité vyhledat zařízení schopné vykonávat funkci border routeru a zprostředkovávat tak nejen lokální komunikaci a správu sítě, ale také komunikaci s internetem. Potřebná byla také podpora komunikace za využití standardu IEEE 802.15.4 a protokolu 6LoWPAN.

Výrobě zařízení odpovídajících těmto požadavkům se věnuje několik společností. Mezi ně patří firma Texas Instruments, nabízející CC2531 USB dongle pro edge router pro 6LoWPAN síť a multi senzorové zařízení SensorTag. Další možností jsou výrobky společnosti NXP. Ta prodává vývojové kity využívající jejich mikrokontroler KW41Z. Dva z nich jsou USB-KW41Z dongle pro tvorbu border routeru pro síť Thread a FRDM-KW41Z desku schopnou mimo jiné měřit teplotu a přes Thread komunikovat.

Jelikož je Thread postaven na protokolu 6LoWPAN a odpovídá standardu IEEE 802.15.4, je vhodnou technologií pro tuto práci. Firma NXP je jednou z hlavních společností zabývajících se vývojem Thread, a aplikací pro něj, a nabízí demo aplikace je využívající pro svá zařízení. NXP má zároveň zájem o spolupráci se studenty, a proto byly zvoleny jejich produkty.

4.3 Výběr operačního systému

Aplikace pro zařízení tvořící Thread síť byly vyvíjeny v programovacím prostředí Kinetis Design Studio 3 IDE na operačním systému Windows 10. Z tohoto důvodu probíhalo původní testování jejich funkčnosti také na Windows 10. Bylo ale naraženo na problém v komunikaci mezi počítačem a sítí Thread. Počítač ani zařízení se nebyli schopni navzájem spojit přes ping ani mezi sebou otevřít socketovou komunikaci. Proto bylo zkušební programů přesunuto na operační systém Raspbian a zařízení Raspberry Pi. Na tomto systému ale stále nefungovalo posílání dat ze zařízení v síti na

Raspberry Pi. Tyto problémy byly způsobovány službou Firewall, která komunikaci blokovala. Poslední změna byla na Ubuntu MATE. Zde se potíže už neopakovaly a bylo úspěšně navázáno oboustranné spojení. Ubuntu MATE byl tedy zvolen jako operační systém pro tuto práci.

4.4 Výběr komunikačního protokolu

Pro komunikaci v sítích internetu věcí se využívá několik různých protokolů. Mezi ně patří například AMQP, XMPP, MQTT a CoAP. Tyto technologie zasílají zprávy v různých formátech. AMQP posílá proud bytů, XMPP komunikuje přes XML a oba jsou postavené na TCP. MQTT také přenáší data přes TCP, zároveň ale potřebuje prostředníka, který překládá a předává zprávy ze serveru klientům. CoAP je založen na UDP a podobá se náročnějšímu protokolu HTTP. Protože CoAP užívá UDP a zařízení od NXP jej podporují, a mají vytvořené demo aplikace s jeho využitím, byl CoAP zvolen pro tuto práci.

CoAP

CoAP je zkratka pro Constrained application protocol, definovaný v RFC 7252³¹. Tento komunikační protokol poskytuje možnost interakce mezi dvěma zařízeními v podobě žádosti a odpovědi, s pomocí příkazů GET, POST, PUT a DELETE, podobně jako protokol HTTP. Je zaměřený na zařízení s malým množstvím ROM a RAM a na sítě s vysokou ztrátovostí a nízkou spotřebou. Proto je vhodný pro sítě jako 6LoWPAN, a tím i pro tuto práci. Protokol v základu využívá UDP a díky tomu umožňuje použití unicast a multicast komunikace. Díky podobnostem s HTTP je možné tyto dva protokoly snadno propojit.

Nabízí také rozšíření CoAP Observe, umožňující sledování měnících se zdrojů na serveru. Díky tomuto lze na senzorových zařízeních vytvořit servery, na které se mohou ostatní zařízení připojit jako klienti. Server si určí, jaké typy dat vysílá a klienti si určí, jaké typy dat chtějí odebrat. Server pak periodicky vysílá nová data všem svým klientům, na základě jejich požadovaného typu.

Knihovny a demo aplikace od firmy NXP pro Thread poskytují nástroje pro využití CoAP i CoAP Observe.

4.5 Problémy při návrhu sítě zařízení

Původní návrh pro tuto síť by umožnil spuštění všech aplikací stisknutím tlačítek na zařízeních. Nejdříve by byl USB dongle zapojen do Raspberry Pi, a pak by na něm bylo dvakrát stisknuto tlačítko SW1, což by vytvořilo Thread síť. Poté by bylo nejdříve jednou stlačeno tlačítko SW4 na FRDM kitu, čímž by bylo vyvoláno připojení do sítě, a dále znovu zmáčknuto SW4 pro započatí vysílání dat. Aby tento návrh byl funkční, bylo by potřeba nastavit Thread síťovému rozhraní na Raspberry Pi statickou IPv6 adresu. Takto nastavená adresa ale způsobovala ztrátu veškeré komunikace mezi zařízeními. Byly provedeny pokusy ji nastavit přes terminál i grafické rozhraní Ubuntu MATE, zároveň byl také upraven nástroj NetworkManager, aby ignoroval dané rozhraní. Žádné z odzkoušených řešení nevyřešilo tento problém, a proto bylo usouzeno, že neleží chyba na straně nastavení adresy, ale v driverech, které adresu nesprávně ukládají do zařízení.

³¹ The Constrained Application Protocol (CoAP). *Internet Engineering Task Force (IETF)* [online]. ISSN 2070-1721.

Z tohoto důvodu musel být návrh upraven. Po nastartování sítě byla přečtena dynamická IPv6 adresa, přiřazená rozhraní, a pak byla předána jedním ze dvou způsobů do USB donglu. První způsob dále umožňoval spuštění stiskem tlačítek, ale vyžadoval vepsání IPv6 adresy do kódu aplikace běžící na FRDM kitu, který ji přes CoAP zprávu přeposlal USB donglu. Druhý způsob využíval shell rozhraní na FRDM kitu, které na USB donglu v módu border routeru chybělo. Přes toto rozhraní byla FRDM kitu sdělena získaná adresa a zároveň započata celá aplikace. Kit pak opět přes CoAP zprávu oznámil tuto adresu USB donglu.

4.6 Výběr Cloudové platformy

Pro podrobnější testování byly zvoleny platformy Amazon Web Services a Microsoft Azure. Obě platformy nabízí NoSQL databáze, vizualizaci dat, aplikace běžící v Cloudu, které mohou reagovat na příchozí data a zkušební dobu, po kterou je lze využívat zdarma. Testovací aplikace byly vytvořeny tak, aby byly spustitelné na Raspberry Pi 2 B. Obě aplikace zasílají do databáze v Cloudu čísla, která jsou po jejich příchodu kontrolována Cloudovými aplikacemi v AWS Lambda a Microsoft Azure Application Service, a pokud přesáhnou nastavený limit, odesílají e-mailovou notifikaci na testovací e-mail.

Amazon Web Services

AWS nabízí NoSQL databázi DynamoDB³², která podporuje ukládání dat ve formátech dokumentů a párů klíčů a hodnot. Tato databáze je propojená se službou AWS Lambda³³, což umožňuje jednoduché a efektivní programování triggerů reagujících na příchozí data do databáze. Samotné DynamoDB i triggerů v AWS Lambda se dají programovat v několika jazycích, mezi ně patří například Java, Node.js, C# nebo Python. Pro testovací aplikaci byl zvolen jazyk Python u obou služeb. Dále také lze získaná data vizualizovat ve službě QuickSight, ta ale zatím není propojená přímo s DynamoDB, a proto bylo napojení vytvořeno přeposláním dat z DynamoDB do úložiště Redshift, které už s QuickSight propojené je.

Python knihovna pro ovládání DynamoDB se nazývá Boto3 a Amazon k ní vytvořil dokumentaci. Tato dokumentace mimo jiné obsahuje návody pro instalaci a nakonfigurování všeho potřebného na klientském zařízení, ukázkové řešení řady problémů a rady jakým typickým chybám se vyvarovat. Kromě dokumentací pro jednotlivé funkce a služby AWS Amazon poskytuje zákazníkům návody, jak využít celou jejich platformu a jak ovládat jejich webovou konzoli.

Pro ovládání AWS služeb je potřeba na klientském zařízení nainstalovat a nakonfigurovat AWS Command Line Interface. Mezi potřebné konfigurace patří také určení přístupového ID kódu a tajného přístupového klíče, které jsou vygenerovány automaticky po vytvoření účtu ve webové konzoli. Musí se také nastavit region, ve kterém se nachází datacentrum, které zařízení bude využívat. Tyto informace pak používá knihovna Boto3 pro přístup do databáze.

Samotná aplikace běží v nekonečném cyklu a očekává vstup z konzole. Na vstup povoluje vepsat slovo end pro ukončení cyklu nebo celé číslo. Po získání celého čísla si aplikace zjistí aktuální

³² What is Amazon DynamoDB? *AWS Documentation* [online]. [cit. 2017-01-20]. Dostupné z:

<http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>

³³ What is AWS Lambda? *AWS Documentation* [online]. [cit. 2017-01-20]. Dostupné z:

<http://docs.aws.amazon.com/lambda/latest/dg/welcome.html>

čas a datum, které využije pro vytvoření jedinečného klíče, a tento pár klíče a hodnoty pak pošle jako JSON dokument do databáze. Po uložení hodnoty do databáze se spustí AWS Lambda trigger vytvořený a spuštěný v Cloudu. Trigger si zjistí příchozí hodnotu a porovná ji se stanoveným limitem. Pokud hodnota tuto hranici přesahuje, trigger pošle přes protokol smtp e-mailové upozornění, jinak žádnou reakci nevyvolá.

Hodnoty zaslané do databáze jsou pak zobrazitelné přes službu QuickSight v podobě grafů. Z důvodu nekompletního propojení DynamoDB a QuickSight se musí nastavit pravidelné zálohování DynamoDB databáze do úložiště Redshift. QuickSight pak aktualizuje svá data na základě dat uložených v Redshift a obnovuje všechny grafy automaticky.

Microsoft Azure

Od firmy Microsoft je možné využít NoSQL databázi DocumentDB³⁴, která data ukládá ve formátu kolekci dokumentů. Databáze je s jistými jazykovými omezeními propojitelná se službou Application Service, díky které lze naprogramovat obdobu triggeru z AWS Lambda. DocumentDB je ovladatelná přes Python, Javu, .NET, .NET Core a Node.js, program v Application Service je možné napsat v C#, JavaScriptu, F# a v některých případech i v Pythonu. Application Service trigger nelze napojit přímo na DocumentDB, je však možnost využít propojení se službami Service Bus nebo Queue storage, které dokáží přijímat zprávy vyvolávající spuštění triggerů. Dále lze spojit databázi DocumentDB s programem Microsoft Power BI umožňujícím tvorbu grafů na základě jejich dat.

Dokumentace pro knihovny od Microsoftu se zásadně liší jazyk od jazyku. Například pro .NET dokumentace obsahuje návody na vytvoření databáze a kolekce ve webové konzoli a příklady kódu na ovládání jak dokumentů, tak kolekci i databáze samotné. Podrobně popisuje jednotlivé kroky vytváření konzolové aplikace a je udržována v aktuálním stavu. Dodatečné příklady kódu jsou pak v Github repositáři pro DocumentDB. Naopak pro Python obdoba návodu na tvorbu konzolové aplikace neexistuje. Na místo něj Microsoft poskytuje pouze návod pro vytvoření webové aplikace v Python Flask, který obsahuje starou syntaxi nahrazenou v roce 2015. V Github repositáři jsou ukázkové kódy pro konzolovou aplikaci využívající nejnovější knihovny, kódy jsou ale jen pro ovládání databáze a kolekce, ne pro ovládání dokumentů. Pro vývoj triggerů v Application Service lze vygenerovat šablonu, která by měla tvorbu aplikací usnadnit. Šablona pro jazyk .NET a napojení na službu Service Bus je ale vygenerována tak, že se data vyslaná na Service Bus ukládají do řetězce. Při posílání dat z .Net programu na počítači je však nutné použít třídu BrokeredMessage³⁵ a jejímu objektu upravit jmenný prostor přes DataContract³⁶ na prostor využívaný v Application Service. Z tohoto důvodu byla šablona nefunkční³⁷ a musela být upravena pro správné přijímání BrokeredMessage objektů.

³⁴ Úvod do DocumentDB. *Microsoft Azure* [online]. [cit. 2017-01-20]. Dostupné z:

<https://docs.microsoft.com/cs-cz/azure/documentdb/documentdb-introduction>

³⁵ BrokeredMessage Class. *Microsoft Azure* [online]. [cit. 2017-01-20]. Dostupné z:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.servicebus.messaging.brokeredmessage>

³⁶ Using Data Contracts. *Microsoft Azure* [online]. [cit. 2017-01-20]. Dostupné z:

[https://msdn.microsoft.com/en-us/library/ms733127\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms733127(v=vs.110).aspx)

³⁷ Service Bus and BrokeredMessage. *StackOverflow* [online]. [cit. 2017-01-20]. Dostupné z:

<https://stackoverflow.com/questions/41039785/how-do-i-use-a-documentdb-database-input-binding-with-a-service-bus-trigger-bind>

Testovací aplikace pro border router byla nejprve vytvořena v jazyku .NET. Díky programovacímu prostředí Microsoft Visual Studio, nabízenému pro studenty zdarma, bylo jednoduché instalovat knihovny balíčky pro ovládání DocumentDB. Bohužel, tyto balíčky zatím nebyly vytvořeny pro Mono³⁸, které umožňuje kompilaci a spouštění .NET programů na Linuxových operačních systémech. Proto byla tato testovací aplikace využita pouze pro seznámení se s vývojem pro DocumentDB, a nabyté zkušenosti byly využity při tvorbě druhé testovací aplikace v Pythonu. Instalace knihoven pro Python na operačním systému Raspbian byla složitější než ve Visual Studiu pro .NET, a kvůli některým nepřesným informacím a chybám v instalačních souborech bylo potřeba provést několik přeinstalací, než všechny moduly začaly fungovat správně.

Obě aplikace pro připojení do databáze potřebují přístupový klíč, vygenerovaný po vytvoření databáze ve webové konzoli a URI odkaz na databázi. Dále pro přístup do databáze je potřeba její ID a pro přístup do kolekce je potřeba ID kolekce. Z těchto ID se skládají ID řetězce, díky kterým přístupové funkce ví, kam přistupovat. Pro připojení do Queue Storage a Service Bus je opět potřebný klíč, vygenerovaný při jejich vytvoření, a název zvolený uživatelem.

Aplikace napsaná v .NET běží po určenou dobu v cyklu a v nastaveném intervalu generuje náhodné číslo v určitém rozmezí a zjistí si aktuální čas a datum. Z času a data vytvoří jedinečné ID pro nový dokument a tohle ID a vygenerovanou hodnotu uloží do objektu třídy, která je serializuje do JSON formátu, a tento objekt pošle do DocumentDB databáze. Dále pak uloží hodnotu ID do objektu třídy se specificky nastaveným jmenným prostorem, vyžadovaným službou Application Service, a z tohoto objektu vytvoří objekt třídy BrokeredMessage a pošle jej na frontu Service Bus. Zpráva poslaná na Service Bus spustí .NET trigger vytvořený v Application Service který dokáže ze zprávy získat ID dokumentu, napojit se do DocumentDB a přečíst nejnovější přidanou hodnotu. Pokud tato hodnota přesahuje nastavený limit, trigger pošle přes protokol smtp upozornění na určený e-mail, jinak nijak nereaguje.

Aplikace vytvořená v Pythonu funguje podobně. Na rozdíl od .NET aplikace běží nekonečně, ne po určenou dobu, ID a hodnotu neukládá do objektu žádné třídy, ale do slovníku, jehož názvy indexů určí názvy položek v dokumentu v databázi a místo na Service Bus posílá zprávy na Queue Storage, která funguje obdobně. Trigger v Application Service je opět napsaný v .NET a zprávu dokáže přečíst přímo jako řetězec, který pak využije jako ID pro přečtení dokumentu v databázi.

Služba Power BI nakonec čte hodnoty z databáze a vytváří z nich grafy. Pro tyto grafy lze nastavit časový interval, který mají zobrazit, například poslední týden nebo měsíc. Microsoft nabízí tři aktualizací intervaly, týdenní, denní a hodinový. Týdenní a denní jsou zdarma, hodinový jen pro placené uživatele a umožňuje maximálně 8 aktualizací za den.

Porovnání použitých platforem

Vytvářením testovacích aplikací a prací s oběma platformami byly získány cenné zkušenosti jak s platformami, tak celkově se způsobem, jak fungují systémy Platform-as-a-Service a Application-as-a-Service.

Cloudové platformy jsou nabízejícími firmami prezentovány jako způsob, jak ušetřit peníze, ale pro projekt malých rozměrů, jako tato bakalářská práce, jsou ceny za využívání jejich služeb příliš vysoké. Často jsou nabízeny způsoby, jak využít dané platformy zdarma, ale pro tento projekt nejsou dostatečné. V případě Amazon Web Services bylo potřeba použít placenou službu Redshift, u

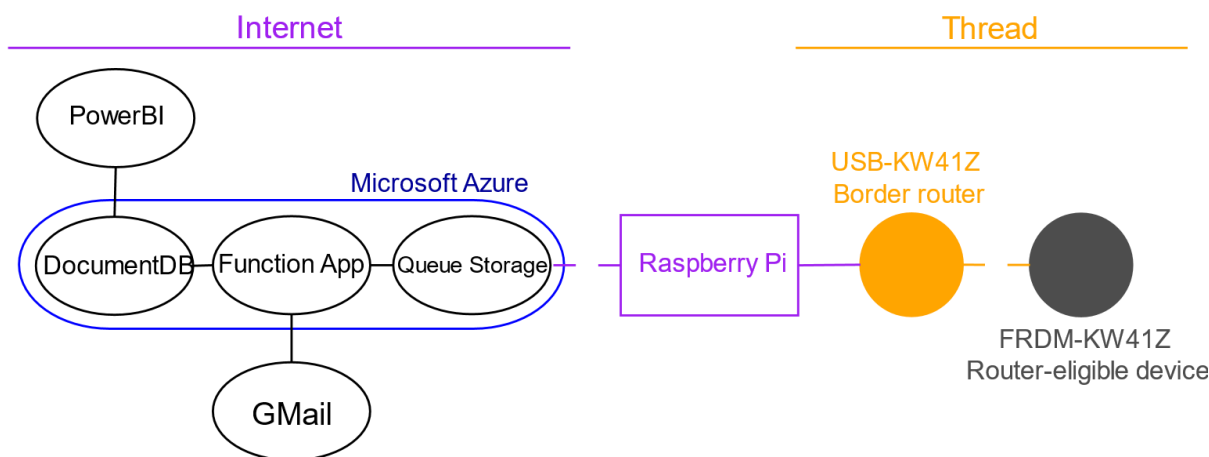
³⁸ About Mono. *Mono* [online]. [cit. 2017-01-20]. Dostupné z: <http://www.mono-project.com/docs/about-mono/>

Microsoft Azure trvala zkušební doba pouze jeden měsíc a následný měsíční kredit 25 euro vystačil na provoz projektu po dobu celého měsíce jen při použití jediné kolekce dokumentů.

Protože je většina platforem relativně mladých a výrobci jim stále dodávají nové funkce a služby, jejich vnitřní propojení zatím není dokonalé. Při tvorbě testovacích aplikací bylo naráženo na překážky vyplývající z těchto nedokonalostí. Nejzávažnější z nich byla chybějící možnost přímo propojit Amazon DynamoDB s Amazon QuickSight. Nedostatky u Microsoft Azure se daly vyřešit jednodušeji, a proto bylo Microsoft Azure zvoleno jako konečná platforma.

5 Implementace a testování

Tato kapitola popisuje finální verzi bakalářské práce a její testování. Projekt se dá rozdělit do několika částí, koncové zařízení vytvořené z FRDM-KW41Z, border router skládající se z USB-KW41Z a Raspberry Pi 2 B a databázi a její správu v Cloudu, využívající služby DocumentDB, Function App a Queue Storage z platformy Microsoft Azure a vizualizační nástroj PowerBI.



Obr. 5.1: Rozložení finální verze sítě

5.1 Koncové zařízení FRDM-KW41Z

Demo aplikace pro Thread poskytované k zařízením od firmy NXP jsem částečně upravil, aby odpovídaly potřebám této bakalářské práce. Mnou upravené a přidáné funkce v programu běžícím na FRDM-KW41Z se starají o měření a posílání teploty a oznámení své IP adresy a přidělené fronty v Queue Storage do border routeru. Tyto funkce se nazývají APP_ReportTemp, SHELL_AppStart, APP_GetTempData a APP_ObserveServerUpdate.

Funkce APP_ReportTemp je vyvolána stisknutím tlačítka SW4 na desce, pokud je již zařízení připojeno do sítě. Funkce nejdříve složí zprávu, skládající se z několika řetězců. První z nich značí, jaký typ dat bude dané zařízení vysílat, další pak délku ID přidělené fronty, délku IP adresy Raspberry Pi, samotné ID fronty a IP adresu. Dále funkce spustí CoAP Observe server a složenou zprávu vyšle přes CoAP protokol do Thread sítě, což spustí APP_CoapAddNewCb na USB-KW41Z.

SHELL_AppStart je využívána pro možnost spuštění celého systému přes shell prostředí. Funguje téměř stejným způsobem jako APP_ReportTemp. Na rozdíl od ní pouze nemá IP adresu Raspberry Pi pevně uloženou v kódu, ale získá ji při zadání příkazu appstart a dané adresy do shellu.

APP_GetTempData vyvolá funkci pro přečtení aktuální hodnoty teploty na desce, převede ji z 32 bitového čísla na 8 bitové a předá jej ukazateli na 8 bitová čísla. Tento ukazatel pak vrátí jako návratovou hodnotu.

APP_ObserveServerUpdate je po spuštění CoAP Observe serveru periodicky volána v zadaném časovém intervalu. Přes APP_GetTempData získá aktuální teplotu a tu porovná s poslední naměřenou, která je po inicializaci programu nastavena na INT8_MAX. Dále inkrementuje počítadlo časových intervalů, uběhlých od poslední aktualizace teploty. Pokud se nově naměřená teplota nerovná poslední nebo uběhl maximální povolený čas mezi aktualizacemi, uloží novou teplotu jako poslední, vynuluje počítadlo intervalů a všem naslouchajícím klientům vyšle novou aktualizaci teploty.

5.2 USB Dongle USB-KW41Z

Na tomto zařízení jsem vytvořil funkci APP_CoapAddNewCb a upravil APP_ObserveRcvData. Starají se o předávání dat přijatých z koncového zařízení do Raspberry Pi za pomoci socketů.

Funkce APP_CoapAddNewCb je callback funkce, která je vyvolána CoAP zprávou žádající o zařazení nové IP adresy a ID fronty do slovníku na Raspberry Pi. Funkce na vstupu přijme ukazatel na void, přes který může číst řetězec, složený buď v APP_ReportTemp nebo v SHELL_AppStart, a rozdělí ho zpět na jeho podčásti. Dále vytvoří řetězec z čísla nula, určující, že zpráva, kterou zašle na Raspberry Pi, požaduje zařazení nové IP adresy a ID fronty do slovníku. Poté zjistí adresu zařízení, od kterého přijalo CoAP zprávu, a uloží ji dalšího řetězce. V dalších krocích uloží do řetězců délku zjištěné IP adresy a délku ID fronty, které, pokud obsahuje pouze jednu cifru, přidá na začátek nulu. Pak všechny tyto kusy slepí dohromady. Na začátek vloží typ zprávy, za něj délku IP adresy, délku názvu fronty, IP adresu a název fronty. Nakonec slepenou zprávu pošle přes UDP socket na Raspberry Pi a spustí CoAP Observe klienta naslouchajícího aktualizacím dat daného typu z nově přidaného zařízení.

Typ zprávy	Délka IP adresy	Délka ID fronty	IP adresa	ID fronty
------------	-----------------	-----------------	-----------	-----------

Obrázek 5.2: Formát zprávy pro zařazení no IP adresy a názvu fronty do slovníku

Funkce APP_ObserveRcvData se spustí, když CoAP Observe server vyšle aktualizaci teploty. Tu přečte jako osmibitovou celočíselnou hodnotu přes vstupní ukazatel na void. Dále vytvoří několik řetězců, první s číslem jedna, značící, že se zasílá nová hodnota do databáze, pak s IP adresou serveru, který teplotu vyslal, a nakonec s teplotou samotnou. Zjistí také délku adresy a počet číslic v teplotě, které, pokud má jen jednu cifru, na začátek přidá nulu. Nakonec vše spojí do jednoho řetězce, nejprve typ zprávy, pak délku IP adresy, počet cifer v teplotě, IP adresu a teplotu. Tuto zprávu přes UDP pošle na Raspberry Pi.

Typ zprávy	Délka IP adresy	Délka teploty	IP adresa	Teplota
------------	-----------------	---------------	-----------	---------

Obrázek 5.3: Formát zprávy pro zařazení zaslání nové teploty do databáze

5.3 Raspberry Pi 2 B

Pro Raspberry Pi jsem si zvolil operační systém Ubuntu MATE. Po vytvoření Thread sítě se na Raspberry Pi spustí program, napsaný v jazyce Python, který se stará o předávání dat získaných z Thread sítě do Cloudu. Program využívá knihovnu SocketServer, pro vytvoření mnohovláknového UDP serveru, a třídu QueueService z knihovny azure.storage.queue, poskytovanou firmou Microsoft, pro komunikaci s Cloudem. Na začátku vytvoří objekty tříd QueueService a ThreadedUDPServer. Prvnímu nastaví název a klíč k účtu Queue Service, a druhému určí port a IP adresu, na kterých je potřeba, aby program naslouchal, a spustí UDP server na nekonečně dlouhou dobu.

Dále jsem vytvořil dvě třídy, `CloudManagement` a `ThreadedUDPRequestHandler`. `CloudManagement` má pouze jednu statickou metodu, `send_qmsg`, která se stará o posílání dat do fronty `Queue Storage`. Na vstup dostane ID fronty a ID řetězec a hodnotu, ze kterých vytvoří dokument ve formátu JSON. Přes dříve vytvořený objekt třídy `QueueService` pak spustí metodu `put_message`, které předá název fronty a JSON dokument, a ta jej pošle do fronty.

`ThreadedUDPRequestHandler` obsahuje metodu `handle`, která je vyvolána v samostatném vlákně při každém přijmutí dat na socketu. Z příchozího řetězce nejprve přečte první znak, určující, zda se jedná o zařazení nové IP adresy do globálního slovníku adres nebo o přeposílání dat do databáze v Cloudu. V obou případech zprávu rozdělí zpět na podčásti, znázorněné v obrázcích 5.2 a 5.3. V případě zařazení nové IP adresy uzamkne slovník a adresu do něj vloží, při přeposílání hodnoty ji převede na celé číslo a zjistí aktuální datum a čas, které uloží do řetězce, pro použití jako ID dokumentu v databázi. S pomocí slovníku adres pak volá `send_qmsg` s potřebnými parametry.

5.4 Cloud

Správa databáze

Využil jsem možnost propojení funkcí vytvořených ve `Function App` s dalšími částmi platformy. Spojení jsem sestavil v integračním prostředí, umožňujícím určit trigger, vstup a výstup funkce. Jako trigger jsem zvolil přijmutí dat na frontě v `Queue Storage`, za vstup dokument `LastUpdate`, udržující informaci o poslední poslané notifikaci a výstupem jsem nastavil vložení nového dokumentu do databáze `DocumentDB`. Funkci ve `Function App` jsem napsal v jazyce `.Net`.

Po jejím vyvolání funkce nejprve projde přijatý JSON dokument a uloží jeho ID do řetězce `newTimeStr` a hodnotu do proměnné `newValue`. Pak s jejich pomocí vytvoří nový dokument, který po ukončení uloží do databáze. Dále převede `newTimeStr` a časový řetězec z dokumentu `LastUpdate` do typu `DateTime` a vytvoří z nich proměnné `newTime` a `lastTime`. Od `newTime` odečte `lastTime` a výsledek této operace uloží do proměnné `timeDif`, typu `TimeSpan`. Poté, pokud se nově naměřená teplota rovná poslední poslané v notifikaci, nastaví proměnnou `minDif` na delší časovou dobu, jinak na kratší. Následně zjistí, jestli je `timeDif`, reprezentující čas, který uběhl od poslední notifikace, delší než `minDif`. Pokud ano, zkontroluje hodnotu nově naměřené teploty. V případě, že je mimo určené hranice, vytvoří notifikační řetězec toto oznamující. Jestliže je v pořádku, podívá se, zda byla poslední notifikace poslána kvůli špatné hodnotě, a když ano, vytvoří notifikační řetězec, sdělující, že teplota je již v normě. Nakonec podle potřeby odešle notifikační e-mail nebo pouze skončí.

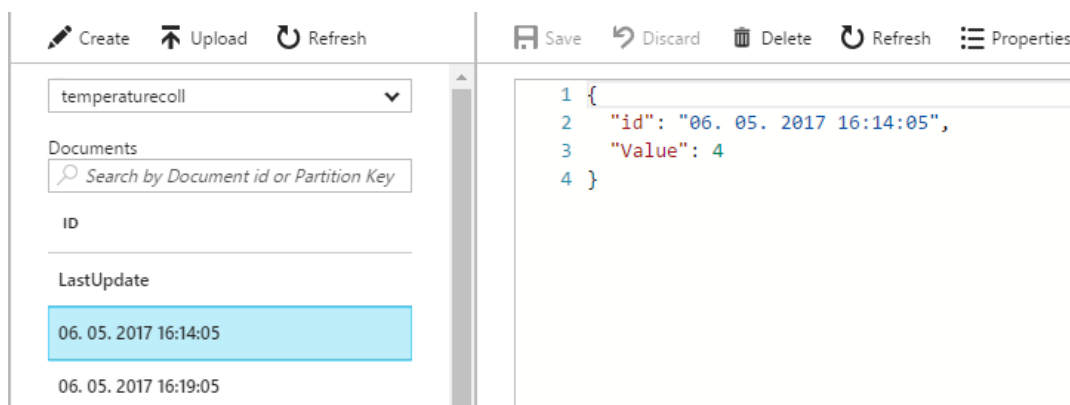
Vizualizace dat

Pro vizualizaci dat jsem použil program `Power BI` nabízený společností `Microsoft`. Nástroj má funkční napojení i na databázi `DocumentDB`. Bylo mu potřeba nastavit zdroj dat, což bylo možné přes URI odkaz na databázi, sestavený při jejím vytvoření. Dále vyžadoval určení názvu databáze a kolekce, ze které měl čerpat dokumenty. Ze získaných dokumentů pak udělal tabulku se dvěma sloupci, s ID a s hodnotou teploty. Sloupci s ID bylo nutné říct, aby ignoroval dokument `LastUpdate`. Pak mohl být celý sloupec nastaven na typ `data` a času a sloupec s hodnotou teploty na typ celého čísla. To umožnilo tvorbu časových filtrů pro grafy. Nakonec bylo zapotřebí nastavit časy pro automatickou aktualizaci dat.

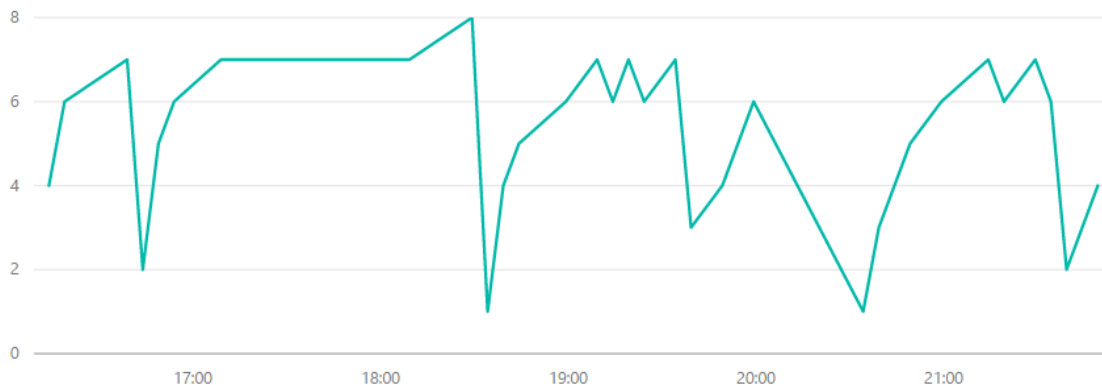
5.5 Testování

Měření teploty

Cílem testování bylo prokázat funkčnost celé práce, a verifikovat naměřené hodnoty. Teplota byla měřena v lednici od firmy LG, nastavené na udržování 3 °C. Referenční teplota byla získávána zařízením Escort 22 TC Calibrator/Thermometer. Interval měření teploty byl nastaven na 5 minut, maximální doba mezi aktualizacemi teplot na 60 minut. Delší doba mezi posíláním notifikací byla 60 minut, kratší 20 minut. Teplotní limity byly určeny jako 0 °C minimum a 4 °C maximum. Z hodnot byl tvořen graf, interval automatické aktualizace dat v něm byl 3 hodiny. Měření bylo prováděno po dobu 5 hodin a 35 minut.



Obr. 5.4: Způsob uložení dat v databázi



Temperature notification - The temperature is now ok. Time of measurement: 06. 05. 2017 20:34:21, measured value: 1.

Temperature notification - The temperature is too high! Time of measurement: 06. 05. 2017 19:54:20, measured value: 5.

Obr. 5.7: Notifikační e-maily

Ze získaného grafu je patrné, že lednice prochází chladicími cykly. Po jednom z těchto cyklů byly otevřeny její dveře, což způsobilo rychlé zvýšení teploty, jak je znázorněno v detailu grafu na obrázku 5.6. Jelikož ale byla vyslána notifikace v čase 16:39, oznamující, že teplota dosáhla hodnoty 7 °C, a následné zchlazení a opětovné oteplení lednice proběhlo během 15 minut poté, nebyly zaslány notifikace ani o spravení teploty, ani o jejím opakovaném překročení maxima. Pak se od 16:54 do 17:09 teplota neměnila, a v 17:09 vystoupala na 7 °C, kde zůstala až do 18:29. Další notifikace tedy byla odeslána až v 18:09, po uplynutí hodinových intervalů pro měření a notifikace.

Tímto testováním byla ověřena správná funkčnost programů, které jsem implementoval, a referenčním měřením teplot byla potvrzena přesnost senzoru. Byl ale také objeven problém s USB donglem, který způsoboval pád sítě po náhodně dlouhé době běhu, od 5 minut až po 5 hodin, a zabraňoval zpětnému připojení koncového zařízení. Pro odblokování sítě bylo potřeba USB dongle restartovat a celou síť vytvořit znovu. Potíže by mohly být způsobovány například vzájemným rušením mezi 802.15.4 a WLAN, jelikož obě technologie využívají kanály se stejnými frekvencemi³⁹. Síť byla testována i v prostředí bez wi-fi rušení, kde fungovala stabilně. K těmto testům byla upravena frekvence měření teplot na 30 vteřin a systém neztratil schopnost komunikovat ani po stovkách provedených měření a zaslání do databáze.

Spotřeba energie

Na zařízení FRDM-KW41Z byl měřen odběr energie. Při napětí 3.3 V bylo po spuštění zařízení a při blikání LED diod odebráno 50 mA, po zhasnutí LED diod a připojování k síti 40 mA a po úspěšném připojení a rozsvícení červené indikační LED diody 44 mA. Při kapacitě použité power banky 8000 mAh, by tak zařízení mělo vydržet asi týden nepřetržitého běhu. Spotřeba je zvýšena mikrokontrolerem K22F, LED diodami a dodatečnými periferiemi, které jsou na zařízení využity, protože se jedná o demonstrační vývojový kit. V případě dalšího rozvíjení práce by bylo zapotřebí vyrobit zařízení obsahující jen ty části, které jsou nezbytné pro jeho funkčnost. Tímto krokem by se výdrž prodloužila na měsíc nebo déle.

Dosah signálu

U sítě byl také testován dosah signálu. Koncové zařízení si udrželo připojení k border routeru i na vzdálenost 50 metrů, což je dostačující pro využití v chytrých domácnostech. Problémy se neobjevily ani při komunikaci přes zed'.

³⁹ *Co-existence of IEEE 802.15.4 at 2.4 GHz: Application Note* [online]. [cit. 2017-05-10]. Dostupné z:

http://www.nxp.com/documents/application_note/JN-AN-1079.pdf

6 Závěr

Cílem této práce bylo studium internetu věcí a technologií vhodných pro jeho implementaci, zejména protokolu 6LoWPAN a technologie Thread. S pomocí těchto technologií bylo součástí zadání vytvořit bezdrátovou síť schopnou komunikace s internetem a Cloudem. Tyto cíle byly splněny a řešení bylo otestováno dlouhodobým během a sledováním systému.

Mezi hlavní výhody vytvořeného projektu patří nízká spotřeba koncového zařízení, zajištěná využitím úsporných technologií 6LoWPAN, CoAP a UDP, a používáním vysílače zařízení pouze v případě změny teploty nebo po uplynutí potřebného časového intervalu. Implementace je dále rozšiřitelná i na několik desítek zařízení snímajících různé informace pouhým přidáním obslužných funkcí pro konkrétní typy dat. Díky využití cenově dostupných zařízení je projekt vhodný pro tvorbu chytrých domácností. Spojení s NoSQL Cloudovou databází umožňuje užití velkého množství senzorů kontrolujících různé typy dat a usnadňuje změny ve způsobu měření a ukládání informací, a zároveň zprostředkovává dostupnost naměřených hodnot kdekoliv, kde je připojení k internetu. V neposlední řadě také zajišťuje přehled průběhu změn sledovaných hodnot přes pravidelně aktualizované grafy, zobrazitelné přes webové rozhraní, a e-mailové notifikace.

Do budoucna je několik aspektů, které by bylo vhodné upravit nebo studovat nadále. Například by bylo potřeba objevit a opravit chyby v software, poskytovanému k využitým zařízením, které způsobovaly problémy při nastavování pevné IP adresy, a také nedokonalosti stojící za pády sítě. Dále by bylo užitečné síť rozšířit o několik dalších senzorů, například pro měření vlhkosti hlíny v květináčích, kontrolu otevření oken a dveří, ovládání osvětlení nebo sledování spotřeby energie v obytném domě. Z pohledu uživatelského rozhraní by bylo účelné vytvořit kupříkladu mobilní aplikaci, umožňující zobrazení a ovládání několika grafů nebo i samotných senzorů a zařízení. Na straně Cloudu by pak bylo patřičné zhodnocení cenové dostupnosti nabízených služeb, a jejich využitelnost pro velmi malé projekty, a případné vyhledání cenově dostupnějších možností. Z důvodu časových omezení při vytváření bakalářské práce ale nemohly být všechny tyto nápady realizovány.

Literatura

[1] VERMESAN, Ovidiu (ed.) a Peter FRIESS (ed.). *Internet of things: converging technologies for smart environments and integrated ecosystems*. Aalborg: River Publishers. ISBN 978-879-2982-735.

[2] *An Introduction to the Internet of Things (IoT)* [online]. [cit. 2015-12-27]. Dostupné z: http://www.cisco.com/web/solutions/trends/iot/introduction_to_IoT_november.pdf

[3] OLSSON, Jonas. *6LoWPAN demystified* [online]. 2014 [cit. 2015-12-28]. Dostupné z: <http://www.ti.com/lit/wp/swry013/swry013.pdf>

[4] EVANS, Dave. *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything* [online]. 2011 [cit. 2015-12-28]. Dostupné z: http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[5] CLARKE, Ruthbea Yesner. *Smart cities and internet of everything: The foundation for developing next-generation citizen services* [online]. 2013 [cit. 2015-12-28]. Dostupné z: <https://pdfs.semanticscholar.org/8a67/62e4847685fd1ce8a9fe4018dce3703cd47d.pdf>

[6] SPONSOR, LAN/MAN Standards Committee of the IEEE Computer Society. IEEE standard for local and metropolitan area networks: Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) [online]. New York: Institute of Electrical and Electronics Engineers, 2011. ISBN 978-073-8166-841.

[7] ERGEN, Sinem Coleri. ZigBee/IEEE 802.15.4 Summary [online]. 2004 [cit. 2015-12-30]. Dostupné z: <http://users.ece.utexas.edu/~valvano/EE345L/Labs/Fall2011/Zigbeeinfo.pdf>

[8] HUI, J (ed.). RFC6282: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks [online]. 2011. ISSN 2070-1721.

[9] ARM. ARM Architecture Reference Manual: ARMv7-A and ARMv7-R edition [online]. [cit. 2015-12-30]. Dostupné z: https://silver.arm.com/download/ARM_and_AMBA_Architecture/AR570-DA-70000-r0p0-00rel2/DDI0406C_C_arm_architecture_reference_manual.pdf

[10] HASSAN, Qusay F. *Demystifying Cloud Computing* [online]. 2011 [cit. 2016-12-26]. Dostupné z: <http://static1.1.sqspcdn.com/static/f/702523/10181434/1294788395300/201101-Hassan.pdf?token=KuQ711b4oxK2IWlsogkRvalQSfc%3D>

[11] MELL, Peter a Timothy GRANCE. *The NIST Definition of Cloud Computing* [online]. 2011 [cit. 2016-12-26]. Dostupné z: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

[12] GROLINGER, Katarina, Wilson A HIGASHINO, Abhinav TIWARI a Miriam AM CAPRETZ. Data management in Cloud environments: NoSQL and NewSQL data stores. *Journal of Cloud Computing: Advances, Systems and Applications*. 2013, **2013**(2:22). DOI: 10.1186/2192-113X-2-22.

[13] HECHT, Robin a Stefan JABLONSKI. *NoSQL Evaluation: A Use Case Oriented Survey* [online]. 2011 [cit. 2016-11-20]. Dostupné z:
<https://pdfs.semanticscholar.org/b316/2e0faf4ed736da6b7c043d5a4b8549d10772.pdf>

[14] *Thread Stack Fundamentals* [online]. 2015 [cit. 2016-12-26]. Dostupné z:
http://threadgroup.org/Portals/0/documents/whitepapers/Thread%20Stack%20Fundamentals_v2_public.pdf

[15] FARAHANI, Shahin. Zigbee wireless networks and transceivers. Elsevier, 2008. ISBN 978-0-7506-8393-7.