

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Smart senzory pre zber dát
Diplomová práca

2015

Bc. Jozef Mocnej

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Smart senzory pre zber dát
Diplomová práca

Študijný program: Hospodárska informatika
Študijný odbor: 9.2.10 Hospodárska informatika
Školiace pracovisko: Katedra kybernetiky a umelej inteligencie
Školiteľ: prof. Ing. Iveta Zolotová, CSc.
Konzultant: Ing. Tomáš Lojka

2015 Košice

Bc. Jozef Mocnej

Abstrakt v SJ

Cieľom tejto diplomovej práce je poukázať na vývoj v oblasti zberu a spracovania dát v priemysle. V teoretickej časti sú analyzované najnovšie trendy v informačných technológiách, sú popísané ich hlavné prínosy a zároveň výzvy. Ide o pojmy ako Priemysel 4.0, Internet vecí, či smart továrne. Takisto je rozobratý komunikačný kanál Internetu vecí a sú definované najpoužívanejšie protokoly spolu s ich porovnaním výhod a nevýhod. V praktickej časti je potom vytvorený návrh architektúry reagujúci na tieto priemyselné trendy s hlavným dôrazom na smart senzor ako prostriedok generovania dát s informačnou hodnotou. Pri realizácii sú popísané konkrétne vykonané kroky aj s odôvodnením ako sa k tomu dospelo. V závere sú zhrnuté nadobudnuté poznatky a uvedené možnosti pokračovania vo vývoji.

Kľúčové slova v SJ

Internet vecí, Priemysel 4.0, Smart senzory, MQTT protokol, Zber dát

Abstrakt v AJ

The aim of this diploma thesis is to point out the evolution of data acquisition and control in the Industry. The theoretical part provides an overview of the latest trends in information technology and describes their main benefits but also challenges. Topics like Industry 4.0, Internet of Things and Smart factories are introduced there. Furthermore, the communication channel for Internet of Things is explored and the most used protocols are defined together with the comparison of their advantages and disadvantages. In the practical part, the architectural design is created as a response to these Industry trends with the main focus on smart sensor as a tool for generating data with information value. Specific steps are then described in the realization part. The end of this study summarizes acquired knowledge and outlines further possibilities for a continuation in development.

Kľúčové slova v AJ

Internet of Things, Industry 4.0, Smart sensors, MQTT protocol, Data acquisition

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra kybernetiky a umelej inteligencie

ZADANIE
DIPLOMOVEJ PRÁCE

Študijný odbor: **9.2.10 Hospodárska informatika**

Študijný program: **Hospodárska informatika**

Názov práce:

Smart senzory pre zber dát
Smart sensor for data acquisition

Študent: **Bc. Jozef Mocnej**
Školiteľ: **prof. Ing. Iveta Zolotová, CSc.**
Školiace pracovisko: **Katedra kybernetiky a umelej inteligencie**
Konzultant práce: **Ing. Tomáš Lojka**
Pracovisko konzultanta: **Katedra kybernetiky a umelej inteligencie**

Pokyny na vypracovanie diplomovej práce:

1. Popísať najnovšie trendy v oblasti zberu dát a monitorovania a ich spojitosť s IaRS.
2. Popísať protokoly pre prenos dát s nimi súvisiace spôsoby predspracovanie a spracovania dát .
3. Analyzovať, navrhnuť a popísať architektúru pre zber dát a riadenie na základe smart senzorov.
4. Vytvorenie aplikácie podľa návrhu. Analyzovať a vyhodnotiť výsledky získané z realizovaného riešenia.
5. Zhrnúť získané poznatky a zhodnotiť dosiahnuté výsledky.
6. Vypracovať dokumentáciu podľa pokynov vedúceho práce (napr. hlavná časť 40 strán a viac, prílohy podľa charakteru práce, tlačaná forma v nerozoberateľnej väzbe, elektronická forma hlavnej časti a príloh).

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 30.04.2015
Dátum zadania diplomovej práce: 31.10.2014


.....
prof. Ing. Peter Sinčák, CSc.
vedúci garantujúceho pracoviska




.....
prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 12. mája 2015

.....

vlastnoručný podpis

Podakovanie

Na tomto mieste by som rád vyjadril podakovanie svojej školiteľke pani prof. Ing. Ivete Zolotovej, CSc. za jej profesionálny prístup a vedenie. Ďalej by som chcel poďakovať svojmu konzultantovi Ing. Tomášovi Lojkovi za jeho neustálu ochotu a nápomocné rady, ktoré ma usmernili pri písaní tejto diplomovej práce. A v neposlednom rade patrí vďaka aj mojej priateľke za podporu.

Obsah

Zoznam obrázkov	8
Zoznam tabuliek	9
Zoznam symbolov a skratiek	10
Úvod	12
1. Najnovšie trendy v priemysle	13
1.1. Priemysel 4.0	13
1.1.1. Kyber-fyzikálne systémy	14
1.1.2. Vlastnosti Priemyslu 4.0	15
1.1.3. Kompatibilita CPS s Priemyslom 4.0	16
1.2. Internet vecí	18
1.3. Smart továrne	19
2. Protokoly pre prenos dát	22
2.1. IoT protokoly	22
2.1.1. MQTT	23
2.1.2. CoAP	25
2.1.3. XMPP	25
2.1.4. MQTT vs CoAP vs XMPP	26
2.2. OPC UA	27
3. Analýza a návrh architektúry pre zber dát na základe smart senzorov	30
4. Vytvorenie aplikácie podľa návrhu	35
5. Záverečné zhrnutie	48
Záver	49
Zoznam použitej literatúry	50
Prílohy	52

Zoznam obrázkov

Obr. 1 Vývoj priemyslu.....	14
Obr. 2 Priame systémové rozšírenie	17
Obr. 3 Systémové rozšírenie mikrokontrolérovou doskou	17
Obr. 4 Rozšírenie smart senzormi a aktuátormi	18
Obr. 5 Tri technologické schopnosti IIoT	19
Obr. 6 Kategorizácia IoT protokolov	22
Obr. 7 Publish/Subscribe komunikácia	24
Obr. 8 Metóda IP Multicast	25
Obr. 9 Sieť senzorov	31
Obr. 10 Komunikácia senzorov s brokerom	32
Obr. 11 Ukladanie dát z brokera na server	33
Obr. 12 Konečný návrh architektúry	34
Obr. 13 Osi gyroskopu v mobile.....	36
Obr. 14 Základná obrazovka Android aplikácie	37
Obr. 15 Pripojenie na broker	38
Obr. 16 Hodnoty osí gyroskopu v posielaných správach	39
Obr. 17 Doručené správy na brokeri.....	39
Obr. 18 Používaná databáza na serveri.....	41
Obr. 19 Windows služba.....	42
Obr. 20 Uložené hodnoty v databáze	42
Obr. 21 Základná obrazovka webovej aplikácie	43
Obr. 22 Optimálne hodnoty senzora	44
Obr. 23 Obrazovka na zadávanie dĺžky trvania procesu učenia	44
Obr. 24 Odpočítavanie do ukončenia procesu učenia	44
Obr. 25 Neštandardné situácie systému aj s popisom riešenia.....	45
Obr. 26 Chybový stav s popisom riešenia	45
Obr. 27 Chybový stav bez existujúceho popisu riešenia	45
Obr. 28 Formulár na zadanie riešenia chybového stavu	46
Obr. 29 Systém bez chýb	46

Zoznam tabuliek

Tab. 1 Porovnanie protokolov	27
------------------------------------	----

Zoznam symbolov a skratiek

AMQP	A dvanced M essage Q ueuing P rotocol
CAN	C ontroller A rea N etwork
CoAP	C onstrained A pplication P rotocol
COM	C omponent O bject M odel
CPS	C yber P hysical S ystems
D2D	D evice- T o- D evice
D2S	D evice- T o- S erver
DCOM	D istributed C omponent M odel
DDS	D ata D istribution S ervice
IEC	I nternational E lectrotechnical C ommission
IETF	I nternet E ngineering T ask F orce
IIoT	I ndustrial I nternet o f T hings
IoT	I nternet o f T hings
IP	I nternet P rotocol
IT	I nformation T echnology
M2M	M achine- T o- M achine
MQTT	M essage Q ueuing T elemetry T ransport
OLE	O bject L inking and E MBEDDING
OPC	O LE for P rocess C ontrol
OPC UA	O LE for P rocess C ontrol U nified A rchitecture
OT	O perational T echnology
PLC	P rogrammable L ogic C ontroller
QoS	Q uality o f S ervice
S2S	S erver- T o- S erver

SCADA	S upervisory C ontrol A nd D ata A cquisition
TCP	T ransmission C ontrol P rotocol
UART	U niversal A synchronous R eceiver/ T ransmitter
UDP	U ser D atagram P rotocol
WLAN	W ireless L ocal A rea N etwork
XMPP	E xtensible M essaging and P resence P rotocol

Úvod

Svet informačných technológií je extrémne dynamický. Nové technológie a možnosti prichádzajú každý deň, zatiaľ čo tie staré sa pomaly vytrácajú. Lepšia dostupnosť hardvéru, ktorý je stále výkonnejší, nové softvérové riešenia, ktoré sú jednoduchšie realizovateľné, to je len ukážka toho, čo nám pokrok prináša. S príchodom zmien však vznikajú aj výzvy, ako tieto trendy čo najlepšie využiť.

V dnešnej dobe sme svedkami exponenciálneho vzrastu zariadení pripojených na Internet. Hovorí sa dokonca o novej priemyselnej revolúcii, ktorá zmení náš pohľad na doterajšie využitie informačných technológií. K Internetu ľudí postupne pribudne aj Internet vecí. Práve tieto malé zariadenia generujúce enormné množstvo dát sú potenciálnou bránou k novej ére využitia Internetu. K dosiahnutiu tohto stavu je však ešte potrebné prekonať niekoľko prekážok.

Medzi najdiskutovanejšie témy patrí otázka, ako tieto zariadenia spoločne prepojiť a umožniť im vzájomnú komunikáciu. Ako riešenie sa ukazujú komunikačné protokoly špeciálne navrhnuté na zariadenia s obmedzenými výpočtovými zdrojmi a na siete s rôznou kvalitou prenosu dát. Tieto protokoly chápu nároky Internetu vecí a snažia sa podľa toho prispôbiť.

Druhou a možno ešte diskutovanejšou otázkou je, čo všetko nám vzájomná komunikácia medzi zariadeniami vie priniesť. Tu sa dostáva k slovu kreativita pri vytváraní nápadov, pretože sa vstupuje do doteraz neprebádaných vôd. Možnosti sú prakticky neobmedzené a je len na ľuďoch, čo všetko z potenciálu Internetu vecí dokážu vyťažiť.

V tejto práci bude nastolená problematika moderného zberu a spracovania dát v súlade s najnovšími trendmi. Cieľom je vytvoriť smart senzor, ktorý namiesto obyčajných dát posielaných v pravidelných intervaloch bude odosielať hotové informácie a to vtedy, keď pre nás budú mať význam. Zároveň bude potrebné vyriešiť, ako tieto informácie zo smart senzorov zbierať a uložiť na ďalšie analýzy. Takisto je zámerom vytvoriť ukážku systému schopného adaptovať sa na prostredie a využiť to vo svoj prospech. Pôjde teda o kompletný návrh a realizáciu architektúry od generovania dát až po ich použitie. V závere potom budú zhodnotené nadobudnuté poznatky a výsledky štúdie.

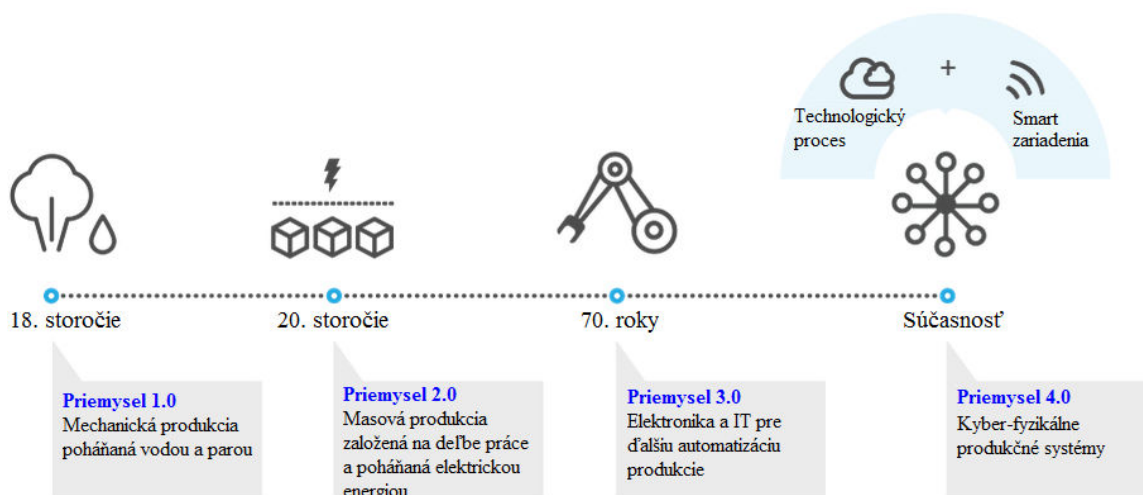
1. Najnovšie trendy v priemysle

Informačné technológie napredujú závratnou rýchlosťou, na Internet sa pripája stále väčšie množstvo zariadení, ktoré produkujú ešte väčšie množstvo dát a kto sa chce adaptovať, musí týmto trendom porozumieť. V tejto kapitole si preto bližšie predstavíme najperspektívnejšie vízie budúcnosti v oblasti priemyslu.

1.1. Priemysel 4.0

Priemysel v zmysle profesionálnej výroby existuje po tisícročia, už od čias výstavby prvých miest. Bola to však prvá priemyselná revolúcia, ktorá transformovala manuálnu prácu na prácu strojov a dala priemyslu podobu, akú poznáme dnes. Stalo sa tak príchodom zariadení poháňaných vodou a parou v období 1760 až 1840. Začiatkom dvadsiateho storočia prišla druhá revolúcia a to vďaka uvedeniu masovej produkcie založenej na delbe práce poháňanej elektrickou energiou. Boli to časy osobností ako Henry Ford či bratia Baťovci, ktorí dokázali svoje produkty vyrábať v množstvách predtým nemysliteľných. V sedemdesiatych rokoch dvadsiateho storočia prišli na pracovisko počítače umožňujúce rýchle výpočty a merania, čo spôsobilo tretiu priemyselnú revolúciu. A je zrejmé, že IT pokračovalo v transformácii výrobných procesov až dodnes. [1]

V súčasnosti však znova zažívame niečo veľkolepé. Požiadavky na priemyselnú výrobu sa menia, zvyšuje sa dopyt po produktoch, ktoré presne spĺňajú nároky zákazníka a zároveň spoločnosti čelia rastúcemu tlaku na výrobu v konkurencieschopných cenách. Na prispôbenie sa týmto novým podmienkam je preto potrebné posunúť informačné technológie na ešte vyššiu úroveň ako kedykoľvek predtým, vytvorením nových možností v procesoch produkcie. Tento stav je možné dosiahnuť prepojením reálneho a virtuálneho sveta, teda vzájomnou kombináciou komunikačných, IT, dátových a fyzikálnych elementov. Hovoríme o takzvaných kyber-fyzikálnych produkčných systémoch, ktoré sú kľúčom k priemyselnej revolúcii 4.0. [2]



Obr. 1 Vývoj priemyslu

1.1.1. Kyber-fyzikálne systémy

V dnešnej dobe môžeme hovoriť o štvrtej priemyselnej revolúcii v spojení kyber-fyzikálnymi systémami (cyber-physical systems – skratka CPS). Tieto systémy sú priemyselné automatizované systémy umožňujúce prepojenie prevádzky fyzikálneho sveta s výpočtovými a komunikačnými štruktúrami. Oproti tradičným vnoreným systémom (embedded systems), ktoré sú nadizajnované ako samostatné zariadenia, zámerom CPS je vzájomné prepojenie viacerých zariadení. CPS idú s trendom mať informáciu vždy na dosah ruky, čo je v dnešnom prepojenom svete priam nevyhnutné. [3]

Embedded systémy, napríklad ako smartfóny, autá či domáce spotrebiče, sú neoddeliteľnou súčasťou moderného života. Napriek tomu však iba zopár z nich dokážeme ovládať na diaľku. Bolo by ale veľmi vhodné, ak by sme napríklad mohli cestou domov zapnúť kúrenie, takže dom by bol v čase príchodu dostatočne vyhriaty. Alebo ak by kávovar začal variť kávu, kým sme ešte v posteli, ušetrilo by nám to ráno cenné minúty strávené inak čakaním. Tento vzdialený prístup spracovania dát by navyše mohol byť použitý na údržbu týchto systémov. Napríklad informácie zo vzdialenej diagnostiky by mohli pomôcť servisným technikom priniesť správne náradie a vymeniť pokazenú súčiastku. Prípadne systémy by si mohli danú súčiastku aj automaticky objednať a to s pomocou vhodnej komunikačnej infraštruktúry. Jednoducho, oblastí využitia CPS je mnoho. Spomenúť sa dá napríklad inteligentná domácnosť, medicínske vybavenie, bezpečnosť na cestách v podobe rôznych asistenčných systémov v automobiloch, ale aj podnikové informačné a riadiace systémy. [3]

Tradičný embedded systém pozostáva z kontrolnej jednotky a jedného alebo viac mikrokontrolérov, ktoré ovládajú senzory a aktuátory umožňujúce interakciu s reálnym svetom. Na vytvorenie CPS z embedded systému je potrebné definovať komunikačné rozhranie, cez ktoré bude

prebiehať výmena dát, či už medzi systémami navzájom alebo napríklad medzi systémom a cloudom. Práve výmena dát je najdôležitejšia funkcionálna CPS, nakoľko dáta tak môžu byť spájané a vyhodnocované centrálné. Inými slovami, CPS je embedded systém, ktorý umožňuje posilať a prijímať dáta cez sieť. CPS pripojený na Internet sa často označuje aj ako „Internet vecí“, viď kapitola 1.2. [3]

1.1.2. Vlastnosti Priemyslu 4.0

Ako už bolo povedané, priemyselná revolúcia 4.0 je založená na použití kyber-fyzikálnych systémov. Integrácia počítačových technológií, ktoré pripájajú produkty na Internet, umožňuje dosiahnutie nových inovatívnych služieb, popri inom napríklad vzdialenú diagnostiku, údržbu, prevádzku atď., cenovo efektívnym a účinným spôsobom. Okrem toho pomáha realizovať nové biznis modely, prevádzkové koncepty a celkovo sa zamerať viac na zákazníka a jeho individuálne potreby. Cieľom Priemyslu 4.0 je rozvinúť digitálne továrne, ktoré majú byť charakterizované týmito vlastnosťami: [3]

- **Smart sieťovanie**

Automatizované systémy a zariadenia, interné logistické systémy a prevádzkové zásobovanie budú dôkladne prepojené s technológiami ako drôtové a bezdrôtové komunikačné služby, smart aktuátory a senzory či telekomunikačné technológie. To im umožní priamy prístup k procesom a službám na vyššej úrovni. Takisto to otvára dvere na úplne nové inovácie s pridanou hodnotou a na biznis modely, ktoré podporujú optimálne využitie zdrojov a smart prevádzku. [3]

- **Mobilita**

Mobilné zariadenia, ako sú smartfóny a tablety, zasiahnu do priemyselnej automatizácie. Poskytnú totiž časovo a priestorovo nezávislý prístup k procesom a službám automatizovaných systémov, čím vznikne nová dimenzia v diagnostike, údržbe a prevádzke týchto systémov. [3]

- **Flexibilita**

Priemysel 4.0 umožní vysokú flexibilitu rovnako vo vývoji, diagnostike a údržbe ako aj v prevádzke automatizovaných systémov. Pri vývoji týchto systémov bude možné zvoliť si najlepšiu ponuku z veľkej skupiny dodávateľov komponentov, modulov a služieb. Diagnóza sa zasa zautomatizuje využitím prístupu k Big Data. Informácie budú môcť byť získané na vyžiadanie, inteligentne použité a spojené tak, aby sa mohla dosiahnuť automatizovaná diagnóza. Náhradné diely potom bude možné objednať automaticky u najlacnejších výrobcov, čím sa zmenší počet pracovníkov potrebných na vyriešenie problému. [3]

- **Integrácia zákazníkov**

S Priemyslom 4.0 bude možné upravovať produkty špecifickým a individuálnym požiadavkám zákazníkov. Automatizované systémy dvadsiateho prvého storočia by sa mali vedieť prispôbiť potrebám a schopnostiam používateľov všetkých vekových skupín. Napríklad moderné predajné automaty na lístky budú poskytovať viaceré spôsoby prevádzky, aby umožnili ich použitie aj ľuďom s rôznym zdravotným postihnutím. Automatizované systémy budú podporovať ľudí vo všetkých situáciách a asistovať im v každej fáze života, aby tak mohli zostať zdraví a mobilní. [3]

- **Nové inovatívne obchodné modely**

Výroba v budúcnosti bude distribuovaná a flexibilná. Vzniknú nové vývojové procesy, infraštruktúry a služby. Produkty sa stanú modulárnymi a konfigurovateľnými, takže sa budú vedieť prispôbiť špecifickým požiadavkám. [3]

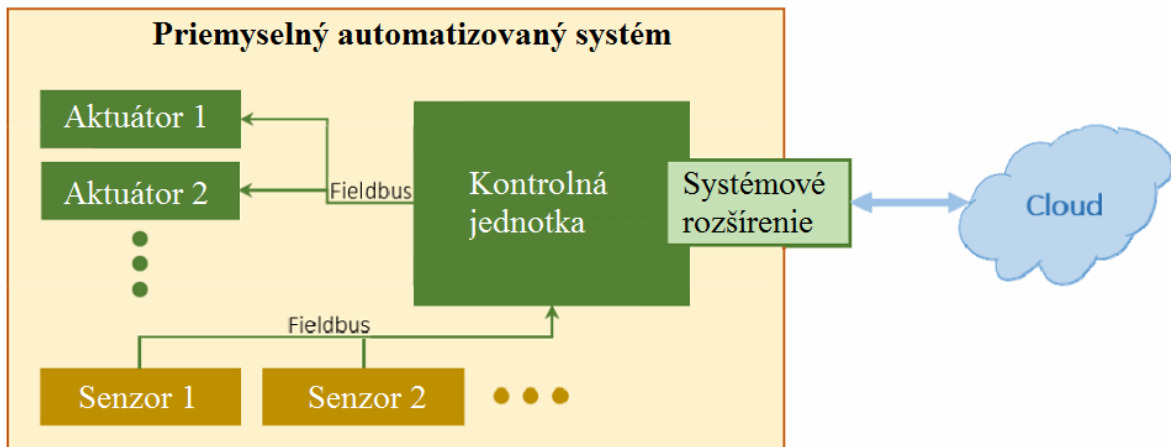
Priemysel 4.0 prináša mnoho výziev, ktoré je potrebné zdolať ďalším výskumom. Napríklad ako stanoviť spoľahlivosť a bezpečnosť produktov, ktorých vývoj je distribuovaný, ako ochrániť dáta, alebo ako zaručiť nedotknuteľnosť know-how a súkromia. Za týmto účelom je potrebné vytvoriť nové koncepty a technológie, ktoré umožnia dôveryhodnú spoluprácu medzi mnohými skupinami. Okrem toho musia byť predefinované aj viaceré etické, právne a sociálne otázky. [3]

1.1.3. Kompatibilita CPS s Priemyslom 4.0

Definovaním komunikačného rozhrania a pripojením na Internet alebo podobnú sieť sa z embedded systému stáva CPS. Na dosiahnutie tohto stavu je v kontexte vývoja možné použiť jeden z nasledujúcich prístupov: [3]

- **Priame systémové rozšírenie**

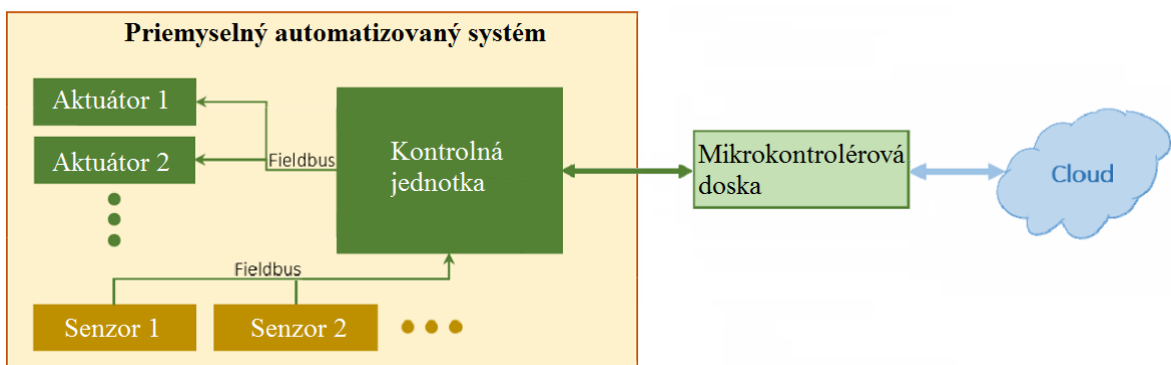
V tejto individualizovanej variante riešenia je embedded systém rozšírený o komunikačné rozhranie umožňujúce prístup na Internet a jeho softvér sa upraví tak, aby vedel komunikovať cez Internet, napríklad s cloudom. V tomto prípade všetky signály senzorov musia byť posielané do cloudu cez kontrolnú jednotku a aktuátory sú ovládané cez implementované metódy. [3]



Obr. 2 Priame systémové rozšírenie

- **Systémové rozšírenie mikrokontrolérovou doskou**

V tomto prípade je do riešenia zahrnutá mikrokontrolérová doska obsahujúca rôzne komunikačné rozhrania ako CAN, UART, WLAN, Ethernet atď. Tá je pripojená k embedded systému a preberá komunikáciu so sieťou. To si však vyžaduje jednotné rozhrania, cez ktoré môže byť doska k embedded systému pripojená a zároveň softvér dosky musí byť pre každý systém samostatne upravený. Nemusí sa však zakaždým prepisovať celý kód, ale stačí prepracovať mapovanie, takže v konečnom dôsledku je relatívne jednoduché preniesť túto variantu na rôzne systémy. [3]

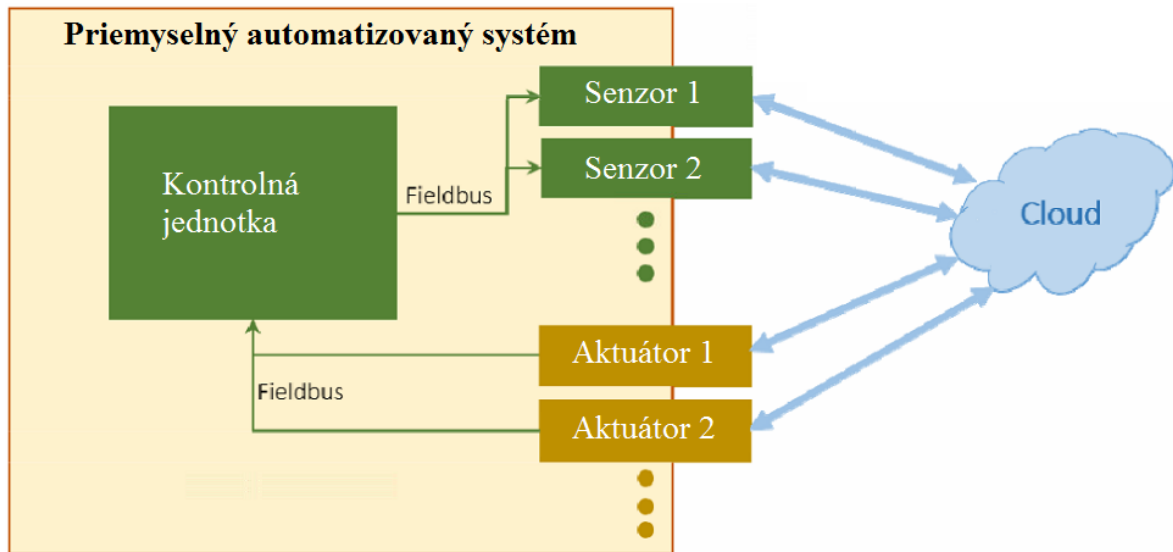


Obr. 3 Systémové rozšírenie mikrokontrolérovou doskou

- **Rozšírenie smart senzormi a aktuátormi**

Pri tradičných embedded systémoch sú senzory a aktuátory prepojené s kontrolnou jednotkou prevádzkovými zbernicami (fieldbuses). V takýchto systémoch spracováva signál kontrolná jednotka. Naproti tomu smart senzory by prevzali spracovávanie signálu na seba a smart aktuátory by nezávisle kontrolovali svoj status a v prípade potreby ho opravili. Takéto smart senzory a aktuátory by prenášali dáta nielen do kontrolnej jednotky, ale aj do cloudu, kde by sa mohli centrálnie spracovať. Pri tomto

riešení je však potrebné počítať s veľkým prenosom dát a netreba ani podceniť cenové náklady takýchto smart senzorov a aktuátorov. [3]



Obr. 4 Rozšírenie smart senzormi a aktuátormi

1.2. Internet vecí

Internet vecí (Internet of Things - IoT) je vízia prepojiť fyzikálne objekty s virtuálnym svetom kombináciou ich schopnosti vnímať, zbierať dáta, spracovať ich, analyzovať a distribuovať v masívnej miere. V konečnom dôsledku tak všetko bude prepojené so všetkým a to kedykoľvek a kdekoľvek. Táto konvergencia virtuálneho a fyzikálneho sveta môže radikálne vylepšiť používateľskú skúsenosť a zároveň priniesť nové, doposiaľ nepoznané možnosti. [4]

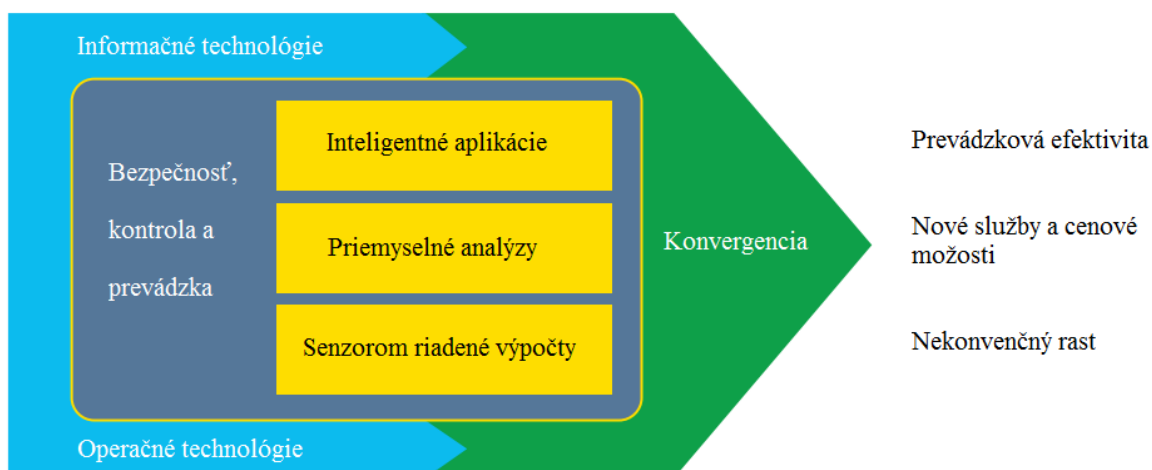
Termín „Internet vecí“ bol prvýkrát zdokumentovaný v roku 1999 a rozumie sa pod tým pokročilá konektivita zariadení, systémov a služieb, ktorá ide za machine-to-machine (M2M) komunikáciu a pokrýva široké spektrum protokolov, domén a aplikácií. Pod pojmom „vecí“ sa dajú predstaviť najrôznejšie zariadenia so všetkých možných oblastí, od smart domácich spotrebičov cez automobily so zabudovanými senzormi až po medicínske monitorovacie implantáty. Oblasť využitia je skrátka naozaj široká. [4]

Potenciál Internetu vecí si spoločnosti uvedomili aj v spojení s Priemyslom 4.0. Hoci pôvodne bol Priemyselný Internet vecí (Industrial Internet of Things - IIoT) zamýšľaný ako cesta k zlepšeniu prevádzkovej efektivity, ukazuje sa, že firmy môžu výborne profitovať aj v spojení IIoT ako nástroja na hľadanie rastu v neočakávaných príležitostiach. V každom prípade, Priemyselný Internet vecí je výrazný trend s nezanedbateľným vplyvom na globálnu ekonomiku. [4]

Na využitie plných výhod IIoT však spoločnosti budú musieť zvládnuť tri technologické schopnosti: [4]

1. senzorovo riadené výpočty,
2. priemyselné analýzy,
3. aplikácie inteligentných strojov.

Spojením týchto troch schopností sa vytvorí náhrada za dnešné oddelené svety informačných technológií (IT) a operačných technológií (OT). Infraštruktúra, ktorá sa vyvíjala oddelene a špecificky podľa obchodníka, sa postupne nahradí za platformy schopné spolupráce. Tým vznikne nová generácia inteligentných zariadení a otvoria sa nové možnosti na rast. [4]



Obr. 5 Tri technologické schopnosti IIoT

1.3. Smart továrne

Priemysel 4.0 a IoT ovplyvnili budúcnosť továrni výrazným spôsobom. Na vytvorenie smart továrne, kde funguje vysoko flexibilná a adaptívna priemyselná automatizácia, klasický koncept hierarchickej štruktúry IARS nebude stačiť. Namiesto toho sa k slovu dostanú kyber-fyzikálne systémy, ktoré svojou decentralizovanou inteligenciou prepoja senzory a aktuátory s Internetovými zdrojmi. Tým vznikne možnosť samo-optimalizácie a rekonfigurácie, čo zaručí vysokú schopnosť adaptácie na meniace sa podmienky. Smart továrne budú mať v Priemysle 4.0 tieto charakteristiky: [5]

- **Masová úprava na zákazku**

Produkčné procesy budú spĺňať rôzne podmienky na produkčné objednávky. To umožní vytvárať na produktoch individuálne úpravy a zmeny na poslednú chvíľu. Takisto bude možné vyrábať v malých množstvách a stále pritom dosahovať zisk. [5]

- **Flexibilita**

Inteligentné produkčné procesy a samo-konfigurácia budú zvažovať rozličné aspekty ako čas, kvalitu, cenu či ekologické dopady. [5]

- **Optimalizované robenie rozhodnutí**

Robiť správne rozhodnutia v danom okamihu je kľúčom k úspechu na trhu. IoT poskytne transparentnosť takmer v reálnom čase (napríklad zistenie stavu produkcie), takže umožní optimalizáciu naprieč všetkými stránkami oblasti výroby a tým zlepši továrenskú efektívnosť. [5]

- **Nové plánovacie metódy pre továrne**

S lepšou optimalizáciou procesov na rôznych úrovniach v reálnom čase a od prípadu k prípadu bude možné efektívnejšie plánovať chod celej továrne. [5]

- **Vytváranie hodnôt z big data**

Analýzy veľkého množstva dát vygenerovaných IoT zariadeniami prinesú nové vylepšenia a hodnoty. Napríklad bude možné porozumieť správaniu strojov v rôznych okamihoch a potom poskytnúť lepšiu údržbu či vylepšiť ich efektívnosť. [5]

- **Vytváranie nových služieb**

IoT otvorí nové cesty vytvárania služieb a hodnôt pre zákazníkov a to pred kúpou ako aj po nej. [5]

- **Vzdialené monitorovanie**

IoT technológia umožní zahrnúť tretie strany (napr. dodávateľov) do monitorovania, prevádzky a údržby, vďaka čomu budú môcť prísť s novými službami šitými na mieru. [5]

- **Automatizácia a zmena postavenia človeka**

Produkčnú prevádzku bude možné optimalizovať s minimom zásahov od človeka. To prispeje k lepšej efektivite a zníženiu chýb a premrhanej energie či iných zdrojov. [5]

- **Aktívna údržba**

Systémy na monitorovanie prevádzky a zbieranie dát v reálnom čase budú mať pozitívny dopad na zlepšenie aktívnej údržby. Ak budú senzormi merané hodnoty mimo normy, vykonajú sa preventívne kroky k zabráneniu nehody. [5]

- **Pripojené dodávateľské reťazce**

IoT pomôže výrobcovi získať hlbšie pochopenie informácií z dodávateľských reťazcov, ktoré budú dodávané v reálnom čase. Pripojením dodávateľom k dátam zo strojov a zariadení budú môcť všetky strany lepšie pochopiť vzájomne závislosti, tok materiálu a časy výrobných cyklov. [5]

- **Manažment energií**

Na zefektívnenie spotreby energie je potrebné mať prehľad o jej spotrebe na úrovni produkcie a strojov. IoT bude vedieť poskytovať dáta v reálnom čase a vykonávať rozhodnutia založené na schopnostiach strojov a v spolupráci s externými službami. [5]

Na základe týchto charakteristík potom továrne budú vedieť nielen optimalizovať svoj chod, ale aj vytvoriť inteligentné produkty, ktoré budú schopné: [4]

- Iniciovať úlohy a komunikovať s inými zariadeniami.
- Prispôbiť svoje používateľské rozhranie, odporúčania a pohyby na splnenie zákazníckych preferencií.
- Posilniť svoje funkcionality cez vylepšenie softvéru a tým poskytnúť zákazníkovi ďalšie benefity.
- Naučiť sa ako znížiť prevádzkové náklady.
- Optimalizovať svoj prínos a produktivitu.
- Predísť haváriám a poruchám počas prevádzky.
- Vykonať opatrenia v neistých alebo nepriaznivých podmienkach.

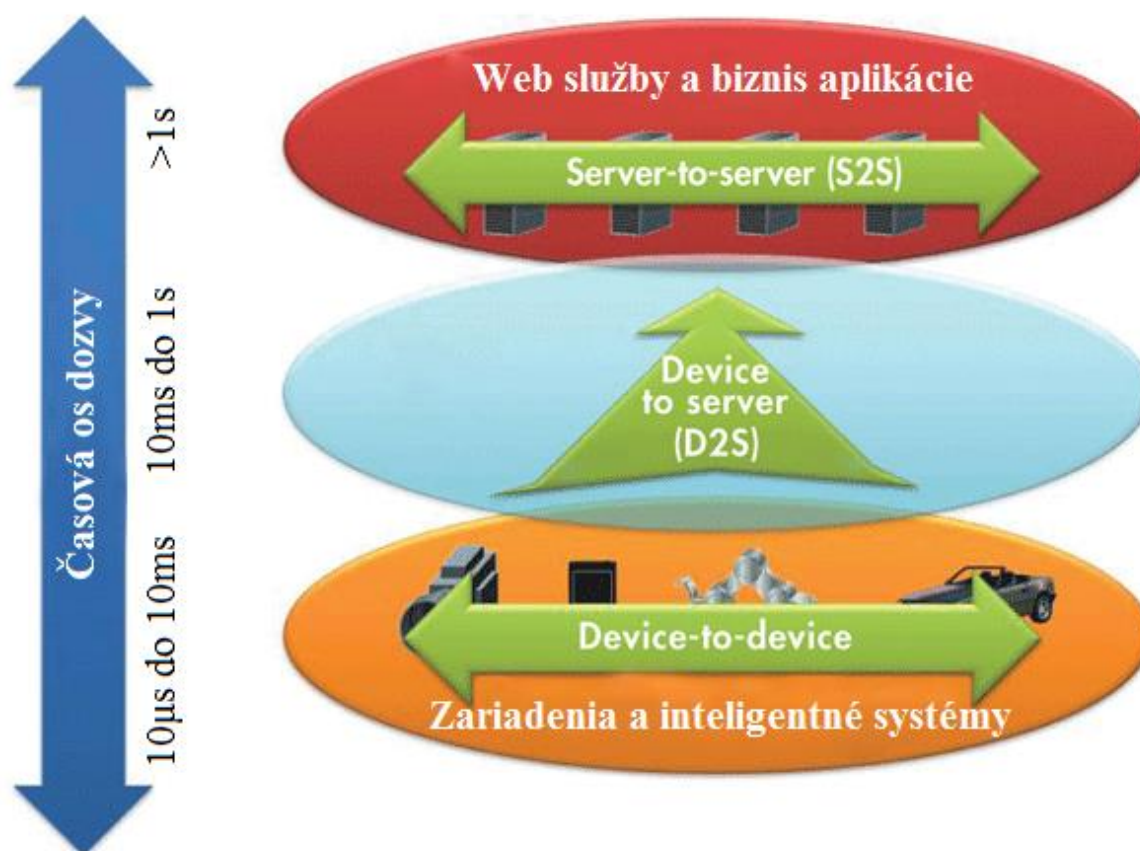
Je dôležité podotknúť, že IoT sa veľmi rýchlo vyvíja a tento progres si uvedomujú aj spoločnosti, vďaka čomu do oblasti investujú značné prostriedky. Niektoré z vyššie popísaných predstáv už preto v určitej miere existujú, niektoré sú na dosah ruky a pri niektorých predstavách je ešte potrebné vykonať ďalší výskum. V každom prípade je isté, že všetky tieto zmeny prídu skôr, než sa nazdáme.

2. Protokoly pre prenos dát

Internet vecí a Priemysel 4.0 so sebou prinášajú mnoho príležitosti, ale zároveň aj mnoho výziev. Jednou z nich je aj otázka, ako všetky zariadenia vzájomne prepojiť a umožniť im komunikáciu zmysluplným spôsobom. V tejto kapitole sa preto bližšie pozrieme na najpoužívanejšie protokoly pre prenos dát a vytvoríme si jasnú klasifikáciu, ktoré protokoly sú na čo vhodné a v čom sa vzájomne odlišujú.

2.1. IoT protokoly

Dnešný Internet podporuje stovky protokolov a s príchodom IoT ich bude o ďalšie stovky viac. Pre ľahšiu orientáciu je preto vhodné zatriediť protokoly do kategórií podľa ich oblasti využitia. V jednoduchosti by sme ich mohli definovať ako protokoly využívajúce sa na komunikáciu medzi zariadeniami (device-to-device – D2D), protokoly využívajúce sa na posielanie pozbieraných dát zo zariadenia na server (D2S) a protokoly využívajúce sa na zdieľanie dát medzi servermi (S2S). Samozrejme v skutočnosti to nie je také zrejmé a mnoho protokolov kooperuje medzi týmito triedami, ale pre potreby porozumenia je táto schéma dostačujúca. [6]



Obr. 6 Kategorizácia IoT protokolov

Z tohto pohľadu potom môžeme najpoužívanejšie protokoly rozdeliť do tried nasledovne: [6]

Server-to-server

- AMQP

Device-to-server

- MQTT
- CoAP
- XMPP

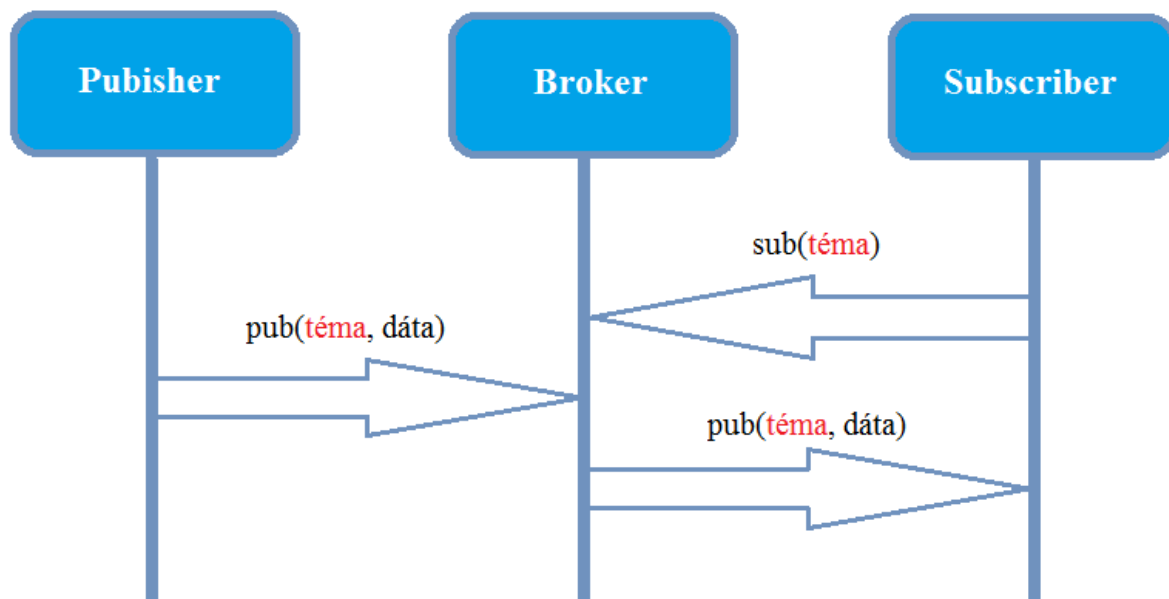
Device-to-device

- DDS

Vzhľadom na zámer tejto diplomovej práce sa bližšie budem zaoberať protokolmi D2S. Z tejto kategórie sú najznámejšie protokoly MQTT, CoAP a XMPP, preto si medzi nimi spravíme porovnanie. Je ešte potrebné podotknúť, že žiadny z týchto protokolov nijakým spôsobom nerieši predspracovanie a spracovanie dát, ale slúžia čisto iba ako komunikačné kanály na prenos dát v definovanej forme.

2.1.1. MQTT

MQ Telemetry Transport (MQTT) je ľahký protokol určený na posielanie správ. MQTT je založený na báze publish/subscribe, kde podstata spočíva v tom, že odosielateľ (publisher) a odoberateľ (subscriber) si neposielajú správy priamo, ale využívajú na komunikáciu sprostredkovateľa (broker). Funguje to teda spôsobom, že odosielateľ pošle správu na broker, vždy na určitú tému a odoberateľ sa môže k tejto téme prihlásiť a následne prijať obsah. [7]



Obr. 7 Publish/Subscribe komunikácia

MQTT bol vytvorený v roku 1999 zamestnancami IBM a vďaka tomu, že ide o otvorený, nenáročný a ľahký protokol, ktorý sa navyše jednoducho implementuje, je ideálnym riešením pre obmedzené prostredia, kde: [7]

- je drahá sieť, má malú priepustnosť alebo je nespoľahlivá,
- sa využívajú CPS s limitovanými výkonnostnými zdrojmi.

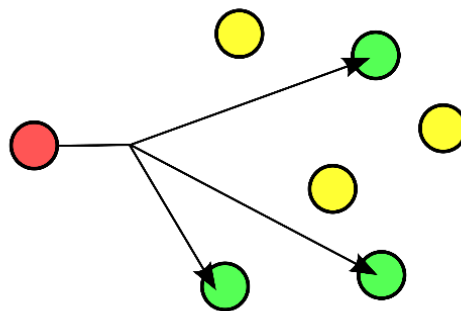
Vlastnosti tohto protokolu zahŕňajú: [7]

- Spomínaný publish/subscribe vzor, ktorý umožňuje one-to-many distribúciu správ.
- Používanie TCP/IP na poskytovanie základného pripojenia do siete.
- Tri kvality služby (Quality of service - SoQ) pre doručenie správ:
 - Nanajvýš raz – vhodné, kde sú správy doručované podľa najlepšieho výkonu základnej TCP/IP siete. Správa sa môže stratiť alebo duplikovať, preto je tento level dobré využiť tam, kde strata jednej správy je akceptovateľná, pretože ďalšia bude odoslaná v blízkej dobe.
 - Aspoň raz – vhodné, kde je potrebné mať istotu, že správa bude doručená. Môže však nastať duplikácia správy.
 - Presne raz – vhodné, kde správy musia byť doručené presne iba raz. Tento level je napríklad dobré využiť pri fakturačných systémoch, kde strata alebo duplicita správ by mohla viesť k účtovaniu nesprávnych poplatkov.
- Malý transportný overhead (fixná dĺžka hlavičky sú iba 2 bajty) a protokolové výmeny minimalizované na zníženie zaťaženia siete.

- Mechanizmus na notifikáciu zainteresovaných strán, ak nastane neobvyklé odpojenie klienta.

2.1.2. CoAP

Constrained Application Protocol (CoAP) je softvérový protokol určený na použitie s veľmi jednoduchými zariadeniami a s cieľom umožniť im komunikáciu cez Internet. Pod jednoduchými zariadeniami môžeme rozumieť rôzne senzory, prepínače, ventily a podobné komponenty, ktoré potrebujú byť monitorované a riadené vzdialene, cez sieť. Na to CoAP využíva request/response model a celý protokol bol navrhnutý tak, aby umožnil jednoduchý preklad do HTTP pre ľahšiu interakciu s webom. Takisto však spĺňa aj požiadavky IoT ako jednoduchosť, nízky overhead a one-to-many distribúcia správ cez IP multicast. [8]



Obr. 8 Metóda IP Multicast

Medzi vlastnosti CoAP protokolu patrí: [8]

- Komunikácia prebieha cez UDP.
- Podpora URI a content-type.
- Podpora pre vyhľadávanie zdrojov poskytovaných známymi CoAP službami.
- Jednoduché prihlásenie sa k zdroju a odoslanie notifikácie.
- Jednoduché kešovanie na základe max-age.

2.1.3. XMPP

Extensible Messaging and Presence Protocol (XMPP) je komunikačný protokol založený na XML. Pôvodne bol pomenovaný Jabber a jeho účelom bolo vytvoriť otvorený štandard pre posielanie správ takmer v reálnom čase, ale vďaka tomu, že je rozšíriteľný, našiel si uplatnenie aj v publish/subscribe systémoch, pri signalizovaní audio a video hovorov, posielaní súborov alebo pri hraní. Svoje miesto si vybudoval aj v IoT. [9]

XMPP ponúka tieto nasledujúce vlastnosti: [9]

- Je otvorený – XMPP je voľný, verejný, ľahko pochopiteľný a je zadarmo.
- Ide o štandard – IETF formalizovala XMPP ako uznanú technológiu na posielanie správ.
- Je overený – prvé Jabber/XMPP technológie sa objavili už v roku 1998 a odvtedy s nimi pracovalo už stovky vývojárov. XMPP je stabilný, čoho dôkazom sú milióny ľudí využívajúcich aplikácie založené na XMPP.
- Je decentralizovaný – architektúra XMPP siete je podobná emailom, čoho výsledkom je, že ktokoľvek si môže zriadiť vlastný XMPP server a ponúknuť používateľom kontrolu nad ich komunikáciou.
- Je bezpečný – hocijaký XMPP server môže byť v prípade potreby izolovaný od verejnej siete a takisto bola do jadra XMPP špecifikácií implementovaná bezpečnosť cez SASL a TLS.
- Je rozšíriteľný – využitím výhod XML si každý môže dobudovať vlastnú funkcionality. Na zachovanie interoperability najpoužívanejšie rozšírenia spravuje XMPP Standards Foundation, ale organizácie si v prípade potreby môžu udržiavať vlastné úpravy.
- Je flexibilný – XMPP aplikácie okrem posielania správ zahŕňajú správu siete, nástroje na spoluprácu, zdieľanie súborov, hranie hier, vzdialené monitorovanie systémov, webové služby a oveľa viac.
- Je diverzifikovaný – široké množstvo spoločností a open-source projektov používa XMPP na vytvorenie a nasadenie aplikácií a služieb pracujúcich v reálnom čase.

2.1.4. MQTT vs CoAP vs XMPP

Každý z týchto protokolov je dobre známy a široko používaný. Aj keď však v princípe riešia spoločnú vec, komunikáciu po sieti, každý z protokolov má odlišný prístup a je preto vhodný pre určitú oblasť, odlišnú od ostatných. V skratke by sme ich mohli zhrnúť nasledovne: [10] [11]

- MQTT je many-to-many komunikačný protokol pre posielanie správ cez centrálného brokera. Keďže MQTT má podporu pre stále spojenie, je vynikajúcim riešením ako komunikačný kanál pre dáta v reálnom čase.
- CoAP je vhodný na zariadenia s nízkym výkonom a do siete, ktorá nie je stabilná, predovšetkým ak je v nej pripojený veľký počet senzorov a iných zariadení. Protokol je však primárne určený na komunikáciu one-to-one a jeho hlavným využitím je prenos informácií o stave medzi klientom a serverom.
- XMPP má svoje korene v instantnom posielaní správ a aktuálne je dobrý uchádzač na širokú správu bielej elektroniky, ako sú práčky, sušičky, chladničky a tak ďalej. Keďže je

však založený na XML, predpokladá trvalé TCP spojenie a chýba mu efektívne binárne kódovanie, nie je veľmi praktický pri použití v sieťach so zlou kvalitou a pri zariadeniach s veľmi nízkym výpočtovým výkonom.

Protokol	MQTT	CoAP	XMPP
Transport	TCP	UDP	TCP
Posielanie správ	Publish/Subscribe	Request/Response	Publish/Subscribe
Využitie	Prenos správ a živých dát cez dlhotrvajúce spojenia	Prepojenie zariadení s obmedzenými zdrojmi	Vzdialená správa bielej elektroniky

Tab. 1 Porovnanie protokolov

2.2. OPC UA

V priemyselnej oblasti, kde je potrebné využívať PLC zariadenia, by zaujímavou alternatívou na komunikačný protokol mohol byť OLE for Process Control Unified Architecture (OPC UA). OPC UA je platformovo nezávislý štandard pre bezpečnú a spoľahlivú priemyselnú komunikáciu, ktorý umožňuje výmenu dát medzi produktami rôznych výrobcov a rôznych platforiem. Je totiž založený na špecifikáciách vyvinutých v úzkej spolupráci s výrobcami, používateľmi, výskumnými inštitúciami a konzorciou, aby tak zabezpečil zdieľanie informácií naprieč heterogénnymi systémami. Cieľom je zaručiť identitu klientov a serverov a odolať útokom. OPC UA definuje sadu služieb, ktoré servery môžu poskytovať a jednotlivé servery potom služby, ktoré podporujú, špecifikujú klientom. Klienti tak môžu pristupovať k aktuálnym aj historickým dátam, prípadne môžu byť upozornení na významnú zmenu alarmom či udalosťou. OPC UA je teda vďaka svojej portabilite a funkčnosti ideálnym riešením v prostredí, kde je potrebné prepojiť zariadenia PLC moderným spôsobom a umožniť im efektívnu komunikáciu. [12]

OPC UA bol vytvorený nadáciou OPC a ide pritom o nástupcu otvoreného štandardu OPC. Oproti svojmu predchodcovi sa však významne odlišuje, pretože ide o posun z platformovo závislého riešenia využívajúceho Microsoft technológiu COM/DCOM na platformovo nezávislú, servisne orientovanú architektúru. OPC UA teda prináša mnoho výhod, napríklad: [13]

- **Je platformovo nezávislý**

OPC UA je navrhnutý na platformovú nezávislosť. To ho jasne oddeľuje od klasického OPC, pri ktorom museli byť použité Microsoft technológie. OPC UA teda podporuje aj iné operačné systémy, ale takisto sa táto nezávislosť týka aj hardvérových zariadení, ktoré môžu bežať na rozdielnych procesorových architektúrach. [13]

- **Obsahuje rozšírenú bezpečnosť informácií**

OPC UA má bezpečnosť zabudovanú priamo v protokole, aplikačnému administrátorovi dovoľuje vybrať si z viacerých bezpečnostných alternatív, presne podľa požiadaviek na systém a prostredie, v ktorom bude použitý. Protokol umožňuje plné šifrovanie správ či aktiváciu autentifikácie. [13]

- **Vytvára reálnu integráciu od prevádzkovej úrovne až po exekutívnu**

Integrácia informácií medzi systémami rôznej úrovne prevádzky bola vždy výzva. OPC UA ale vďaka zabezpečenej komunikácii a doménovo špecifickým informačným modelom umožňuje neobmedzenú a bezproblémovú integráciu. Navyše, vďaka faktu, že komunikácie môže byť bezpečná a zúžená na jeden port, je možné otvoriť určité časti automatizovanej siete na priamy prístup zo systémov vyšších úrovní. [13]

- **Je založený na IEC štandarde**

OPC UA je štandardizovaný ako IEC 62541, čo znamená, že protokol je prístupný na preskúmanie kýmkoľvek. V praxi to znamená, že nie je viazaný na žiadnu špecifickú technológiu, napr. ako DCOM, ale namiesto toho definuje alternatívne transportné protokoly, ktoré môžu byť použité na doručenie správ. A v budúcnosti budú navyše môcť byť jednoducho pridané nové alternatívy, bez modifikácie iných častí protokolu. [13]

- **Stavia na zjednodušenej architektúre**

Integrácia OPC UA do aplikácie je v porovnaní s klasickým OPC relatívne priamočiara. Na začiatku sa implementuje základná funkcionálna a keď je potrebné pridať ďalšie vlastnosti, využijú sa rovnaké základné stavebné bloky a nemusí sa začínať odznova. [13]

- **Zahrňa veľké množstvo doménovo špecifických add-onov**

Flexibilita štandardu OPC UA spočíva vo vybranom informačnom modeli, ktorý poskytuje rámec pre vytvorenie a vystavenie informácií definovaných obchodníkom štandardným spôsobom. Táto flexibilita rozširuje použitie OPC UA štandardu z obyčajného komunikačného protokolu na široký výber komunikačných riešení špecifických pre rôzne aplikačné prípady použitia. [13]

- **Zakladá sa na jasnej a technicky neobmedzujúcej špecifikácii**

OPC UA je navrhnutý tak, aby obmedzoval aplikáciu v najmenšej možnej miere. Jedným z hlavných kritérií pri jeho návrhu totiž bola ľahká rozšíriteľnosť a preto OPC UA definuje pravidlá, ako rozšíriť základnú špecifikáciu a pritom zachovať súčinnosť s ďalšími časťami celku. [13]

- **Umožňuje rozšíriteľné riešenia**

Kým platformová nezávislosť je o voľnosti pri nasadzovaní, rozšíriteľnosť je o voľnosti pri aplikačnom dosahu. Ide však o úzko prepojenú oblasť, nakoľko aplikácie bežiace na limitovanom hardvéri sú zvyčajne aj skromnejšie na funkcionality. Klasický OPC väčšinou fungoval na desktopoch, OPC UA má však za cieľ podporovať aplikácie s rôznou mierou funkcionality a tak pokryť všetko od malých embedded systémov až po široko škálovateľné počítačové systémy. [13]

- **Umožňuje vytvárať riešenia, ktoré sú nadčasové**

Nakoľko OPC UA sa stal IEC štandardom, predajcovia a systémoví integrátori majú dôvod upustiť z ich vlastných riešení a prejsť na štandardizovanú komunikáciu. Podporovať ich v tom budú aj samotní koncoví používatelia, pre ktorých tak bude budúca úprava jednoduchšia. Výhoda tohto štandardu je tá, že neukazuje na žiadnu konkrétnu platformu ani techniku, ktorá by musela byť použitá. [13]

- **Lahko sa nasadzuje**

S dnešnými modernými nástrojmi a programovacími jazykmi je trend vytvárať riešenia, ktoré nie sú limitované na jeden operačný systém alebo špecifické knižnice. Pri OPC UA sa na to myslelo a preto je možné vytvoriť bezpečné riešenia tak, ako si to vývoj vyžaduje. [13]

3. Analýza a návrh architektúry pre zber dát na základe smart senzorov

Udržať krok s trendmi nie je jednoduché. Ak sa však správne využijú vo svoj prospech, je možné v tradičných oblastiach vytvoriť nové pridané hodnoty. Faktom je, že už teraz vzrastá počet pripojených zariadení k Internetu a toto číslo ešte exponenciálne porastie. Sensory budú monitorovať každý faktor potenciálne ovplyvňujúci výrobu a ich dáta budú slúžiť k nájdeniu nových, doposiaľ nevyužitých príležitostí. Viac senzorov teda znamená viac dát, ale takisto aj väčšie zaťaženie siete, zložitejšiu infraštruktúru a vyššie energetické nároky.

Typickým javom v priemysle je senzor napojený na PLC, ktorý získava dáta o stave snímaného objektu a tie sú potom ďalej posielané do databázy. V klasických architektúrach je riadenie centralizované, kde systém SCADA v pravidelných intervaloch posiela PLC zariadeniam požiadavku na dáta a tie sú potom odoslané v odpovedi. V tých novších je snaha o určitý stupeň decentralizácie, kde sa snaží predísť posielaniu dát spôsobom požiadavka/odpoveď. Takéto riešenia sa dajú dosiahnuť buď implementáciou vhodného protokolu (napr. OPC UA) alebo sa medzi PLC a SCADA vkladajú ďalšie zariadenia, napríklad priemyselné smerovače eWON. V každom prípade sú však dáta zaznamenávané v definovaných intervaloch.

Je ale mnoho senzorov, ktoré snímajú javy meniace sa pomalšie, prípadne javy, ktoré sa menia v nepravidelných intervaloch. Príkladom môžu byť senzory merajúce teplotu, tlak, prítomnosť nejakej látky, pohyb a tak ďalej. Pri týchto javoch nás nezaujíma informácia, že sa nič nezmenilo, zaujíma nás čo sa zmenilo a ako. Bolo by preto vhodné, ak by sme sa nemuseli senzoru pravidelne dopytovať, či nastala nejaká zmena, ale senzor nás na to sám upozornil.

Predmetom tejto diplomovej práce bol návrh a realizácia smart senzoru, ktorý by posielal hodnoty iba v prípade potreby. Išlo by teda o nový stupeň decentralizácie až na samotné senzory, ktoré by boli schopné vyhodnotiť situáciu a informovať o tom okolie. Takéto smart senzory by potom neposielali iba dáta o snímanom objekte, ale hotové informácie. Napríklad, že teplota klesla o stupeň. Alebo tlak stúpol o jeden hektopascal. Alebo objekt sa pohol.

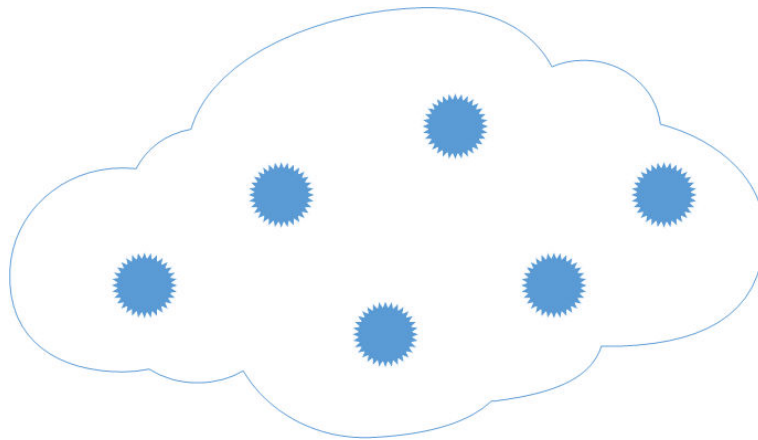
Smart senzory by mali mnoho výhod. V prvom rade by sa rapídne znížilo zaťaženie siete, cez ktorú by sa neposielali dáta bez informačnej hodnoty. Po ďalšie by sa znížila energetická náročnosť spotrebovaná posielaním dát. Zmysel to má predovšetkým pri senzoroch na baterky, ktoré by sa vybíjali pomalším tempom a na jedno nabitie vydržali dlhšie. A v neposlednom rade, v databáze by sme mali dáta s informačnou hodnotou.

Riešenie spočíva v implementovaní do senzora algoritmus na výpočet entropie. Entropia je miera neusporiadanosti vo fyzikálnych systémoch, resp. množstvo informácie, ktoré je možné získať pozorovaním neusporiadaných systémov. Informačnú entropiu prvýkrát zaviedol Claude E. Shannon v roku 1948. Odvtedy sa tejto oblasti venovalo viacero pánov a jedným z nich je aj Constantino Tsallis, ktorý v roku 1988 prišiel so vzorcom:

$$S_{\alpha} = \frac{1}{\alpha - 1} * \left(1 - \sum_{i=1}^n p_i^{\alpha} \right), \alpha \geq 0, \alpha \neq 1 \quad 1.$$

kde $\{p_i\}$ je množina pravdepodobností a α je parameter ovplyvňujúci výsledné vyhodnotenie entropie. Implementovaním tejto rovnice je možné vypočítať informačnú hodnotu zo skúmanej vzorky dát. [14]

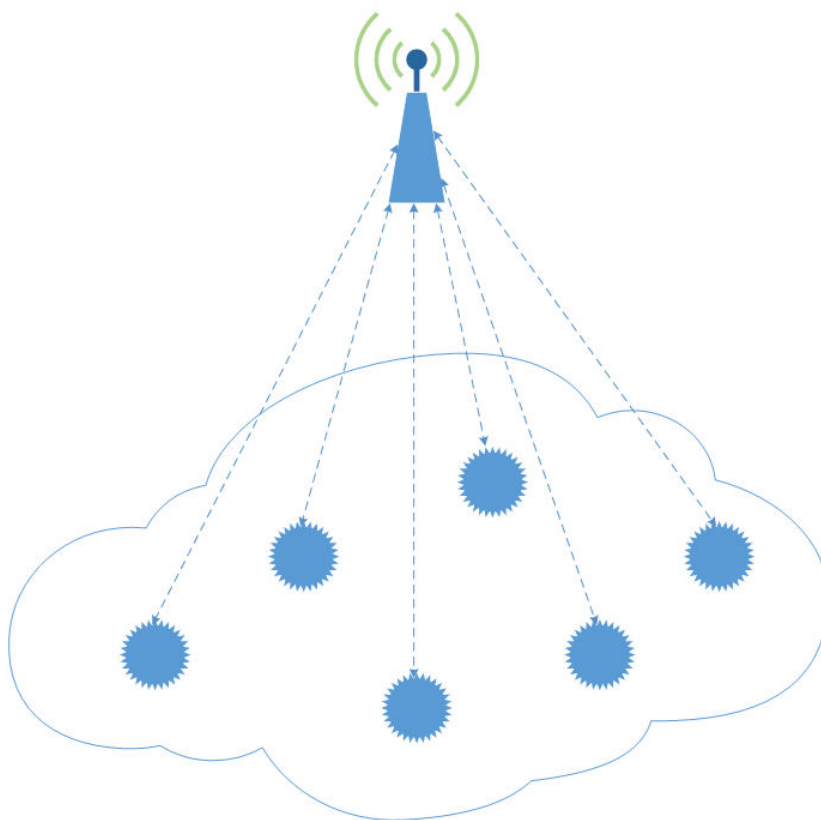
Cieľom je teda vytvoriť jednoduchý algoritmus, ktorý zo snímaných dát bude vypočítavať informačný zisk a v prípade, že výsledok prekročí stanovenú senzitivitu, informácia sa odošle do databázy. Takto dosiahneme posielanie iba tých hodnôt, ktoré sú potenciálne užitočné a majú význam pre ďalšiu analýzu. Ak navyše prispôbime algoritmus výpočtovému výkonu kyberfyzikálnych systémov, je možné aktuálne dostupným senzorom dodať novú pridanú hodnotu.



Obr. 9 Sieť senzorov

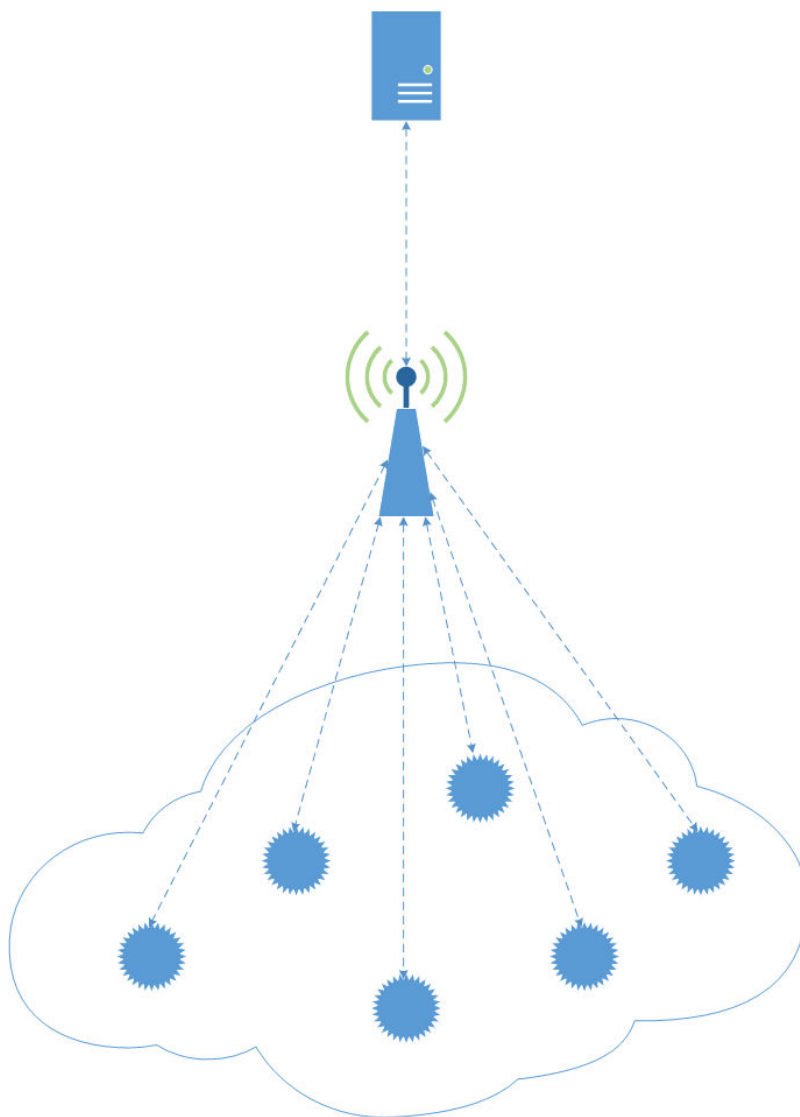
Ďalším krokom je vytvoriť komunikačný kanál, cez ktorý budú smart senzory posilať informácie. Na to sa výborne hodí nejaký z IoT protokolov, ktoré sú špeciálne prispôbené týmto prípadom použitia. Požiadavky sú bezpečnosť, nízka energetická náročnosť a vysoká efektivita. Po analýze uvedenej v kapitole 2 som sa rozhodol pre protokol MQTT, ktorý funguje na báze publish/subscribe a komunikácia prebieha cez broker. Smart senzory teda budú posilať informácie na jedno miesto a nebudú pritom viazané na žiadnu konkrétnu databázu či aplikáciu. Navyše tento prístup umožňuje obojsmernú komunikáciu, čím vzniká priestor na ďalšie inovácie a nápady,

napríklad vytvoriť komunikáciu medzi senzormi navzájom, prípadne získať dáta na požiadanie. Prepojenie smart senzormi s brokerom cez MQTT je znázornené na nasledujúcom obrázku.



Obr. 10 Komunikácia senzorov s brokerom

Komunikácia cez MQTT znamená preposielanie správ na určité témy. Ak smart senzory budú odosielať informácie na definované témy, je potrebná takisto služba, ktorá bude správy odchytať a pracovať s nimi ďalej. V mojom návrhu je to služba bežiaci na serveri, ktorá uloží hodnoty do databázy. Jej funkcia bude jednoduchá, bude sledovať definované témy a zaznamenávať každú správu o zmene snímaných objektov.



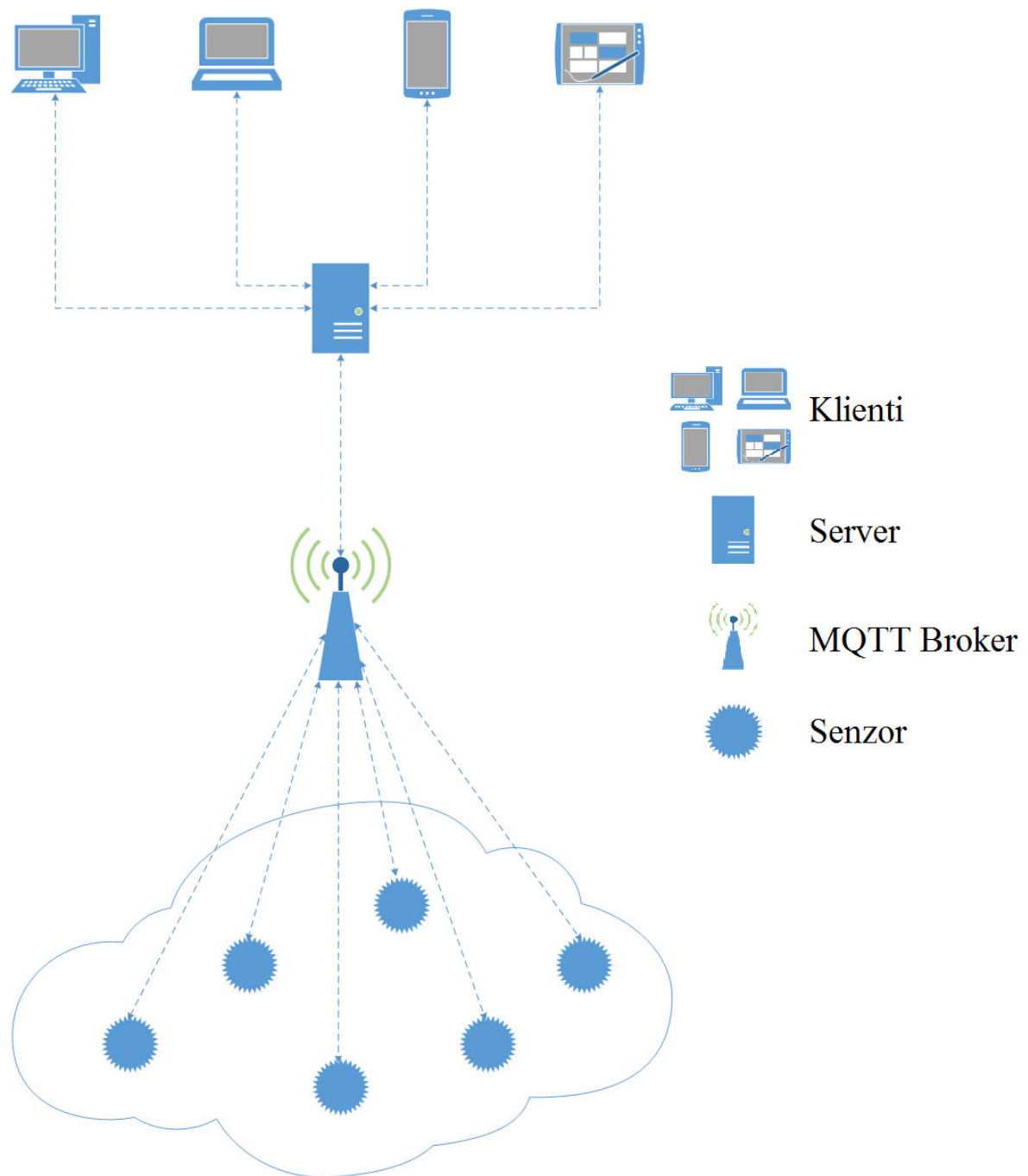
Obr. 11 Ukladanie dát z brokera na server

Daná architektúra je teda kompletným návrhom riešenia na zber dát zo smart senzorov. Senzor vygeneruje informáciu, odošle ju cez MQTT protokol na broker, odtiaľ správu prečíta služba bežiaca na serveri a uloží hodnotu do databázy. V rámci tejto práce som sa ale ešte venoval jednému kroku a to vyhodnoteniu a vizualizácii získaných informácií.

Medzi hlavné trendy v priemysle, uvádzané v kapitole 1.1.2, patrí aj samospráva a automatizované riešenie vzniknutých situácií. Takáto funkcionality má viacero výhod, medzi inými vyššiu efektivitu pri riešení problémov, skrátený čas strávený opravou a šetrenie zdrojov vynaložených na hľadanie príčiny. Navyše ak by sa systém sám naučil čo je správne a čo nie, zvýšilo by to pridanú hodnotu ešte výraznejšie.

Mojim návrhom je vytvoriť ukážku, ako by to mohlo vyzerieť a ukázať, aké reálne výhody takéto riešenie vie priniesť. Vytvorím preto jednoduchú webovú aplikáciu, ktorá naučí systém adaptovať

sa na prostredie a následne v prípade výnimiek sa naučí, ako ich nabudúce riešiť. Pôjde teda o akýsi prototyp, akým smerom by sa mohla správa systémov uberať. Celkovo okrem zberu dát zo smart senzorov tak bude navrhnutá architektúra schopná aj automatizovane detegovať a podávať riešenia na vzniknuté odchýlky od normálu.



Obr. 12 Konečný návrh architektúry

4. Vytvorenie aplikácie podľa návrhu

V predchádzajúcej kapitole bola popísaná architektúra na zber a vizualizáciu dát získaných zo smart senzorov. V tejto kapitole budú popísané jednotlivé kroky, ktoré bolo potrebné vykonať. Nakoľko riešenie pozostáva z viacerých častí, v popise budem postupovať chronologicky tak, ako boli vytvárané. Na záver potom zhodnotím dosiahnuté výsledky a uvediem možnosti pokračovania vo výskume tejto problematiky.

Primárnym cieľom práce bolo vytvoriť smart senzor. Pod pojmom smart senzor sa rozumie zariadenie schopné posilať hotové informácie určené na ďalšiu analýzu. Ako už bolo popísané, výhod takéhoto senzoru je mnoho:

- hodnoty o snímanom objekte sú posielané iba vtedy, ak pre nás majú informačnú hodnotu,
- frekvencia posielania informácií sa určuje dynamicky podľa potreby,
- menej posielaných dát s nízkou informačnou hodnotou znamená menšie zaťaženie siete,
- menej odoslaných dát zároveň znamená menšia energetická náročnosť na prevádzku senzora (výhodné najmä pri zariadeniach na batérie),
- v databáze sú hodnoty, ktoré pre nás majú význam,
- dáta s vyššou informačnou hodnotou sú vhodnejšie na ďalšie analýzy,
- decentralizácia rozhodovania klesá až na samotný senzor.

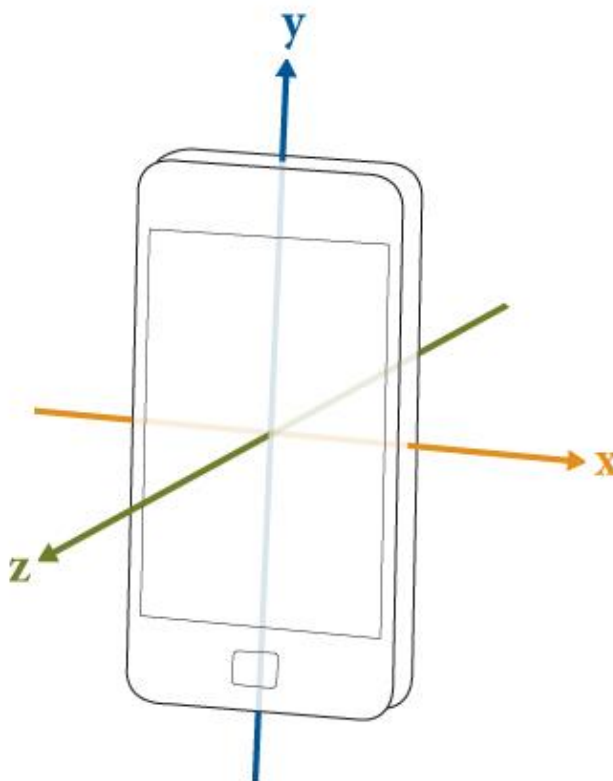
Prínos riešenia je teda jasný. Predovšetkým v oblastiach, kde sa snímajú javy menia nepravidelne alebo kde sú obmedzené zdroje, by využitie smart senzorov mohlo znamenať veľkú pridanú hodnotu.

Pri tvorbe smart senzoru však bolo potrebné vyriešiť viacero otázok:

- aký senzor použiť,
- aký algoritmus použiť na výpočet informačnej hodnoty,
- aké prípadné rozširujúce funkcionality doňho implementovať,
- ako odoslať informáciu na ďalšie spracovanie.

Odpoveď na otázku „aký senzor použiť“ je jednoduchá, vhodný je každý senzor. Či už meria teplotu, tlak, vlhkosť, rýchlosť alebo prítomnosť nejakých látok. Dokonca aj kamery zaznamenávajúce obraz by boli riešiteľné. Ja som sa však zameral na dáta v textovej forme a z tohto

plynul aj ďalší postup. Vzhľadom na dostupnosť technických možností som sa nakoniec rozhodol využiť senzor zo smartfónu a konkrétne mi na ukážku poslúžil gyroskop.



Obr. 13 Osi gyroskopu v mobile

Gyroskop je senzor na meranie orientácie zariadenia, ktorú určuje na základe zemskej gravitácie. Senzor meria polohu v troch osiach, X, Y a Z, podľa ktorých vieme vyhodnotiť natočenie zariadenia. V mojej práci poslúžil ako vhodný kandidát na implementáciu algoritmu, zaujíma ma vždy aktuálna orientácia zariadenia. To znamená, že ak sa s mobilom nehýbe, nepotrebujem ani posielať dáta a naopak, v prípade aktivity chcem byť presne informovaný.

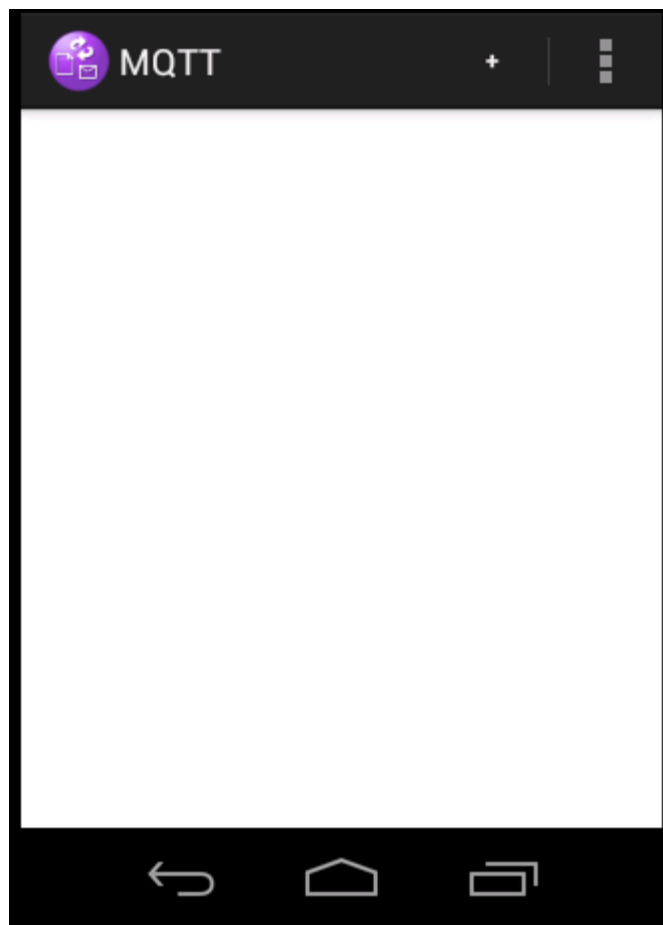
Na výskum bol použitý mobil s operačným systémom Android a tým pádom sa pracovalo v programovacom jazyku Java. Základom Android aplikácie bolo snímanie dát z gyroskopu, výpočet informačnej hodnoty a odoslanie informácie cez komunikačný kanál na ďalšie spracovanie.

Za komunikačný kanál som po zvážení viacerých alternatív vybral protokol MQTT. Ako už bolo spomenuté v predchádzajúcich kapitolách, ide o publish/subscribe protokol, pri ktorom komunikácia prebieha cez broker. Tento protokol je výhodný z viacerých hľadísk:

- je prispôsobený na zariadenia s nízkym výpočtovým výkonom,
- má malý overhead,

- podporuje nepretržité spojenie a zdieľanie živých dát,
- funguje aj v sieťach s pomalou rýchlosťou alebo častými výpadkami,
- umožňuje obojsmernú komunikáciu,
- je možné nastaviť kvalitu služby pre doručenie správ,
- obsahuje mechanizmus na notifikáciu, ak by zariadenie vypadlo.

MQTT je navyše celkom populárny protokol a tak naňho existujú otvorené implementácie v rôznych jazykoch. Známym projektom implementujúcim klienta je Paho [15] a na broker existuje projekt Mosquitto [16]. Aby som teda nadviazal na existujúce riešenia a nevytváral všetko nanovo, stiahol som voľne dostupnú klientskú aplikáciu na Android [17], ktorá obsahovala základný dizajn a pre mňa dôležité metódy na posielanie a prijímanie správ.



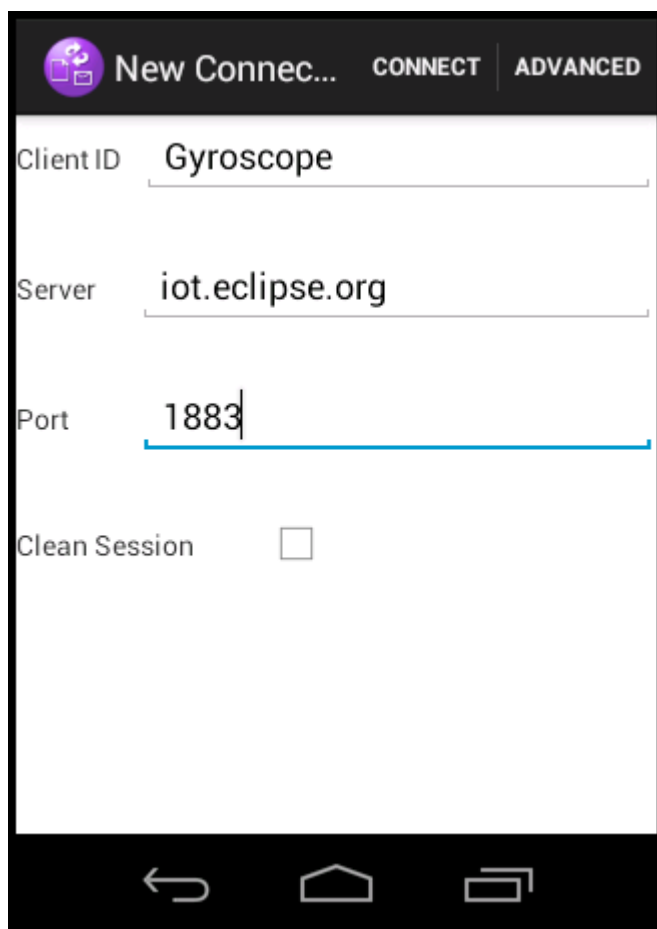
Obr. 14 Základná obrazovka Android aplikácie

V aplikácii je možné využiť všetky základné funkcionality protokolu MQTT ako je prihlásenie sa na broker, posielanie správ na definovanú tému a takisto ich odoberanie. V mojom prípade dizajn nebol podstatný, pretože priemyselné senzory grafické rozhranie nepotrebujú, ale implementované metódy na prácu s MQTT mi prišli vhod.

Podobným spôsobom, t.j. využitím dostupných možností, som vyriešil aj otázku MQTT brokera. Je k dispozícii hotový inštalačný program pre viaceré platformy [16], ale takisto je možné na testovanie použiť aj voľne prístupný sandbox server, na ktorý sa dá pripojiť s týmito údajmi:

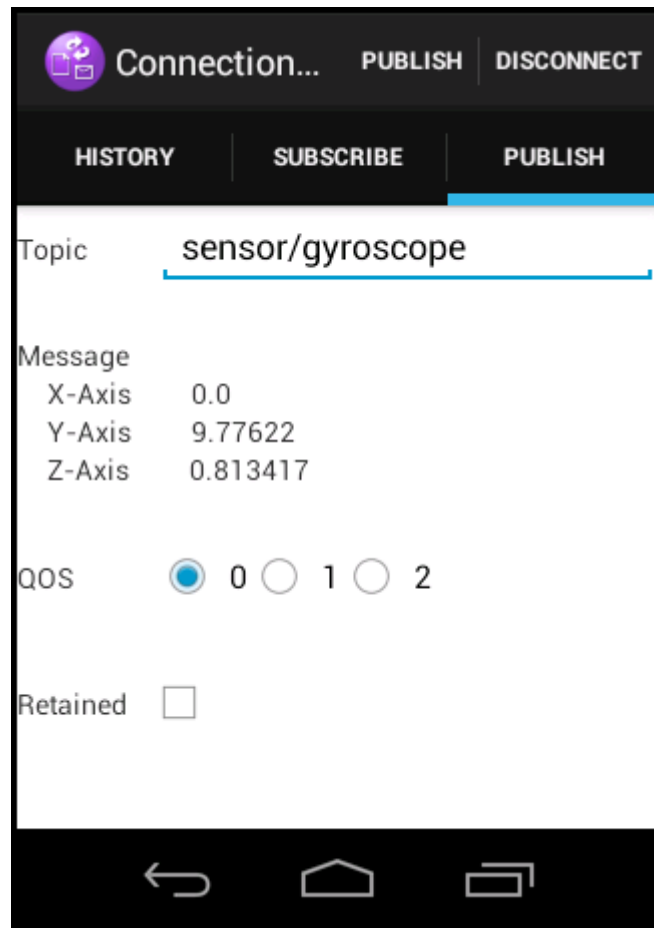
- Hostname: *iot.eclipse.org*
- Port: *1883*

Tento broker je prevádzkovaný skupinou organizácií a jednotlivcov, ktorí spoločne vytvorili ekosystémom Eclipse IoT za účelom vývoja, podpory a prijatia IoT technológií [18]. A keďže som pri návrhu architektúry chcel oddeliť broker od servera, na ktorom bude databáza, a k dispozícii som mal iba jeden server, použil som na ukážku práve tento voľne prístupný MQTT broker.



Obr. 15 Pripojenie na broker

Po využití dostupných riešení som mal k dispozícii aplikáciu schopnú komunikovať cez MQTT protokol a broker, cez ktorý komunikácia prebieha. Ďalším krokom bolo zaregistrovať do aplikácie senzor gyroskop a vyplniť obsah správ jeho hodnotami. Nastavil som, že všetky tri osi sa budú posielat' na jednu tému a budú vo formáte X/Y/Z. Výstupom bola aplikácia schopná odosielať v správach aktuálnu orientáciu zariadenia.



Obr. 16 Hodnoty osí gyroskopu v posielaných správach

```

Connected to tcp://m2m.eclipse.org:1883 with client ID SampleJavaV3_subscribe
Subscribing to topic "sensor/gyroscope" qos 2
Press <Enter> to exit
Time: 2015-04-26 23:41:36.148 Topic: sensor/gyroscope Message: -0.9942854/2.4516625/10.378705 QoS: 0
Time: 2015-04-26 23:41:38.194 Topic: sensor/gyroscope Message: 0.44947147/5.9248514/7.4230895 QoS: 0
Time: 2015-04-26 23:41:40.122 Topic: sensor/gyroscope Message: -0.9125633/5.7750273/7.6137743 QoS: 0

```

Obr. 17 Doručené správy na brokeri

V tejto fáze prišli na rad otázky „aký algoritmus použiť na výpočet informačnej hodnoty“ a „aké prípadné rozširujúce funkcionality doňho implementovať“.

Mieru informácie z dát vieme zistiť pomocou vzorcov na výpočet entropie. Touto problematikou sa zaoberali mnohí matematici a vďaka tomu existuje viacero rovníc, ktoré sa od seba mierne odlišujú prístupom a počtom vstupujúcich faktorov. Ja som vyskúšal dva vzorce: [14]

- Shannonovú entropiu

$$H(S_i) = - \sum_{j=1}^n p_j \log p_j \quad 2.$$

kde $\{p_i\}$ je množina pravdepodobností a

- Tsallisovu entropiu

$$S_{\alpha} = \frac{1}{\alpha - 1} * \left(1 - \sum_{i=1}^n p_i^{\alpha} \right), \alpha \geq 0, \alpha \neq 1 \quad 1.$$

kde $\{p_i\}$ je množina pravdepodobností a α je parameter ovplyvňujúci výsledné vyhodnotenie entropie.

Pravdepodobnosti p_i sa vypočítavajú podľa vzorca:

$$p_i = \frac{m_i}{n}, i = 1, 2, 3 \dots n \quad 3.$$

kde m_i je počet výskytov i -tej hodnoty z množiny a n je veľkosť množiny.

Vzhľadom na obmedzené zdroje senzorov bolo ale vhodnejšie ponechať iba jednu rovnicu a tak som sa rozhodol ďalej pracovať s Tsallisovou entropiou, ktorá oproti riešeniu Shannona obsahuje navyše parameter α . Ten pri bol experimentálne stanovený na hodnotu 1.2. Veľkosť množiny som definoval na $n=5$ a išlo teda o akýsi zásobník posledných piatich najaktuálnejších hodnôt, kde príchod novej hodnoty nahradil vždy tú najstaršiu.

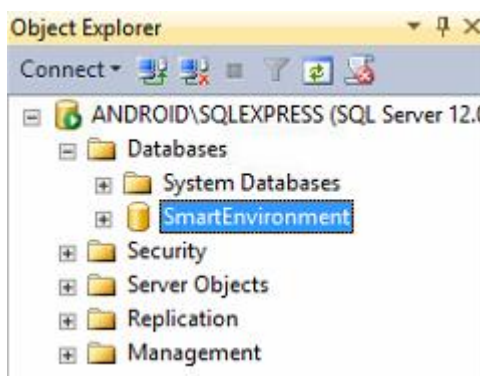
Pri tejto príležitosti je však potrebné spomenúť vlastnosť gyroskopu. Snímanie tohto senzora je na mnoho desatinných miest a ani pri úplnom pokoji nie je výstup statický, ale každá z osí osciluje okolo nejakej hodnoty. Tento jav by zapríčiniť, že iba málokedy by sme nadobudli v množine viac rovnakých hodnôt a teda pravdepodobnosti by boli vždy nízke (konkrétne $1 / 5 = 0.2$) a výsledná entropia vysoká. Je preto žiadúce, aby sa nastavila určitá miera akceptovateľnosti, vďaka ktorej by sa hodnoty rozdielne od určitého desatinného čísla považovali za rovnaké. Táto miera samozrejme závisí od požadovanej citlivosti detekcie. V mojom prípade som mieru akceptovateľnosti nastavil na hodnotu 0,5, čo znamená, že za zmenu orientácie zariadenia budem považovať každý posun na osi o viac ako 0,5. Zmenu do 0,5 beriem ako akceptovateľné kmitanie okolo jednej hodnoty.

Keď bol algoritmus na výpočet entropie hotový, ostávalo určiť senzitivitu senzoru. Tá sa skladá z hranice, od ktorej pre mňa informačná hodnota začína byť zaujímavá a stanovenia minimálnej frekvencie posielania správ. V mojom prípade som hranicu nastavil na 0.9 a minimálnu frekvenciu na 2 sekundy. V praxi to znamená, že každá entropia nad 0,9 je dostatočne hodnotná a senzor ju odošle v správe, ale zároveň tieto správy musia byť v rozmedzí minimálne 2 sekúnd. Tým sa zaručí, že budem informovaný pri každej zmene a prípade potreby sa tak bude diať dvojsekundovým intervale.

Po naprogramovaní týchto krokov som mal vytvorený smart senzor schopný vypočítavať entropiu z dát a podľa nastavenej senzitivity ju posilať cez MQTT na broker. Na záver som uvažoval

o prípadných rozširujúcich funkcionalitách. Keďže MQTT umožňuje obojstrannú komunikáciu, bolo by napríklad možné prijímať príkazy z vyšších úrovní systému, ako pokyny na zapnutie/vypnutie senzora či odoslanie dát na vyžiadanie. Takisto tu ale vzniká príležitosť aj na úplne nové riešenia, spomeniem napríklad vytvorenie zdieľanej inteligencie medzi senzormi navzájom. Išlo by o funkcionality ako kontrola správnosti merania, detekcia a riešenie neočakávaných situácií, spoločné zameranie sa na špecifický objekt a podobne. Vzájomná komunikácia má mnoho výhod a vie priniesť doteraz nepoznané možnosti. Pri premýšľaní o nových nápadoch je však potrebné neignorovať výpočtové kapacity týchto zariadení. Tým nechcem povedať, že ich výkon nič nedokáže, práve naopak, len je skrátka potrebné tento fakt brať do úvahy. Aj z tohto dôvodu považujem rozširujúce funkcionality skôr za nadstavbu smart senzorov, ktoré závisia od konkrétnych prípadov a špecifických nárokov. Mojim cieľom bolo zamerať sa na vytvorenie smart senzoru schopného odosielať hotové informácie, ale vzhľadom na potenciál ďalších možností nechávam „dvere otvorené“.

Ďalším krokom bol zber generovaných informácií. Smart senzor posiela cez MQTT správy na broker, ale odtiaľ ich je potrebné odoberať a niekam uložiť. Na to výborne poslúžil server v škole. Nainštaloval som teda naňho relačný databázový systém, konkrétne SQL Server 2014 Express, a vytvoril databázu pre účely tejto práce.



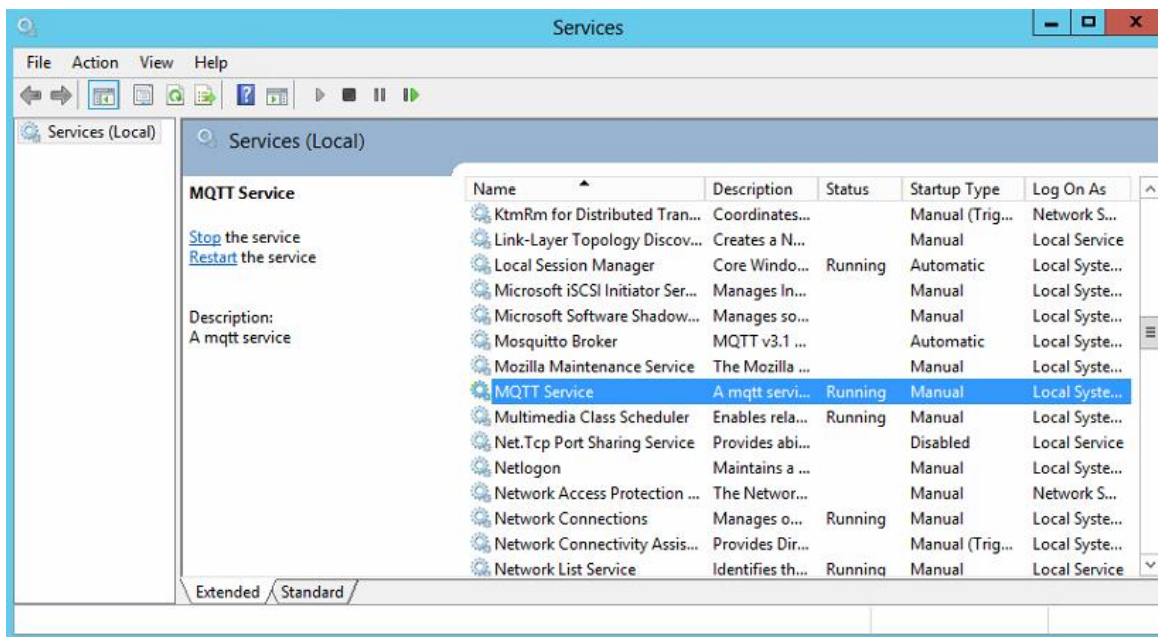
Obr. 18 Používaná databáza na serveri

Aby som však dostal správy z brokera do databázy, potreboval som byť k téme permanentne prihlásený. Rozhodol som sa preto vytvoriť windows službu, ktorá bude bežať na pozadí a bude vykonávať nasledovné funkcie:

- prihlási sa na broker ako odoberateľ definovanej témy,
- prijaté správy konvertuje do vhodnej podoby,
- hodnoty osí gyroskopu uloží do databázy.

Na komunikáciu cez MQTT protokol som znova siahol po otvorenej implementácii, tentokrát išlo o klienta pre platformu Windows [19]. Potrebne metódy som vložil do novovytvorenej windows služby, čím som mohol jednoducho naprogramovať prihlásenie sa na broker a odoberanie správ. Stiahnuté správy sa potom transformujú do vhodnej podoby a uložia do tabuľky v databáze.

Keď bola windows služba hotová, mohol som ju nainštalovať a spustiť.



Obr. 19 Windows služba

ID	Date	XAxis	YAxis	ZAxis	AlertCategory_ID
33	2015-04-26 14:48:08.263	2.982856	6.347082	6.1563973	NULL
34	2015-04-26 14:48:09.857	-1.1441092	4.930566	8.3765135	NULL
35	2015-04-26 14:48:11.233	-0.027240695	3.8273177	8.771504	NULL
36	2015-04-26 14:48:15.207	-0.10896278	0.29964766	9.833891	NULL

Obr. 20 Uložené hodnoty v databáze

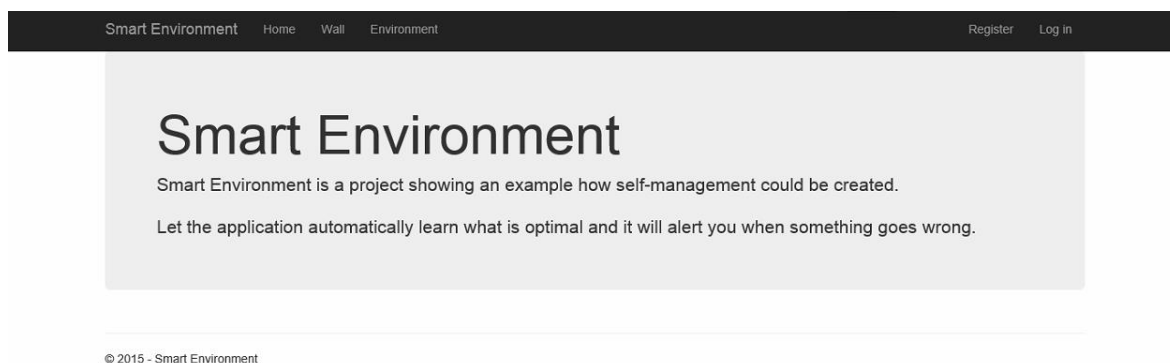
Tým bol zber informácií dokončený a ja som mohol prejsť k poslednej časti tejto práce, vytvoreniu samosprávy prostredia. Ako bolo uvedené v návrhu architektúry, cieľom bolo poukázať na výhody riešenia držiaceho krok s trendmi Priemyslu 4.0. Automatizácia detekcie vzniknutých neštandardných situácií a poukázania na ich riešenie môže byť pre celkový systém mimoriadne prospešná, pretože znižuje čas trvania problému a takisto šetrí zdroje. Chcel som teda vytvoriť ukážku systému, ktorý sa adaptuje na zadané prostredie, naučí sa pravidelne vykonávané procesy a keď potom nastane neštandardná situácia, zdeteguje ju a zapamätá si, ako ju v budúcnosti riešiť. Tento koncept predpokladá splnenie nasledujúcich podmienok:

- procesy v prostredí sa cyklicky opakujú,

- behom učenia nenastane neštandardná situácia,
- keď potom systém odhalí neštandardnú situáciu, používateľ ho naučí ako takýto problém nabudúce riešiť.

Koncept som sa rozhodol implementovať do webovej aplikácie, využil som preto technológiu ASP .NET MVC. Tá mala obsahovať tieto funkcionality:

- algoritmus na učenie sa procesov,
- algoritmus na detekciu chýb,
- možnosť nájsť problém zobrazíť používateľovi,
- umožniť používateľovi zadať riešenie,
- v budúcnosti mu zobrazíť rovnakú chybu aj s popisom na jeho riešenie.



Obr. 21 Základná obrazovka webovej aplikácie

Algoritmus na učenie sa procesov vychádza z aktuálnych hodnôt generovaných počas behu systému. Z nich sa potom aplikácia naučí, aké cykly sa opakujú a čo môže považovať za správne. Rozhodol som sa pre učenie z aktuálnych hodnôt, pretože pri mojej tvorbe som nemal k dispozícii historické dáta. Musí tak byť splnená podmienka, že počas behu učiaceho algoritmu nemôže nastať chyba. V opačnom prípade by bola zahrnutá medzi správne hodnoty a učiaci sa algoritmus by musel byť spustený odznova. Takéto riešenie je vhodné pri nasadzovaní systému do stále nového prostredia, v ktorom nie je možnosť učiť sa z historických údajov. Ak však historické dáta sú k dispozícii, alternatívou by mohlo byť aj učenie z nich za pomoci využitia metód dolovania v dátach.

Učiaci algoritmus som implementoval do databázy ako trigger, vzhľadom na rýchlosť práce s dátami. V mojom prípade údaje získavam zo smart gyroskopu a tak sa systém v algoritme učí minimálne a maximálne hodnoty osí, ktoré senzor nadobúda. Naučené poznatky potom uloží do tabuľky.

ID	Parameter	MinVal	MaxVal
1	XAxis	-0.53119355	6.578628
2	YAxis	-1.0215261	7.9270425
3	ZAxis	-2.8330324	9.874752

Obr. 22 Optimálne hodnoty senzora

Naprogramovaný trigger je aktivovaný z aplikácie pri spustení procesu učenia, pričom dĺžku trvania procesu si používateľ môže nastaviť v závislosti od prostredia.

The screenshot shows the 'Environment' section of a web application. At the top, there is a navigation bar with 'Smart Environment', 'Home', 'Wall', and 'Environment' links, along with 'Register' and 'Log in' buttons. The main heading is 'Environment' with a subtext: 'Let the environment learn what is optimal. Just set time period for the learning process and everything else will be done.' Below this, there is a form with 'Learning Period' set to '20' and 'Unit Of Time' set to 'Seconds'. A 'Start learning process' button is visible at the bottom of the form. The footer indicates '© 2015 - Smart Environment'.

Obr. 23 Obrazovka na zadávanie dĺžky trvania procesu učenia

This screenshot shows the same 'Environment' interface as before, but with a modal dialog box titled 'Learning process' open in the center. The dialog contains the text 'The learning process has started. It will finish in:' followed by a digital timer showing '00' for hours, '00' for minutes, and '15' for seconds. Below the timer, the labels 'HOURS', 'MINUTES', and 'SECONDS' are displayed. A 'Close' button is located at the bottom right of the dialog. The background interface is dimmed.

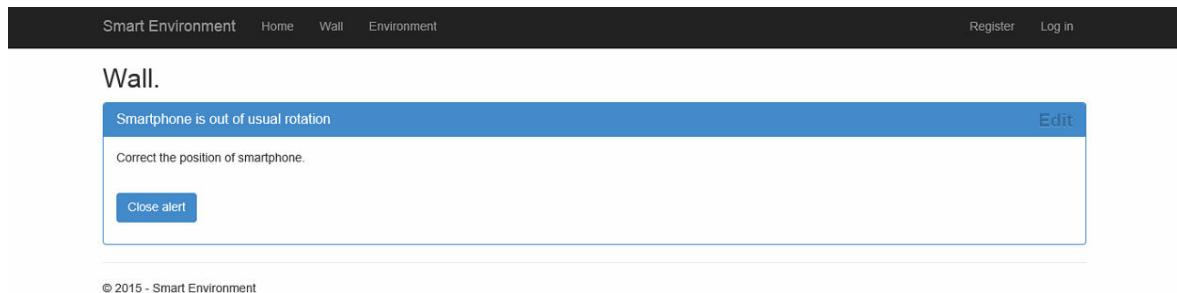
Obr. 24 Odpočítavanie do ukončenia procesu učenia

Po ukončení procesu učenia sa spustí druhý trigger, ktorý je zodpovedný za detekciu neštandardných situácií. Ten porovnáva prichádzajúce informácie s naučeným rozsahom a v prípade, že je nejaká hodnota mimo normu, nastane upozornenie. Systém najprv prejde tabuľku s doterajšími chybami a pokúsi sa nájsť zhodu.

ID	XAxisStatus	YAxisStatus	ZAxisStatus	IsSolved	SolutionTitle	SolutionDescription
4	2	1	0	0	Smartphone is out of usual rotation	Correct the position of smartphone.

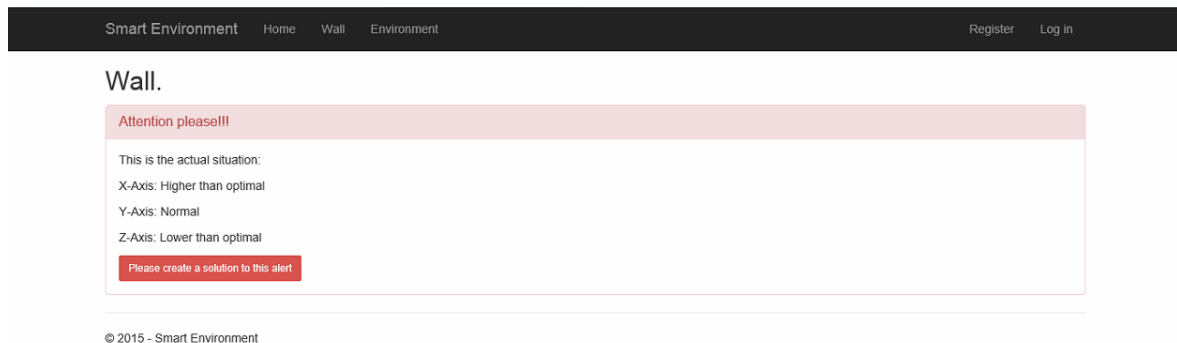
Obr. 25 Neštandardné situácie systému aj s popisom riešenia

Ak už daný problém niekedy vznikol, systém zobrazí používateľovi naučené riešenie.



Obr. 26 Chybový stav s popisom riešenia

Ak taký problém ešte nenastal, používateľovi sa zobrazí dôvod neštandardnej situácie aj s požiadavkou na zadanie riešenia. To sa potom uloží do databázy pre budúcu potrebu.



Obr. 27 Chybový stav bez existujúceho popisu riešenia

The screenshot shows a web application interface with a dark header containing 'Smart Environment', 'Home', 'Wall', 'Environment', 'Register', and 'Log in'. The main content area is titled 'Wall.' and contains an alert box with the text 'Attention please!!!' and 'This is the actual situation: X-Axis: Higher than optimal, Y-Axis: Normal, Z-Axis: Lower than optimal'. A red button says 'Please create a solution to this alert!'. A modal window titled 'Create a solution to this alert' is open, displaying a table of axis statuses and input fields for a solution title and description.

X Axis status	Higher than optimal
Y Axis status	Normal
Z Axis status	Lower than optimal

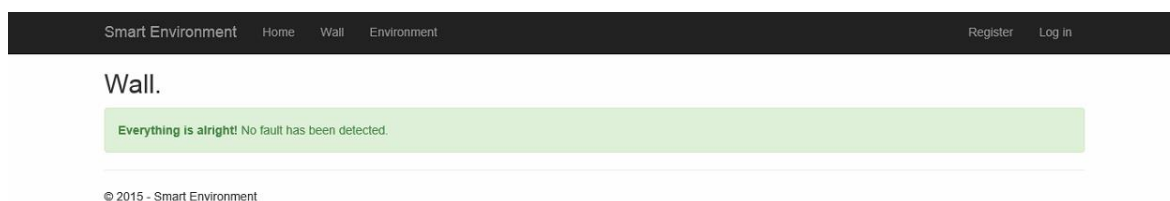
Solution Title:

Solution Description:

Obr. 28 Formulár na zadanie riešenia chybového stavu

Riešenie je pritom možné v prípade potreby aj opätovne editovať.

Po oprave problému používateľ jednoducho chybový stav ukončí, no systém už naňho bude nabudúce pripravený.



Obr. 29 Systém bez chýb

Vytvorená aplikácia je samozrejme iba ukážka. Ide o reakciu na trend v oblasti automatizácie a samosprávy prostredia, pričom môj návrh je aplikovateľný aj v prípadoch, kde nie sú k dispozícii historické dáta. Má však ešte mnoho nevyriešených otázok, ale takisto aj možností na rozšírenie. Ako príklad na ďalšie funkcionality uvediem:

- schopnosť predvídať blížiaci sa vznik neštandardných situácií sledovaním príznakov,
- automatizovať vykonávanie riešení na neštandardné situácie,
- pomocou dolovania v dátach hľadať podobnosti medzi jednotlivými chybami a navrhovať riešenia aj na nové, no podobné situácie,
- schopnosť adaptovať sa na nové podmienky aj po ukončení učiaceho algoritmu.

A týchto možností je oveľa viac. Znova to však závisí od konkrétnych požiadaviek na systém. No je isté, že automatizácia je jeden z hlavných trendov Priemyslu 4.0 a vývoj sa bude uberať aj týmto smerom.

Zhrnutím, pri realizácii tejto práce som využil viaceré technológie. Na vytvorenie smart senzoru bol použitý gyroskop z Android zariadenia, do ktorého som implementoval algoritmus na výpočet entropie. Pomocou protokolu MQTT som odosielať správy zo zariadenia na broker tvoriaceho bránu pre všetkých klientov. Na zber a ukladanie posielených informácií mi poslúžila windows služba nainštalovaná na školskom serveri, ktorej funkciou bolo odoberanie správ z brokera a ukladanie hodnôt do databázy. Nakoniec som vytvoril ukážku využitia zbieraných informácií v podobe webovej aplikácie, ktorá sa dokázala adaptovať na používané prostredie a učiť riešenia na prípadné odchýlky.

Na záver by som rád podotkol, že celá komunikácia prebieha cez Internet. Z charakteru práce preto toto riešenie nie je vhodné pre:

- monitorovanie a riadenie systémov kritických na odozvy v milisekundách – pri komunikácii cez Internet je potrebné počítať s približne sekundovou odozvou,
- monitorovanie a riadenie systémov rizikových z hľadiska bezpečnosti – pri Internete je vždy potrebné počítať s možnosťou krátkodobého výpadku,
- monitorovanie systémov, pri ktorých je potrebné zbierať dáta v pravidelných intervaloch – aj keď je možné do smart senzora implementovať odoslanie dát na vyžiadanie, jeho primárnym účelom je posielanie dát v prípade potreby.

A naopak, riešenie je vhodné najmä pre:

- monitorovanie javov meniacich sa v nepravidelných intervaloch,
- monitorovanie javov, pri ktorých využívajú senzory s obmedzenými zdrojmi,
- systémy, kde je potrebná komunikácia na diaľku,
- siete so spoľatným objemom prenosu dát,
- systémy, pri ktorých je žiadúci určitý stupeň automatizácie.

Pri zavádzaní nových systémov je vždy potrebné ujasniť si požiadavky a následne zhodnotiť klady a zápory ponúkaných možností. Málokedy totiž existuje univerzálne riešenie na všetko. Podobne je to aj pri tejto práci, sú oblasti, kde sa tento návrh hodí viac a kde menej. V každom prípade ale nepochybujem o tom, že by si pri aktuálnych trendoch rozvoja IoT a Priemyslu 4.0 dokázal nájst svoje uplatnenie v širokej oblasti použitia.

5. Záverečné zhrnutie

Počas vypracovania tejto diplomovej práce boli vykonané nasledovné kroky:

- Analýza najnovších trendov v oblasti priemyslu a zbere dát
- Popis najpoužívanejších komunikačných protokolov v spojitosti IoT a Priemyslom 4.0
- Analýza a návrh architektúry pre zber dát a riadenie na základe smart senzorov
- Vytvorenie smart senzoru ako zariadenie schopné posielat' hotové informácie
- Implementovanie komunikačného protokolu MQTT na prenos správ
- Vytvorenie databázy aj s potrebnými tabuľkami
- Naprogramovanie windows služby ako spôsob na ukladanie prijímaných hodnôt do databázy
- Vizualizácia využitia dát v rámci webovej aplikácie schopnej adaptovať sa na prostredie a zaznamenávať riešenia pri vzniku neštandardných situácií
- Definovanie triggerov v databáze pre potreby implementácie algoritmov využívaných webovou aplikáciou
- Vyhodnotenie vypracovaného riešenia a návrh možností na ďalšie pokračovanie

Záver

Ako bolo v tejto práci ukázané, vzrast počtu zaradení pripojených na Internet bude hrať v najbližšom období prím. Priemyselná revolúcia 4.0 spolu s Internetom vecí vytvorí prostredie, v ktorom sa otvoria brány na príchod nových myšlienok a nápadov. Je načas vytvárať návrhy, ktoré pôjdu za hranice tradičných riešení používajúcich sa už niekoľko desaťročí. Nemusí ísť pritom nutne o ich náhradu, ale skôr o rozšírenie a ponúknutie nových možností.

Cieľom štúdie bolo ukázať architektúru zahrňujúcu najnovšie trendy. Kľúčovým prvkom bol smart senzor, ktorý je schopný posilať hotové informácie a to na základe výpočtu entropie z generovaných dát. Stav snímaného objektu je teda odoslaný iba vtedy, ak to pre nás má význam. Takéto riešenie nielenže znižuje zaťaženie siete, ale takisto šetrí zdroje senzoru a zásobuje systém hodnotami, ktoré sú dôležité pre ďalšiu analýzu. Smart senzor pritom ako komunikačný kanál využíva protokol navrhnutý špeciálne pre Internet vecí a to konkrétne MQTT. Ten funguje na báze publish/subscribe a posielanie správ prebieha cez bránu nazývanú broker. Vďaka tomu je umožnená obojstranná komunikácia so senzorom a správy je možné posilať aj na veľké vzdialenosti. Z brokera je potom možné informácie odoberať a ukladať ich do databázy. Takto je zaručený zber a uloženie dát v súlade s najmodernejšími kritériami Internetu vecí. V záverečnej časti bola navyše prezentovaná ukážka možnosti využitia dát v prospech adaptácie systému na prostredie, v ktorom je implementovaný. Celkovo bola teda vytvorená kompletná architektúra, ktorá rieši problematiku od generovania dát až po ich použitie.

Ako pri každej novej téme, aj tu je mnoho možností na rozšírenie. Vďaka využitiu protokolu umožňujúceho komunikáciu v oboch smeroch je tu dobrý predpoklad na implementáciu ďalších funkcionalít. Niektoré konkrétne návrhy boli spísané v práci, napríklad prijímanie príkazov smart senzorom z vyšších úrovní systému, alebo vytvorenie zdieľanej inteligencie medzi senzormi navzájom, ale priestor na nové myšlienky je prakticky neobmedzený. Mojim zámerom bolo poukázať na referenčný vplyv Internetu vecí v priemysle ako na potenciálny zdroj novej pridanej hodnoty.

Zoznam použitej literatúry

- [1]. BRETTEL, Malte, et al.: How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective. World Academy of Science, Engineering and Technology, 2014, Vol. 8.
- [2]. Accenture Digital Industry 4.0. Accenture. 2014 [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://www.accenture.com/microsites/digital-industry/Pages/home.aspx#/digital>>.
- [3]. JAZDI, N. Cluj-Napoca: Cyber physical systems in the context of Industry 4.0. IEEE, 2014. ISBN 978-1-4799-3731-8.
- [4]. DAUGHERTY, Paul, BANERJEE, Prith and ALTER, Allan E.: Accenture IIoT. Accenture. 2014 [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Driving-Unconventional-Growth-through-IIoT.pdf>>.
- [5]. SHROUF, F., ORDIERES, J. and MIRAGLIOTTA, G.: Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. Bandar Sunway : IEEE, 2014.
- [6]. SCHNEIDER, Stan: Understanding The Protocols Behind The Internet Of Things. Electronic Design. 9. Október 2013 [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://electronicdesign.com/embedded/understanding-protocols-behind-internet-things>>.
- [7]. International Business Machines Corporation (IBM); Eurotech. MQTT V3.1 Protocol Specification. IBM. [Online] 2010. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>>.
- [8]. SHELBY, Z., et al.: IETF Tools. IETF Tools. Jún 2014 [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://tools.ietf.org/pdf/rfc7252.pdf>>. ISSN 2070-1721.
- [9]. XMPP Standards Foundation. XMPP Technology Overview. XMPP. 10. Október 2010 [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://xmpp.org/about-xmpp/technology-overview/>>.
- [10]. JAFFEY, Toby: MQTT and CoAP, IoT Protocols. Eclipse. Február 2014 [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <http://eclipse.org/community/eclipse_newsletter/2014/february/article2.php>.

- [11]. DUFFY, Paul: Beyond MQTT: A Cisco View on IoT Protocols. Cisco Blogs. 30. Apríl 2013 [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://blogs.cisco.com/ieo/beyond-mqtt-a-cisco-view-on-iot-protocols>>.
- [12]. OPC Foundation. OPC Unified Architecture. Prosysopc. [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<https://downloads.prosysopc.com/downloads/opc-brochure-pioneer-v2.pdf>>.
- [13]. FREJBORG, Andreas, et al.: Top 10 reasons why to choose OPC UA over OPC. Prosysopc. Máj 2013 [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <https://downloads.prosysopc.com/downloads/automation_xx_seminar_opcua_connects_your_systems.pdf>.
- [14]. ZOLOTOVÁ, Iveta and LOJKA, Tomáš: Online Data Stream Mining in Distributed Sensor Network. Košice : WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS, 2014, Vol. 13. E-ISSN 2224-266X.
- [15]. Eclipse paho. Eclipse. [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<https://eclipse.org/paho/>>.
- [16]. Eclipse Mosquitto. Eclipse. [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://eclipse.org/mosquitto/>>.
- [17]. Eclipse paho Android. Eclipse. [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<https://eclipse.org/paho/clients/android/>>.
- [18]. IoT Eclipse Sandboxes. IoT Eclipse. [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://iot.eclipse.org/getting-started#sandboxes>>.
- [19]. Eclipse Mosquitto Download. Eclipse. [Online]. [cit. 25. Apríl 2015]. Dostupné na internete: <<http://eclipse.org/mosquitto/download/#windows>>.

Prílohy

- Príloha A: CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe
- Príloha B: Používateľská príručka
- Príloha C: Systémová príručka