

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

**Vizualizační systémy pro zpracování a
analýzu dat z IQRF MESH sítí**

**Visualization systems for processing and
analyzing data from IQRF MESH
networks**

Zadání diplomové práce

Student: **Bc. Radek Byrtus**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2612T041 Řídicí a informační systémy

Téma: **Vizualizační systémy pro zpracování a analýzu dat z IQRF MESH sítí**
Visualization Systems for Processing and Analyzing Data from IQRF
MESH Networks

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Seznámení se s technologií IQRF.
2. Analýza dostupných vizualizačních systémů pro zpracování dat z IQRF MESH sítí.
3. Návrh a realizace demonstrační úlohy měření environmentálních veličin pomocí IQRF technologie na platformě RPI.
4. Návrh a realizace propojení měřicího a vybraného vizualizačního systému.
5. Implementace subsystému pro zpracování a analýzu dat do vybraného vizualizačního systému.
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] KREIDL, Marcel. *Měření teploty, senzory a měřicí obvody*. 1. vyd. Praha: BEN - technická literatura, 2005. 240 s. ISBN 80-7300-145-4.
- [2] MICRORISC. *IQRF Quick start guide*. [online]. © 2017 [cit. 2017-03-10]. Dostupné z: <http://www.iqrf.org>.
- [3] MICRORISC. *IQRF Cloud Technical guide*. [online]. © 2015 [cit. 2015-09-21]. Dostupné z: <http://www.iqrf.org>.
- [4] MICRORISC. *TR-72D datasheet*. [online]. © 2017 [cit. 2017-03-22]. Dostupné z: <http://www.iqrf.org>.
- [5] MICRORISC. *IQRF OS v4.00D User's guide for TR-7xD*. [online]. © 2017. [cit. 2017-05-29]. Dostupné z: <http://www.iqrf.org>.
- [6] MICRORISC. *DPA Framework Technical Guide v3.00*. [online]. © 2017 [cit. 2017-03-14]. Dostupné z: <http://www.iqrf.org>.
- [7] MICRORISC. *IoT-StarterKit-01 User's guide*. [online]. © 2017 [cit. 2017-06-02]. Dostupné z: <http://www.iqrf.org>.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

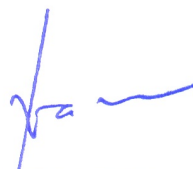
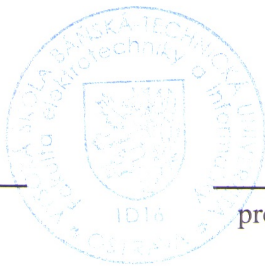
Vedoucí diplomové práce: **Ing. Radovan Hájovský, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2018

.....Byntus.....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 30. dubna 2018

.....*Bjntao*.....

Rád bych na tomto místě vyjádřil poděkování všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Zejména bych pak rád vyjádřil přímé poděkování:

- Ing. Radovanovi Hájovskému, Ph.D. za směřodátne konzultace, podané rady, vedení a korekturu textu diplomové práce.
- doc. Mgr. Jiřímu Dvorskému, Ph.D. za zpracování typografické knihovny *diploma* pro L^AT_EX.
- Ing. Zdeňku Slaninovi Ph.D. za poskytnutí úložných prostor pro data z IQRF MESH síť.
- Své rodině a přítelkyni za podporu, jenž mi během vypracování prokázali.

Abstrakt

Tato diplomová práce se zabývá vizualizačními systémy pro vizualizování, zpracování a analýzu naměřených dat z IQRF MESH sítí. Stěžejní bod práce tedy je zpracování pilotního projektu na vybrané vizualizační platformě zahrnující aplikaci různých statistických, analytických i predikčních metod nad vizualizovanými daty. Součástí praktické části této diplomové práce je také návrh a realizace demonstrační úlohy pro měření environmentálních veličin, jež bude součástí, spolu s Raspberry Pi, měřicího systému, na jehož výstupu budou k dispozici v podobě databázové tabulky vstupní data pro vybraný vizualizační systém.

Klíčová slova: Elektronika, DS18B20, BMP280, BME280, DHT11, DHT22, teplota, vlhkost vzduchu, atmosférický tlak, IQRF, MESH, DPA, C, python, Java, nod, koordinátor, modul, analýza, vizualizace, zpracování, data, environmentální veličiny, databáze, SQL, noSQL, MQTT, Watson IoT, Watson Analytics, Apache Spark

Abstract

This diploma thesis is intended analysis of the visualization systems for processing and analysis data from IQRF MESH networks. The main part of the final thesis is the preparation of a pilot project on selected visualization platform, including the integration of the subsystem for data processing, analysis and prediction. Part of the practical part of this thesis is also the desing and realization of a demonstration task based on IQRF technology, which will form, together with Raspberry Pi, measurement system. This measurement system will store the data into a database, which then serves as the data source for the selected viausualisation platform.

Key Words: Electronics, DS18B20, BMP280, BME280, DHT11, DHT22 temperature, relative humidity, air pressure, IQRF, MESH, DPA, C, python, Java, node, coordinator, module, analysis, visualization, processing, data, enviromental quantity, database, SQL, noSQL, MQTT, Watson IoT, Watson Analytics, Apache Spark

Obsah

	Strana
Seznam použitých zkratek a symbolů	10
Seznam obrázků	11
Seznam tabulek	13
Seznam výpisů zdrojového kódu	14
1 Úvod	15
2 Technologie IQRF	17
2.1 Hardware	17
2.1.1 Komunikační pásmo	19
2.2 Software	19
2.3 Zprávy DPA protokolu	21
2.4 Síťová topologie a MESH	22
3 Použité technologie v rámci DP	24
3.1 Raspberry Pi	24
3.2 SQL databáze	25
3.3 MQTT a MQTT broker	26
3.3.1 Podoba zpráv	28
4 Analýza vizualizačních systémů	29
4.1 Grafana	30
4.1.1 Možnosti vizualizace	30
4.1.2 Možnosti analýzy	30
4.1.3 Licence	31
4.2 Node-RED	32
4.2.1 Možnosti vizualizace	33
4.2.2 Možnosti analýzy	33
4.2.3 Licence	34
4.3 MS Azure a Power Bi	34
4.3.1 Možnosti vizualizace	34
4.3.2 Možnosti analýzy	35
4.3.3 Licence	36
4.4 IBM Bluemix a Watson IoT	37
4.4.1 Možnosti vizualizace	37

4.4.2	Možnosti analýzy	39
4.4.3	Licence	40
4.5	Srovnání vizualizačních systémů	40
5	Realizace demonstrační úlohy	43
5.1	Hardwarové zapojení	45
5.1.1	Senzor Dallas DS18B20	46
5.1.2	Senzor DHT11	47
5.1.3	Senzor BME280	48
5.2	Programy uzlů	49
5.2.1	Uzel 1 - BME280	49
5.2.2	Uzel 2 - DS18B20, DHT11	50
5.3	Realizace měřicího systému	50
5.3.1	Služba iqrfd-daemon	51
5.3.2	Služba mqtt-to-sql	52
5.3.3	MQTT zprávy	54
6	IBM Bluemix a Watson IoT	56
6.1	Propojení s databází	56
6.1.1	Datová pumpa	57
6.1.2	Připojení k Watson IoT	58
6.2	Vizualizace	59
6.2.1	Datová historie	60
6.2.2	Notifikace, alarmy	62
6.3	Datová analýza	63
6.3.1	Node-RED	63
6.3.2	Watson Analytics	74
7	Závěr	79
	Literatura	81
	Přílohy	82
A	Seznam příloh na CD	83
B	Tabulka s porovnáním jednotlivých modulů transceiverů	84

Seznam použitých zkratek a symbolů

IQRF	– Nízkopříkonová bezdrátová technologie pro přenos malých dat
MESH	– Peer-to-peer architektura bezdrátové sítě, smyčková síť, uzlová síť
SQL	– Standardizovaný strukturovaný dotazovací jazyk
IoT	– Internet věcí (Internet of Things)
VF	– Oblast vysokých kmitočetů (High frequencies)
MCU	– Mikrokontrolér (Microcontoller)
OS	– Operační systém (Operational system)
EEPROM	– Elektronicky vymazatelná paměť pouze pro čtení (Electrically erasable programmable read-only memory)
SoC	– System on Chip
SPI	– Serial Peripheral Interface
QoS	– Quality of service - korektura, potvrzování předání
MQTT	– Message Queuing Telemetry Transport
JSON	– Javascriptový objektový zápis
BSON	– Binární javascriptový objektový zápis
WA	– Watson Analytics; Služba dostupná na cloudu IBM Bluemix
MB	– Mega Byte; Mega Bajt; 8 bitů
RPi	– Jednodeskový počítač Raspberry Pi
MS	– Společnost Microsoft
USB	– Universal Serial Bus
LED	– Ligth emitting diode
RAM	– Random access memory
ROM	– Read only memory
CPU	– Processor; Procesor
Wi-Fi	– Wireless Fidelity
CSI	– Camera serial interface
DSI	– Display serial interface
SMTP	– Simple Mail Transfer Protocol
BETA	– Beta verze produktu; Produkt, na kterém je již opravena většina chyb, nicméně je pořád nestabilní

Seznam obrázků

1	Vlevo: IQRF transceivery řady TR-7xD; Vpravo: SIM modul DK-EVAL-04A . . .	17
2	Vlevo: Sada DS-START-04; Vpravo: Konektor pro RPi KON-RASP-01	18
3	Blokové schéma transceiveru TR-72D	19
4	Architektura softwaru	20
5	Dopad vysílacích úrovní a provozních režimů na energetickou náročnost	21
6	Ukázka ovládání světla za pomoci DPA příkazu ¹	22
7	Ukázka síťové topologie	23
8	Propojení jednotlivých technologií v měřicí systém	24
9	Jednodeskový počítač Raspberry Pi, model 3B	25
10	Možný scénář struktury za použití MQTT	26
11	Rozdíl mezi úrovněmi QoS ²	28
12	Vlevo: Ukázky dashboardů; Vpravo: Aplikace varování	31
13	Ukázka programového kódu v Node-RED ³	32
14	Ukázka jednoduché vizualizace v Node-REDu ⁴	33
15	Ukázka struktury IoT systému v MS Azure	34
16	Ukázka zpracované vizualizace ve službě Power Bi ⁵	35
17	Možné propojení služby "Stream analytics"napříč celou platformou	36
18	Další dostupné služby v rámci řešení pro svět IoT	36
19	Servisní dashboard cloudové platformy IBM Bluemix	37
20	Vizualizační dashboard na platformě IoT Watson	38
21	Správce vizualizačních dashboardů	38
22	Ukázkové propojení světa IoT s IoT Watson společnosti IBM	39
23	Několik bloků funkcí pro propojení Node-REDu s platformou IoT Watson i dalšími službami	40
24	Cena analyzování dat na platformě IoT Watson	40
25	Struktura IQRF sítě	43
26	Koordinátor připojený na jednodeskový počítač Raspberry Pi	44
27	Uzly IQRF sítě s prototypy senzorických shiedlů	44
28	Připojení transceiveru koordinátora k Raspberry Pi	45
29	Zjednodušené schéma připojení čidel k uzlům	45
30	Diagram a senzor DS18B20 v pouzdře T092 ⁶	46
31	Bitová reprezentace dat z senzoru DS18B20 (ZN - znaménko)	47
32	Senzory - vlevo: DHT11; vpravo: DHT22	47
33	Bitová reprezentace odpovědi senzoru DHT11	48
34	Modul se senzorem BME280 a vnitřní diagram	48
35	Stavový a vývojový diagram programu v uzlu 1	50
36	Struktura služby mqtt-to-sql	52

37	Struktura tabulky a její obsah	54
38	Struktura MS s bloky datové pumpy a IBM Watson	57
39	Část programu datové pumpy v Node-RED řešící vyčítání a formátování	57
40	Nastavení bloku funkce Watson IoT	58
41	Úspěšné propojení s platformou Watson IoT	59
42	Karta s průběhy zaznamenaných teplot z IQRf sítě	60
43	Další druhy vizualizačních karet informující o naměřených hodnotách	60
44	Rozšíření: Historical Data service	61
45	Zpracování notifikací pro čidlo DS18B20	62
46	Podoba přijatých notifikačních emailů	62
47	Zachycení výpadku dat a zaslání varování pomocí emailu	63
48	Reakce systému na vytažení senzoru DHT11	63
49	Flow kód pro hledání zpracování minim a maxim v Node-REDu	65
50	Vizualizační karta maximálních hodnot teplot	65
51	Karty maxima, minima, ar. průměru a mediánu veličin čidla BME280	67
52	Vizualizační karta směrodatných odchylek	68
53	Vizualizační karta procentuálního zastoupení vadných řádků v rámci dne	70
54	Vizualizační karty prezentující intervaly a zastoupení hodnot v jejich rámci	72
55	WA: Porovnání naměřených teplot	74
56	WA: Nejčastější hodnota veličiny	75
57	WA: Datová asociace	76
58	WA: Vliv na veličinu, či proměnnou	76
59	WA: Četnost naměřených hodnot v rámci intervalů	77
60	WA: Část podmínkového stromu pro predikci hodnoty	77
61	WA: Počet výskytů hodnoty v rámci vstupních dat	78
62	WA: Dopad změn veličin	78
63	Tabulka transceiverů s porovnáním	84

Seznam tabulek

1	Strukturovaná zpráva	22
2	Porovnání možností vizualizačních systémů	41

Seznam výpisů zdrojového kódu

1	Ukázka vložení dat do SQL databáze	25
2	Ukázková JSON zpráva	28
3	Insert dat do MySQL databáze	53
4	JSON request na paměť RAM	54
5	JSON response na request 0x05 RAM	54
6	JSON request na hodnotu DS18B20	55
7	JSON response s hodnotou DS18B20	55
8	dataRow připravený k odeslání na Watson IoT	58
9	Kód na hledání maxima a minima v přijatých datech	64
10	Metoda pro výpočet aritmetického průměru	65
11	Metoda pro hledání mediánu	66
12	Metoda pro hledání nejvyšší četnosti	66
13	Metoda pro výpočet rozsahu hodnot	67
14	Metoda pro výpočet rozptylu hodnot	67
15	Metoda pro výpočet směrodatné odchylky hodnot	68
16	Metoda pro výpočet absolutní průměrné odchylky hodnot od průměru	68
17	Metoda pro výpočet standardizovaného skóre hodnot	69
18	Funkce pro kontrolu a přidělení dat	69
19	Funkce dělicí hodnoty dle podmínek do intervalů	70
20	Formátovací funkce koláčových grafů	72
21	Ukázka dotazu	74
22	Porovnání a srovnání dat	74
23	Nejčastnější výskyt naměřené hodnoty	75
24	Asociace naměřených hodnot	75
25	Vzájemné ovlivňování veličin a proměnných	76
26	Četnost v rámci intervalů	77
27	Vytvoření podmínkového stromu	77
28	Dotaz na množství výskytů hodnot	78
29	Dotaz na dopad a význam veličiny	78

1 Úvod

Způsobů nakládání a využití naměřených dat je v současné době více, než kdykoliv předtím. Mnohá rozhodnutí ohledně zdokonalení výrobního procesu, či kvality měřicího systému je založeno právě na samotných naměřených datech. Naměřená data se podrobují různým statistickým a analytickým metodám, jež podají žádaný výsledek, na základě kterého se uskuteční rozhodnutí. Zpracovaná data je nutné také vhodně prezentovat.

V dnešní praxi se setkáváme čím dál častěji s nasazováním tzv. vizualizačních systémů na výrobní procesy, ba i celé podniky. Tyto vizualizační systémy fungují zpravidla na bázi cloudových platforem a poskytují nespočet různých funkcionalit zahrnující i zmíněnou problematiku zpracování a analýzy dat.

Předmětem této diplomové práce tedy je seznámení se s těmito vizualizačními platformami a na jedné z nich poté vypracovat samotné jádro diplomové práce. To tkví ve zpracování vizualizačního systému s aplikací různých statistických, analytických metod na datech. Bude-li to možné, aplikována bude také datová predikce.

Využitelná data pro vizualizaci a analýzu bude potřeba získat. K tomuto účelu bude realizována také IQRF MESH síť jako nutný základ pro samotné jádro práce. Technologie IQRF má své kořeny u nás v České republice a je vyvinuta zejména pro účely ovládání a řízení, mimo jiné ji lze také využít právě pro měření a sběr dat.

Cílem praktické části této diplomové práce je zpracování vizualizační aplikace v podobě online dashboardu na vybrané cloudové platformě, kdy v pozadí budou data podrobena zpracování, analýze, či predikci, aby se tímto využil jejich maximální potenciál, pod kterým se v reálném světě může skrývat například zefektivnění sledovaného výrobního procesu, či včasné varování před hrozící havárií.

Diplomová práce je složená z kapitol, ve kterých je zahrnuta teorie, všeobecné informace o použitých technologiích, analýza vizualizačních systémů, realizace měřicího systému a samotného vizualizačního subsystému na vybrané cloudové platformě. V příloze jsou uvedeny veškeré návody a instalační skripty, které provázely vypracováním praktické části. Dále na CD s přílohami jsou uvedeny samotné zdrojové kódy, či dodatečné informace, které nebylo vhodné uvést přímo do vypracování.

Na seznámení s technologií IQRF dojde v kapitole první. Uvedeny zde jsou nejdůležitější informace o této technologii a představeny jsou jednotlivé dílčí záležitosti jako hardwarové a softwarové prostředky.

Kapitola druhá je věnována použitým technologiím v rámci vypracování této diplomové práce. Zmíněny zde jsou mimo jiné technologie jako MQTT, databázový systém, či Raspberry Pi, které je významnou částí měřicího systému.

Analýze dostupných vizualizačních systémů je věnována kapitola čtvrtá. Vybrány a analýze podrobeny zde jsou čtyři vizualizační systémy pro zpracování dat z IQRF MESH sítí včetně platforem, ke kterým jsou vázané.

V kapitole páté dojde na návrh a realizaci IQRF MESH sítě, která je důležitým základem pro vypracování následujících bodů zadání této diplomové práce. Podrobně zde je popsána zvolená architektura měřicího systému a jeho realizace. K této kapitole se také částečně váže obsah zadání bodu č. 4, jelikož zde dojde na zpracování služby, která zpracovává surová data z IQRF MESH sítě a následně je ukládá do databáze, jež poslouží jako datový zdroj pro vybraný vizualizační systém.

Dořešení otázky propojení databáze a vybraného vizualizačního systému je popsáno v kapitole šesté. Stěžejní částí této kapitoly je popsání návrhu a realizace subsystému pro zpracování, analýzu a vizualizaci dat z IQRF MESH sítě, kdy v kapitole sedmé dojde na zhodnocení dosažených výsledků a samotný závěr diplomové práce.

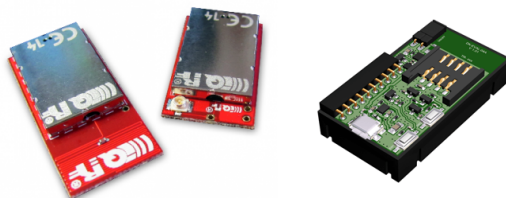
V poslední části zvané *Přílohy* jsou podány veškeré dodatečné informace, jež provázely celým průběhem vypracování této diplomové práce. Vysázeny zde jsou obsáhlejší zdrojové kódy, instalační skripty, obsáhlé tabulky, či velké obrázky.

2 Technologie IQRF

Tato technologie, jež má kořeny u nás v České Republice ve firmě MICRORISC s.r.o, je platforma pro přenos malých objemů dat za spotřebování minimálního množství elektrické energie. Konektivita dosahuje vzdáleností až několika kilometrů - nasadit ji lze prakticky kdekoliv. Momentálně již našla uplatnění v oblastech jako je automatizace, průmysl, telemetrie, či chytrých budov a měst. Platforma je navržena, aby mohla být používána s jakýmkoliv elektronickým zařízením. Společně s možností vzdáleného přístupu, monitoringu, sběru dat a zároveň velice snadné implementaci, dělá z IQRF více než vhodného kandidáta pro nasazení v širokém okruhu aplikací. Jednoduše řečeno můžete využít této technologie jak doma v rámci monitoringu teploty, tak na několika kilometrech např. železniční tratě pro potřeby měření vibrací. Řadí se také mezi významné technologie využitelné v rámci IoT, kdy dokáže velice efektivně využívat topologie MESH a data doručuje s vysokou spolehlivostí i díky vlastnímu konceptu řízení sítě.

2.1 Hardware

Celá topologie sítě si zakládá na speciálních malých transceiverech. Na tomto modulu se nachází vše pro uskutečnění bezdrátového přenosu - MCU s operačním systémem a perifériemi, EEPROM a VF část s anténou. Samozřejmě pak tedy je i vhodný konektor kompaktních rozměrů. MICRORISC se v tomto případě inspiroval SIM kartami a využil stejného konektoru také u svých modulů. Modifikací transceiverů je však více. Kromě již zmíněných rozměrů SIM karty má MICRORISC ve svém portfoliu také transceivery určené pro SMT montáž, či připojení skrze konektor RJ-45.



Obrázek 1: Vlevo: IQRF transceivery řady TR-7xD; Vpravo: SIM modul DK-EVAL-04A

Samotná hardwarová i softwarová stránka transceiverů prošla markantním vývojem od roku 2005. Níže můžete vidět výčet řad, které provází vývojem⁷ celé IQRF platformy.

- Řady TR-01A, TR-11A, TR-21A a TR-3xB

Tyto řady se již dnes řadí mezi archivní. Jedná se o první transceivery, které vznikly. Změny se odehrávaly spíše v oblasti softwarové - bylo odstraněno mnoho chyb, naopak vývoj OS a DPA šel kupředu a přinesl s sebou mnohá vylepšení. Poslední verzi OS, kterou tyto moduly podporovaly je v2.10 z roku 2009. Po stránce hardwarové pak moduly až na výjimky nabízely standardní dvě LED diody, či 5-6 vstupně-výstupních portů.

⁷Více na: <https://www.iqrf.org/technology/os-history>

- Řada TR-5xD

Jedná se o řadu transceiverů, kde můžeme pozorovat nejvíce podobností s řadou aktuální. I proto, že z této řady vzešla aktuálně nabízená řada TR-7xD. Nabízí výkonnější verzi MCU oproti svým předchůdcům, vylepšenou oblast pamětí a operačního systému. Obsahuje také precizní teplotní senzor, dále dvě LED diody a 6 vstupně-výstupních portů. Operační systém během aktivního období těchto transceiverů povýšil na verzi v3.80D z roku 2016 a jsou i nadále podporované.

- Řada TR-6xD

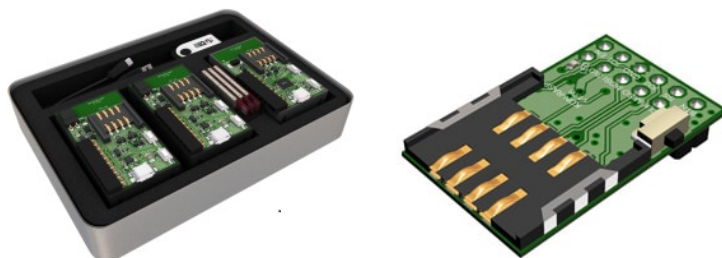
Tato řada se stala poněkud atypickou. Zahrnuje pouze jediný transceiver pro UART - Wireless M-Bus převod. Později byla tato řada nahrazena transceiverem TR-72D-WMB, či TR-76D-WMB aktuální řady.

- Řada TR-7xD

Aktuální produkční řada TR-7xD toho má s řadou TR-5xD hodně společného. Rozdílem je vyšší vysílací výkon, sériová EEPROM paměť a rozšířená hardwarová část o pájecí plošky či konektory.

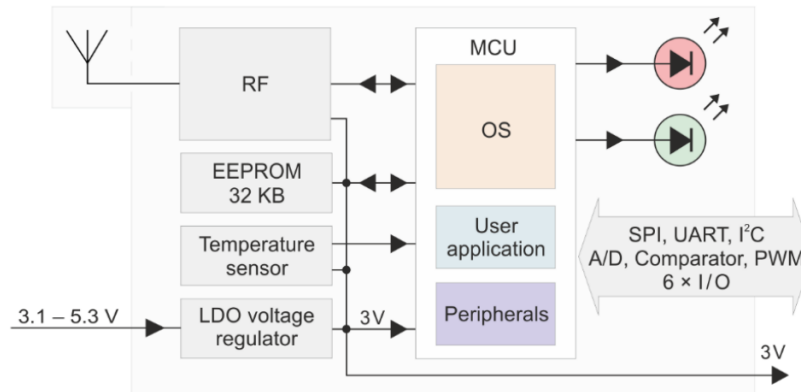
Mezi transceivery jednotlivých řad nejsou rozdíly až tak markantní. Většinou jde o rozdílnou montáž, počet pinů, podporované vysílací pásmo, napájecí napětí, či spotřebu (o ohledem na možné provozní režimy). Kompletní přehled řady TR-7xD můžeme vidět v tabulce na stránce 63.

IQRF MESH síť si bude zakládat na modulech transceiverů řady TR-72D, jež se společně s dalším vybavením jako jsou vývojové kity DK-EVAL, microUSB kabel a USB flash disk nachází v tzv. startovací vývojové sadě DS-START-04. Společně se sadou bude v rámci diplomové práce ještě využít konektor KON-RASP-01 pro připojení modulu transceiveru k prototypové platformě Raspberry Pi. Toto řešení umožní sběr, předzpracování a uložení naměřených dat přímo na zvolené hardwarové platformě, kdy dojde ke zjednodušení a zrychlení celého řetězce na rozdíl od klasického řešení IQRF sítě s branou (gateway). Data, která následně budou k dispozici jako datový zdroj pro vybraný vizualizační systém, již budou v před připravené formě a práce s nimi tak bude jednodušší.



Obrázek 2: Vlevo: Sada DS-START-04; Vpravo: Konektor pro RPi KON-RASP-01

V praktické části budou tvořit IQRF MESH síť celkem tři transceivery TR-72D. Pro potřeby měření a komunikace s vnějším světem disponují moduly transceiverů několika vstupně-výstupními porty, čili je k nim možno připojit libovolný senzor enviromentálních veličin. Dva transceivery budou vsazeny do vývojového kitu DK-EVAL-04A, který nabízí akumulátor, LED diody a tlačítka uživatelské, či systémové. Samozřejmostí kitu DK-EVAL-04 pak jsou piny pro vstupní napájení V_{cc} , napájení externích obvodů pomocí baterie V_{bat} a signálová, napájecí zem GND . Popisem vstupně-výstupních portů se budu dále zabírat v sekci měření enviromentálních veličin.



Obrázek 3: Blokové schéma transceiveru TR-72D

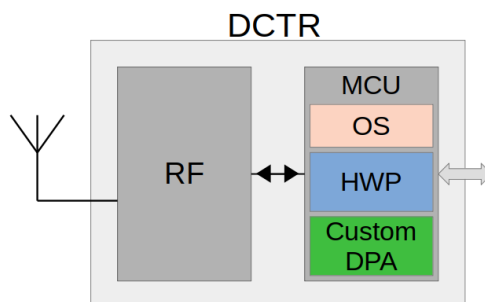
2.1.1 Komunikační pásmo

Pro komunikaci a přenos dat jsou využívána bezlicenční pásma. Frekvenční interval těchto pásem se liší dle jednotlivých zemí. U nás a v celé EU je to 868 MHz, v USA 916 MHz a v Číně 433 MHz. Jelikož jsou tato pásma bezlicenční, musí DPA protokol a komunikace odolat možnému rušení. Samozřejmostí je využití modulace signálu. V pásmu 868 MHz lze pro komunikaci využívat až 62 různých kanálů, jež jsou oddělené po 100 kHz.

2.2 Software

Softwarovým základem, tzv. spodní vrstvou, transceiveru je operační systém (OS). Ten je do modulu nahrán už od výroby, kdy do verze OS v3.00 nebylo možné tento operační systém povýšit. Nad operačním systémem se nachází hardwarový profil modulu, který specifikuje roli transceiveru v rámci IQRF sítě. Po nahrání příslušného profilu se z něj stane buďto uzel(node) nebo koordinátor(coordinator). V další vrstvě se nachází obsluha DPA protokolu, což je komunikační vrstva transceiveru. Stará se o komunikaci a předávání zpráv mezi jednotlivými moduly. Zde je nutné se zmínit o tom, že je možné do této vrstvy zasahovat uživatelským programem. Programátor je schopen pomocí volby Custom DPA handler přejít do stavu, kdy dokáže pomocí obsluhy stavů DPA handleru řídit program, doplňovat to vše o svůj kód a maximálně tak přizpůsobit chování transceiveru svým potřebám. Pokud ovšem tuto možnost nepotřebuje, využít

ji nemusí a DPA handler je řešen na základní úrovni hardwarového profilu. Vrstvou nejvyšší je samozřejmě vlastní uživatelský program.



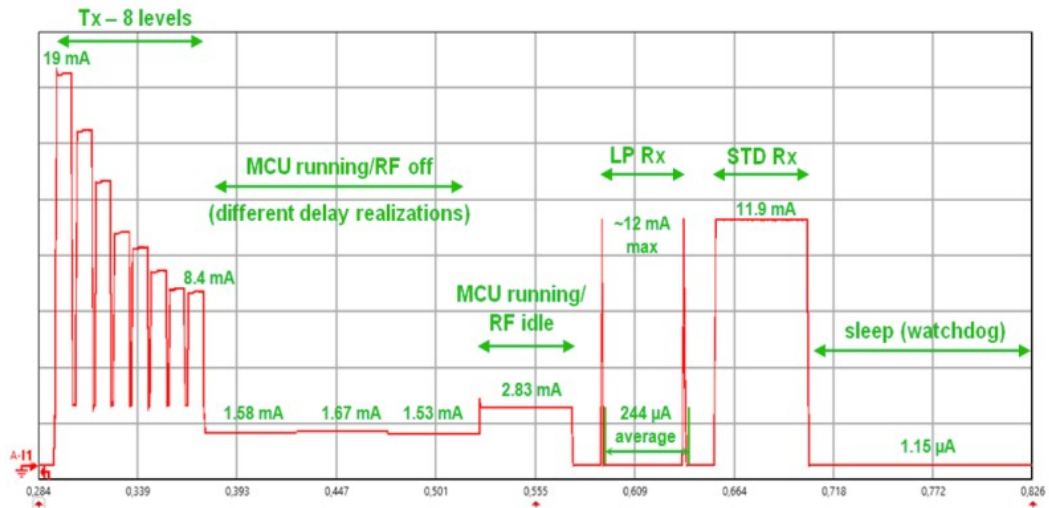
Obrázek 4: Architektura softwaru

Uživatelský program, společně s upravenou obsluhou DPA protokolu, je psán v jazyce C. Společnost MICRORISC poskytuje kromě základní podpory pro své produkty také širokou škálu různých ukázkových programů od jednoduché obsluhy tlačítka na vývojovém kitu DK-EVAL, až po rozsáhlé ukázkové programy pro obsluhu sítě s modifikovaným DPA řízením, včetně uplatnění režimů se sníženou spotřebou. K tomuto účelu také vydává vlastní IQRF IDE⁸, které programátorovi usnadňuje nakládání se soubory, základní práci se sítí, testování. Prostředí je intuitivní, přehledné a obsahuje širokou škálu různých nástrojů. Samotný kód se pak upravuje prostřednictvím externího editoru, kompiluje a nahrává se však skrze IDE, popřípadě je zde možnost v použití příkazové řádky.

DPA protokol a OS transceiverů nabízí celou řadu různých možností jak ještě více zefektivnit jejich použití. V rámci napájení jsou k dispozici tři režimy a to STD, LP a XLP⁹. V IQRF síti, jež bude zpracována v navržena a realizována v rámci diplomové práce bude použit standardní režim STD. Také již zmíněný DPA protokol nabízí mnohé v oblasti Custom DPA handleru. Ku příkladu bych zde vyzvedl možné použití intervalového dotazování zvaného *pooling*. Na následující obrázku č.5 můžeme vidět, jak se promítají jednotlivé provozní režimy na celkové spotřebě v čase. Samozřejmostí je pak fakt, že čím úspornější mód je zvolen, tím složitější bývá programování, jelikož se musí programátor potýkat s řešením obsluhy DPA protokolu.

⁸Dostupné z: <https://www.iqrf.org/technology/iqrf-ide/iqrf-ide-gui>

⁹STD - Standard; LP - Low Power, XLP - Extra Low Power



Obrázek 5: Dopad vysílacích úrovní a provozních režimů na energetickou náročnost

2.3 Zprávy DPA protokolu

DPA protokol pro komunikaci v IQRF MESH síti využívá strukturovaných zpráv. Každá zpráva obsahuje pět základních parametrů a sice:

- NADR - Adresa zařízení v síti

Jedná se o adresu zařízení v IQRF síti o podobě $2B^{10}$ čísla. Některé z adres jsou rezervované pro určité záležitosti. Ku příkladu uveďme adresu koordinátora, která je vždy 0x00, či broadcastu 0xFF. Dále se v oficiálním datasheetu dočteme, že ačkoliv je adresa velikostně 2B, významově vyšší bajt se nebere v úvahu.

- PNUM - Číslo periferie

Toto číslo určuje, se kterou periferií cílového zařízení hodláme komunikovat. Kromě základních striktně daných adresy např. pro EEPROM, RAM, LED, UARD, apod., zde máme periferie uživatelské a rezervované. Uživatelskou periferií volíme tehdy, není-li standardní (zabudovaná). Jedná se ku příkladu o mnou použitý konfigurovatelný teploměr Dallas 18B20, jež komunikuje po 1-wire sběrnici. PNUM je o velikosti 1B.

- PCMD - Číslo příkazu

Specifikuje, jaký příkaz se na dané cílové periferií hodláme uplatnit. Příkazy jsou jak uživatelsky přizpůsobitelné, tak i pevně dané. Zahrnují se zde příkazy pro čtení, zápis, restart, či konfiguraci, jež jsou vázané přímo na konkrétní cílové zařízení. Opět se jedná o 1B číslo.

¹⁰B - Bajt, 8 bitů = 1 Bajt

- HWPID - Číslo hardwarového profilu

Další 2 bajty strukturované zprávy specifikují unikátní funkcionalitu zařízení. Máme-li tedy na dvou uzlech totožné teplotní senzory, přiřadíme sensorům stejné HWPID a můžeme tohoto údaje následně využít pro vyčtení obou teplot jedním příkazem - požadavek se uplatní konkrétně pouze na nich, ostatní senzory jej budou ignorovat. Rozsah je od 0x0000 do 0xFFFF, kdy čísla 0x0001 až 0xYYYYE (Y - libovolné) jsou rezervována pro zařízení certifikovaná a 0xYYYYF pro uživatelská.

- PData - Předávaná data

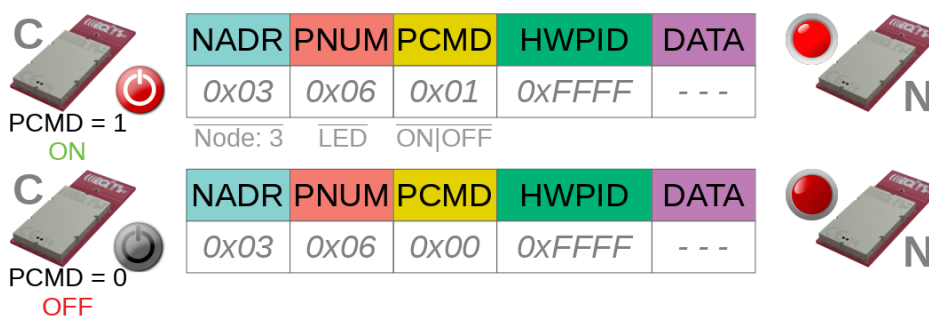
Tato část může narůst až na 56B. Jedná se o konkrétní data, např. vyčtenou teplotu, jež uzel vrací koordinátorovi jako odpověď na jeho dotaz.

Celá konkrétní zpráva třeba pro zápis do paměti RAM může vypadat takto:

Tabulka 1: Strukturovaná zpráva

NADR	PNUM	PCMD	HWPID	PData	Typ zprávy
0x0002	0x20	0x00	0xFFFF	NULL	Request
0x0002	0x20	0x00	0xFFFF	0xFF.67.02.04.02	Confirmation
0x0002	0x20	0x80	0x0F00	0x00.6A.02.20.32.33	Response

V oblasti standardních HWPID, či PNUM pak napoví následující obrázek, kdy se pomocí zaslání zprávy ovládá v modulu integrovaná LED dioda:

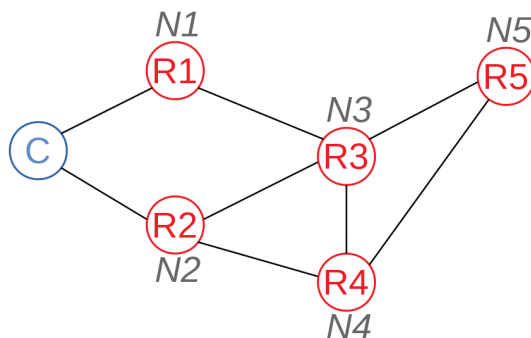


Obrázek 6: Ukázka ovládání světla za pomoci DPA příkazu¹¹

2.4 Síťová topologie a MESH

Jak již bylo zmíněno, topologie IQRf sítě si zakládá na uzlech a koordinátorovi. Koordinátor je v síti pouze jeden. Ten synchronním způsobem komunikuje s uzly, jež pod sebou spravuje v rámci své sítě. Uzlů je pak možno mít až 239. Tento počet uzlů ve většině aplikací vystačí.

¹¹Dostupné z: <https://www.root.cz/clanky/bezdratovy-prenos-s-iqrf-komunikacni-protokol-dpa/>



Obrázek 7: Ukázka síťové topologie

Na obrázku č.7 můžeme vidět ukázkou topologie IQRF sítě. Úplně vlevo vidíme jednoho koordinátora a dále několik uzlů. Při započetí práce s moduly je nutné všechny uzly tzv. naboundovat ke koordinátorovi, aby jednotlivým uzlům přiřadil adresy a tedy věděl, že patří k němu. Poté je možné uzly rozmístit kde je potřeba a spustit mapování sítě, které stanoví topologii. Můžeme si tedy povšimnout, že ne všechny uzly jsou v dosahu koordinátora, ale komunikovat s nimi lze přes některé uzly. Odtud také pramení spolehlivost této sítě, neboť je-li jeden z uzlů mimo provoz, koordinátor má stále možnost komunikovat se všemi ostatními jinou cestou. Podmínkou avšak je rozumné rozmístění jednotlivých uzlů - tedy aby každá cesta byla jistěna nejméně jednou další.

Pod pojmem MESH¹² pak rozumíme takovou síť, která si zakládá na peer-to-peer přístupu. Z jejího principu vyplývá hned několik výhod jako:

- Zastupitelnost

V případě zničení, či výpadku jednoho aktéra MESH sítě jej může v rámci komunikace nahradit člen jiný.

- Úspora pásma

Komunikační pásmo je využito pouze tehdy, kdy je to potřeba. K tomu slouží tzv. časová okna, tedy každý modul komunikuje v určený čas a ostatní naslouchají.

- Náklady a údržba

Komunikační MESH síť se jednoduše zakládá a také udržuje. Vyplývá to z použití samotného komunikačního protokolu, který se stará o komunikaci (routing) v síti.

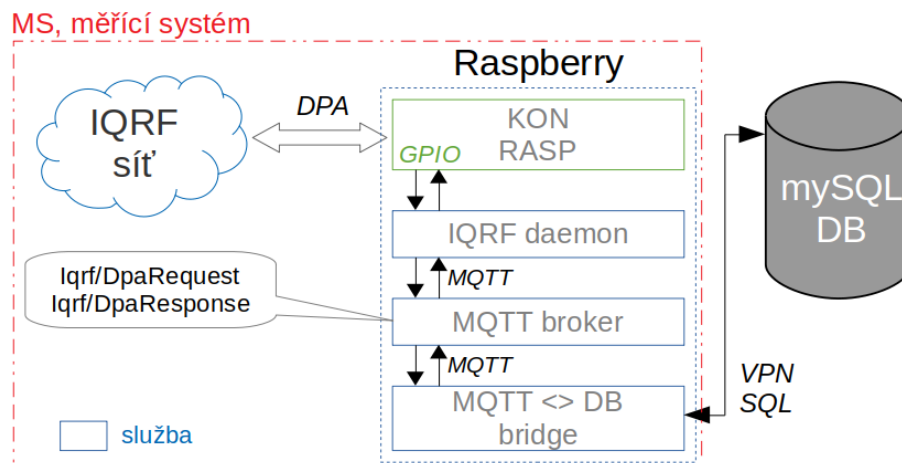
- Dosah sítě

Jelikož se každý síťový aktér může účastnit přenosu informací, markantně se tím zvětšuje dosah celé MESH sítě.

¹²Zdroj: <https://www.marigold.cz/item/mesh-site-p2p-architektura-v-bezdratovych-sitich>

3 Použité technologie v rámci DP

V této kapitole dojde na představení jednotlivých částí, které budou použity v praktické části diplomové práce. Jedná se jak o prostředky fyzické, tak softwarové. Bude zde slovo například o jednodeskovém počítači Raspberry Pi, SQL databázích, službách a komunikačních protokolech. Základní části měřicího systému lze vidět na následujícím diagramu.



Obrázek 8: Propojení jednotlivých technologií v měřicí systém

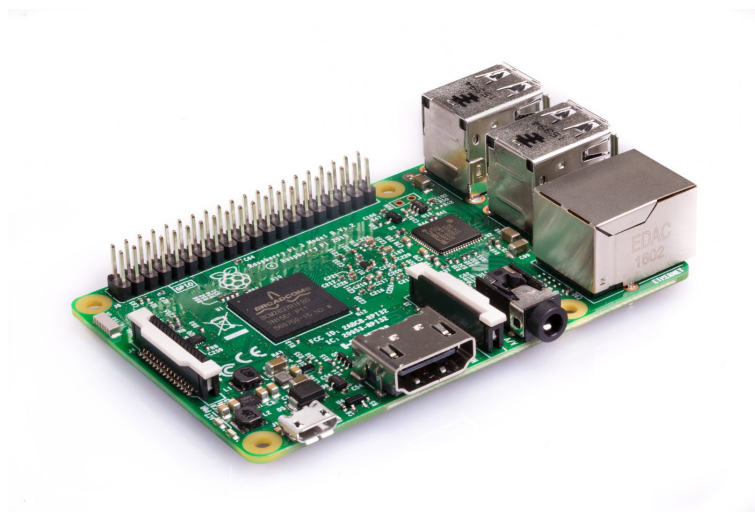
V kapitole 2.1 bylo zmíněno, že IQRF síť musí disponovat branou, skrze kterou bude zasílat naměřené informace k dalšímu zpracování. K tomuto účelu bylo zvoleno právě Raspberry, které v kombinaci s konektorem pro IQRF transceiver vytvoří bránu, pomocí níž bude možné získaná data předat dále. Toto ovšem není jediná záležitost. Raspberry bude tyto surová data předpracovávat, jelikož samotné moduly transceiveru jsou navrženy tak, aby za co nejnižší spotřeby data vyčetly a odeslaly. Ušetří se tím výpočetní výkon na jednotlivých uzlech (nodech) IQRF sítě, které by jinak tento krok obsáhly také. Následně se předzpracovaná data vloží do vzdálené databáze a budou tak připraveny pro vizualizační systém.

3.1 Raspberry Pi

Jedná se o jednodeskový počítač vyvinutý nadání Raspberry Foundation pro potřeby výuky. Momentálně je k dispozici několik variant s různým výkonem a nabízenou výbavou. Pro praktickou část mé diplomové práce mi byl zapůjčen model 3B, kdy je tento jednodeskový počítač osazen CPU SoC BCM2837¹³ od společnosti BroadCom. Procesoru sekunduje 1GB RAM a k dispozici máte kromě standardních rozhraní jako jsou USB, HDMI, CSI, DSI, Ethernet také důležitou 40p lištu se vstupně-výstupní GPIO sběrnici. Nechybí také slot pro microSD kartu s operačním systémem, či nově také u modelu 3B Bluetooth a Wi-fi. Nadace Raspberry Foundation pro své jednodeskové počítače také vydává vlastní linuxové distribuce, které jsou navíc

¹³SoC - System on Chip, vedle CPU čip také integruje veškeré bezprostřední řadiče, periferie, apod.

díky obrovské komunitě velice dobře odladěné a podporované širokou škálou softwarových a hardwarových výrobců. V rámci OS máme možnost sáhnout i po Windows 10 v IoT verzi, či po distribuci RiscOS pro real-time aplikace.



Obrázek 9: Jednodeskový počítač Raspberry Pi, model 3B

Nyní k samotnému významu Raspberry v rámci měřicího systému. Na GPIO sběrnici bude připojen konektor KON-RASP-01 pro transeiver modul koordinátora, kdy skrze sběrnici SPI bude komunikovat se službou `iqrf-daemon`, jež poběží na pozadí operačního systému. Ve své podstatě z Raspberry Pi udělá bránu (gateway). Službu včetně návodů pro instalaci a konfiguraci vydává¹⁴ společnost MICRORISC. Dále na pozadí OS Raspberry Pi poběží služba MQTT brokeru, jež bude obhospodařovat MQTT protokol, či zprávy a také služba pro příjem, zpracování a uložení záznamu do databáze zvaná `mqtt-sql-bridge`.

3.2 SQL databáze

V rámci měření, sběru a dalšího zpracování dat z IQRF sítě bude nutné data někde uchovávat. K tomuto účelu se bezesporu hodí určitá forma databáze. Pod pojmem databáze rozumíme určitý prostor, či nástroj k uchování a organizaci dat. V případě praktické části práce se budeme věnovat SQL relační databázi na vzdáleném MySQL serveru¹⁵. Pod pojmem SQL pak rozumíme strukturovaný dotazovací jazyk, jež v sobě implementuje problematiku výrokové matematiky a jež se používá pro manipulaci s daty v databázích. Následuje výpis s ukázkou vložení dat do tabulky za pomoci jazyka SQL:

```
INSERT INTO <Data01> VALUES (<ID>, <NAZEV>, <DATUM>, <TEPLOTA>, <VLHKOST>);
```

Výpis 1: Ukázka vložení dat do SQL databáze

¹⁴Dostupné z: <https://github.com/iqrfsdk/iqrf-daemon>

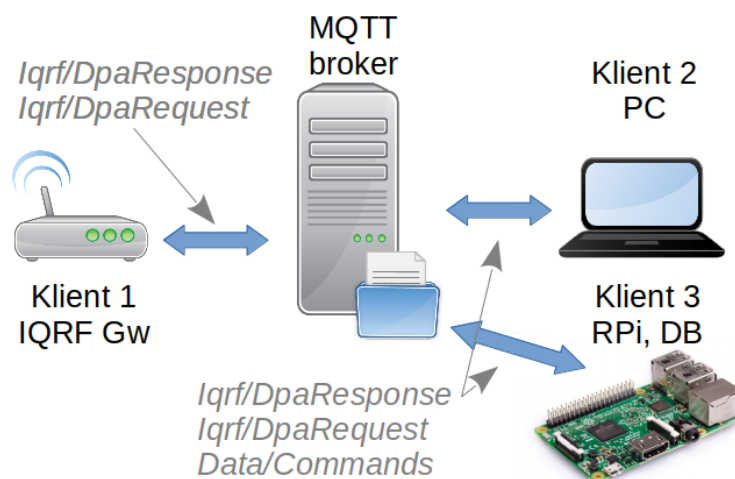
¹⁵Jedná se o formu zprostředkovatele přístupu k databázi

kde <Data01> je název tabulky, <ID> unikátní primární klíč k řádku, <NAZEV>, <DATUM>, <TEPLOTA> a <VLHKOST> pak jednotlivé hodnoty. Otázka primárního klíče je velice důležitá, neboť tento klíč musí být, jak již bylo psáno, unikátní. Musí zaručovat ojedinelost jednotlivých řádků v celém spektru databáze a jeho nesprávné generování může být zdrojem duplicitních, ba i přepsaných záznamů. Proto v praktické části své diplomové práce budu věnovat několik odstavců i této problematice, aby byla zaručena řádná kvalita záznamů v databázi. Jak si můžeme také ve výpisu kódu č.1 povšimnout, jazyk si opravdu zakládá na velice srozumitelné formě zápisu za pomoci klíčových slov.

Pro potřeby uchovávání dat mi byla poskytnuta školní projektová databáze Enviro běžící na MySQL serveru, kdy je nutné, aby Raspberry Pi bylo připojeno do akademické sítě. Jelikož je ovšem praktická část zpracovávána mimo tuto akademickou síť, je nutné využít připojení skrze VPN¹⁶. V databázi bude vytvořena tabulka BYR0034-IQRF, jejíž struktura bude přizpůsobena obsahu naměřených dat. Nezbytnou součástí tabulky budou sloupce pro primární klíč, datum a čas, či číslo uzlu, ze kterého naměřená data pochází.

3.3 MQTT a MQTT broker

Mezi jednotlivými službami, jež poběží na pozadí OS Raspberry Pi (totéž platí i pro komunikaci mezi zařízeními) je nutné provozovat předávání zpráv. V posledních letech se kvůli velkému zájmu o svět IoT pracovalo na zcela novém způsobu komunikace a předávání zpráv, či dat. V softwarových dílnách firmy IBM tak vznikl komunikační protokol MQTT. Jedná se o velice jednoduchý, datově nenáročný komunikační protokol, bez jehož znalosti se dnes ve světě IoT neobejdete. Koncept je velice jednoduchý - celá věda spočívá v jednom centralizovaném bodě, tzv. brokeru, který řídí předávání zpráv mezi klienty. Na následujícím obrázku je pak k vidění zjednodušená komunikační struktura MQTT.



Obrázek 10: Možný scénář struktury za použití MQTT

¹⁶Virtuální privátní síť; Více na: https://cs.wikipedia.org/wiki/Virtuální_privátní_síť

V MQTT probíhá přenos za pomoci TCP a používá se komunikační vzor čtenáře a písaře. Zprávy jsou tříděny do tzv. témat (topic) a v rámci aktérů zde máme dvě základní role: zařízení buď publikuje v daném tématu (publish), to znamená, že posílá data brokeru, který je ukládá a distribuuje dalším zařízením, nebo je přihlášeno k odběru tématu či témat (subscribe), a broker pak všechny zprávy s daným tématem posílá do zařízení. Jedno zařízení samozřejmě může najednou být v některých tématech publisher, v jiných subscriber.

- Publisher, neboli písař

Má-li zařízení, či služba tuto roli znamená to, že publikuje v daném tématu např. *Iqrf/Response*. Broker tuto publikaci přijme a poskytne ji čtenářům, kteří jsou přihlášení k odběru tohoto tématu. Ano, čtenářem se může stát i samotný písař.

- Subscriber, neboli čtenář

Je-li zařízení, či služba čtenář, tak ve chvíli, kdy si zažádá v přihlášených tématech o obsah (např. v *Iqrf/Response*), dostane jej. Data se z tématu pro dané zařízení smažou, ovšem v duplicitní podobě jsou připravena pro zařízení, které je také přihlášeno k odběru tohoto tématu.

Samozřejmě to znamená, že jedno zařízení může být zároveň jak čtenář, tak i písař do jiných témat. Je to pouze na vůli programátora, jak své zařízení, či službu nakonfiguruje a naprogramuje.

MQTT minimalizuje množství jak dat servisních, tak i dat nenesoucích nezbytně nutné informace. Chceme-li nějakou servisní úroveň komunikace zavést, máme možnost. Jedná se o parametr QoS, pomoci něhož jsme schopni ovlivnit, do jaké míry se bude samotný přenos ověřovat. Obecně platí, že broker odesílá zprávu na té úrovni, na které zpráva přišla, s možností úroveň snížit, umí-li klient pouze nižší QoS. Rozeznáváme pak tři úrovně QoS:

- QoS 0 (at most once)

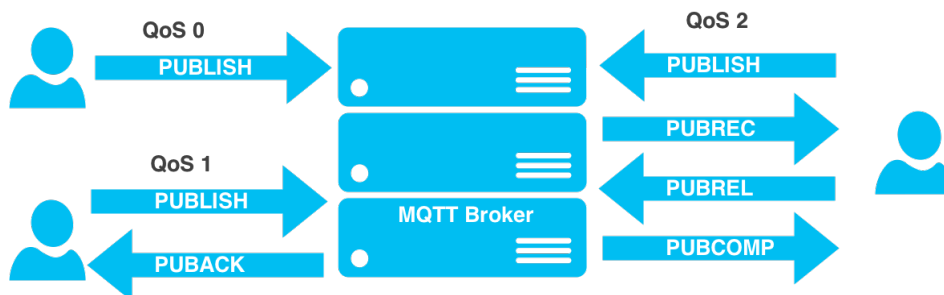
Publisher pošle zprávu PUBLISH brokeru a o víc se nestará. Stejným způsobem proběhne také zaslání zprávy subscriberům.

- QoS 1 (at least once)

Publisher zašle brokeru zprávu PUBLISH a broker ji poskytne dále subscriberům, ovšem může se stát, že je doručena vícekrát. Jakmile odběratelé potvrdí přijetí zprávou PUBACK, broker zprávu odstraní a pošle PUBACK publisherovi zpět.

- QoS 2 (exactly once)

Publisher zasílá brokeru zprávu PUBLISH. Ten ji, stejně jako v předchozím případě, přijme a posílá ji odběratelům, kdy publisherovi vrátí zprávu PUBREC (tedy potvrzení přijetí). Publisher odpoví zprávou PUBREL, broker zprávu smaže a potvrdí zprávou PUBCOMP.



Obrázek 11: Rozdíl mezi úrovněmi QoS ¹⁷

3.3.1 Podoba zpráv

Obsah samotných zpráv není nijak standardizovaný. MQTT je tzv. *payload agnostic* komunikace, což znamená, že obsahem jsou zkrátka libovolná binární data o libovolném formátu, která jsou přenesena. Nejčastěji se využívá formátů jako JSON¹⁸, BSON¹⁹, či textové zprávy. Tedy až na velikost jednotlivých zpráv, což je hranice v aktuální verzi stanovena na 256MB.

```

{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}
}

```

Výpis 2: Ukázková JSON zpráva

¹⁸Podrobnější informace: <https://www.zdrojak.cz/clanky/json-jednotny-format-pro-vymenu-dat/>

¹⁹Podrobnější informace: <https://en.wikipedia.org/wiki/BSON>

¹⁷Dostupné z: <https://zoetrope.io/tech-blog/brief-practical-introduction-mqtt-protocol-and-its-application-iot/>

4 Analýza vizualizačních systémů

Oblast vizualizačních systémů pro zobrazení, zpracování a analýzu dat je opravdu široká. V rámci analýzy vizualizačních systémů pro zpracování dat z IQRF sítí se nabízí na desítku řešení od různých dodavatelů softwaru. Takový typický vizualizační systém pro zpracování dat z měřicího systému má několik základních prvků:

- Vizualizace

Bez této vlastnosti si vizualizační systém nelze představit. Jedná se pracovní plochu, na které jsou naskládány většinou bloky s grafickými prvky, jež vizualizují poskytnutá data. Může se jednat o grafy průběhů v jakékoli podobě, o grafy statistického významu, grafy koláčové a sloupcové, tabulky, či jiné textové a indikační prvky. Samozřejmostí jsou pak prvky v podobě tlačítek, posuvníků, vstupních polí umožňující operátorovi různé zásahy.

- Datové uložení - cloud

Data, která chceme vizualizovat je nutné mít někde uložené. Mnohé vizualizační systémy poskytují jak vlastní úložné prostory pro vizualizovaná data, tak i konektory pro vnější databázové systémy. Vizualizační systém Grafana má ku příkladu vlastní cloud zvaný GrafanaCloud, v případě společnosti IBM je to IBM Cloud pro IoT Watson. Mezi konektory můžeme najít například připojení k vzdáleným databázím mySQL, InfluxDB, OpenTSDB a dalším.

- Varování a notifikátory

Významným prvkem vizualizačních systémů je také systém varování a notifikátorů. Operátor by měl mít možnost zajistit, aby byl informován, vyskytne-li se problém. K tomuto účelu slouží systém varování (Alerts), v němž je možné nastavit různé limitní hladiny veličin, kdy překročení povolených limitů bude zaslán pomocí notifikačního subsystému (Notifications) zpráva, či obslužný příkaz.

- Analýza nad daty

Sem spadá processing dat. Jedná se především o zpracování dat na úrovni statistických a analytických metod. Řadíme zde například jak základní matematické úkony v podobě maxima, minima, mediánu, průměru, tak i metody pokročilé datové analytiky, strojového učení, či datových věd. Samozřejmostí je pak oblast predikovatelnosti dat, kdy je možné odhalit problém ještě dříve, než nastane například na bázi znalosti trendu dat.

- Pluginy, rozšíření

Velká část funkcionality vizualizačního systému může být soustředěna do tzv. rozšíření. Může se jednat jak o vyhotovené funkce, tak i různé konektory, prvky pro vizualizační plochu a podobně. Je zde možnost zpracovat také vlastní pluginy a různá rozšíření. Ve

vizualizační platformě, společnosti Microsoft, Power Bi je to tzv. *R script*²⁰ který umožňuje zpracovat vlastní vizualizační prvky.

- Možnost lokálního, či cloudového řešení

Tato záležitost se týká místa nasazení vizualizačního systému. V určitých aplikacích postačí, aby byl vizualizační systém přímo na cílové hardwarové platformě spolu s databází a zpracováním dat z měřicího systému. Z pohledu bezpečnosti, rozšitelnosti, pracnosti a nebezpečí vypovězení hardwaru se však toto řešení může jevit problémově. Z tohoto důvodu stále vznikají komplexní cloudová řešení, která zastřešují všechny výše popsané body a řeší zmíněné problémy.

Mezi lokální i cloudová řadíme např. Nodered se svou grafickou nástavbou v podobě vizualizačního Dashboardu, kdy jej lze provozovat jak přímo lokálně třeba na Raspberry Pi, tak i ve spojení např. s platformou IoT Watsonu v cloudovém řešení BlueMix společnosti IBM. Jako druhý příklad zde můžeme uvést vizualizační systém Grafana.

4.1 Grafana

Vizualizační systém společnosti Grafana Labs je možné nasadit jak cloudově, tak i ve své lokální verzi. Podporovaných hardwarových platforem je hned několik a sice to jsou Windows, Linux, Mac OS či virtuální Docker. V cloudové verzi pak zákazník neřeší žádnou hardwarovou kompatibilitu a své řešení má dostupné online. Mezi nejvýraznější zákazníky, či partnery se řadí společnosti jako PayPal, e-bay, Intel, GameForge, či server Booking.com.

4.1.1 Možnosti vizualizace

Základní vlastnost jako každého jiného vizualizačního systému je samotná vizualizace. Grafana nabízí širokou paletu různých indikačních a ovládacích grafických prvků. Je-li potřeba, je zde možnost sáhnout po různých rozšiřujících komponentách, které umožní nasadit například digitální hodiny, seznamy prvků, apod. Pakliže se to pro konkrétní daný případ hodí, na webu²¹ Grafany je dostupných mnoho již hotových pracovních ploch (dashboardů).

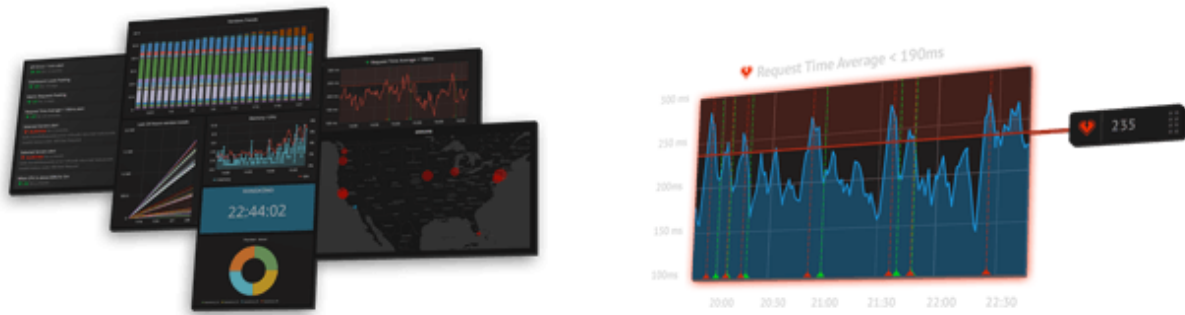
4.1.2 Možnosti analýzy

Samotná Grafana nabízí základní možnosti, jak zpracovat data. Jedná se převážně o jednoduché matematické operace jako jsou maximum, minimum, aritmetický průměr a podobně. K dispozici je také filtrování dat, které dle zadaných podmínek vyfiltruje určitá data z celku. Zajímavější však jsou nadstavby, které lze v rámci analýzy dat využít. Jednou z nich je nadstavba zvaná Graphite²², která umožňuje na data aplikovat různé funkce a pokročilé datové filtry.

²⁰Více na: <http://community.powerbi.com/t5/R-Script-Showcase/bd-p/RVisuals>

²¹Dostupné z: <https://grafana.com/dashboards>

²²Více zde: <https://grafana.com/plugins/graphite>



Obrázek 12: Vlevo: Ukázky dashboardů; Vpravo: Aplikace varování

Samozřejmostí pak je možnost využití varování a notifikací se zasláním zprávy např. na email, slack, či webhook. Určit možno je limitní hodnoty, jejich četnost a dále uchovávat historii varování. Obsahově lze měnit také samotné notifikace - lze je tak využít i k zaslání příkazu.

4.1.3 Licence

Jak již bylo psáno výše, Grafanu lze provozovat jak lokálně, tak i na cloudu Grafana Labs. V rámci lokálního nasazení se zde nabízí tři možnosti:

- Licence lokální bez podpory

Již nadpis napovídá, že tato volba je naprosto zdarma. Licence není omezující jak ze strany počtu nasazení, tak ze strany vizualizace v počtu dashboardů, varování a notifikací. Ovšem je absolutně bez jakékoliv podpory a prémiových pluginů.

- Licence lokální s podporou PROFESSIONAL

Kromě všech výše zmíněných kladných vlastností nabízí podporu s garancí pomoci při kritických problémech do dalšího pracovního dne. Dále umožňuje lokální nasazení jednoho prémiového pluginu, účast na webinaru (webovém semináři určeného pro výuku) a požádat o emailovou podporu. Cena začíná na \$7,200.

- Licence lokální s podporou ENTERPRISE (podnikové nasazení)

Oproti licenci profesionální nabízí: garanci pomoci v kritických případech do 2h, dále neomezené používání prémiových pluginů, privátní real-time spolupráci v rámci vlastního týmu, podporu na telefonu, možnost odškodnění, nasazení vlastního loga v rámci vizualizace a účast na spolupráci v rámci budoucích vydáních Grafany. Cena se odvíjí dle velikosti nasazení.

Ve verzi cloudové Vám pak společnost Grafana Labs nabízí celkem tři možnosti:

- Licence Zdarma

Zahrnuje možnost používání pouze jednoho uživatelského účtu s až 5 dashboardy. Komunikace je dále zabezpečena pomocí SSL šifrování.

- Licence pro malé týmy

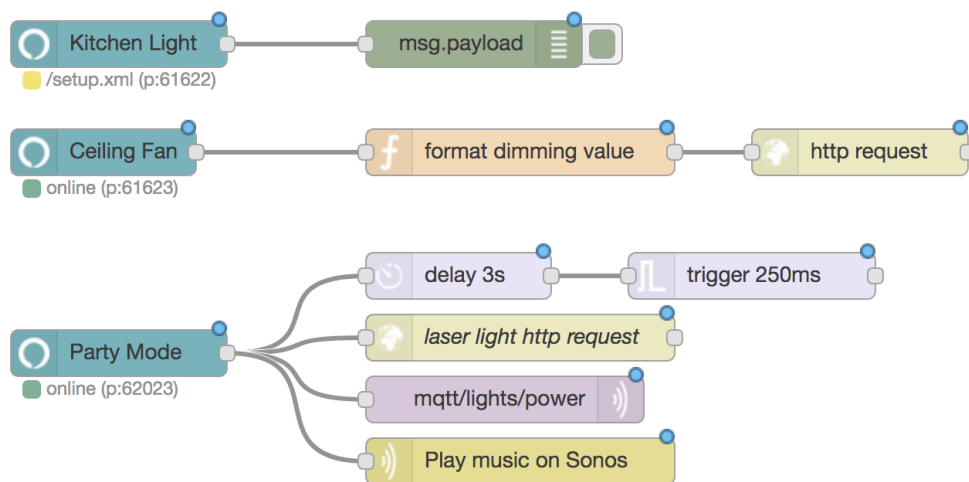
Počet uživatelských účtů vzrostl až na 10 (zdarma) s možností aktivace dalších za cenu \$4, dále není limitován jak počtem dashboardů, tak i varování. Samozřejmostí je opět šifrovaná komunikace navíc s možnou podporou a řešení problémů v rámci uživatelské komunity. Cena se pohybuje od \$39 za měsíc.

- Licence s podporou

V základě se jedná o stejnou licenci jak v předchozím případě zde navíc s možností vlastní domény a autentizačních procedur (LDAP, OAuth a další), či podpory ze strany společnosti Grafana Labs a použití prémiových pluginů. Cena se pohybuje od \$90 za měsíc.

4.2 Node-RED

Samotný Node-RED není vizualizační systém, ale vývojovým programovacím jazykem na principu data-flow. Vyvinula jej na základě jazyka Node.js společnost IBM a umožňuje propojit zařízení, různé aplikační vrstvy a online služby dohromady jako součást konceptu IoT. Platforma Node-REDu umožňuje programování přímo v prohlížeči za pomoci jednak bloků se striktně danou funkcionalitou tak i bloků pro kód v Java scriptu. Je-li Node-RED součástí cloudového řešení nějaké korporace, je zvykem, že ona korporace nabízí portfolio vlastních knihoven a funkcí, jež umožní propojení Node-REDu s jejími službami. Vytvořené blokové schéma programu se uchovává pomocí JSON formátu.



Obrázek 13: Ukázka programového kódu v Node-RED²³

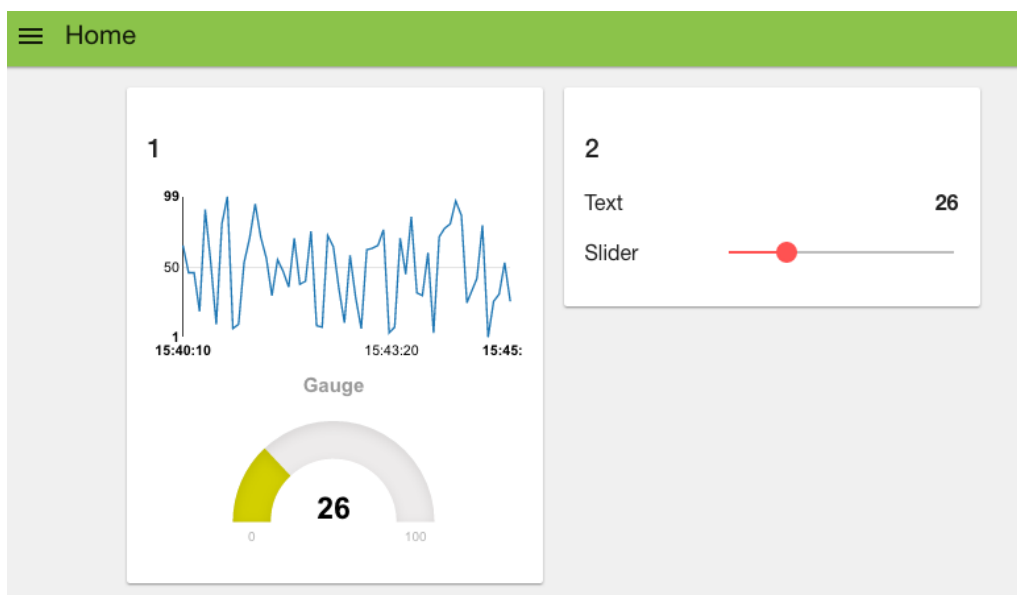
Platformu Node-REDu pak lze provozovat opět jak lokálně, tak i v rámci cloudové platformy. Stáhnout a provozovat jej lze na jakékoli lokální platformě, jelikož samotný JavaScript

²³Dostupné z: <https://flows.nodered.org/node/node-red-contrib-alexa-local>

je multiplatformní programovací jazyk. Po nainstalování se do programovacího prostředí přistupuje pomocí webového prohlížeče na místní adrese pod portem 1880 - samozřejmě je mít spuštěnou službu Node-REDu.

4.2.1 Možnosti vizualizace

Node-RED má tu vlastnost, že je možné jej rozšířit o knihovny funkcí. Jednou z nich je knihovna zvaná "*node-red-dashboard*", která programátorovi umožní přidat do programu ovládací a indikační prvky jako jsou grafy, tlačítka, posuvníky a podobně. K vloženým prvkům připojí drát s daty, které chce zobrazit a z Node-REDu máte rázem vizualizační systém. K čelnímu panelu se dostanete pomocí dopsání `/ui/` do adresní řádky prohlížeče.



Obrázek 14: Ukázka jednoduché vizualizace v Node-REDu²⁴

Grafické prvky pak lze co do vzhledu a jednotlivých popisů dále upravovat.

4.2.2 Možnosti analýzy

Využít můžeme jednak JavaScriptu v samostatném bloku, elementárních funkcí, knihoven či možností cloudové platformy, ke které je Node-RED provozován. Jedná se o cloudové platformy společností Microsoft (MS Azure), IBM (Watson IoT, Analytics) a dalších. Mezi knihovnými funkcemi v samotném Node-REDu lze najít například bloky pro jednoduché matematické a statistické funkce, jež jsou k dispozici v knihovně *node-red-contrib-datakitjs*²⁵. V tomto případě je to tedy spíše o tom, kde je samotná služba Node-RED provozována a má-li možnost programátor využít služeb cloudových platform velkých korporací jako Microsoft, či IBM.

²⁴Dostupné z: <http://developers.sensetecnic.com/article/a-node-red-dashboard-using-node-red-dashboard/>

²⁵Dostupné z: <https://flows.nodered.org/node/node-red-contrib-datakitjs>

4.2.3 Licence

Lokální verze Node-REDu je zcela zdarma, ovšem podléhá licenčnímu ujednání, jež je sepsáno nadací Apache Software²⁶. V případě cloudového řešení je potřeba se obrátit na licenční ceníky a ujednání konkrétních cloudových platform společností jako Microsoft, IBM, či Amazon. Většina z nich nabízí své služby v zkušebním (trial) módu, jež je dostupný po určitou dobu a spojení Node-REDu s konkrétní cloudovou platformou tak lze omezeně vyzkoušet.

4.3 MS Azure a Power Bi

Microsoft Azure je komplexní sada cloudových služeb, které vývojáři a odborníci na IT využívají k sestavování, nasazování a správě aplikací prostřednictvím naší globální sítě datových center. Díky integrovaným nástrojům, DevOps a Marketplace získáte podporu pro efektivní vytváření čehokoli od jednoduchých mobilních aplikací po aplikace v internetovém měřítku. Mimo jiné slibuje zvýšení produktivity vývojářů, bezpečnosti, konzistentnost, či vytváření aplikací nové generace (smart - chytrých). Samozřejmě pak je samostatná služba věnující se konkrétně světu IoT včetně vizualizační platformy zvaná *Sada Azure IoT*. Zahrnuje několik již předhotovených řešení, připojení obrovského množství zařízení přes širokou škálu komunikačních protokolů s možností využití libovolného OS, či analýzu a vizualizaci sesbíraných dat i ve velkých objemech.



Obrázek 15: Ukázka struktury IoT systému v MS Azure

4.3.1 Možnosti vizualizace

Základní možností jak vizualizovat data v prostředí platformy MS Azure je nástavba *PowerBi*, či *ThingLabs*. Jedná se o samostatné služby, které lze v rámci platformy propojit se službou jinou, například analytickou zvanou *Stream Analytics*. Výsledná zpracovaná vizualizace může nabýt následující podoby:

Službu PowerBi včetně datového připojení lze provozovat i na místním lokálním zařízení. K dispozici už musíte mít ovšem data zpracovaná a připravená k výsledné vizualizaci. Lokálně

²⁶Dostupné z: <https://www.apache.org/licenses/>



Obrázek 16: Ukázka zpracované vizualizace ve službě Power Bi²⁷

zpracovanou vizualizaci lze také nahrát zpět na cloudovou platformu. Pro administraci a správu dat pak máte k dispozici různé programy, které Vám umožní z lokálního zařízení přistupovat do cloudové databáze. Závěrem - Microsoft se obecně snaží používání vlastní platformy co nejvíce zjednodušit. Propojení jednotlivých služeb mezi sebou je tedy velice snadné a takřka ke všemu nabízí obrázkové průvodce s návody.

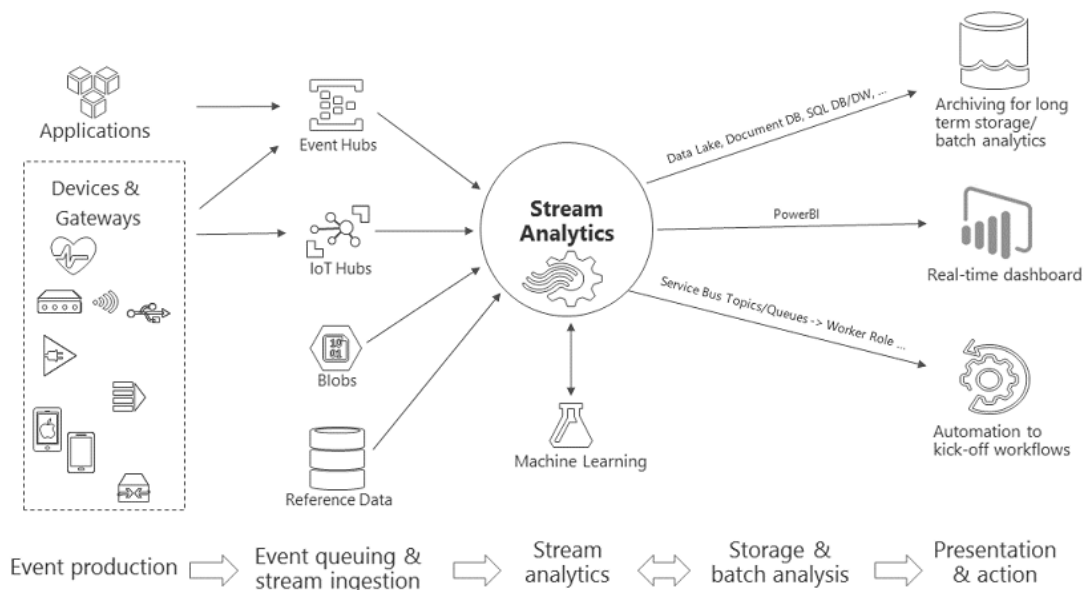
4.3.2 Možnosti analýzy

Platforma MS Azure nabízí opravdu obsáhlou škálu různých služeb pro zpracování a analýzu dat, včetně cloudových úložišť a podobně. Jako služba pro zpracování a analýzu dat ze světa IoT Microsoft přímo nabízí službu, která již byla zmíněna, a sice "*Stream Analytics*".

Nabízí řadu využití, mezi které přímo řadí například:

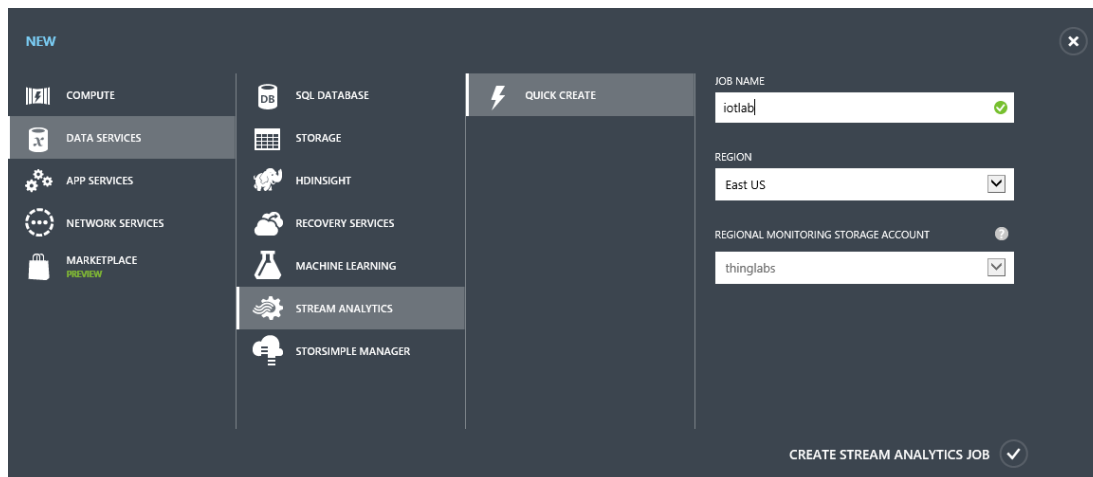
- Ekonomická, či obchodní analýza akcí, včetně zabudování výstrah.
- Detekování možných podvodů, dat a identifikaci ochran.
- Zabudovanou datovou analýzu dat ze senzorů a aktuátorů.
- Webovou analýzu *clickstream*.

²⁷Dostupné z: <https://www.pubnub.com/tutorials/microsoft-power-bi/streaming-business-data-to-dashboards/>



Obrázek 17: Možné propojení služby "Stream analytics" napříč celou platformou

Pokud by možnosti této služby bylo nedostačující, lze využít mnoha různých dalších služeb jako *HDInsight*, *Data Lake Analytics*, či *Machine Learning Studio*. Samozřejmostí je pak zahrnutí vlastních výpočetních a analytických nástrojů s použitím virtualizovaných pracovních stanic, či využití bloku pro zápis kódu v C#, či JavaScriptu.



Obrázek 18: Další dostupné služby v rámci řešení pro svět IoT

4.3.3 Licence

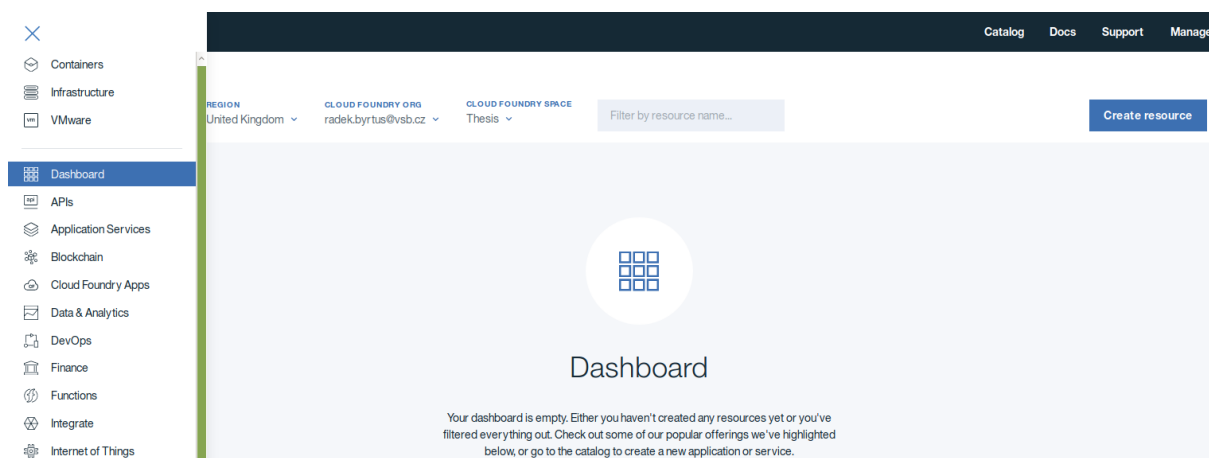
V případě této cloudové platformy je to velice složité. Samozřejmostí je nabídka verze *TRIAL* a licence zdarma, která je velice omezená. Mimo to však každá jak z uvedených, tak i služeb, které dále tato platforma nabízí má svůj licenční systém s několika úrovněmi na bázi předplatného.

Stojíte-li tedy o nasazení nějaké služby, musíte si platit pravidelné předplatné, aby Vám tato služba nebyla deaktivovaná.

Ještě krátce k licenci zkušební - v návaznosti na nedávné odezvy ze strany nespokojených uživatelů se Microsoft uvolil a poskytuje nyní 12 měsíční zkušební dobu, po kterou můžete do značné míry využívat skoro veškerých služeb v omezené míře. Některé pak úplně zdarma. Navíc pro potřeby nasazení placených máte k dipozici kredit \$170. Ovšem osobně musím varovat před situací, která nastane, jakmile kredit, či zkušební doba pomine. Pro přístup a využívání platformy po Vás totiž společnost Microsoft bude chtít vyplnit údaje k Vaší kreditní kartě, kde Vám bude po ukončení oné zkušební doby, či přečerpání kreditu bude účtovat veškerá předplatná a to i v případě, že tuto platformu a zahrnuté služby ve vašem projektu vůbec nevyužíváte.

4.4 IBM Bluemix a Watson IoT

Tato cloudová platforma společnosti IBM se v mnoha ohledech podobá konkurenčnímu řešení od společnosti Microsoft. V rámci svého cloudu nabízí bleskové vytváření, nasazení a administraci vytvořených cloudových aplikací s orientací na velké korporace. Stejně jako MS Azure nabízí k správě všech služeb přehledný dashboard s možností přidání dalších²⁸. Pro potřeby IoT se nabízí služba *Watson Internet of Things*, která v sobě zahrnuje vše potřebné pro vytvoření a nasazení aplikace pro svět IoT, zejména pak cloudové úložiště, služba pro zpracování a analýzu dat, či vizualizační platformu. Také stejně jako v případě MS Azure je zde možné využít služby Node-RED a propojit ji s další součástí IBM Cloudu, je-li to umožněno.



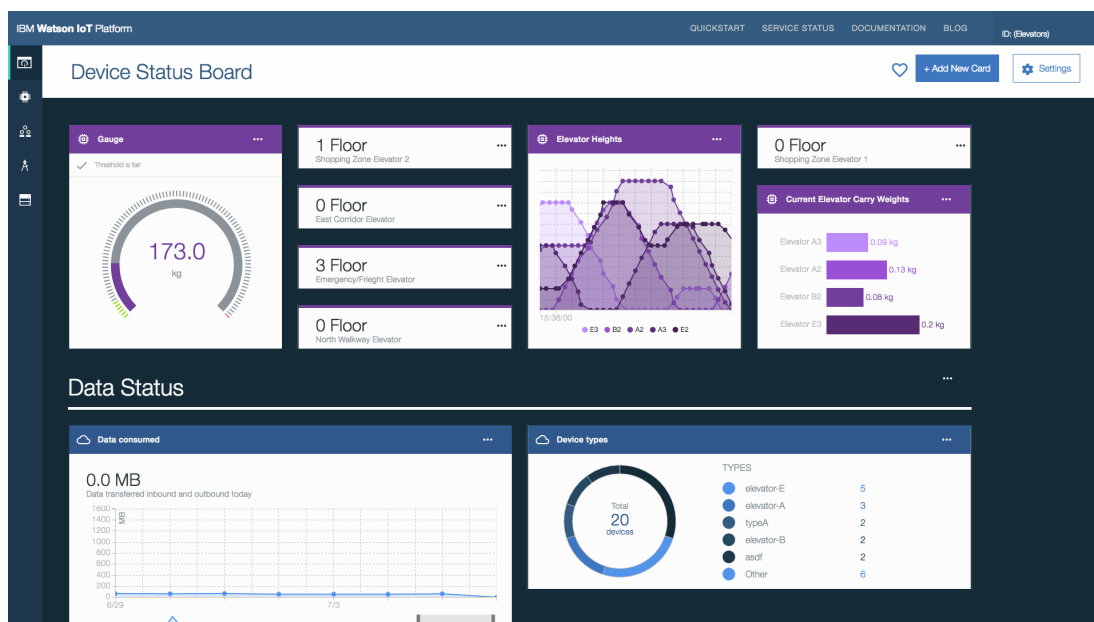
Obrázek 19: Servisní dashboard cloudové platformy IBM Bluemix

4.4.1 Možnosti vizualizace

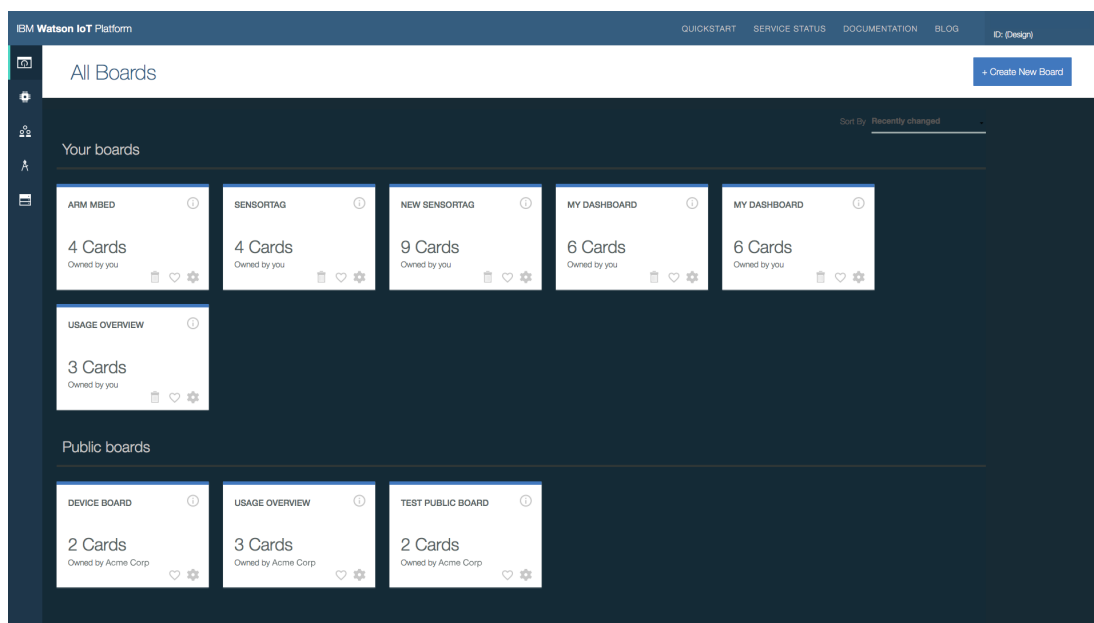
K vizualizaci jen krátce, jelikož se ani zde nejedná o nic nového oproti nabídce konkurence. IBM IoT Watson obsahuje vizualizační službu, kdy si uživatel může ve správci dashboardu

²⁸Kompletní katalog zde: <https://console-regional.ng.bluemix.net/catalog/>

vytvořit několikero různých vizualizačních obrazovek viz. níže:



Obrázek 20: Vizualizační dashboard na platformě IoT Watson



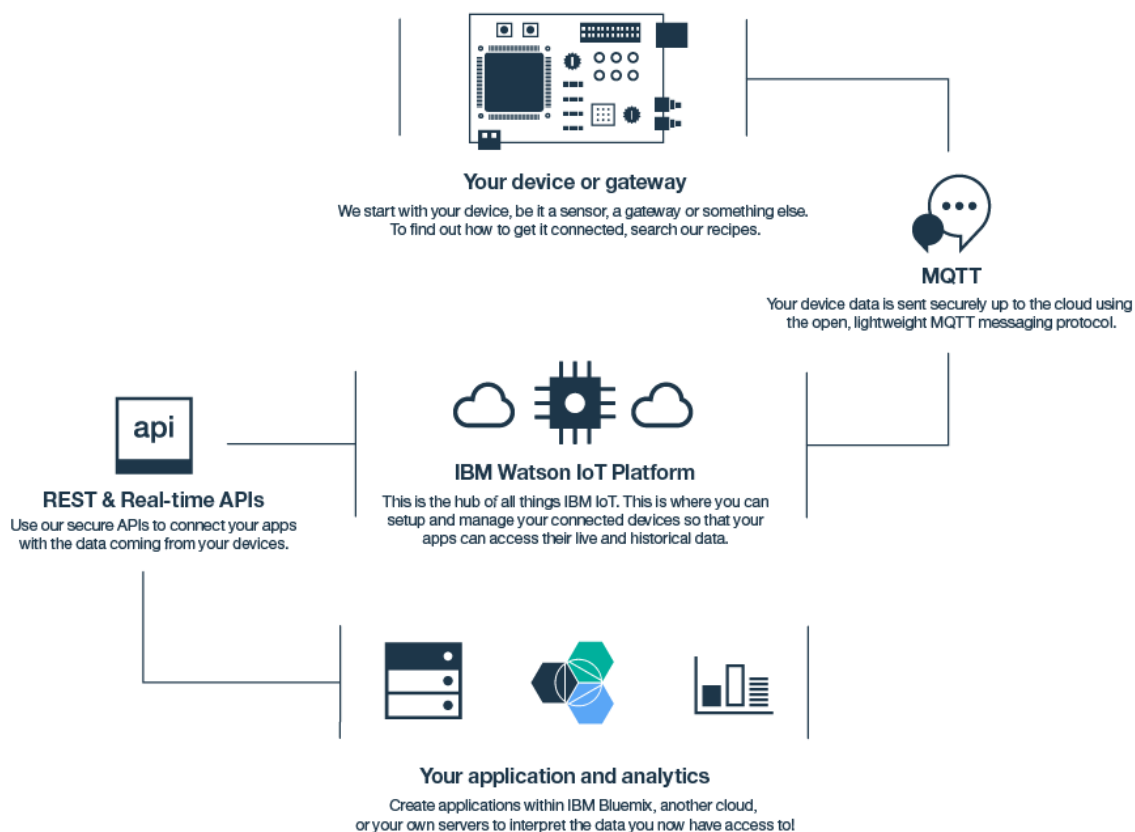
Obrázek 21: Správce vizualizačních dashboardů

Pak je zde ještě možnost zprovoznit dashboard v rámci Node-REDu, ovšem toto řešení považuji za bezúčelné. Co však dashboardu v rámci IoT Watson chybí je možnost použití ovládacích prvků. Je nutné podotknout, že služba Watson IoT se nyní nachází ve stádiu vývoje. Každý měsíc zde přibývají a naopak ubývají funkcionality. Ke dni 9. 3. 2018 například stále nefungují alarmy stavy a se musí řešit skrze Node-RED. Samozřejmostí je pak automatické ukládání

přijatých informací pro historické trendy v noSQL Cloudant databázi, jež se vytvoří zároveň s instancí služby Watson IoT.

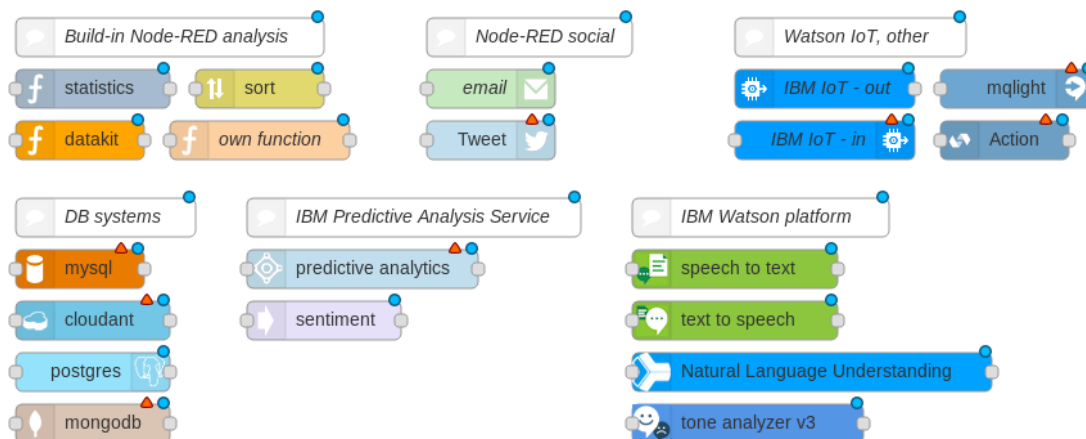
4.4.2 Možnosti analýzy

Ve své podstatě nabízí takřka stejné portfolio služeb jako konkurenční řešení od společnosti Microsoft. Jedná se zejména o služby jako *Data monitoring a analytics*, *Streaming and social analytics*, analýza řeči, tónů, také zvuků a podobně. Ve spojení s Watson IoT je pak možné využít služby *IBM Watson Data Platform* a *IBM Watson Analytics*.



Obrázek 22: Ukázkové propojení světa IoT s IoT Watson společnosti IBM

Využít lze také již zmíněný Node-RED, navíc s konektory pro jednotlivé služby z IBM Bluemix, jako pro predikce, analýzy, databázové systémy, či různé druhy marketingových analyzátorů. Na následujícím obrázku je vyobrazeno několik konektorů a funkcí dostupných v rámci Node-REDu na platformě IBM Bluemix.



Obrázek 23: Několik bloků funkcí pro propojení Node-REDu s platformou IoT Watson i dalšími službami

4.4.3 Licence

V rámci využívání služeb na platformě IBM Bluemix se nabízí samozřejmě licence zdarma, kdy jste omezeni například časovým intervalem nebo množstvím zpracovaných dat. Ve většině případů se pak jedná o množstevní licence na data, výpočetní výkon a podobně, kdy se účtuje po 1MB zpracovaných, či přenesených dat. Ceny se odvíjí například dle následujícího scénáře:

Tiers	Pricing
1 - 449,999	€0.002257 EUR/Megabyte Analyzed
450,000 - 6,999,999	€0.001579 EUR/Megabyte Analyzed
7,000,000+	€0.000316 EUR/Megabyte Analyzed

Obrázek 24: Cena analyzování dat na platformě IoT Watson

Většinu služeb také nemáte v licenci zdarma vůbec k dispozici a spoléhat se tak musíte na několik vybraných modulů.

4.5 Srovnání vizualizačních systémů

V rámci této podkapitoly dojde na zhodnocení analyzovaných vizualizačních systémů. Posuzování bude prováděno na základě několika kritérií jako například zda-li je vizualizační systém možné provozovat lokálně, či je vázán na cloudovou službu. Dále jaké nabízí možnosti vizualizace a analýzy dat, či zda-li je možné použít zásuvných modulů. Zohledněny budou také licenční podmínky, ceny a tarify jednotlivých vizualizačních systémů. K některým bodům v tabulce bude také přiděleno číselné hodnocení v rozsahu 1-5, které popisuje úroveň podpory daného okruhu, kdy hodnocení číslem 1 je nejlepší a číslem 5 nejhorší. Znamená to, že pokud bude daný okruh

hodnocen číslem např. 1, poskytuje v rámci něj základní i rozšířené možnosti zpracování. Je-li hodnocení 4, poskytuje v daném okruhu jen základní funkcionalitu.

Příklad posuzování okruhu alarmů a notifikací:

- Hodnocení 1, či velmi dobré řešení

Vizualizační systém poskytuje v rámci svého nasazení subsystém zabezpečující ošetření alarmních stavů a zaslání notifikací skrze email, či sociální síť. Dále s možností propojit tento subsystém se službou jiného vydavatele typu SmartGrid, SendGrid, apod. Předpokladem je také možnost zakomponování akčního zásahu, který případný alarmní stav řeší.

- Hodnocení 2, či dobré řešení

Subsystém řešící notifikace a varování při alarmních stavech nelze propojit s jinou službou typu SendGrid, či zakomponování akčního zásahu je poněkud složitější záležitostí.

- Hodnocení 3, či postačující řešení

Akční zásah nelze jednoduše zakomponovat do notifikačního systému a musí se řešit jiným způsobem. Vizualizační systém umí pouze základní threshold aktuálních hodnot se zasláním emailového upozornění.

- Hodnocení 4, či řešení na prahu použitelnosti

Akční zásah se realizuje pomocí jiné služby. Služba pro zaslání notifikací a varovných emailů o překročení mezí poskytuje pouze základní funkcionalitu.

- Hodnocení 5, či nefunkční, neumožňující

Služba je aktuálně vyvíjena, tedy nefunkční, či zcela chybí. Celý systém notifikací a varování je řešen pomocí jiných služeb.

Tabulka 2: Porovnání možností vizualizačních systémů

Okruh	Grafana	Node-RED	MS Azure	IBM Bluemix
Platforma	lokální, cloudová	lokální, cloudová	cloudová	cloudová
Alarmy, notifikace	Ano, 1	Ano, 3	Ano, 2	Ano, 5
Vizualizace	Ano, 1	Ano, 2	Ano, 2	Ano, 4
Analýza dat	Ano, 2	Ano, 4	Ano, 3	Ano, 4
Moduly	Ano, 2	Ano, 3	Ano, 1	Ano, 3
Úložiště	Ano, 1	Ano, 4	Ano, 1	Ano, 1
User-friendly UI	Ano, 1	Ano, 2	Ano, 1	Ano, 2
Česká lokalizace	Ano	Ano	Ano	Ne
Trial, verze zdarma	Ano, omezená	Ano, dle cloudu	Ano	Ano
Placená verze	od 7.200USD/rok	dle cloudu	dle tarifu	dle tarifu

Platforma, na které je vizualizační systém provozován je buďto lokální, či cloudová. Některé vizualizační systémy poskytují možnost nasazení jak v lokální verzi na vašem zařízení, tak v cloudové verzi. Jiné jsou nasazeny pouze jako služby na cloudové platformě.

Moduly pro zpracování alarmů na základě mezních hodnot a poskytování oznámení skrze email a ostatní služby byly veskrze na velmi dobré úrovni, až na službu Watson IoT cloudové platformy IBM Bluemix, kde je tato služba nyní ke dni 27. 3. stále mimo provoz a způsob na zprovoznění této služby v BETA verzi nefunguje.

V rámci okruhu vizualizačních dashboardů byly posuzovány možnosti, jaké jednotlivé prvky vizualizace poskytovaly. Je-li možnost přizpůsobovat obsah, vzhled, osy, apod. Opět všechny systémy až na službu Watson IoT mají rozmanité možnosti přizpůsobení prvků vizualizace. Vizualizační prvky služby Watson IoT však poskytují pouze základní zobrazovací schopnosti bez dalších možností, mnohem vhodnější je proto využít instance služby Node-RED, která má mnohem více možností v rámci zpracování vizualizace.

Okruh možností datové analýzy na datech byl posuzován v rámci konkrétní služby, která řešila příjem a vizualizaci dat. Vizualizační systém Grafana a Node-RED z toho vycházejí nejlépe, neboť v rámci nich lze nasadit základní statistické a datové zpracování. Služba Power BI platformy MS Azure dovoluje z datového zdroje vybírat maximum, minimum, či spočítat průměrnou hodnotu, kterou následně zobrazí do vybraného prvku vizualizace. Watson IoT však v této věci naprosto spoléhá na služby jako Node-RED, Apache-Spark apod, které ji zpracovaná data k vizualizaci poskytnou.

Modularita je také velice důležitý okruh. Umožňuje rozšířit vizualizační systém o další funkcionalitu umožňující např. pokročilé datové zpracování, možnosti analýzy, prvky vizualizace či propojení se službou jiné společnosti.

Všechny zmíněné vizualizační systémy a služby poskytují možnost propojení s mnoha typy databázových serverů. Co se týče uchovávání dat v rámci samotného vizualizačního systému, tak zde figuruje pouze základní ukládání dat v podobě dočasné cache, či globálních proměnných. Tyto data se však při restartování služby ztratí.

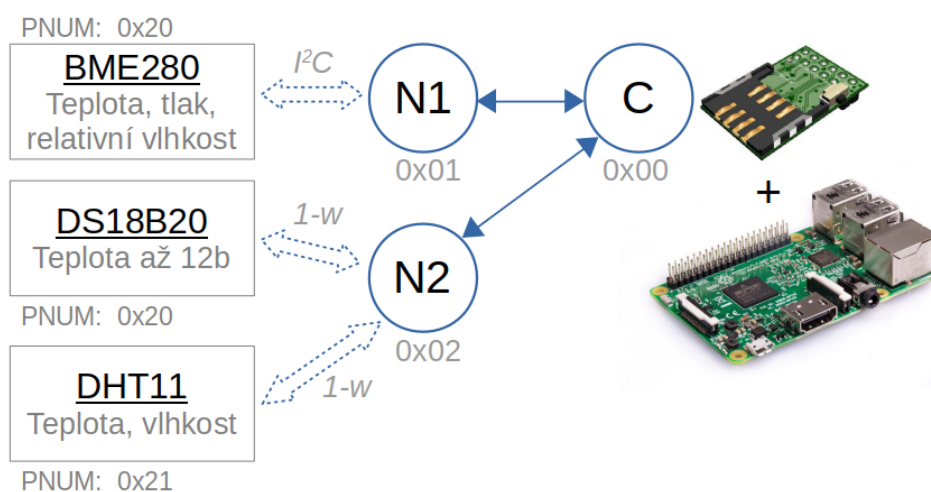
Moderní vizualizační systémy se také snaží o co nejvyšší možný komfort a intuitivnost svého prostředí. Tyto vlastnosti usnadňují a urychlují vývoj nasazení, který v rámci nich vzniká a značně také přispívají ke konkurenceschopnosti celého vizualizačního systému. Neméně důležitá je také srozumitelnost a lokalizace prostředí.

Posledním faktorem, který byl posuzován jsou ceny jednotlivých vizualizačních systémů a doba licencí zdarma.

Výstupem této kapitoly má být také výběr vizualizačního systému, na kterém bude posléze vypracována praktická část této diplomové práce. Pro tento účel bylo vybráno nasazení vizualizačního systému na platformě IBM Bluemix, kde vznikne pilotní projekt, v jehož rámci dojde na vypracování vizualizace a integraci subsystému pro zpracování a analýzu dat. Bylo tak učiněno v návaznosti na započatou spolupráci mezi VŠB-TU Ostrava a společností IBM, kdy nám byla poskytnuta trial verze na 6 měsíců.

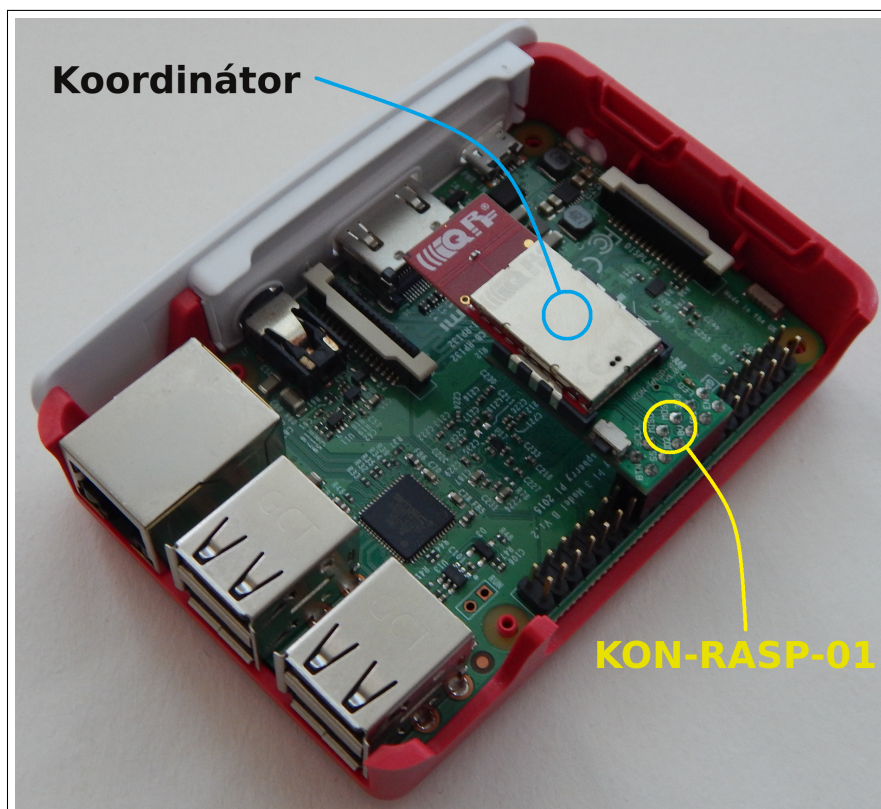
5 Realizace demonstrační úlohy

Tato část diplomové práce bude věnována návrhu a realizaci senzorické IQRF MESH sítě, jež bude společně s Raspberry Pi tvořit měřicí systém. Základem budou dva IQRF transeivery v roli uzlů (nodů) a koordinátor v podobě jednoho transeiveru, jež bude připojen na Raspberry Pi za pomoci již zmíněného konektoru KON-RASP-01. K uzlům budou připojeny senzory rozličných environmentálních veličin jako je atmosférický tlak, teplota, či relativní vlhkost. K tomuto účelu mi byly zapůjčeny senzory Dallas DS18B20 a Bosch BME280, kdy dále využito také senzoru DHT11. Senzorům bude dále věnována vlastní podkapitola. Struktura IQRF MESH sítě bude velice jednoduchá a vyobrazená je na následujícím obrázku:

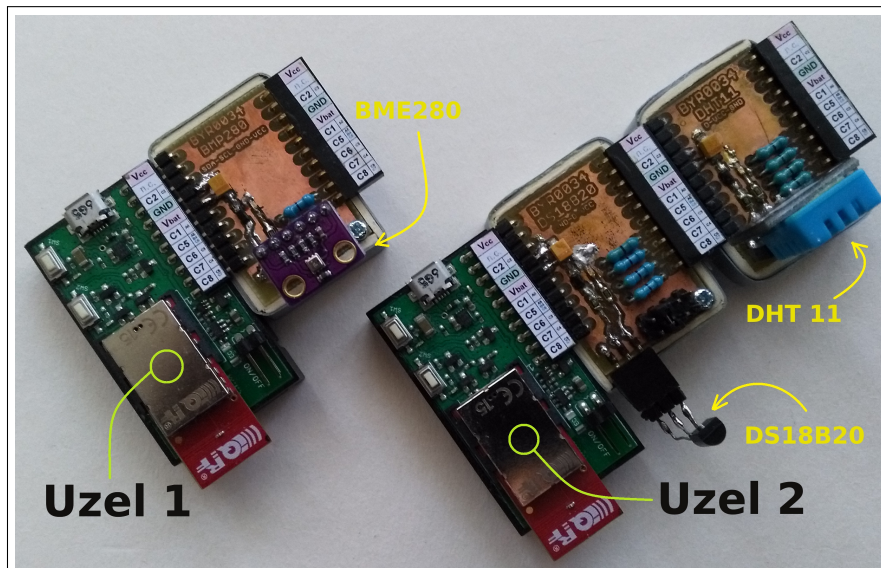


Obrázek 25: Struktura IQRF sítě

K IQRF síťovému koordinátorovi tedy budou připojené celkem dva uzly s adresami 0x01 a 0x02. Na každém z uzlů bude připojen jeden, či více senzorů environmentálních veličin pod uživatelskou periférií na adresách začínajících 0x20. Pro senzor DS18B20 již MICRORISC rezervoval vlastní HWPID 0x0007, jelikož stejný senzor je použit v rámci jejich starter kitu. HWPID senzoru BME280 je pak nspecifikováno, zůstane tedy pod obecnou hodnotou 0xFFFF. Diagram č.25 také napovídá, že DS18B20 komunikuje po 1wire sběrnici a BME280 pomocí sběrnice I²C. Nyní k samotnému fyzickému zapojení a popisům. Na následujících obrázcích jsou k vidění fotografie realizované IQRF sítě.



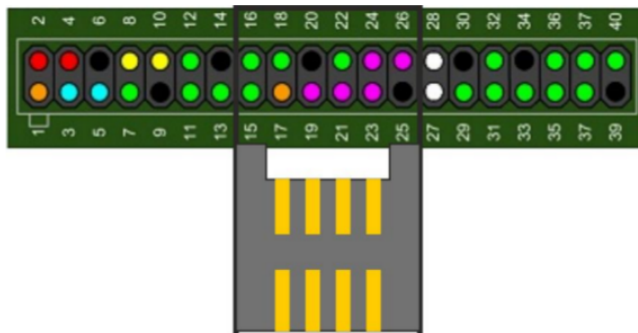
Obrázek 26: Koordinátor připojený na jednodeskový počítač Raspberry Pi



Obrázek 27: Uzly IQRF sítě s prototypy senzorických shiedlů

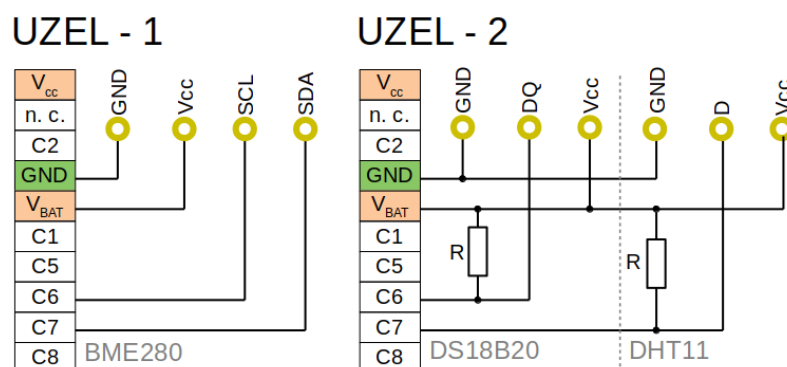
5.1 Hardwarové zapojení

Transceiver modul koordinátora je, jak již bylo několikrát zmíněno, připojen přes KON-RASP-01 na 40 pinové GPIO Raspberry Pi a to následujícím způsobem:



Obrázek 28: Připojení transceiveru koordinátora k Raspberry Pi

K dispozici totiž musí mít SPI sběrnici, po které komunikuje se službou iqrfd-daemon na straně OS Raspberry, napájení, zem, či klasické vstupně-výstupní porty. V případě uzlů IQRF sítě je situace následující:



Obrázek 29: Zjednodušené schéma připojení čidel k uzlům

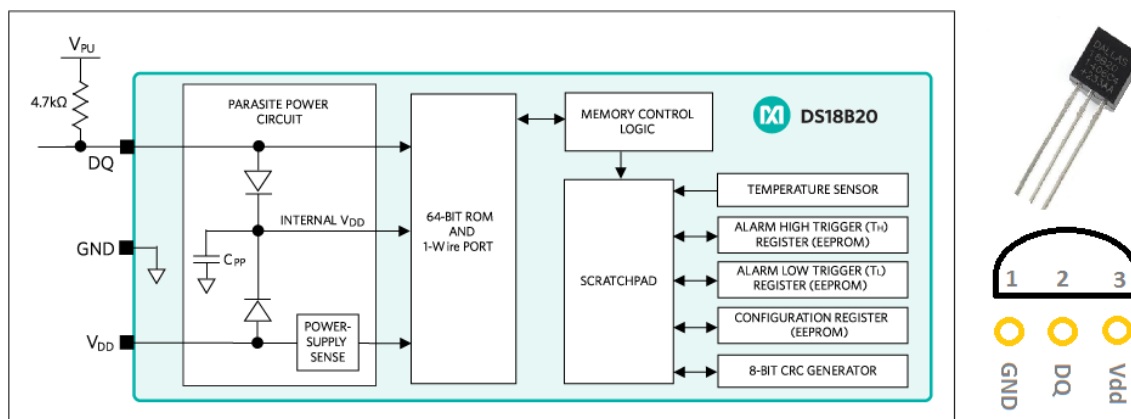
Senzor BME280 je připojen k uzlu 1, který pro komunikaci s ním využívá již zmíněné sběrnice I^2C . Potřeba je tedy kromě vodiče napájecího, či zemnicího také vodič pro hodinový signál a vodič datový. MCU osazené v IQRF modulech nabízí řešení I^2C sběrnice na pinech C6 (SCL) a C7 (SDI) a proto jsou právě tyto využity pro připojení senzoru BME280.

K uzlu 2 jsou pak připojeny senzory DS18B20 a DHT11. K senzoru DS18B20 jsou vedeny kromě nezbytného vodiče signálového a zemnicího také vodič napájecí, aby zbytečně nedocházelo k parasite-power jevu. Signálový vodič použitý pro 1-wire sběrnici je připojen k portu C6 (u MCU je to pak pin C3). Samozřejmostí je pak pull-up rezistor, aby bylo komunikaci vůbec možné uskutečnit. U senzoru DHT11 je situace takřka stejná jen s rozdílem v datovém vodiči, který je veden z portu C7 (MCU C4). V případě obou senzorů můžeme datový vodič připojit

ke kterémukoliv digitálnímu I/O, jelikož je 1-w sběrnice implementována pouze na softwarové úrovni.

5.1.1 Senzor Dallas DS18B20

Toto digitální teplotní čidlo společnosti Dallas Semiconductor zprostředkovává měření teploty v 9 až 12 bitové přesnosti. Poskytuje také triggerovací signál v případě, že programátor nastaví teplotní alarmy. Jak již bylo psáno, komunikuje skrze 1-wire sběrnici, která byla vyspecifikována toutéž společností. Pro připojení se tak používají pouhé dva vodiče - signálový a zemnicí, kdy samotné napájení je tento snímač schopný získat z vodiče signálového - parasite power. K uzlu bude připojen pomocí tří vodičů a to napájecím V_{bat} , zemnicím GND a datovým DQ .



Obrázek 30: Diagram a senzor DS18B20 v pouzdře T092 ²⁹

Každý vyrobený kus tohoto senzoru má unikátní 64-bitový kód uložený v ROM paměti. Řeší se tím možnost zapojit více stejných senzorů na jednu 1-wire sběrnici. Aplikaci pak najde téměř všude, kde je potřeba sledovat aktuální teplotu okolí.

5.1.1.1 Klíčové vlastnosti

- Měření teploty v rozsahu od -55°C do $+125^{\circ}\text{C}$.
- Přesnost $\pm 0.5^{\circ}\text{C}$ v rozsahu -10°C do $+85^{\circ}\text{C}$.
- Potřebné pouze dva vodiče, piny pro připojení k MCU.
- Konverze teploty s 12-bit přesností trvá maximálně 750ms.

Po obdržení zprávy z uzlu 2 ve vytvořené službě bude nutné surová data podrobit zpracování. Celkem to budou 2B s následující datovou reprezentací dle tabulky:

³⁹Odkaz: <https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html>

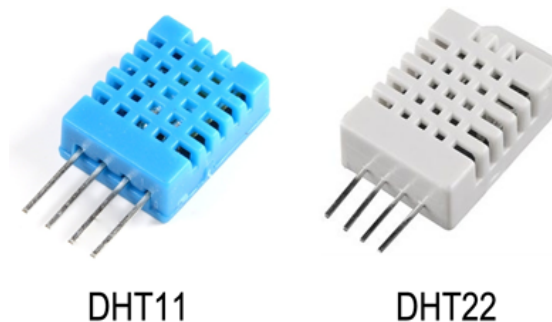
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LSB	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MSB	ZN	ZN	ZN	ZN	ZN	2^6	2^5	2^4

Obrázek 31: Bitová reprezentace dat z senzoru DS18B20 (ZN - znaménko)

5.1.2 Senzor DHT11

Jedná se o základní digitální senzor společnosti Adafruit Industries pro měření teploty okolí a relativní vlhkosti vzduchu. K nízké ceně se váže fakt, že jsou tyto senzory pomalé a taky nepřesné, ovšem i tak mohou dobře sloužit například kutilům. Skládá se ze dvou částí a sice z kapacitního snímače vlhkosti a termistoru. Jelikož se jedná o senzor s digitálním rozhraním, je zde přítomen také jednoduchý čip, jež obstarává převod analogové hodnoty na digitální a také samotnou komunikaci například s MCU. K uzlu bude připojen pomocí tří vodičů a to napájecím V_{bat} , zemnicím GND a datovým D .

Senzor DHT11 lze velice snadno (takřka kus za kus) zaměnit také za DHT22. Jedná se o přesnější senzor měřící v širších rozsazích. Pinout je u obou senzorů stejný. Datová reprezentace je však v případě senzoru DHT22 odlišná. Služba pro zpracování dat z IQRF sítě však bude připravena na oba typy senzoru.



Obrázek 32: Senzory - vlevo: DHT11; vpravo: DHT22

5.1.2.1 Klíčové vlastnosti

- Měření teploty okolí v rozsahu od 0°C do $+50^{\circ}\text{C}$ při nepřesnosti až $\pm 2^{\circ}\text{C}$.
- Měření relativní vlhkosti vzduchu v rozsahu 20% až 80% při maximální nepřesnosti až $\pm 5\%$.
- Vyčtení naměřené hodnoty nelze provádět častěji než 1x za vteřinu.

Komunikace je zde opět řešena 1-w sběrnicí. Master komunikaci zahajuje startovací sekvencí, na kterou mu senzor DHT11 odpovídá a následně zasílá 40b, které reprezentují následující data:

Nutno podotknout, že nastavení vyššího rozlišení (až 20 bitů) zde nepřinese kýžený efekt v podobě vyšší přesnosti naměřené veličiny, jelikož samotný senzor není schopen až takto titěrný rozdíl ať už v teplotě, či tlaku zaznamenat. Převzorkování ale můžeme snížit dopad šumu při měření. Dále je nutné senzor vhodně dle situace nastavit a brát v úvahu kalibrační konstanty, jež budou k dispozici v RAM (PNUM: 0x05) paměti uzlu. Bez zahrnutí těchto kalibračních konstant by naměřené hodnoty zcela vůbec neodpovídaly realitě. Výslednou hodnotu tedy získáme až po přepočtení hodnoty naměřené.

Datovou reprezentaci v rámci registrů senzoru BME280 lze dohledat v souboru BME280.pdf, jež je uveden v přílohách na CD. Pro její rozsáhlost ji zde uvádět nebudu.

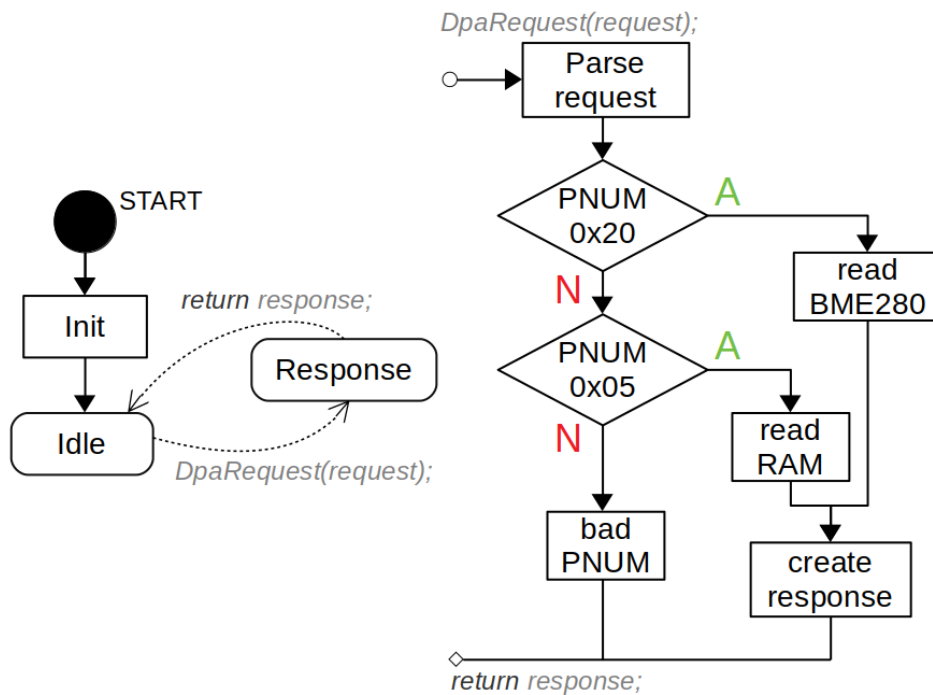
5.2 Programy uzlů

Tato podkapitola je věnována programovému obsahu IQRF uzlů. Programový kód je psán v jazyce C a v rámci jeho implementace je nutné používat *Custom DPA handler* v konfiguraci jednotlivých uzlů. Z toho vyplývá, že se programátor musí zabývat mimo jiné také obsluhou DPA protokolu v rámci svého programového řešení. Naštěstí jsou společností MICRORISC s.r.o poskytovány dobře zpracované předlohy a tak samotná implementace DPA protokolu je snadnou záležitostí. Programy jednotlivých uzlů se dále věnují implementaci potřebné sběrnice a práci se senzorem, do které spadá jeho konfigurace a vyčtení hodnot.

5.2.1 Uzel 1 - BME280

Program nahraný na uzlu 1 v sobě implementuje obsluhu I^2C sběrnice, aby bylo možné se senzorem BME280 komunikovat. Po připojení napájení k modulu se automaticky zahájí činnost *Init*, ve které dojde k inicializaci sběrnice a odeslání konfigurace do senzoru. Proběhne-li odeslání korektně, dojde k prvotnímu vyčtení přepočtených koeficientů, jež se uloží do RAM paměti (adresa 0x05). Uzel poté zůstává v nečinnosti až do doby, než je na něj skrze modul koordinátora zaslána zpráva, jež vyvolá událost. Zpracování této události v uzlu 1 popisuje stav *Response* v následujícím diagramu.

Zpracování události tedy probíhá tak, že se přijatý požadavek od koordinátora rozkouskuje a prověří se hodnota PNUM, která určuje s jakou periferií se bude pracovat. Je-li tato periferie přítomna, přistoupí se ke kontrole parametru PCMD. Ta je v případě programu i požadavku vždy 0, protože v každém z případů hodláme data vyčíst. Na základě parametru PNUM tedy pracujeme v případě uzlu 1 buďto s pamětí RAM (0x05), či samotným senzorem (0x20). Když se jedná o hodnotu 0x20, programový kód vyčte ze senzoru aktuální naměřené hodnoty environmentálních veličin, poskládá zprávu a pošle ji. V případě čtení z paměti RAM (0x05) dojde k vyčtení počtu zadaných bajtů a opět se složí zpráva a odešle se jako odpověď. Samozřejmostí je, že pokud dorazí požadavek s neplatnými parametry, dojde o odeslání odpovědi s kódem chyby.



Obrázek 35: Stavový a vývojový diagram programu v uzlu 1

5.2.2 Uzel 2 - DS18B20, DHT11

Program uzlu dva v sobě implementuje dvě 1-w sběrnice. První slouží pro obsluhu senzoru Dallas DS18B20 a druhá pro obsluhu senzoru DHT11. Po připojení IQRF modulu k napájení se opět zahájí inicializační činnost, kdy se inicializují obě sběrnice a v případě senzoru DS18B20 se na sběrnici zašlou konfigurační sekvence, jež nastaví přesnost snímání teploty. Ta může být buďto 9 nebo 12 bitová.

Sběrnice druhá pak slouží k prostému nastartování komunikace s DHT11 startovací sekvencí a vyčtení 40b, ve kterých jsou k dispozici hodnoty teploty a relativní vlhkosti vzduchu. Nutným požadavkem je kontrola přijatých dat, jelikož se v nich často nachází chyba. Pokud data neprojdou testovacím součtem, je nutné vyčítat znovu.

Senzor DS18B20 je registrován na adrese PNUM 0x20, DHT11 pak na 0x21.

5.3 Realizace měřicího systému

Pojem měřicí systém kompletně zastřešuje demonstrační úlohu, tedy IQRF MESH síť spolu se zpracováním na Raspberry Pi a uložení hodnot do SQL databáze na mySQL serveru. Struktura tohoto měřicího systému je patrná z obrázku č. 8. V této kapitole dojde na jeho popis.

Z jednodeskového počítače Raspberry Pi je nutné udělat síťového koordinátora. To lze uskutečnit dvěma způsoby:

1. Připojením programátoru s modulem koordinátora pomocí USB.
2. Použitím konektoru KON-RASP-01 a GPIO sběrnice Raspberry Pi.

Pro tento případ byl, jak již bylo několikrát avizováno, vybrán druhý způsob, jelikož je toto řešení praktičtější. Modul koordinátora, jež je za pomoci konektoru KON-RASP-01 připojen přes rozhraní SPI k jednodeskovému počítači Raspberry Pi, slouží jako komunikační brána pro spojení s IQRF sítí. Nahrán je na něm pouze základ v podobě hardwarového profilu pro koordinátora. Moduly uzlů musí být samozřejmě řádně přihlášeny (nabudnované) k modulu koordinátora. Komunikaci s ním obhospodařuje služba `iqrf-daemon` spuštěná na pozadí OS Raspberry Pi, jejíž vydavatelem je samotná společnost MICRORISC s.r.o. Díky této službě je možné s IQRF sítí možné komunikovat skrze MQTT. Nutnost je tedy mít k dispozici MQTT broker, který bude zprostředkovávat MQTT komunikaci.

Funkci MQTT brokeru zastane aplikace `mosquitto`³⁰. Role brokeru v měřicím systému je přijímat publikované zprávy do témat a předávat je odběratelům, jež jsou v k tomuto tématu přihlášení k odběru. Odběratel může být například zařízení, či program připojený k brokeru přes softwarový konektor. Publikovat a odebírat zprávy bude také služba `mqtt-to-sql`, jež bude pro potřeby této diplomové práce zpracována. Díky tomu bude moci žádat jednotlivé uzly o zaslání naměřené hodnoty na senzoru a zprávy s výsledky si pak vyzvedávat.

Přijatá data z MQTT brokeru jsou v nezpracovaném (surovém) stavu. Programy jednotlivých uzlů totiž, jak již bylo psáno v jejich popisech, zprostředkovávají pouze nastavení a vyčtení hodnot příslušných senzorů. Ve službě `mqtt-to-sql` proto jsou zakomponovány algoritmy pro zpracování dat, které se nakonec uloží do MySQL databáze, jež slouží jako datový zdroj pro další bod této diplomové práce. Zmíněná MySQL databáze je umístěna ve školní síti na serveru `rc112.vsb.cz`. Pro potřeby ukládání a vyčtení dat z prostředí mimo školu je tedy nutné využít VPN, jelikož přístup k databázi z veřejného sektoru není prozatím možný.

5.3.1 Služba `iqrf-daemon`

Služba `iqrf-daemon` komunikuje s modulem koordinátora pomocí SPI. Aby bylo možné službu využívat, je nutné ji správně nastavit. Jedná se zejména o konfigurační soubory `IqrfInterface.json` a `MqttMessaging.json`, ve kterých se nachází nastavení MQTT klienta (port a adresa brokeru, témata, úroveň QoS, podoba zprávy), či samotné služby (povolení MQTT, mód běhu služby, apod.). Máme-li vše potřebné nastaveno, je možné publikovat zasílat na MQTT broker příkaz ve formě JSON struktury, který si služba `iqrf-daemon` vyzvedne. Následně jej zpracuje a proběhne-li všechno v IQRF síti tak jak má, publikuje služba po chvíli odpověď do zvoleného tématu.

Důležitá nastavení v souboru `IqrfInterface.json`:

- **IqrfInterface** : Konfigurace rozhraní pro modul koordinátora (`/dev/spidev0.0`)

Důležitá nastavení v souboru `MqttMessaging.json`:

³⁰Více zde: <https://mosquitto.org/>

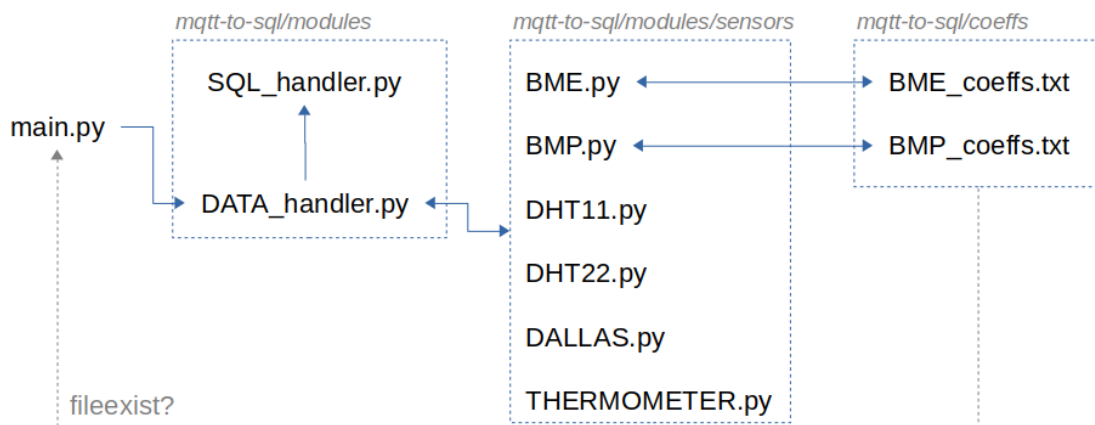
- **BrokerAddr** : IP adresa a port brokeru (127.0.0.1:1883)
- **TopicRequest** : Téma pro požadavky (Iqrf/DpaRequest)
- **TopicResponse** : Téma pro odpovědi (Iqrf/DpaResponse)
- **User** : Uživatelské jméno v případě, je-li potřeba
- **Password** : Uživatelské heslo v případě, je-li požadováno
- V tomto souboru je nastavení pro dva MQTT kanály. První z nich je věnován lokálnímu brokeru, druhý pak lze využít pro spojení s veřejným brokerem.

Služba `iqrf-daemon` se automatickým po svém spuštění přihlásí k odběru do zadaných témat na poskytnutém MQTT brokeru.

5.3.2 Služba `mqtt-to-sql`

Programový kód vytvořený pro účely této diplomové práce je ve výsledku služba běžící na pozadí linuxového operačního systému. Psána je ve skriptovacím jazyce *python* ve verzi 2.7. Mezi hlavní výhody použití tohoto programovacího jazyku patří rychlost, s jakou je možno vyvinout výslednou aplikaci a také široká komunitní podpora a dostupné knihovny. V této službě pak jsou použity konkrétně knihovny jako *mysql.connector*, či *paho.MQTT* pro práci se vzdálenou mySQL databází a komunikaci s MQTT brokerem. Samozřejmostí je použití funkcí pro práci s časem z knihovny *time* a soubory z knihovny *os.path*.

Struktura služby je následující:



Obrázek 36: Struktura služby `mqtt-to-sql`

- `main.py`

Základní soubor registrovaný jako služba OS. Implementuje v sobě hlavní vlákno a přerušování v rámci obsluhy MQTT protokolu. V hlavním vlákne dochází k pravidelnému odesílání

požadavků o nastavené pauze mezi jednotlivými operacemi. Je-li do téma pro odpověď z IQRf sítě přiřazena zpráva, dojde k přerušení. Zpráva je vyčtena a předána dál ke zpracování metodě *sensor_Data_Handler(Topic, Response)*, jež je importována ze souboru *DATA_handler.py*. Probíhá zde také kontrola přítomnosti souborů s koeficienty. Jsou-li přítomny, nezasílá se zbytečný požadavek pro vyčtení koeficientů.

- *DATA_handler.py*

Pomocí metody *sensor_Data_Handler* se zde data podrobují testování na obsah. Je-li zjištěna obsahová shoda s některých z pravidel, jsou data na základě výsledku přiřazena ke zpracování metodám souborů ze složky *sensors*, jež jsou návratové. Data se porovnávají na základě značky "response", ve které se nacházejí informace o uzlu, dotazované periférii, možné chybě a také zde jsou samotná data určena ke zpracování.

- *SQL_handler.py*

Rutina *insert_Data_toMySQL(Data[], *args)* na obsluhu SQL relace je volána metodou *sensor_Data_Handler* ve chvíli, kdy jsou připravená zpracovaná data k vložení do databáze. Dojde ke zprostředkování připojení k mySQL databázi na serveru rc112.vsb.cz, data se vloží a relace skončí.

- Obslužné metody senzorů

Návratové metody těchto souborů mají za úkol přijatá data zpracovat. Zpracování zde probíhá na úrovni bitů, kdy je nutné každý bit předané zprávy správně reprezentovat. Zejména pak se jedná o data s koeficienty. Soubory se nacházejí ve složce */modules/sensors*.

Zpracovaná data se následně ukládají do mySQL databáze na serveru rc112.vsb.cz, jež je umístěn ve školní síti. Jak již bylo několikrát avizováno, nutné je připojení pomocí VPN, jež zařízení v vnějším prostředí propojí s interní školní sítí. Připojení je realizováno pomocí balíku OpenConnect³¹

Ukládání probíhá za pomoci jednoduchého dotazu o následující podobě:

```
INSERT INTO IQRfData (ID, timestamp, N2_Dallas_T, N2_DHT11_T, N2_DHT11_H,
    N2_Inter_T, N1_BME280_T, N1_BME280_P, N1_BME280_H, N1_Inter_T)
VALUES (%(ID)s, %(Datum)s, %(Dallas_T)s, %(DHT11_T)s, %(DHT11_H)s, %(N2_T)s, %(
    BME_T)s, %(BME_P)s, %(BME_H)s, %(N1_T)s))
```

Výpis 3: Insert dat do MySQL databáze

³¹Dostupné z: <http://www.infradead.org/openconnect/>

ID	timestamp	N2_Dallas_T	N2_DHT11_T	N2_DHT11_H	N2_Inter_T	N1_BME280_T	N1_BME280_P	N1_BME280_H	N1_Inter_T
1	2018-03-09 17:34:37	26.5	27	57	29	24.6561030	985.837680	43.8805268	27
2	2018-03-09 17:35:15	26.5	27	57	29	24.6573625	985.839719	43.3483046	27
3	2018-03-09 17:35:52	27.0	26	58	29	24.6598816	985.842112	43.9154193	27
4	2018-03-09 17:36:30	27.0	26	58	29	24.6567328	985.848804	44.0098577	27
5	2018-03-09 17:37:07	27.0	27	57	29	24.6482309	985.851885	42.2933636	27
6	2018-03-09 17:37:45	27.5	27	56	29	24.6331164	985.857735	41.5914375	27
7	2018-03-09 17:38:22	28.0	27	56	29	24.6170573	985.867107	40.7101964	26.5

Obrázek 37: Struktura tabulky a její obsah

5.3.3 MQTT zprávy

První uvedená zpráva obsahuje dotaz na obsah paměti RAM uzlu 1, kde se nacházejí kalibrační konstanty vyčtené z připojeného čidla BME280. Vyčteno ná být celkem 36B. HWPID je zadáno univerzální

JSON request na RAM uzlu 1

```
{ "response_ts": "",
  "msgid": "1521120698.61",
  "request": "01.00.05.00.ff.ff.00.24",
  "ctype": "dpa",
  "request_ts": "",
  "confirmation_ts": "",
  "type": "raw", }
```

Výpis 4: JSON request na paměť RAM

Následuje odpověď na dotaz pro vyčtení obsahu RAM paměti uzlu 1. Obsahem je tedy kromě hlavičky zprávy DPA protokolu dále 36B, ve kterých se nacházejí vyčtené kalibrační konstanty.

JSON response s koeficienty BME280

```
{ "ctype": "dpa",
  "type": "raw",
  "msgid": "1521120698.61",
  "request": "01.00.05.00.ff.ff.00.24",
  "request_ts": "2018-03-15 13:31:38.612740",
  "confirmation": "01.00.05.00.ff.ff.ff.3a.01.04.01",
  "confirmation_ts": "2018-03-15 13:31:38.661255",
  "response": "01.00.05.80.34.12.00.3d.cb.6e.27.67.32.00.bc.91.51.d6.d0.0b.7a
    .1b.b4.ff.f9.ff.0c.30.20.d1.88.13.4b.43.01.00.1a.2d.03.1e.00.00.00.00",
  "response_ts": "2018-03-15 13:31:38.828260",
  "status": "STATUS_NO_ERROR" }
```

Výpis 5: JSON response na request 0x05 RAM

Dva následující výpisy obsahují nejprve dotaz a poté odpověď týkající se vyčtení aktuálně naměřené hodnoty teploty digitálním teplotním čidlem DS18B20.

JSON request na DS18B20 uzlu 2

```
{ "confirmation": "",
  "response_ts": "",
  "msgid": "1521120662.87",
  "request": "02.00.20.00.ff.ff",
  "ctype": "dpa",
  "request_ts": "",
  "confirmation_ts": "",
  "type": "raw",
  "response": "" }
```

Výpis 6: JSON request na hodnotu DS18B20

JSON response s namerenou hodnotou teploty DS18B20 uzlu 2

```
{ "ctype": "dpa",
  "type": "raw",
  "msgid": "1521120702.37",
  "request": "02.00.20.00.ff.ff",
  "request_ts": "2018-03-15 13:31:42.369569",
  "confirmation": "02.00.20.00.ff.ff.ff.3a.02.04.02",
  "confirmation_ts": "2018-03-15 13:31:42.415727",
  "response": "02.00.20.80.ff.ff.00.43.88.01",
  "response_ts": "2018-03-15 13:31:42.583952",
  "status": "STATUS_NO_ERROR" }
```

Výpis 7: JSON response s hodnotou DS18B20

Mimo samotná užitečná data zprávy dále obsahují informace o průběhu (status), typu (ctype) a id (msgid). Dále také časové značky času odeslání dotazu, potvrzení příjmu dotazu a příjmu odpovědi.

6 IBM Bluemix a Watson IoT

Kapitola šestá je věnována propojení a integraci subsystému pro zpracování a analýzu dat. Jako první záležitost zde bude řešeno propojení SQL databáze s cloudovým vizualizačním systémem IBM Bluemix³². Tím bude umožněn vznik vizualizace, která využije data z SQL databáze jako datový zdroj. Další položkou bude zprovoznění uchovávání hodnot v rámci tzv. historických trendů (grafů). S vyřešením tohoto problému započne fáze implementace subsystému pro zpracování a analýzu dat. Výsledky budou zasílány do vizualizačního subsystému.

6.1 Propojení s databází

Problematiku propojení mySQL databáze s cloudovým systémem IBM Bluemix komplikuje několikero kritérií, které je nutné zohlednit a propojení navrhnout tak, aby bylo spolehlivé.

- Databáze na rc112.vsb.cz je nedosažitelná

Znamená to, že se nelze z IBM Bluemix jednoduše připojit pomocí softwarového konektoru. Správa MySQL serveru totiž potřebuje znát minimálně IP adresu, ze které dochází k připojení na databázi. Tuto IP adresu je však obtížné zjistit, jelikož ani log připojení veřejného MySQL serveru tuto ip adresu nevyzradil. Problémem také je nedosažitelnost samotného MySQL serveru rc112.vsb.cz, kdy je nutné užít VPN. Porty jsou totiž blokovány z důvodu vysokého zabezpečení akademické sítě.

- Licencování modulů a služeb

Struktura cloudového systému je skládána pomocí modulů, či služeb, které jsou nabízeny v katalogu cloudu. Mezi nimi najdeme také několik služeb, jež se věnují problematice VPN připojení. Problém však je s licencováním těchto modulů. IBM Bluemix funguje na bázi paušálu - za všechny moduly se platí částkou dle jejich tarifu, či využití. Licence poskytována školám, tedy i mě, je ve verzi *Trial*. To znamená, že moduly, které nenabízejí tarif *Lite* nebo nejsou přístupné zdarma nemohu do své struktury projektu nasadit. Mezi tyto moduly patří také služby týkající se VPN.

- Další omezení IBM Bluemix

Další neméně povzbudivý fakt je omezení počtu aktivovatelných služeb na 10. Může se to jevit jako dostatečný počet, ale mnohé moduly pro svou funkci vyžadují aktivaci např. 3 služeb.

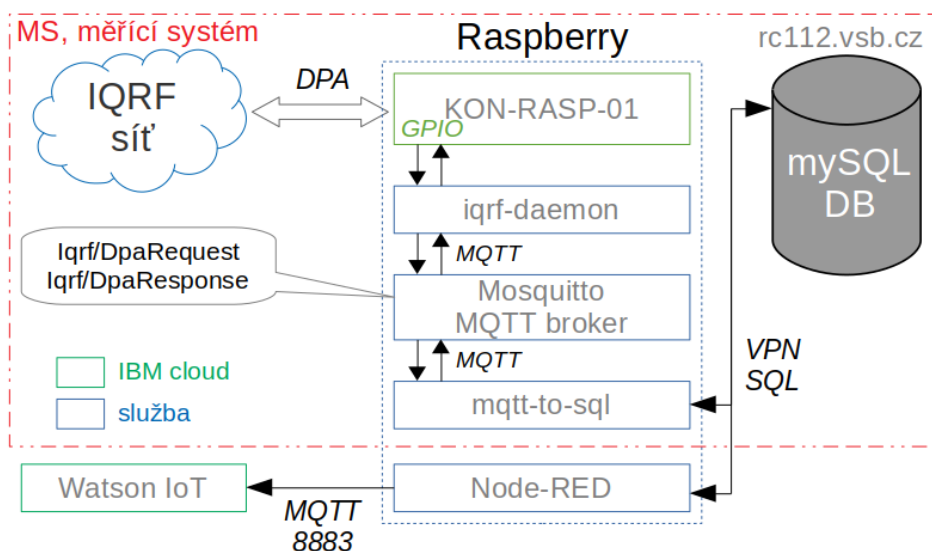
Z důvodu výše popsaných problémů bylo nutné přijít s alternativní cestou, jak odesílat data na IBM Bluemix, potažmo Watson IoT. K tomuto účelu byla na jednodeskovém počítači Raspberry Pi zprovozněna místní instance služby Node-RED. Jelikož je Node-RED vyvinutý

³²Název *IBM Bluemix* byl na jaře roku 2018 při velké aktualizaci zavržen a místo něj se nyní užívá název *IBM Cloud*

společností IBM a navíc pro účely internetu věcí (IoT), nabízí možnost přímého propojení se službou Watson IoT. Raspberry Pi tak bude sloužit zároveň i jako brána propojující mySQL databázi a IBM Bluemix. Programový obsah Node-REDu pak nazvěme datovou pumpou.

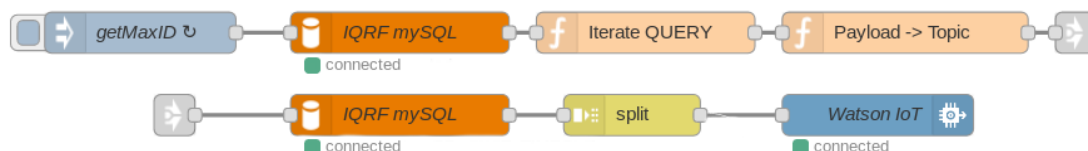
6.1.1 Datová pumpa

Navázání na dosavadní měřicí systém vyobrazuje následující diagram:



Obrázek 38: Struktura MS s bloky datové pumpy a IBM Watson

Samotný programový obsah je jednoduchý. Jedná se o vyčítání dat z SQL databáze pomocí MySQL konektoru, jejich zpracování do požadované podoby a odeslání na službu Watson IoT pomocí správně nastaveného konektoru. Průběžně je nutné kontrolovat poslední vyčtený řádek, aby data na IBM Watson nebyla duplicitní. Důležité je také uchovávat ID posledního řádku z důvodu možného výpadku služby, kdy se všechny programové proměnné vynulují. Z toho důvodu je použit soubor, do kterého se ukládá globální proměnná *lastID* - samozřejmě až po řádném odeslání řádku. Nastane-li restart služby, program naváže na číslo řádku, jež je vyčteno ze souboru. Celý průběh vytváření a zprovoznění služby IBM Watson včetně připojení na ni zachycuje vypracovaný návod - příloha na CD.



Obrázek 39: Část programu datové pumpy v Node-RED řešící vyčítání a formátování

Data připravená k odeslání na IBM Watson pak mají následující podobu.

```
[ { "d": {  
  "ID": 18928,  
  "timestamp": "2018-03-18T18:35:02.000Z",  
  "N2_Dallas_T": "24.5",  
  "N2_DHT11_T": "25",  
  "N2_DHT11_H": "45",  
  "N2_Inter_T": "26.5",  
  "N1_BME280_T": "23.7089340",  
  "N1_BME280_P": "988.455397",  
  "N1_BME280_H": "34.1144734",  
  "N1_Inter_T": "26" } } ]
```

Výpis 8: dataRow připravený k odeslání na Watson IoT

Jedná se o datovou strukturu *dataRow*. Díky tomu bude možné snadno zpracovat data ve vizualizaci. Kdyby byla použita struktura jiná, data jednotlivých odeslání by se ukládala do jiných datových klíčů na příjmu. Systém by totiž nepoznal, že jde o data spadající pod stejný klíč. Výsledkem by byla vizualizace, která by vyobrazovala pouze jeden řádek dat, nikoliv jejich sérii.

6.1.2 Připojení k Watson IoT

Lokální instanci Node-REDu spuštěnou na Raspberry Pi je nutno svázat se službou Watson IoT založenou na IBM Bluemixu. K tomu slouží gateway konektor, na obrázku 39 pojmenovaný jako Watson IoT. Jedná se o funkční blok, který se vnitřně sváže s vytvořeným zařízením, či branou na platformě Watson IoT.

Connect as	Gateway
	<input type="radio"/> Quickstart <input checked="" type="radio"/> Registered
Credentials	BYR0034_GW
Device Type	RaspGW
Device Id	BYR0034_GW
Event type	dataRow
Format	json
Name	Watson IoT

Obrázek 40: Nastavení bloku funkce Watson IoT

Postup založení tohoto zařízení popisuje návod *WatsonIoT-gateway.pdf* v příloze. Toto umožní zasílání dat na Watson IoT. Každé odeslání dat ve formátu *dataRow* tak bude zachyceno a identifikováno pod stejnojmennou událostí. Funkční propojení a zachytávání dat je pak na platformě IoT oznamováno následující formou.

BYR0034_GW		RaspGW	Gateway
Identity	Device Information	Recent Events	State
	Logs	Edge Services	
Device ID	BYR0034_GW		
Device Type	RaspGW		
Date Added	9. bře 2018 13:51		
Added By	byr0034@vsb.cz		
	Connected		
Connection Status	Connection Time: 28. bře 2018 13:55 Client Address: 158.196.195.152 (SecureToken)		

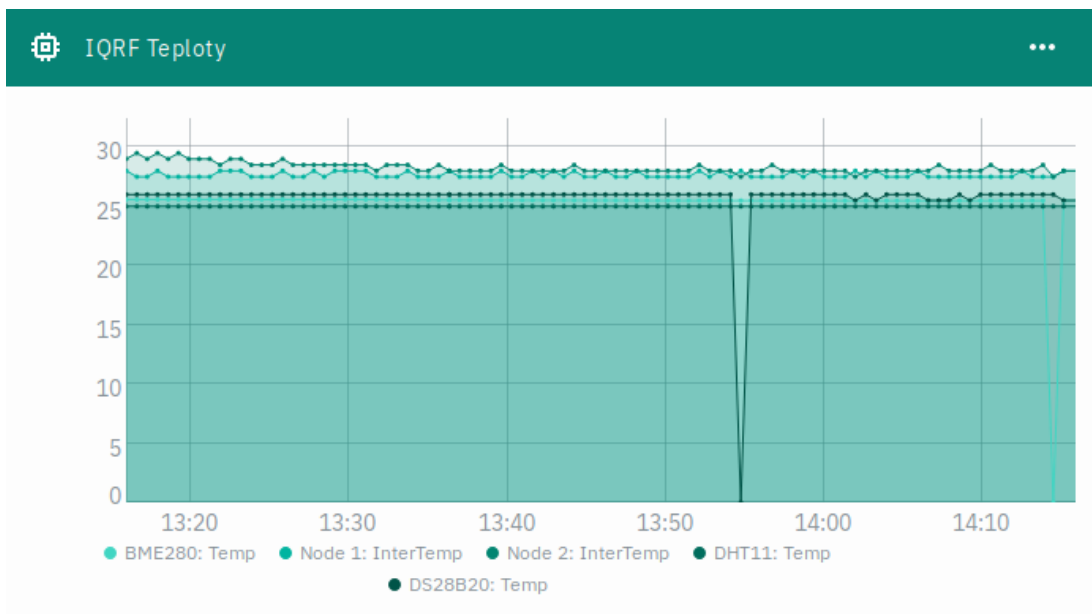
Obrázek 41: Úspěšné propojení s platformou Watson IoT

6.2 Vizualizace

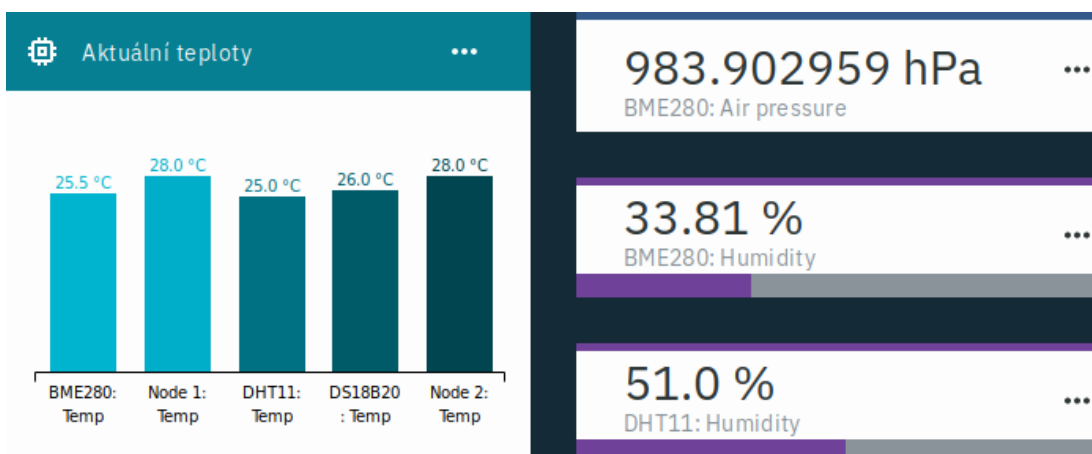
Přijátá data, pokud nejsou vizualizována, či předána ke zpracování uživatelskému programu v rámci instance Node-RED na platformě IBM, se ztratí. Uživatelským programem myslíme jejich zachycení a uložení do databáze. V této podkapitole se však budeme věnovat vizualizaci.

Vizualizační platforma služby Watson IoT si zakládá na tzv. kartách, které představují libovolný vizualizační, či strukturální prvek. Tyto karty jsou shlukovány do složek, což umožňuje oddělit karty například s aktuálními hodnotami od dat vizualizovaných z Node-REDu. Každé složce s kartami pak přísluší vlastní čelní panel vizualizace. Tyto panely si pamatují rozmístění a velikosti jednotlivých karet i skrze změnu rozlišení, potažmo roztažení okna vašeho webového prohlížeče. Čelní panely lze také zpřístupnit veřejnosti - kdokoliv si tak přes poskytnutý odkaz zobrazit čelní panel s vizualizací.

Vizualizování dat se začíná volbou prvku, který bude vybraná data vizualizovat. Pro typ dat, který máme k dispozici z IQRF sítě je nevhodnější využít klasické křivkové grafy, či textové číselníky. Zvolenému prvku následně přiřadíme, kterou, či které datové klíče z *dataRow* má zachytávat a s příštím datovým příjmem se hodnota klíče vykreslí. Časový klíč hodnoty převezmou z okamžiku, kdy byla přijata. Následující obrázek poskytuje náhled na možnou podobu karty.



Obrázek 42: Karta s průběhy zaznamenaných teplot z IQRF sítě



Obrázek 43: Další druhy vizualizačních karet informující o naměřených hodnotách

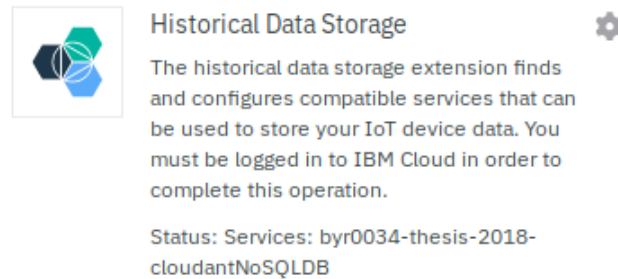
6.2.1 Datová historie

Ukládání dat je jistě nedílnou součástí také tohoto nasazení služby Watson IoT. V případě zobrazení historických trendů, či zpracování dat předešlých dní je nutné mít k dispozici historická data. Datová historie se dá řešit několika způsoby:

- Historical data service

V rámci rozšíření pro Watson IoT je k dispozici zásuvný modul pro ukládání historických dat. Zakomponování tohoto modulu je velice jednoduché, postačí zvolit časový interval, který bude od sebe dělit jednotlivé databáze, dále časovou zónu a prefix názvů vytvářených databází. Po aktivaci jsou přijatá data automaticky ukládána do vaší NoSQL Cloudant

databáze. Takto uložená data jsou ovšem vložena navíc do automaticky generované datové struktury JSON s takéž generovaným názvem samotného dokumentu, což může v řadě případů komplikovat jejich vyčítání z databáze.



Obrázek 44: Rozšíření: Historical Data service

Použít tento modul však vřele doporučuji, neboť bez něj by bylo značně složitější řešit historické trendy s delším jak 7 denním intervalem a s jeho nasazením se pro potřeby historických trendů zkrátka počítá.

- Last Event Cache

Rozšíření aktivovatelné skrze panel nastavení služby Watson IoT. Jedná se o cache, do které se ukládají informace o posledních vyvolaných událostech připojeného zařízení k Watson IoT v určeném časovém intervalu. Tento interval je dlouhý 1 nebo 7 dní a umožní dočasné zachování dat přijatých v rámci událostí skrze konektor zařízení.

Využít této možnosti také doporučuji, neboť řeší ztrátu již zobrazených dat z prvků vizualizace např. při obnově stránky.

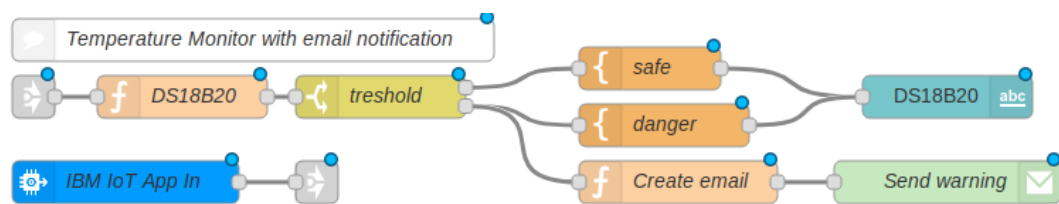
- Použití služby Node-RED

Toto řešení obnáší implementaci vlastního programu do služby Node-RED, kde se skrze konektor k zařízení dají přeposílat přijaté události ze služby Watson IoT. Node-RED nabízí celou řadu konektorů i ke službám IBM Cloudu, což umožňuje implementátorovi rozhodnout se pro nejvhodnější řešení ukládání dat. Bez použití již dvou jmenovaných metod však nelze zajistit kvalitní zpracování vizualizace.

6.2.2 Notifikace, alarmy

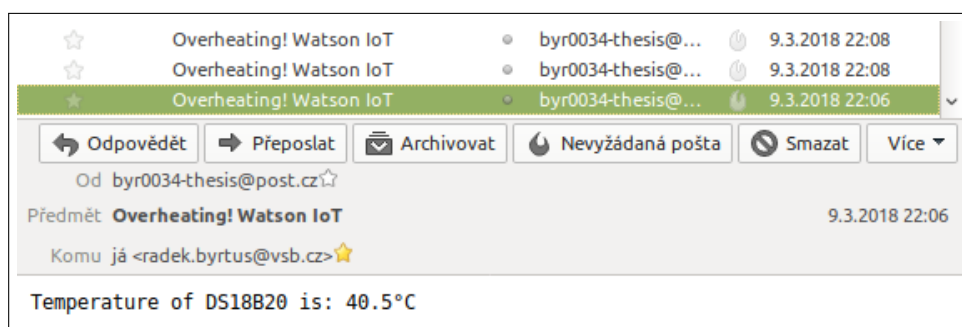
Služba Watson IoT také poskytuje službu pro zaslání notifikace v případě vzniku problému. Takovým problémem může být nebezpečný nárůst sledované teploty, či datový výpadek. V době vypracování této diplomové práce však tato služba prochází dlouhodobou úpravou s nabídkou otestování BETA verze na úrovni implementace pravidel v API Watson IoT. Mé snahy se však zastavily při potvrzení změn v metadatech pro vytvořené zařízení, kde se mají vložit meze veličiny v JSON formátu - tlačítko formuláře nereaguje stejně tak, jak je tomu v případě přidání rozšíření s názvem Custom Cards, které vám do vašeho vizualizačního prostředí umožní přidání vlastních vizualizačních prvků.

Notifikace a alarmy je proto nutné řešit jiným způsobem a tím je použití Node-REDu. Zde je několik bloků funkcí, které lze využít pro zpracování systému notifikací a alarmů. Patří mezi ně blok pro zachycení aktuální příchozí události na zařízení Watson IoT, blok pro porovnávání, či blok vlastní funkce. Nejdůležitějším blokem je však funkce, která umožňuje zaslat skrze SMTP server upozorňující email. SMTP server je třeba volit vlastní, či třetích stran jako např. portálu seznam.cz. Program řešící hlídání aktuální teploty čidla DS18B20 a zaslání emailu vypadá následovně.



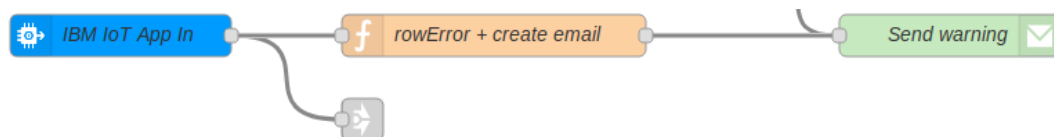
Obrázek 45: Zpracování notifikací pro čidlo DS18B20

Prahovou hodnotu lze volit ovládacím prvkem ve vizualizačním prostředí Node-REDu, kde je také zaslán k vypsání výsledek vyhodnocení. Překročí-li aktuální teplota hodnotu prahovou, vytvoří se email s varováním, který se následně odešle na zadanou emailovou adresu.

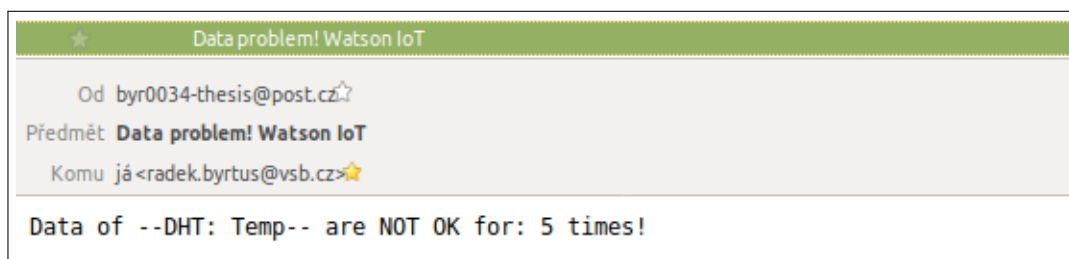


Obrázek 46: Podoba přijatých notifikačních emailů

Zajisté je také potřeba detekovat výpadek dat. To lze uskutečnit za použití čítače. Pokaždé, když budou přijatá data v pořádku, dojde k restartování časového intervalu časovače. Jakmile budou data po několik příchozích událostí špatná, bude odeslán email s informacemi o problému.



Obrázek 47: Zachycení výpadku dat a zaslání varování pomocí emailu



Obrázek 48: Reakce systému na vytažení senzoru DHT11

6.3 Datová analýza

Kapitola věnující se zpracování a analýze dat bude rozdělena do částí. Počet těchto částí je roven počtu využitých služeb v rámci cloudové platformy IBM Bluemix. Mezi nejdůležitější z nich patří zpracování v rámci instance služby Node-RED a využití služby Watson Analytics.

Část věnována službě Node-RED v rámci IBM Bluemix zahrnuje zpracování a analýzy dat za pomoci uživatelského programu. Využity budou jak knihovní, tak vlastní funkce pro výpočet statistických ukazatelů jako jsou *aritmetický průměr*, *medián*, *rozptyl*, *směrodatná odchylka*, *maximum* a *minimum*, či počty vzorků dat v rámci zvolených intervalů. Zpracován bude také ukazatel kvality dat, tedy jaké množství dat je chybové a která z nich jsou chybová nejčastěji.

Zpracování dat v rámci služby Watson Analytics bude obsahovat pokročilejší datové analýzy, jejichž výsledky budou přehledně prezentované vhodnými vizualizačními prvky. Mezi použitými metodami se bude vyskytovat porovnávání vícero veličin, jejich vzájemná asociace, ovlivňování, ba i predikce jejich hodnot za určitých podmínek.

6.3.1 Node-RED

V rámci IBM instance služby Node-RED dojde na aplikaci základních statistických ukazatelů a analytických metod. Zdrojem těchto metod jsou jak knihovní funkce, tak také uživatelský kód psaný v jazyce JavaScript. Analyzována budou jak data uchovaná v databázi, tak aktuálně přijatá v rámci volitelného časového intervalu.

Vstupní data budou funkcím poskytována buďto ve formě pole, či postupně po jednom prvku, kdy tomu tak bude v případě zpracování aktuálních hodnot, které přicházejí postupně. Knihovní funkce jsou na tuto záležitost připraveny, stejně tak tomu musí být u uživatelsky zpracovaných funkcí. Proto se v rámci nich musí pracovat s interní cache, neboli kontextem funkčního bloku, který umožňuje uchování přijatých dat napříč jednotlivými voláními těchto funkčních bloků. Bez využití tohoto kontextu by se při každém volání funkce pro výpočet například maxima musela

vždy nahrát všechna data nanovo a z nichž by se posléze maximum získalo. Výsledné hodnoty pak budou poskytovány na výstup ve vhodném formátu, stejně tak budou pro jistotu uloženy do globálních proměnných, jež jsou dosažitelné v celé instanci Node-RED na platformě IBM Cloudu.

U funkcí knihovnic se problémem také stává zachytit, který prvek z pole je maximální, jelikož tyto funkce nepracují s celou poli datových struktur. Na výstupu se tedy sice dozvíme maximální hodnotu datového pole, ovšem který prvek, či v který čas byl naměřen již nikoliv. Z tohoto důvodu bude do uživatelského programu v rámci funkčních bloků implementováno zachycení času a ID nalezené maximální a minimální hodnoty.

Použití knihovnic funkcí bude vyobrazeno v přílohách této diplomové práce, pro svou blokové rozsáhlosti. Následuje soupiska implementovaných statistických a analytických funkcí v Node-REDu.

6.3.1.1 Maximum a minimum je maximální nebo minimální hodnota vstupního datového vektoru. Zde je vstupní datový vektor podroben porovnání jednotlivých hodnot, na základě něhož vzejde maximální a minimální hodnota.

```
...
BME_T_min = global.get("BME_T_min");
BME_T_max = global.get("BME_T_max");

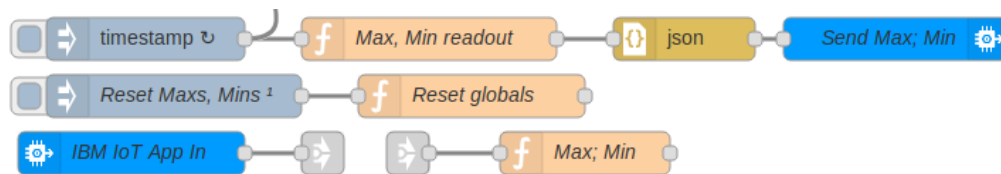
bad_value = "0.0";
data = msg.payload.d;

if (data.N1_BME280_T < BME_T_min) && (data.N1_BME280_T !== bad_value)){
  global.set("BME_T_min", data.N1_BME280_T);
  global.set("BME_T_min_t", data.timestamp);
  global.set("BME_T_min_id", data.id);
}

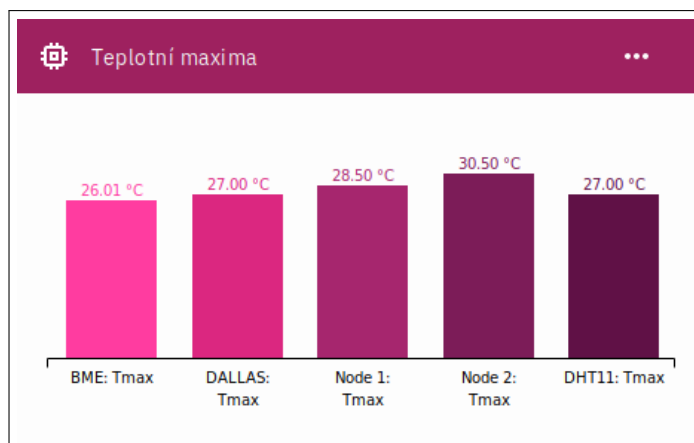
if ((data.N1_BME280_T > BME_T_max)){
  global.set("BME_T_max", data.N1_BME280_T);
  global.set("BME_T_max_t", data.timestamp);
  global.set("BME_T_max_id", data.id);
}
...
```

Výpis 9: Kód na hledání maxima a minima v přijatých datech

Spolu s vyhodnocováním pro ostatní naměřené veličiny jsou data dále odeslána k vizualizaci do služby Watson IoT pod vlastní událostí.



Obrázek 49: Flow kód pro hledání zpracování minim a maxim v Node-REDu



Obrázek 50: Vizualizační karta maximálních hodnot teplot

6.3.1.2 Aritmetický průměr je součtem všech hodnot vstupního datového vektoru vyděleným jejich počtem. Všechny hodnoty ve výpočtu mají stejnou důležitost – váhu.

```
function fc_mean(numbers)
{
  var total = 0,
  i;
  for (i = 0; i < numbers.length; i++) {
    total += numbers[i];
  }
  return total / numbers.length;
}
```

Výpis 10: Metoda pro výpočet aritmetického průměru

6.3.1.3 Medián je hodnota, jež dělí řadu vzestupně, či sestupně seřazených výsledků na dvě stejně početné poloviny. Ve statistice patří mezi míry centrální tendence. Platí, že nejméně 50% hodnot je menších nebo rovných a nejméně 50% hodnot je větších nebo rovných mediánu.

```
function fc_median(numbers)
{
    numsLen = numbers.length;
    numbers.sort();
    if (numsLen % 2 === 0)
    {
        return (numbers[numsLen / 2 - 1] + numbers[numsLen / 2]) / 2;
    }
    else
    {
        return numbers[(numsLen - 1) / 2];
    }
}
```

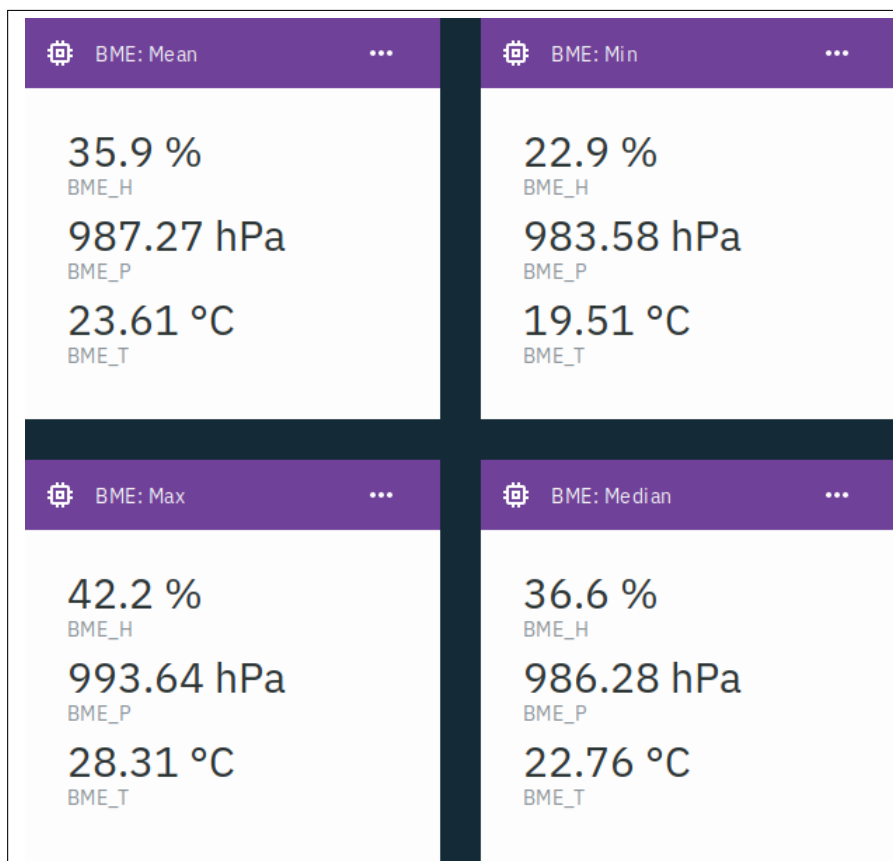
Výpis 11: Metoda pro hledání mediánu

Na následující obrázku 51 jsou vyobrazeny vizualizační karty vypisující maximální, minimální, průměrnou a mediánovou hodnotu veličin statistiky digitálního čidla BME280.

6.3.1.4 Nejvyšší četnost je prvek jeden, či soubor několika prvků, které splňují podmínku největšího počtu výskytů ve vstupním datovém vektoru.

```
function fc_mode(numbers){
    var modes = [], count = [], i, number, maxIndex = 0;
    for (i = 0; i < numbers.length; i++){
        number = numbers[i];
        count[number] = (count[number] || 0) + 1;
        if (count[number] > maxIndex)
            maxIndex = count[number];
    }
    for (i in count) if (count.hasOwnProperty(i)) {
        if (count[i] === maxIndex)
            modes.push(Number(i));}
    }
    return modes;
}
```

Výpis 12: Metoda pro hledání nejvyšší četnosti



Obrázek 51: Karty maxima, minima, ar. průměru a mediánu veličin čidla BME280

6.3.1.5 Rozsah je reprezentován dvěma hodnotami, které reprezentují interval, ve kterém se nacházejí vstupní hodnoty datového vektoru.

```
function fc_range(numbers){
  numbers.sort();
  return [numbers[0], numbers[numbers.length - 1]]; }
```

Výpis 13: Metoda pro výpočet rozsahu hodnot

6.3.1.6 Rozptyl hodnot udává údaj o tom, jak moc jsou hodnoty ve vstupním datovém souboru rozptýleny.

```
function fc_variance(numbers){
  var mean = fc_mean(numbers);
  return fc_mean(numbers.map(function(num) {
  return Math.pow(num - mean, 2);
  }));}
```

Výpis 14: Metoda pro výpočet rozptylu hodnot

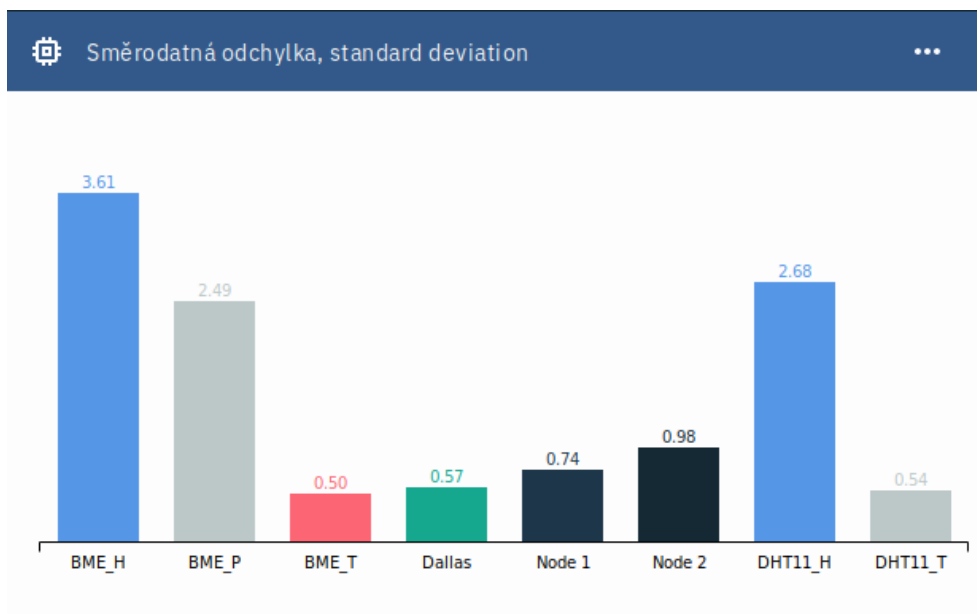
6.3.1.7 Směrodatná odchylka reprezentuje hodnotu, která je definována jako odmocnina z rozptylu hodnot a je mírou variability náhodné veličiny.

```
function fc_standDeviation(numbers){  
  return Math.sqrt(fc_variance(numbers)); }  

```

Výpis 15: Metoda pro výpočet směrodatné odchylky hodnot

Vypočtené hodnoty rozsahu a rozptylu hodnot jednotlivých naměřených veličin jsou posléze přenášeny do vizualizační části k vypsání do příslušné karty.



Obrázek 52: Vizualizační karta směrodatných odchylek

Na vyobrazeném grafu se můžeme dozvědět, že nejvyšší hodnoty směrodatné odchylky drží naměřené hodnoty relativní vlhkosti, což znamená, že se nejméně blíží získanému aritmetickému průměru. Jsou tedy rozesety dále od této hodnoty. Naopak nejbližší aritmetickému průměru se blíží hodnoty naměřených teplot nejspíše z důvodu, že v tento den se teplota v místnosti zásadně neměnila.

6.3.1.8 Průměrná absolutní odchylka je hodnota, která je výsledkem průměrování absolutní odchylky jednotlivých hodnot od aritmetického průměru.

```
function fc_meanAbsoluteDeviation(numbers){  
  var mean = fc_mean(numbers);  
  return fc_mean(numbers.map(function(num){  
    return Math.abs(num - mean);  
  })); }  

```

Výpis 16: Metoda pro výpočet absolutní průměrné odchylky hodnot od průměru

6.3.1.9 Standardizované skóre je pole hodnot standardizovaného skóre, jež je ve statistice označení pro čísla vzniklá lineární transformací z původně naměřených či jinak zjištěných hodnot (označovaných jako hrubé skóre) tak, aby výsledné rozložení mělo předem dané vlastnosti. Nejčastějším příkladem standardizovaného skóre je z-skóre s průměrem 0 a směrodatnou odchylkou 1. Předpokladem pro použití standardizovaných skóre je normální rozdělení původních hodnot.

```
function fc_zScores(numbers) {
  var mean = fc_mean(numbers);
  var standardDeviation = fc_standDeviation(numbers);
  return numbers.map(function(num) {
    return (num - mean) / standardDeviation;
  });
}
```

Výpis 17: Metoda pro výpočet standardizovaného skóre hodnot

6.3.1.10 Chybovost datového vstupu je hodnota prezentující počet chybných měření v rámci vstupního datového vektoru. Tato hodnota dále slouží k výpočtu procentuálního zastoupení chyb v rámci celého datového souboru naměřené veličiny. Tato funkce se také podílí na následné bezchybovosti dat určených ke statistickému zpracování, kdy výstupy těchto funkcí nejsou chybami v naměřených datech ovlivněny.

Využívá se zde mnohočetného výstupu bloků vlastních funkcí, kdy hodnota, která projde šetřením je pomocí metody *send* třídy *node* odeslána příslušným výstupem do vlastní instance statistické a analytické funkce.

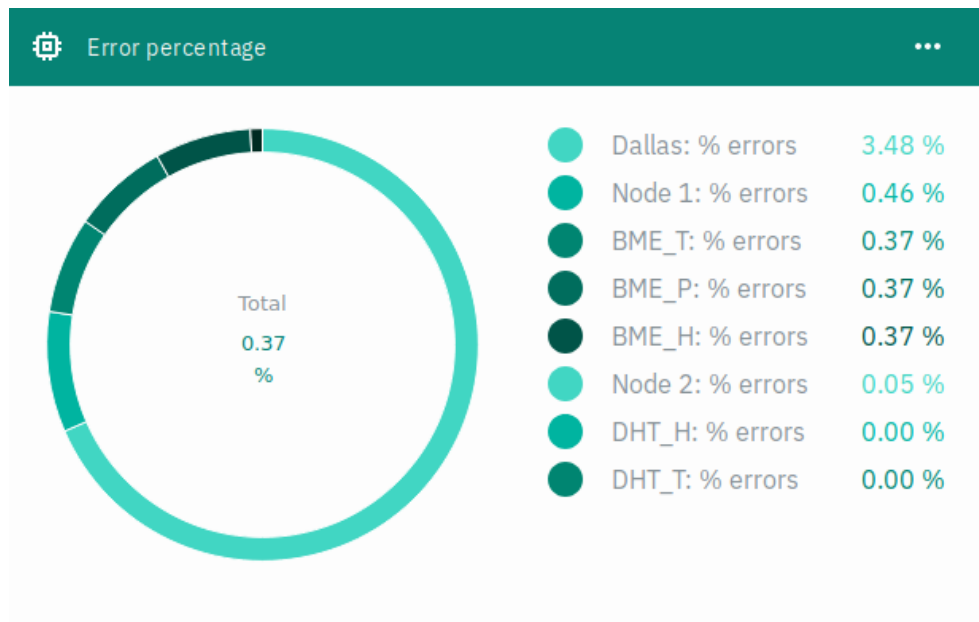
```
data = msg.payload;
delka_dataRow = global.get("rowCount_e"); errors = [0,0,0,0,0,0,0,0];

if(msg.eventType === "dataRow"){
  delka_dataRow++;
  if (data.d.N1_BME280_T !== badValue){
    data_1 = parseFloat(data.d.N1_BME280_T); // Odeslání 1. výstupem
    node.send([{"payload":data_1},null,null,null,null,null,null]);
  }
  else
    errors[0]++;
...

```

Výpis 18: Funkce pro kontrolu a přidělení dat

Zpracované hodnoty se dále zasílají do příslušné vizualizační karty k vyobrazení. K vidění jsou zde jak jednotlivé počty vadných řádků v rámci celého dne, tak také jejich procentuální zastoupení. Na následujícím grafu můžeme právě toho procentuální zastoupení vidět. Nejvyšší výskyt chyb v rámci zvoleného dne vykazuje digitální teplotní čidlo DS18B20.



Obrázek 53: Vizualizační karta procentuálního zastoupení vadných řádků v rámci dne

6.3.1.11 Rozložení hodnot v rámci intervalů lze využít jako ukazatel nejčastěji se vyskytujících hodnot v rámci měření. Hodnoty lze dělit do libovolného počtu intervalů. V rámci tohoto nasazení bude výsledek průzkumu dělen dle šesti kritérií do příslušných intervalů.

```

data = msg.dataSets; msg.dataSets = null; // Příprava
dataLength = data.length; numOfSteps = 6;
// Založení polí výsledků
dataPie = []; dataSteps = []; dataMaxs = []; dataMins = [];
// Vytvoření intervalů
for(var d = 0; d < dataLength; d++){
  l_max = Math.max.apply(Math, data[d]);
  l_min = Math.min.apply(Math, data[d]);
  dataMaxs[d] = l_max; dataMins[d] = l_min;

  l_step = (l_max - l_min) / numOfSteps; dataSteps[d] = l_step;

  l_count = []; l_dataLength = data[d].length;
  actualArray = data[d];

```

```

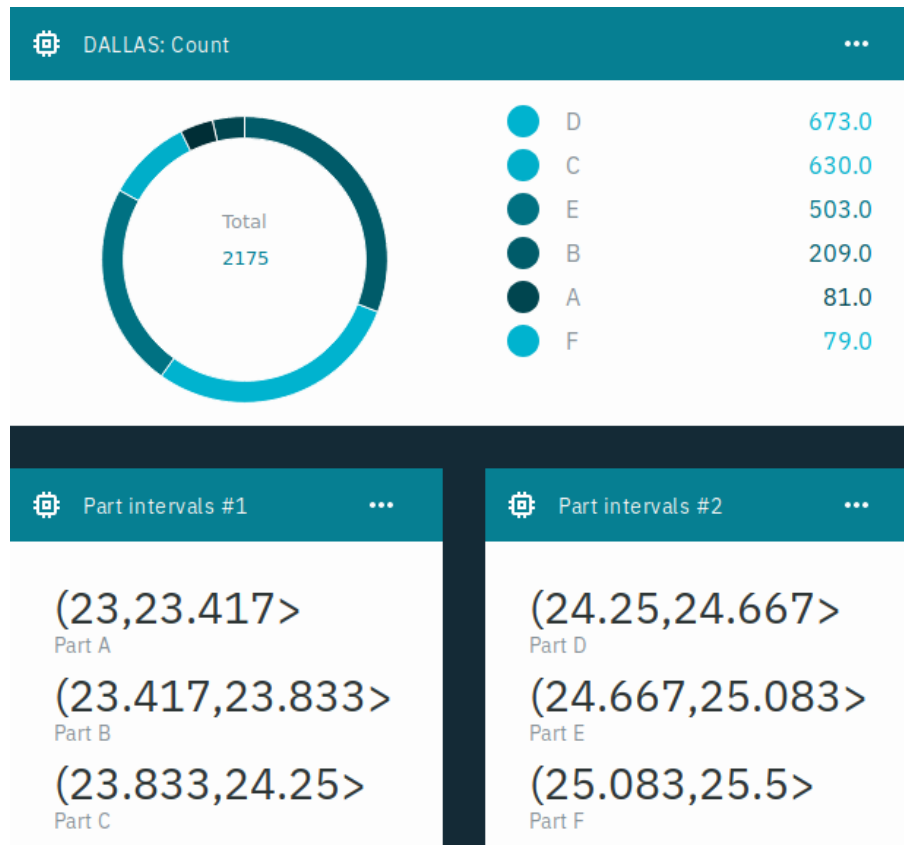
for(var n = 0; n < numOfSteps; n++)
    l_count.push(0);

for(var i = 0; i < l_dataLength; i++){ // Dělení do intervalů
    if((actualArray[i] >= l_min) && (actualArray[i] < (l_min + 1 * l_step)))
        l_count[0]++;
    else if(actualArray[i] >= (l_min + 1 * l_step) && actualArray[i] < (l_min
        + 2 * l_step))
        l_count[1]++;
    else if(actualArray[i] >= (l_min + 2 * l_step) && actualArray[i] < (l_min
        + 3 * l_step))
        l_count[2]++;
    else if(actualArray[i] >= (l_min + 3 * l_step) && actualArray[i] < (l_min
        + 4 * l_step))
        l_count[3]++;
    else if(actualArray[i] >= (l_min + 4 * l_step) && actualArray[i] < (l_min
        + 5 * l_step))
        l_count[4]++;
    else
        l_count[5]++;
}
dataPie.push(l_count);
}

```

Výpis 19: Funkce dělicí hodnoty dle podmínek do intervalů

Výsledkem tedy jsou následující přehledy pro každý vstupní datový soubor. Výstup musí být v rámci optimálního zobrazení rozdělen do více karet, neboť možnosti a přizpůsobení samotných karet je velice omezené.



Obrázek 54: Vizualizační karty prezentující intervaly a zastoupení hodnot v jejich rámci

6.3.1.12 Formátování: koláčové grafy formátuje výstup z funkcí do tvaru vhodného pro odeslání do vizualizačního systému Watson IoT a následného zobrazení v kartách koláčových grafů. Podobné funkce jsou nasazeny i v případě formátování ostatních výstupů např. ze statistických funkcí.

```

dataPie = msg.dataPie; dataSteps = msg.dataSteps; dataMaxs = msg.dataMaxs;
    dataMins = msg.dataMins;
prefixes = ["BME_T", "BME_H", "BME_P", "DHT11_T", "DHT11_H", "DALLAS", "NODE1",
    "NODE2"];

for (var n = 0; n < 8; n++){ // Předzpracování, zaokrouhlení, apod.
    actualPie = dataPie[n]; limits = [];

    for (var i = 0; i < dataPie[0].length; i++)
    {
        low = (dataMins[n] + dataSteps[n] * i);
        high = (dataMins[n] + dataSteps[n] * (i + 1));
    }
}

```

```

    low = Math.round(low * 1000) / 1000;
    high = Math.round(high * 1000) / 1000;

    limits[i] = "(" + low + "," + high + ">";
}
actualPrefix = prefixes[n];
msg = {}; // Vyčištění struktury msg
msg = { // Vložení dat do JSON struktury v msg
payload: JSON.stringify(
{ [actualPrefix]:
  {
    "Part A" : actualPie[0],
    "Part B" : actualPie[1],
    "Part C" : actualPie[2],
    "Part D" : actualPie[3],
    "Part E" : actualPie[4],
    "Part F" : actualPie[5]
  }
}});}
node.send(msg); // Odeslání dat k předání do Watson IoT

```

Výpis 20: Formátovací funkce koláčových grafů

6.3.2 Watson Analytics

Tato služba poskytuje možnost pokročilejší datových analýz za pomoci pokládání strukturovaných vstupních dotazů, které se následně uplatní na množině importovaných dat. V rámci své instance poskytuje pokročilé možnosti vizualizačních prvků, které přehledně reprezentují výsledky zavedených šetření. Ve své podstatě je tato služba zaměřená výhradně pro podnikovou sféru, nic však nebrání analyzovat libovolná vstupní data. Data pak lze získat pomocí konektoru k široké škále databázových systémů. K databázi Cloudant NoSQL DB, ve které jsou uloženy data v rámci vypracování diplomové práce, však konektor k dispozici není - nutné je tedy manuální exportování dat do souboru s příponou .csv a následně je importovat do služby Watson Analytics.

Implementovány zde jsou procesy pro zjišťování vzoru a významu vstupních dat. Díky tomu a automatizované prediktivní analýze, či kognitivních schopností, jako je dialog o přirozených jazycích, lze s daty komunikovat v rozhovorech, jejichž výsledkem jsou odpovědi, kterým lze snadno porozumět. Dále následují možné případy užití včetně podoby zápisu dotazu.

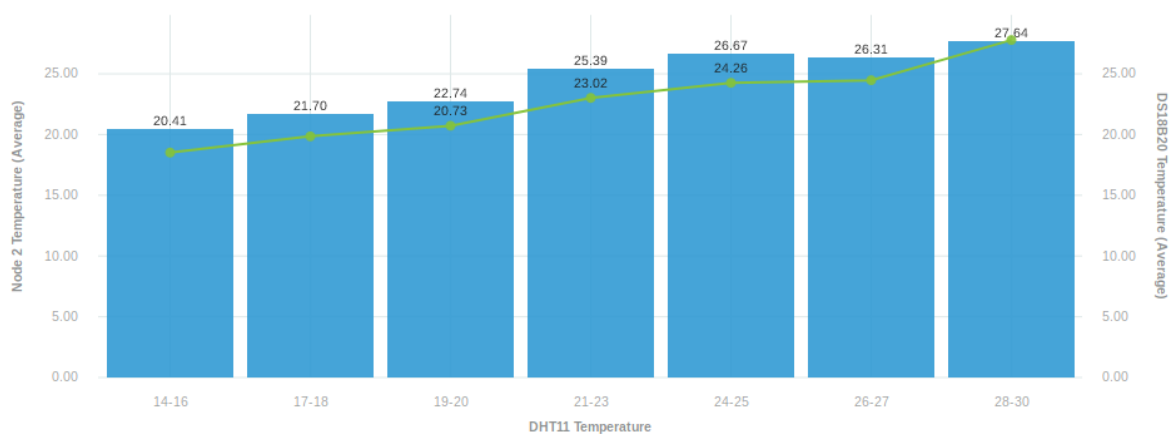
How are values of <BME280: Temperature> and <DHT11: Temperature> associated?

Výpis 21: Ukázka dotazu

6.3.2.1 Porovnání, srovnání je jednou ze základních analýz, kterou lze v rámci služby Watson Analytics použít. Umožňuje přehledně srovnat průměrné hodnoty jednotlivých veličin. V tomto případě jsou porovnávány naměřené hodnoty všech tří externích teplotních čidel.

How do values of <Node 2: Temperature> and <DS18B20: Temperature> compare by <DHT11: Temperature>?

Výpis 22: Porovnání a srovnání dat

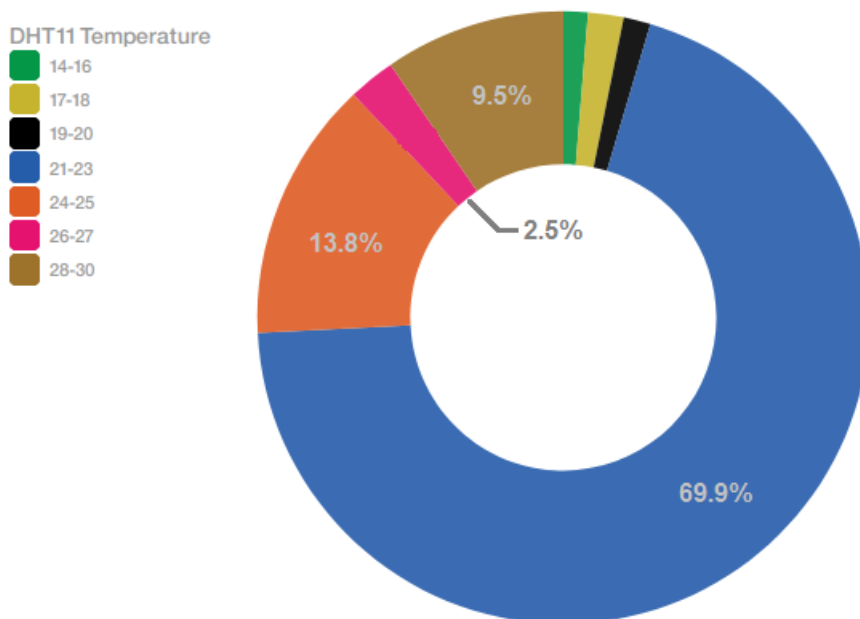


Obrázek 55: WA: Porovnání naměřených teplot

6.3.2.2 Nejčastější hodnota je další ze základních možností analýzy, kterou lze na vstupních datech provést. Systém automaticky dle určených intervalů analyzuje četnost naměřených hodnot v rámci jiné veličiny. V zadaném případě systém zkoumá četnost měření tlaku vzduchu čidla BME280 dle jednotlivých teplotních intervalů na čidlu DHT11.

What is the breakdown of <BME280: Pressure> by <DHT11: Temperature>?

Výpis 23: Nejčastnější výskyt naměřené hodnoty



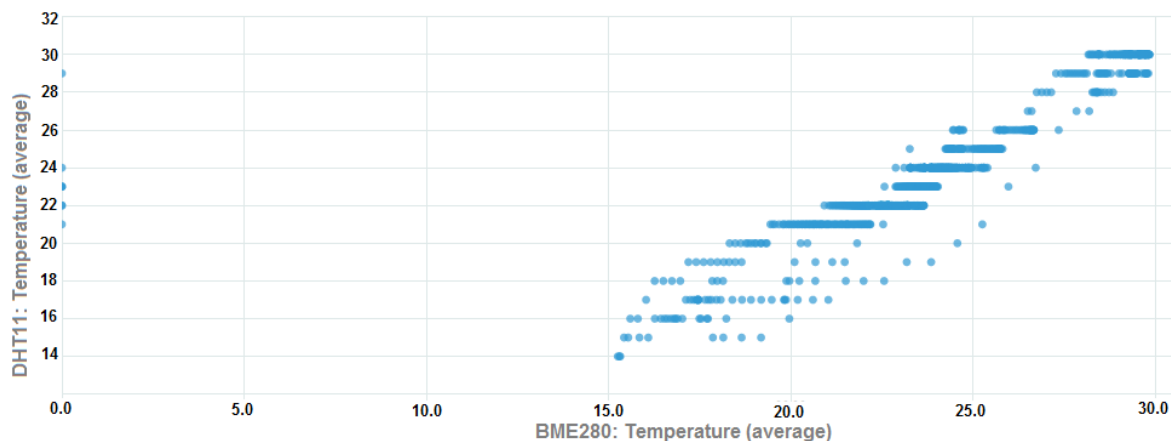
Obrázek 56: WA: Nejčastější hodnota veličiny

6.3.2.3 Asociace umožňuje přehledně zobrazit, jak jsou jednotlivé naměřené hodnoty dvou veličin asociovány co do velikosti.

How are values of <BME280: Temperature> and <DHT11: Temperature> associated?

Výpis 24: Asociace naměřených hodnot

Obecně platí, že by měly jednotlivé body ležet na pomyslné přímce, která těmito body prochází. Každé čidlo však aktuální teplotu naměří jinak, ať už je to z důvodu chyby měření nebo jako v tomto případě různého umístění jednotlivých čidel.

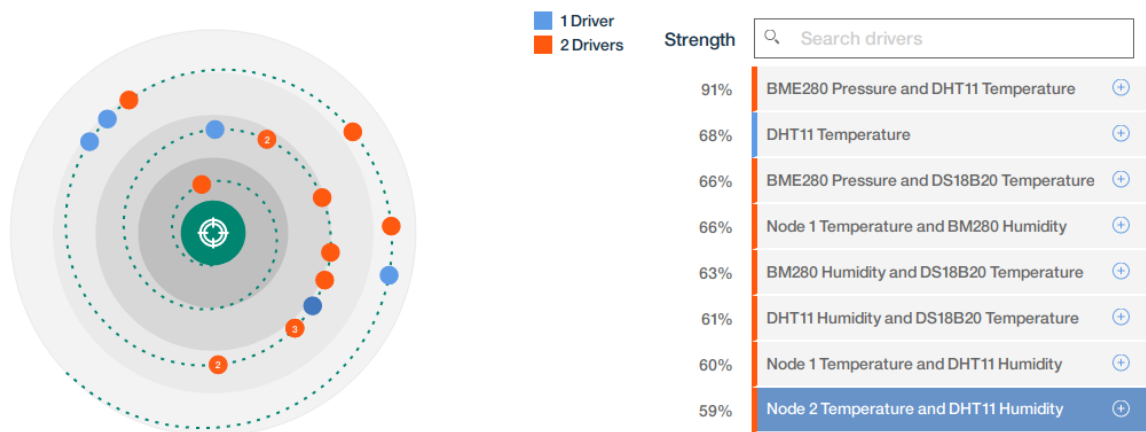


Obrázek 57: WA: Datová asociace

6.3.2.4 Vliv, řízení je ukazatel závislosti veličiny na ostatních veličinách. V případě, který byl zpracován, je teplota naměřená čidlem BME280 nejvíce závislá na naměřeném tlaku vzduchu čidlem BME280 spolu s teplotou naměřenou senzorem DHT11. Právě tato kombinace má nejvyšší hodnocení v predikovatelnosti vývoje teploty naměřené čidlem BME280.

[What drivers](#) <BME280: Temperature>?

Výpis 25: Vzájemné ovlivňování veličin a proměnných

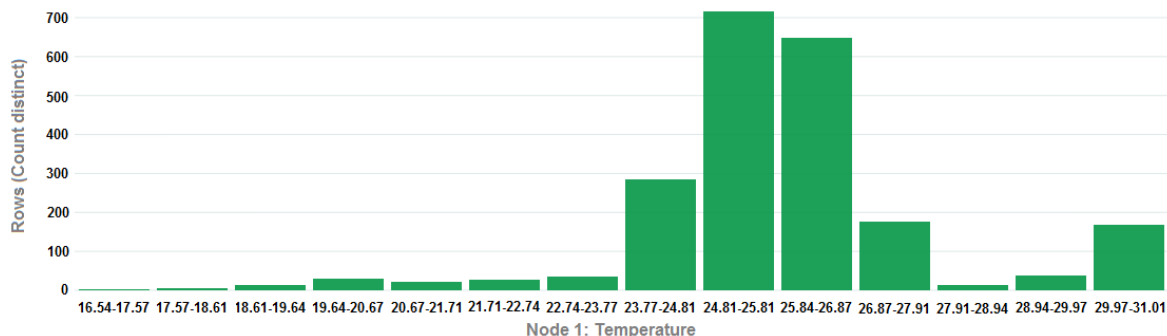


Obrázek 58: WA: Vliv na veličinu, či proměnnou

6.3.2.5 Četnost lze opět považovat za jednu ze základních analytických metod. Jedná se o prosté zjištění, kolik naměřených hodnot určité veličiny svou hodnotou zapadá do stanovených intervalů. Podmínky lze stanovit jak dle svého uvážení, tak pomocí automatického rozdělení rozsahu na několik intervalů.

How does the number of <rows> compare by <Node 1: Temperature>?

Výpis 26: Četnost v rámci intervalů

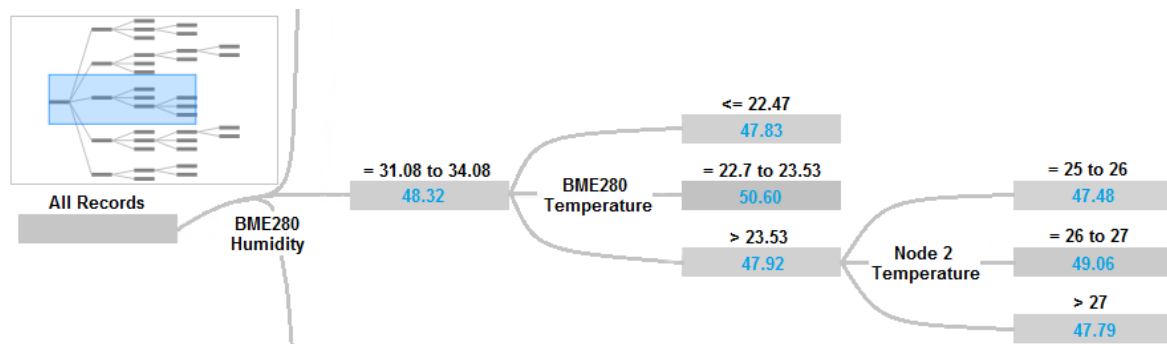


Obrázek 59: WA: Četnost naměřených hodnot v rámci intervalů

6.3.2.6 Podmínkový strom, prediktivní model lze použít k predikci hodnot určité veličiny. Systém automaticky vystaví scénáře, které když nastanou, bude s určenou pravděpodobností hodnota udané veličiny o dané hodnotě, či v určeném rozsahu. Scénáře se vytvářejí na základě importovaných dat, čím více dat je v prostředí podáno, tím přesnější by měly tyto odhady být. V níže zobrazeném případě byl vytvořen prediktivní model pro vlhkost vzduchu naměřenou senzorem DHT11.

What is a predictive model for <DHT11: Humidity>?

Výpis 27: Vytvoření podmínkového stromu

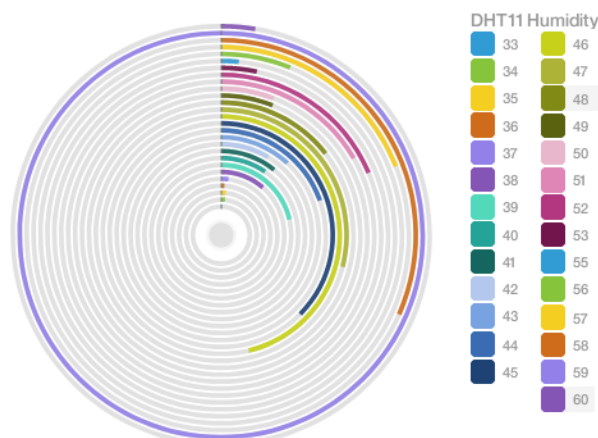


Obrázek 60: WA: Část podmínkového stromu pro predikci hodnoty

6.3.2.7 Rozložení hodnot v rámci vstupních dat umožňuje přehledně zobrazit počty výskytu jednotlivých hodnot jednotlivých veličin. Určit lze buďto konkrétní hodnoty, či opět intervaly. Rozdělení do intervalů bude mnohem vhodnější např. pro naměřené teploty čidlem BME280, jelikož je tato hodnota skoro pokaždé jiná vzhledem k aplikaci přepočtu hodnoty.

What is the value of the number of <timestamp>?

Výpis 28: Dotaz na množství výskytů hodnot



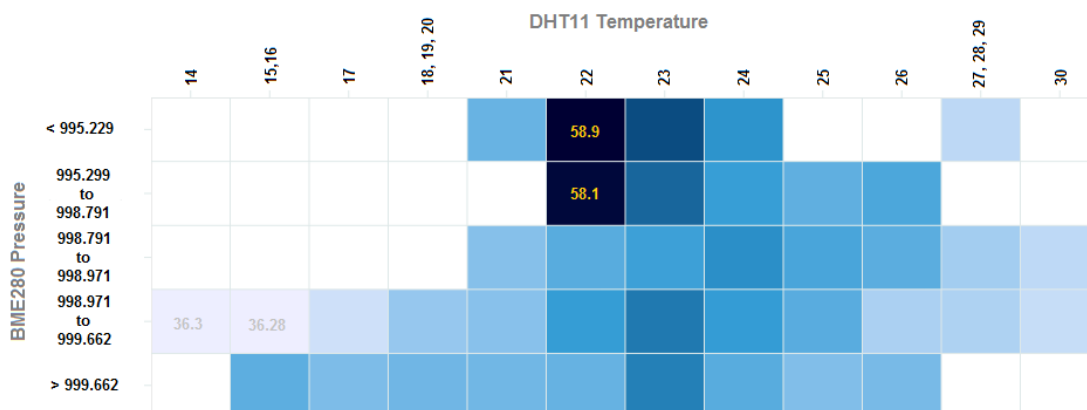
Obrázek 61: WA: Počet výskytů hodnoty v rámci vstupních dat

6.3.2.8 Dopad, význam můžeme považovat za ukazatele významu změny jedné veličiny na jiných. V tomto případě je zkoumán dopad změny tlaku vzduchu a vlhkosti vzduchu na teplotě.

How do <BME280: Pressure> and <BME280: Temperature> impact <DHT11: Humidity>?

What are values of <DHT11: Humidity> for <BME280: Pressure> and <DHT11: Temp>?

Výpis 29: Dotaz na dopad a význam veličiny



Obrázek 62: WA: Dopad změn veličin

7 Závěr

Výstupem této diplomové práce je vytvoření pilotního projektu na vybraném vizualizačním systému, v rámci něhož se budou vizualizovat, zpracovávat a analyzovat data z datového zdroje. O dostupnost vhodných dat se stará navržená a realizovaná IQRF MESH síť s čidly enviromentálních veličin. Ta je složená z tří transeiverů řady TR-7xD a měřeno je celkem osm enviromentálních veličin. Zpracován je celý měřicí řetězec od naměření aktuálních hodnot, přes jejich před-zpracování a uložení do databáze, následované propojením této databáze s vybraným vizualizačním systémem, kde je zpracovaná vizualizace. V rámci vizualizačního systému a jeho nástaveb jsou dále aplikovány různé statistické a analytické metody poskytující mimo jiné také náhled na možnosti těchto služeb.

Důležitou částí této práce je analýza dostupných vizualizačních systémů, na jejich základě byl vybrán jeden z nich, na kterém je zpracována praktická část této diplomové práce. Popsány, analyzovány a porovnány byly celkem čtyři vizualizační systémy a to Grafana, Node-RED, IBM Bluemix (Watson IoT) a Microsoft Azure (Power Bi). Vizualizační systémy si navzájem v rámci svých možností velmi dobře konkurovaly, ať už šlo o srovnání vizualizační, či analytické stránky. V subjektivním hodnocení nejhůře dopadla cloudová platforma IBM Bluemix a to z důvodu aktuálního stavu, kdy jsou nefunkční některé moduly Watson IoT. Vybrána nakonec byla realizace vizualizačního systému na cloudové platformě IBM Bluemix s využitím služeb k tomu určených. Takto rozhodnuto bylo ze dvou hlavních důvodů: společnost IBM navázala s VŠB-TU Ostrava spolupráci a také zde figuruje fakt, že s touto cloudovou platformou naše fakulta nemá mnoho zkušeností. V rozhodnutí dále figurovala také změněná licenční politika Microsoftu v rámci své cloudové platformy.

Pilotní projekt tedy vznikl na cloudové platformě IBM Bluemix. V rozsahu jeho nasazení bylo použito čtyř základních modulů a služeb. Jedná se o Watson IoT, Cloudant NoSQL DB, Watson Analytics a Node-RED. Tyto služby umožnily splnit pátý bod zadání této diplomové práce - integraci subsystému pro zpracování a analýzu dat. Subsystém je integrován v rámci instance služby Node-RED, kde jsou implementovány jak vlastní, tak i knihovní funkce. Zpracovat a analyzovat data je možné jak z NoSQL Cloudant databáze, tak i aktuálně přichozí. Výsledky jsou poté přenášeny na vizualizační dashboard, jež je vytvořen ve službě Watson IoT. Dále jsou data analyzována také prostřednictvím služby Watson Analytics, kde ačkoliv je služba určená pro podnikovou sféru, je možnost podrobit data pokročilým metodám analýzy. Patří mezi ně datová asociace, vzájemné ovlivňování, či vytváření podmínkových stromů umožňujících predikci hodnot. Výsledky těchto analýz jsou však zavádějící, neboť vstupní datový soubor není o potřebné velikosti a nutné je taktéž vytvořit závislost mezi naměřenými hodnotami pomocí IQRF a například počasím, aby výsledky analýz měly hlubší smysl.

Jako součást vypracovávání této diplomové práce také vznikly návody provázející instalací nezbytných součástí, návrh desky plošných spojů pro připojení snímačů k uzlům IQRF sítě a názorné návody usnadňující implementaci řešení na platformě IBM Bluemix.

Hlavním přínosem této diplomové práce je zpracovaný pilotní projekt na vybraném vizualizačním systému, který usnadní implementaci řešení ostatních uživatelů cloudové platformy IBM Bluemix. Přínos mohl být mnohem větší, ovšem nestalo se tak z důvodu výskytu problémů v průběhu vypracování, kdy bylo nutno celkem 3x začít s implementací nanovo. Situace první nastala v okamžik změny licenční politiky společnosti Microsoft. Další dva problémy se odehrály již na cloudové platformě IBM Bluemix, kdy nejprve skončila půlroční licence a k práci se bylo možné dostat pouze pod podmínkou zpoplatnění služby a zadání údajů o platební kartě a v případě druhém se jednalo o interní chybu při deaktivaci modulu.

V návaznosti na dosavadní výsledky diplomové práce lze na rozšířit IQRF MESH síť o další bezdrátové měřicí moduly enviromentálních veličin. K dispozici je čidlo BMP280 a dva senzory DHT22. Programovou část lze rozšířit o automatizovaný inicializační proces komunikující se službou *mqtt-to-sql*, kdy dojde na oznámení o počtu a typu připojených senzorů a čidel. Služba *mqtt-to-sql* musí být aktuálně manuálně upravována v případě přidání dalšího čidla, či uzlu.

Literatura

- [1] TKOTZ, Klaus a Jiří HANDLÍŘ. *Příručka pro elektrotechnika*. 2. dopl. vyd. Haan-Gruiten: Verlag Europa-Lehrmittel, 2014, dotisk, 623 s. ISBN 978-3-8085-3034-4.
- [2] KREIDL, Marcel *Měření teploty, senzory a měřicí obvody*. 2005, 240 s. Praha: BEN - Technická literatura, ISBN 80-7300-145-4.
- [3] MICRORISC s.r.o. *IQRF Quick start guide*. [online]
Dostupné z: <https://www.iqrf.org/support/download&kat=34&ids=235>
- [4] MICRORISC s.r.o. *IQRF Cloud - Technical Guide* [online]
Dostupné z: <https://iqrf.org/technology/iqmesh/technology/iqrf-cloud>
- [5] MICRORISC s.r.o. *IQRF TR-72D Datasheet* [online]
Dostupné z: <https://www.iqrf.org/products/transceivers/tr-72d>
- [6] MICRORISC s.r.o. *IQRF OS v4.02D User's guide for TR-7xD*. [online]
Dostupné z: <https://www.iqrf.org/support/download&kat=35>
- [7] MICRORISC s.r.o. *DPA Framework Technical Guide v3.00* [online]
Dostupné z: <https://www.iqrf.org/technology/dpa/dpa-sw-documentation>
- [8] MICRORISC s.r.o. *IoT-StarterKit-01 User's guide* [online]
Dostupné z: <https://www.iqrf.org/products/development-tools/development-sets/iot-starterkit-01>
- [9] IBM, Inc. *Create Node-RED device simulator* [online]
Dostupné z: https://console.bluemix.net/docs/services/IoT/nodereddevice_sample.html#creating-and-connecting-a-node-red-device-simulator
- [10] IBM, Inc. *Watson Analytics docs* [online]
Dostupné z: https://watson.analytics.ibmcloud.com/wa_docs/help/#/SS4QC9/com.ibm.solutions.wa_an_overview.2.0.0.doc/wa_tut_overview_for_docs.html
- [11] IBM, Inc. *Watson IoT developer docs* [online]
Dostupné z: https://console.bluemix.net/docs/services/IoT/developer_doc_overview.html#developer_doc_overview
- [12] IBM, Inc. *Node-RED developer docs* [online]
Dostupné z: <https://nodered.org/docs/>
- [13] Dallas Semiconductor, Inc.; Maxim Integrated Products, Inc. *Datasheet for DS18B20* [online]
Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

- [14] Bosch Sensortec, Inc. *Datasheet for BME280* [online]
Dostupné z: https://ae-bst.resource.bosch.com/media/__tech/media/datasheets/BST-BME280_DS001-12.pdf
- [15] Adafruit industries, Inc. *Datasheet for DHT11, DHT22* [online]
Dostupné z: <https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf>
- [16] Jaroslav Straka, www.itbiz.cz *Článek na téma IoT a IBM Bluemix* [online] 19. leden 2017 11:30
Dostupné z: <http://www.itbiz.cz/clanky/iot-a-ibm-bluemix>
- [17] www.zoetrope.io *Článek na téma MQTT a IoT* [online]
Dostupné z: <https://zoetrope.io/tech-blog/brief-practical-introduction-mqtt-protocol-and-its-application-iot/>
- [18] Grafana Labs *Vizualizační systém Grafana* [online]
Dostupné z: <https://grafana.com/grafana>
- [19] Martina Litschmannová, VŠB-TU Ostrava *Úvod do statistiky* [online] 2012, 525 s.
Dostupné z: http://www.cs.vsb.cz/ochodkova/courses/MADI/interaktivni_uvod_do_statistiky.pdf
- [20] MICRORISC s.r.o. *IQRF SPI library for Linux* Github [online]. [cit. 2018-04-22].
Dostupné z: <https://github.com/iqrfsdk/clibspi>
- [21] MICRORISC s.r.o. *Samples of MQ and MQTT communication for IQRF daemon* Github [online]. [cit. 2018-04-22].
Dostupné z: <https://github.com/iqrfsdk/iqrf-daemon-examples>
- [22] Martin Strouhal, MICRORISC s.r.o. *Bridge iqrf mqtt (user manual)* Github [online]. [cit. 2018-04-22].
Dostupné z: [https://github.com/iqrfsdk/jlibiqrf/wiki/Bridge-iqrf-mqtt-\(user-manual\)](https://github.com/iqrfsdk/jlibiqrf/wiki/Bridge-iqrf-mqtt-(user-manual))

A Seznam příloh na CD

(i) **DPS**

[Složka] Obsahuje layouts desek plošných spojů pro potřeby připojení čidel a senzorů k uzlům IQRF sítě.

(ii) **IQRF-IDE**

[Složka] Obsahuje projekt vytvořený v prostředí IQRF IDE 4.0. Přítomny jsou zde zdrojové kódy uživatelského programu pro jednotlivé uzly, konfigurační XML soubory, hardwarové profily.

(iii) **mqtt-to-sql**

[Složka] Obsahuje zdrojové kódy služby *mqtt-to-sql*. Je zde uložen také skript VPN a registrační soubor služby pro operační systém.

(iv) **Node-RED: IBM**

[Složka] Obsahuje exportované pracovní plochy s uživatelským programem v rámci instance Node-RED běžící na platformě IBM. Přítomen je zde soubor s požadavky na přítomné knihovny.

(v) **Node-RED: RPi**

[Složka] Obsahuje exportované pracovní plochy s uživatelským programem v rámci instance Node-RED běžící na platformě Raspberry Pi. Přítomen je zde soubor s požadavky na přítomné knihovny.

(vi) **PDF**

[Složka] Obsahuje datasheety, knihy, technické příručky, či návody.

(vii) **Vypracovani-LaTeX**

[Složka] Obsahuje vypracování textu závěrečné práce v prostředí TexStudio. Dále použité obrázky, soubory LibreOffice Draw s kresbami a diagramy.

(viii) **IQRFData_rc112.csv**

[Soubor] Jedná se o data exportovaná ze serveru rc112.vsb.cz, jejichž obsahem jsou naměřená data z IQRF MESH sítě.

B Tabulka s porovnáním jednotlivých modulů transceiverů

Type		TR-72D	TR-75D	TR-76D	TR-77D	TR-78D	
Mounting		SIM	Vertical	SMT	SMT	RJ-45 connector	
Number of pins		8	12	18	18	8	
I/O		6	10	12	12	4	
MCU PIC		16LF1938					
MCU internal memories	Flash	16 K x 14 b					
	RAM	1024 B					
	EEPROM	256 B					
External serial EEPROM		32 KB					
RF IC		SPIRIT1					
RF band [MHz]		868/916/433	868/916	868/916/433	868	868/916	
RF bit rate [kb/s]		19.8					
RF power ¹ [mW]		8					
RF range ^{2 3}		500 m 1100 m ³	500 m 1100 m ³	500 m 1100 m ³	230 m 450 m ³	500 m 1100 m ³	
Typical supply current	Deep sleep	1.7 μ A	56 nA	56 nA	56 nA	TBD	
	Sleep	2.3 μ A	610 nA	610 nA	610 nA	TBD	
	RX	STD	12.1 mA	11.8 mA	11.8 mA	11.8 mA	TBD
		LP	260 μ A	250 μ A	250 μ A	250 μ A	TBD
		XLP	18.5 μ A	16.3 μ A	16.3 μ A	16.3 μ A	TBD
	TX	8.3 - 21.5 mA					
Temperature sensor IC ⁴		MCP9808	-	-	-	MCP9808	
LDO voltage regulator		MCP1700	-	-	-	MCP1700	
Supply voltage [V]		3.1 - 5.3	3.0 - 3.4	3.0 - 3.4	3.0 - 3.4	3.1 - 5.3	
LEDs		2	2	-	2	2	
A/D inputs		2	3	3	3	1	
SAW filter		-	-	-	Yes	-	
RF shielding		Yes					
Coated, UV resistive		-	-	-	-	Yes	
Dimensions [mm]	Standard	25.1 x 14.9	-	15.2 x 14.9	25.1 x 14.9	-	
	TR-xxxA ³	31.8 x 14.9	27.5 x 14.9	23.3 x 14.9	31.8 x 14.9	29 x 40 x 52	
Latest OS version		4.02D	4.02D	4.02D	4.02D	TBD	
Status		<i>Standard</i>	<i>New</i>	<i>Standard</i>	<i>Preliminary</i>	<i>Preliminary</i>	
Type		TR-72D	TR-75D	TR-76D	TR-77D	TR-78D	

Obrázek 63: Tabulka transceiverů s porovnáním