

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-5382-72737

**VYUŽITIE METODIKY IOT V INTELIGENTNÝCH  
BUDOVÁCH**  
**BAKALÁRSKA PRÁCA**

**2016**

**Marek Hrebík**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-5382-72737

**VYUŽITIE METODIKY IOT V INTELIGENTNÝCH  
BUDOVÁCH**  
**BAKALÁRSKA PRÁCA**

Študijný program: Aplikovaná informatika  
Číslo študijného odboru: 2511  
Názov študijného odboru: 9.2.9 Aplikovaná informatika  
Školiace pracovisko: Ústav automobilovej mechatroniky  
Vedúci záverečnej práce: prof. Ing. Štefan Kozák, PhD.

**Bratislava 2016**

**Marek Hrebík**



## ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Marek Hrebík**  
ID študenta: **72737**  
Študijný program: **Aplikovaná informatika**  
Študijný odbor: **9.2.9. aplikovaná informatika**  
Vedúci práce: **prof. Ing. Štefan Kozák, PhD.**  
Konzultant: **Ing. Anton Pytel**  
Miesto vypracovania: **Ústav automobilovej mechatroniky**

Názov práce: **Využitie metodiky IoT v inteligentných budovách**

Špecifikácia zadania:

Cieľom bakalárskej práce je aplikácia najnovších poznatkov v oblasti IoT (Internet of things) pre optimalizáciu procesov v inteligentných budovách.

Úlohy:

1. Analyzujte možnosti využitia IoT, so zameraním na inteligentné budovy.
2. Navrhnite metódy a postupy využitia IoT pre automatizáciu inteligentných budov.
3. Navrhnite systém komunikácie medzi jednotlivými zariadeniami a prvkami v inteligentných budovách (senzory, akčné členy a pod.)
4. Overte prototypovaciu platformu Command Fusion s využitím javascriptu a navrhnite komunikáciu so zariadením LogicMachine, ktoré využíva Lua script a slúži na ovládanie zariadení v inteligentnej budove.

Zoznam odbornej literatúry:

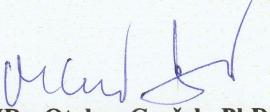
1. Puškár, B. Vznik konceptu inteligentnej budovy. In *Vplyv nových technológií a konštrukcií na tvorbu architektonických diel V : Medzinárodný vedecký zborník recenzovaných vedeckých prác*. 1. vyd. Bratislava: Nakladateľstvo STU, 2012, s. 29–33. ISBN 978-80-227-3850-7.
2. Petrás, D. Climate control. In *Intelligent build and design innovations 2004: Creating the building blocks for forward-looking design*. London : SPG Media Limited, 2004, s. 66–67. ISBN 1-85938-571-0.

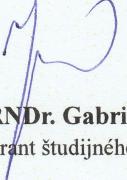
Riešenie zadania práce od: 21. 09. 2015  
Dátum odovzdania práce: 20. 05. 2016



**Marek Hrebík**  
študent



  
**prof. RNDr. Otokar Grošek, PhD.**  
vedúci pracoviska

  
**prof. RNDr. Gabriel Juhás, PhD.**  
garant študijného programu

# SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:

Aplikovaná informatika

Autor:

Marek Hrebík

Bakalárská práca:

Využitie metodiky IoT v inteligentných budovách

Vedúci záverečnej práce:

prof. Ing. Štefan Kozák, PhD.

Miesto a rok predloženia práce:

Bratislava 2016

Práca obsahuje popísané riešenia Internet of Things. Vysvetluje pojem Internet of Things, jeho spektrum využitia. Vymenúva a popisuje spôsoby komunikácie zariadení, vysvetľuje ich štruktúru s ohľadom na použitý middleware a jeho požiadavky. V práci je vysvetlený postup realizácie ovládania objektov v inteligentných budovách pomocou prototypovacej platformy CommandFusion a zariadenia LogicMachine. Rieši vytváranie komunikácie medzi týmito zariadeniami a podáva výklad tohto riešenia. Sústreduje sa na riešenie prostredníctvom HTTP a TCP protokolov. V práci sa pojednáva o riešení komunikácie klient-server a aplikačnej logike. Obsahuje aj stručné zhodnotenie výsledkov a víziu do ďalšieho postupovania.

Kľúčové slová: Internet of Things, LogicMachine, CommandFusion, automatizácia, inteligentné budovy, Lua, JavaScript

# ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA  
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Author:	Marek Hrebík
Bachelor Thesis:	Use of the methodology Internet of Things in intelligent buildings
Supervisor:	prof. Ing. Štefan Kozák, PhD.
Place and year of submission:	Bratislava 2016

Thesis includes described solutions of Internet of Things. Explains concept Internet of Things, its range of use. Thesis enumerates and describes how can devices communicate, explains their structure with respect to used middleware. It is explained realisation process of objects control in smart buildings using prototyping platform CommandFusion and device LogicMachine. Deals with communication creation between this devices and explains interpretation of our solution. Thesis focuses on solutions using HTTP and TCP protocols, client-server communication and application logic. Thesis contains also brief assessment of results and vision for the next process.

Keywords: Internet of Things, LogicMachine, CommandFusion, automation, smart buildings, Lua, JavaScript

## **Podakovanie**

Chcem sa podakovať vedúcemu záverecnej práce, ktorým bol prof. Ing. Štefan Kozák, PhD., konzultantovi Ing. Antonovi Pytelovi za odborné vedenie, rady a prípomienky, ktoré mi pomohli pri vypracovaní tejto bakalárskej práce ako aj Petrovi Verešovi a firme Intelectric s.r.o. za poskytnutie zariadení na realizáciu tejto práce.

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Metodológia Internet of Things</b>	<b>2</b>
1.1 Vznik pojmu Internet of Things . . . . .	2
1.2 Využiteľnosť . . . . .	2
1.3 História . . . . .	3
<b>2 Zariadenia a prepojiteľnosť</b>	<b>5</b>
2.1 Senzory . . . . .	5
2.2 Akčné členy . . . . .	6
2.3 Agregátory . . . . .	6
2.3.1 Riadiace aplikácie . . . . .	6
2.4 Možnosti komunikácie . . . . .	9
2.4.1 Model OSI/ISO . . . . .	10
2.4.2 Lokálne pripojenie . . . . .	11
2.4.3 Pripojenie do siete Internet . . . . .	15
2.5 Middleware . . . . .	18
2.5.1 Požiadavky . . . . .	18
2.6 Architektúry . . . . .	21
<b>3 Nástroje na praktickú realizáciu</b>	<b>23</b>
3.1 CommandFusion . . . . .	23
3.1.1 guiDesigner . . . . .	23
3.1.2 iViewer . . . . .	26
3.1.3 JavaScript API . . . . .	26
3.2 LogicMachine . . . . .	26
3.2.1 Objekty . . . . .	26
3.2.2 Lua script . . . . .	27
3.2.3 Grafické rozhranie LogicMachine . . . . .	28
<b>4 Implementácia</b>	<b>30</b>
4.1 Spôsoby komunikácie . . . . .	30
4.1.1 Tvorba HTTP requestu . . . . .	30
4.1.2 Tvorba TCP pripojenia . . . . .	32
4.1.3 Listener . . . . .	33

4.1.4	Zariadenia ako klient a server . . . . .	33
4.2	Aplikačná logika . . . . .	35
4.2.1	Problém join - skupinová adresa . . . . .	35
4.2.2	Problém dátových typov . . . . .	36
4.3	Zhodnotenie praktickej časti . . . . .	36
<b>Záver</b>		<b>37</b>
<b>Zoznam použitej literatúry</b>		<b>38</b>
<b>Prílohy</b>		<b>I</b>
<b>A</b>	<b>Skupinová adresa na join</b>	<b>II</b>
<b>B</b>	<b>Listener v LogicMachine</b>	<b>III</b>
<b>C</b>	<b>Funkcia odosielajúca stavy tagovaných zariadení</b>	<b>IV</b>
<b>D</b>	<b>Funkcia na prevod joinu na skupinovú adresu</b>	<b>V</b>

## Zoznam obrázkov

Obrázok 1	Spektrum využiteľnosti IoT[18]	3
Obrázok 2	Architektúra technológií Internet of Things	9
Obrázok 3	Diagram HTTP komunikácie	14
Obrázok 4	Schéma LoraWAN komunikácie	18
Obrázok 5	Schéma troch rôznych architektúr	22
Obrázok 6	Vzhľad hlavného panelu	29
Obrázok 7	Nastavenie TCP klienta v CommandFusion	32
Obrázok 8	Sekvenčný diagram nami navrhnutej komunikácie	34

## **Zoznam skratiek a značiek**

IoT - Internet of Things  
HTTP - Hypertext Transfer Protocol  
TCP - Transmission Control Protocol  
USB - Universal Serial Bus  
API - Application Programming Interface  
IEEE - Institute of Electrical and Electronics Engineers  
MIT - Massachusetts Institute of Technology  
D2D - Device-to-Device  
PIR - Passive Infrared Sensor  
GPS - Global Positioning System  
AMD - Advanced Micro Devices  
SD - Secure Digital  
HTML - HyperText Markup Language  
LAN - Local Area Network  
RAM - Random-access memory  
IP - internet protocol  
SLC - Single-level cell  
AV - audio/video  
DIN - Deutsche Industrie-Normen  
DALI - Digital Addressable Lighting Interface  
OSI - Open Systems Interconnection  
ISO - International Organization for Standardization  
MAC - Media Access Control  
LLC - Logical Link Control  
ID - Identification  
PAN - Personal Area Network  
PLC - Programmable Logic Controller  
SSL - Secure Sockets Layer  
W3C - World Wide Web Consortium  
IETF - Internet Engineering Task Force  
GPRS - General Packet Radio Service  
HSDPA - High-Speed Downlink Packet Access  
LTE - Long Term Evolution

LoraWAN - Low Power Wide Area Network

URL - Uniform Resource Locator

ASCII - American Standard Code for Information Interchange

UDP - User Datagram Protocol

## Zoznam algoritmov

1	Ukážka HTTP requestu	31
2	Vytvorenie socketu	34
A.1	Skupinová adresa na join	II
B.1	Listener v LogicMachine	III
C.1	Funkcia odosielajúca stavy tagovaných zariadení	IV
D.1	Funkcia na prevod joinu na skupinovú adresu	V

# Úvod

Metodológia Internet of Things sa čoraz viac dostáva do každodenného života každého človeka. Pomocou tej dokážeme šetrif životné prostredie, uľahčiť si prácu, ako aj pomáhať okoliu. Uplatnenie si nachádza čoraz v širšom kontexte v každom odvetví. Práve preto by sme radi popísali túto metodológiu.

V tejto práci sa pokúsime popísať Internet of Things, ako vznikol, kde a ako dokážeme využívať túto metodológiu a pozrieme sa aj na kúsok histórie. Našou snahou bude objasniť možnosti pripojiteľnosti Internet of Things, definujeme senzory, akčné členy, agregátory a riadiace aplikácie nad nimi. Popíšeme niektoré z možností prepojenia zariadení ako aj pripojenie do siete Internet. Zadefinujeme si vlastnosti vhodného middlevéru a porovnáme základné architektúry. Tieto poznatky nám budú dobrým základom k vytvoreniu vhodnej komunikácie medzi zariadením LogicMachine a prototypovacou platformou CommandFusion. Predstavíme si bližšie LogicMachine, štruktúru objektov, Lua skript rozhranie na tvorbu vlastných skriptov ako aj grafické rozhranie na pohodlné ovládanie. Samozrejme povieme si aj niečo o CommandFusion, ich nástroji guiDesigner a aplikáciu iViewer, aké objekty je možné pridávať do grafického rozhrania a akým spôsobom ich donútiť komunikovať. Poukážeme aj na JavaScript API. Nakoniec prejdeme k realizácii a tvorbe pripojení pomocou HTTP a TCP protokolu. Určíme si aplikačnú logiku, ako by malo naše riešenie fungovať. Pokúsime sa vymyslieť spôsob mapovania joinov a skupinových adries. Na záver si zhodnotíme naše riešenie a zamyslíme sa nad možnosťami rozšírenia a ďalšej aplikácie.

# 1 Metodológia Internet of Things

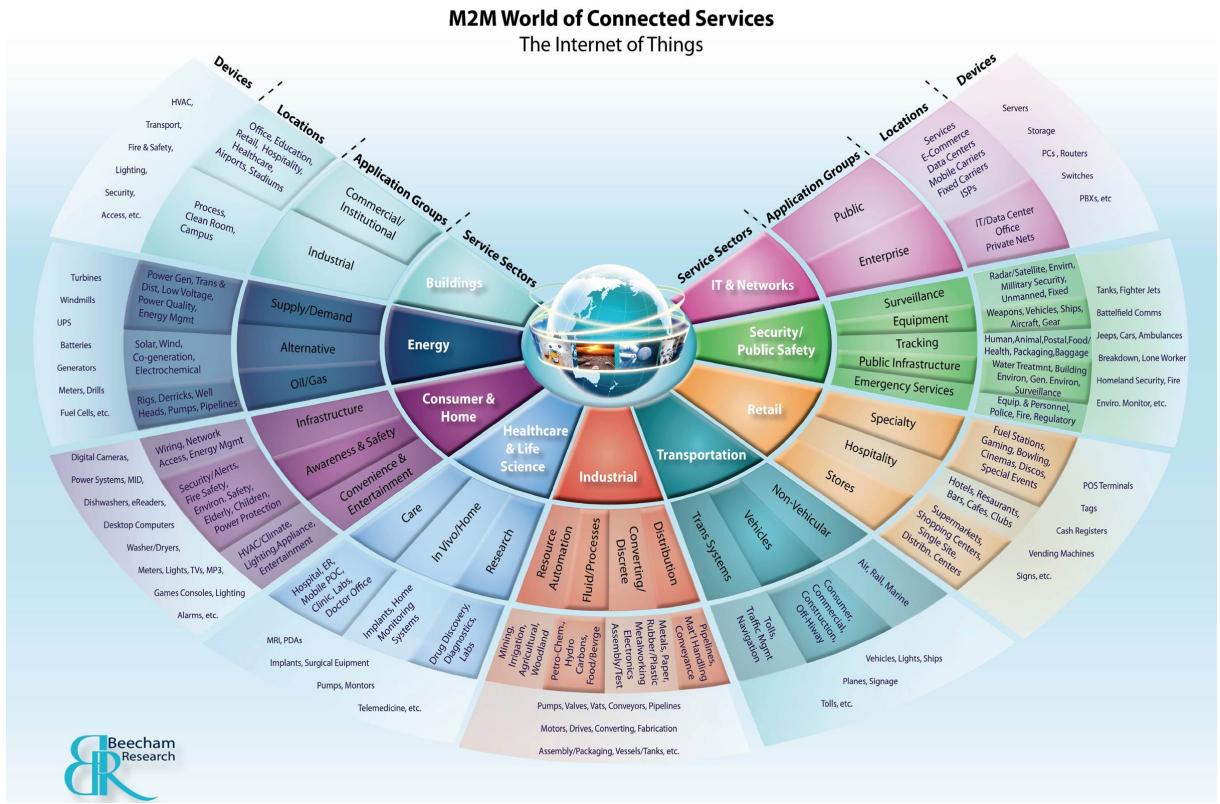
Internet of Things je sieť objektov (vozidiel, budov alebo iných zariadení), ktoré obsahujú elektroniku, softvér alebo senzory schopné vysielať a prijímať signál medzi nimi. Toto internetové pripojenie zabezpečuje zber a výmenu dát v takomto systéme.[47] IoT zabezpečuje ovládanie objektov naprieč existujúcou sietovou infraštruktúrou. Vytvára možnosti integrovania počítačových systémov do reálneho sveta, čo sa odráža na efektivite ako aj ekonomických výhodách.[48] Princípom IoT je využitie senzorov, akčných členov v objektoch, ich konektivita, logika a jej aplikácia na týchto objektoch. Táto technológia je časťou všeobecnejšej triedy cyber-fyzikálnych systémov. Každý objekt je jedinečne identifikovateľný prostredníctvom výpočtového systému v rámci jednej internebovej infraštruktúry.[35] Touto metodológiou sa budeme zaoberať v celej práci, z tohto dôvodu je potrebné objasniť a popísat ako vznikla, popísat časti z ktorých sa skladá, možnosti prepojenia jendotlivých senzorov, akčných členov a agregátorov.

## 1.1 Vznik pojmu Internet of Things

Pojem Internet of Things prvýkrát použil britský podnikateľ Kevin Ashton v roku 1999 ako spoluzakladateľ Auto-ID Labs. Vtedy mal na mysli globálnu siet objektov prepojených pomocou rádfrekvenčnej identifikácie.[50] IoT ponúka pokročilejšiu konektivitu zariadení, systémov a služieb, ktorá presahuje komunikáciu Machine to Machine a pokrýva celý rad protokolov, domén a aplikácií.[39] Prepojenie týchto vstavaných zariadení má využitie na poli automatizácie takmer vo všetkých smeroch, zároveň umožňuje aplikovanie napríklad inteligentných sietí, inteligentných budov a miest.[41]

## 1.2 Využiteľnosť

Slovo "Things" v IoT odkazuje na širokú škálu zariadení využiteľných pri monitorovaní srdcových implantátov. Uplatnenie nájdu aj rôznorodé čipy na hospodárske zvieratá, elektronické senzory v moriach, zabudované senzory v automobiloch a iných objektoch, monitoring patogénov [34], prevádzka zariadení v teréne či na vojenské, alebo záchranné účely.[44] Zároveň ho môžme definovať aj ako neoddeliteľnú zmes hardvéru, softvéru, dát a služieb. [40] Tieto zariadenia zbierajú užitočné informácie pomocou existujúcej technológie a následne dáta nezávisle posielajú do ďalších zariadení.



Obrázok 1: Spektrum využiteľnosti IoT[18]

### 1.3 História

Koncepcia siete inteligentných zariadení bola spomenutá už v roku 1982, kde na Carnegie Mellon University využili upravený automat na Coca-Colu, ktorý bol ako prvé zariadenie tohto druhu pripojený na internet. Toto zariadenie odosielalo informáciu o svojej zásobe nápojov, ktoré malo k dispozícii, ako aj informáciu o tom, či vydaný nápoj bol studený. [33] V súvislosti s históriou IoT je známy referát Marka Weisera z roku 1991 o všadeprítomnej výpočtovej technike s názvom "Počítač 21. storočia", ktorý hovorí o tom, že počítačové výpočty a technológia sa môžu objaviť kdekoľvek. Zmienil sa aj o vízii IoT, takisto ako aj spoločnosti Percom, UbiComp.[38, 49] V roku 1994 popísal Reza Raji koncept IoT v magazíne *IEEE Spectrum*, ako pohyb malých dát do veľkej množiny uzlov, ako možnosť automatizovať všetko od malých domácností až po továrne gigantických rozmerov.[43] V rozmedzí rokov 1993-1996 predstavil Microsoft ich riešenie s názvom *at Work*, ktoré malo prepojiť zariadenia vo firme prostredníctvom komunikačných protokolov, na umožnenie ovládania a informácie o pripojených zariadeniach na operačnom systéme Microsoft Windows. V roku 1999 Bill Joy, spoluzakladateľ

*Sun Microsystems* predstavil komunikáciu Device-to-Device (D2D), ktorý prezentoval na *World Economic Forum* v meste Davos vo Švajčiarsku.[42] V tomto roku sa stal koncept IoT populárny, a to vďaka *Auto-ID Center* na Massachussetskej Univerzite (MIT) a ich publikáciách trhovej analýzy.

## 2 Zariadenia a prepojiteľnosť

Je potrebné objasnenie pojmov senzor, akčný člen, agregátor, pripojiteľnosť a uvedenie príkladu. V tejto časti si podrobnejšie predstavíme tieto prvky. Definujeme senzory, ich typy, použitie, taktiež akčné členy. Pri agregátoroch okrem popisu uvedieme aj konkrétné riešenia od rôznych výrobcov a nakoniec si predstavíme protokoly a typy komunikácie, akou môžu tieto zariadenia byť spojené.

### 2.1 Senzory

Senzor alebo snímač je technické zariadenie fyzickej vrstvy, ktorého úlohou je snímať a reagovať na zmeny prostredia a veličiny, ktorú meria. Senzor premieňa úžitkovú energiu na elektrické dátu. Zmenu tejto veličiny senzor zaznamená a odošle ju ďalej ako výstup. Tieto hodnoty ďalej spracujú agregátor a posielajú informáciu do akčných členov, ktoré na základe hodnôt vykonajú predom definovanú akciu. Senzor premieňa úžitkovú energiu na elektrické dátu. Pod pojmom si môžeme predstaviť širokú škálu zariadení, ktoré merajú rôzne veličiny. Typy senzorov:

- Tlakové senzory
- Teplotné senzory
- Ultrazvukové senzory
- Senzory vlhkosti
- Senzory plynov - chemické
- PIR senzory pohybu
- Senzory zrýchlenia
- Senzory farieb
- Gyroskopické senzory
- Pohybové senzory

V inteligentných budovách nájde využitie každý z menovaných typov senzorov. Tlačidlá, meteostanice, ktoré majú viacero senzorov, ktoré merajú teploty, tlak, vlhkosť, pohybové senzory, PIR senzory. Existujú zariadenia, ktoré obsahujú senzory, akčné členy a zároveň aj agregátor. Takýmto príkladom môže byť mobilný telefón s aplikáciou Waze, ktorá využíva GPS senzor v mobile, zároveň zobrazuje dátu na displeji a sprostredkuje komunikáciu pomocou pripojenia na internet. V tomto prípade môžme povedať, že aj vodič auta využívajúci službu Waze je akčný člen, ktorý mení trasu na základe dát z aplikácie, napríklad pri dopravných zápchach. Do budúcnosti si vieme predstaviť, že takáto aplikácia by ovplyvňovala nie vodiča, ale priamo vozidlo tým, že by samostatne navigovala vozidlo najkratšou trasou, teda dokázala by ovládať aj servomotor na riadení.

## 2.2 Akčné členy

Akčný člen, alebo *actuator* je zariadenie fyzickej vrstvy, ktoré zodpovedá za odozvu na hodnoty, ktoré sú odmerané senzormi. Vo väčšine prípadov je ovládanie akčných členov sprostredkovane prostredníctvom agregátora. Tento akčný člen transformuje formu energie, zväčša elektrický prúd, ale aj vzduchový tlak na ďalšiu akciu. Napríklad stlačením tlačidla na vytiahnutie žalúzií na stene sa odošle signál do agregátora, ktorý ho ďalej sprostredkuje akčnému členu (motor na žalúziách) a ten sa na základe tohto signálu spustí a vykoná akciu vytiahnutia žalúzií. V riešení problematiky inteligentných domov potrebujeme akčné členy, ktoré nám dajú do pohybu žalúziu, zapnú svetlo, kúrenie na základe žiadanej teploty. Teda akčný člen môžme nazvať mechanizmom, ktorým kontrolný systém - agregátor pôsobí na objekty nachádzajúce sa v prostredí, ktoré je takto prepojené.

## 2.3 Agregátory

Agregátory sú zariadenia, ktoré patria do systémovej vrstvy, spájajú akčné členy a senzory prostredníctvom pripojení predstavených v časti 2.4.2, 2.4.3 a odosielajú dátu do riadiacej aplikácie prostredníctvom rôzneho middleveru. Ich úlohou je zoskupiť dátu zo senzoru, spracovať ich a odoslat do akčného členu, ktorý spracuje tieto dátu a vykoná akciu. Niektoré agregátory odosielajú tieto dátu do cloudu alebo do riadiacej aplikácie, ktorá môže byť súčasťou agregátora. V inteligentných budovách je agregátor hlavným prvkom, ktorý spravuje všetky objekty, uchováva informácie o ich stavoch, hodnotách, prijíma signály zo senzorov, spracuje ich a preposiela do akčných členov. Každé zariadenie pripojené v agregátori je jedinečne identifikované.

### 2.3.1 Riadiace aplikácie

V súčasnosti existuje niekoľko možností na výber riadiacej aplikácie, ktorá bude spravovať zariadenia v inteligentnom dome.

Pozrieme sa na niektoré z existujúcich riešení:

**Gira Homeserver** Toto zariadenie pre inteligentné budovy prepája elektrické inštalačie KNX systému do počítačovej siete a siete internet. Sprostredkuje pripojenie prostredníctvom sériového portu RS-232<sup>1</sup>, pripojenie do internetu zabezpečuje štandardný Ethernetový kábel, taktiež disponuje USB vstupom. Takto pripojené zariadenie je schopné ovládať inteligentný dom a objekty v ňom, kamery, audiosystémy, svetlá. Ich riešenie poskytuje aj aplikáciu pre iOS zariadenia, ktoré po pripojení

---

<sup>1</sup>Staršie rozhranie na prijatie počítača a ďalšej elektroniky, bity dát odosielala sériovo, teda postupne za sebou

sa do siete vedia prevziať kontrolu nad celým domom a všetkými ovládateľnými prvkami v ňom. Aplikácia poskytuje informácie o spotrebe energie, vody a plynu v domácnosti, prehľad počasia, taktiež umožňuje prepojenie s vchodovou kamerou a mikrofónom prostredníctvom plug-inov. Z hľadiska zabezpečenia dokáže odoslať informáciu o vlámaní, prípadne pohybe cudzieho objektu v priestore monitorovanom kamerami alebo senzormi pohybu. [7, 6]

**Jung Facility pilot** Zariadenie určené na trvalú prevádzku, montuje sa k rozvádzcačom na lištu. Je to vlastne malý počítač s AMD dvojjadrovým procesorom, štyrmi pripojeniami USB, sieťovým pripojením. Obsahuje aj slot na Micro-SD kartu a DVI pripojenie na obrazovku pre potreby vizualizácie. Vizualizáciu zariadenia je možné realizovať aj cez internetový prehliadač, ktorý podporuje HTML5. Na rozdiel od riešenia od firmy GIRA má dostupnú aplikáciu s názvom *Facility Pilot app* nielen na iOS platofmu, ale aj pre Android smartfóny. Má dve KNX rozhrania, z ktorých je jedno využívané na vizualizáciu celého systému a lepšiu údržbu KNX systému, a druhé KNX rozhranie je určené na komunikáciu prostredníctvom LAN ako KNX IP rozhranie. Oba tieto rozhrania bežia nezávisle, jedno na vizualizáciu vzdilenej údržby a druhé na prevádzku. Vďaka tomuto systému sa eliminujú akékoľvek ďalšie poplatky na dodatočnú kúpu ďalšieho rozhrania. [10]

**Vera Control Vera3 WiFi** a ovládanie domu v jednom. Takto prezentuje svoj produkt firma Vera Control. Okrem toho, že je to router podporujúci *WiFi 802.11n*, sa dokážete pripojiť do lokálnej siete prostredníctvom *Z-Wave*. Výrobca garantuje pripojiteľnosť viac ako 200 rôznych zariadení a aplikácií. Disponuje 500MHz procesorom MIPS SoC s RAM 128MB a pamäťou 64MB (32MB Flash NAND + 32MB SDRAM). Vďaka tejto pamäti si môžete ukladať scény a aplikácie pre váš dom. Zariadenie taktiež ponúka jednoduchú inštaláciu a ovládanie pre laikov, ovládanie kamier, vrátnikov, zabezpečenia, prehľad spotreby domácností, dôležité upozornenia. [27]

**Fibaro Home Center** Systém od poľskej spoločnosti Fibaro je mozgom automatizovaných systémov *Fibaro systems* alebo iných systémov ktoré komunikujú prostredníctvom pripojenia Z-Wave a na ovládanie pomocou externých zariadení využíva HTTP protokol. Obsahuje aj možnosť programovania správania objektov v jazyku Lua. Ich vizualizácia má intuitívne graficé rozhranie. Hardvérové vybavenie sa skladá z procesoru Intel Atom s frekvenciou 1,6 GHz s 1GB RAM a 2GB SLC pevného disku. Zaujímavostou je Recovery disk s kapacitou 4GB, ktorý zabezpečuje

udržiavanie zálohy systému. Technológia *Linked Devices* umožňuje kombinovať viacero zariadení do jedného virtuálneho zariadenia, čo zjednoduší ovládanie prostredníctvom jedného ovládacieho prvku. Zaujímavou funkcionalitou je geolokácia zariadení prostredníctvom GPS. V aplikácii na mape zobrazuje polohy ostatných ovládacích zariadení. Fibaro Home Center je možné rozšíriť o ďalšie zariadenia ich značky.[45]

**Samsung Hub** Samsung Hub je jeden z produktov rady *SmartThings*. Pomocou Hubu je možné ovládať senzory, svetlá, zámky, kamery. Komunikáciu zabezpečujú protokoly ZigBee, Z-Wave, WiFi a LAN pripojenie. Ovláda sa pomocou aplikácie, ktorú si nainštalujete do svojho Android, Windows Phone alebo iOS zariadenia. Inštalácia zariadenia je veľmi jednoduchá. [25]

**Loxone Miniserver** Ďalšie zo zariadení, Loxone Miniserver podporuje ovládanie inteligentných budov pomocou iOS, Android zariadení. Má podporu aj pre webové prehliadače. Pracuje aj s KNX zbernicou. Do siete sa dokáže pripojiť pomocou ich vlastnej technológie Loxone-Link a LAN. Loxone Air je technológia nahradzujúca Loxone-Link, pomocou ktorej dokáže zariadenie ovládať bezdrôtovo budovu a pripojiť zariadenia bez potreby pripájania a vedenia kabeláží, v Loxone Miniserveri je dostupná ako hardvérové rozšírenie. Dokáže spravovať AV Recievery, počítače, telefóny a ďalšie zariadenia, ktoré sú sieťovo orientované. Zabudovanie Miniserveru je riešené pomocou štandardnej DIN koľajnice. Miniserver ukladá všetky dátá lokálne, čo môžeme považovať za bezpečnejšie ako ukladanie do clouдовých služieb. [16]

**Embedded systems LogicMachine** Existuje viacero verzií LogicMachine, najnovšie a aktuálne modely sú napríklad LogicMachine Re:actor V3, LogicMachine 4. Je to ovládacia jednotka, ktorá podporuje vstupy formátov (záleží od typu zariadenia) RS-485, 1-wire, KNX TP1, USB 2.0<sup>2</sup>, EnOcean, DALI, Modbus. Základné funkcionality spočívajú v IP routingu, vizualizácii, programovaniu a vytváraniu brán medzi zariadeniami rôznych štandardov. Má funkcionalitu vzdialenej diagnostiky, a ovládania IP routerov ktoré sú zapojené v odlišných inštaláciách. Rôzne funkcionality na zaslanie e-mailu pri chybe zbernice, pretečeniu buffera, prípadne stavu pripojených zariadení. Obsahuje Redis in-memory databázu typu klúč-hodnota , do ktorej je možné ukladať rôzne objekty a ich hodnoty pre ďalšie použitie. Novou funkcionalityou, ktorá prišla s najnovším firmvérom je tvorba vizualizácie vrátane pridávania fontov, predprogramovaných prechodov pri zmene okna vo webovom prehliadači.

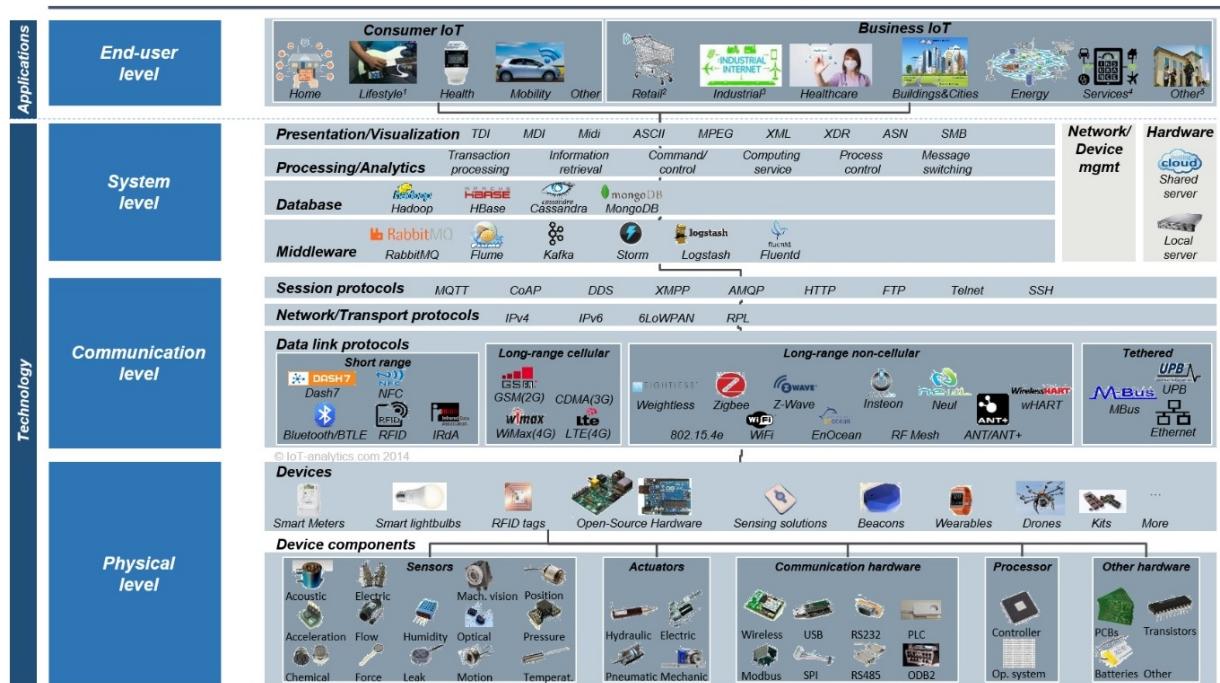
---

<sup>2</sup>Používa sa na ukladanie zálohy, databázy skriptov.

Tvorba vizualizácie je jednoduchá aj pre laikov, keďže systém má v sebe už zabudované šablóny, kde stačí pridať ďalšie objekty. Užívateľ si vie sám namodelovať ako bude vyzerat jeho ovládacia aplikácia pre jeho inteligentný dom. Programovanie je ďalšou z výhod tohto produktu. Využíva sa tu skriptovací jazyk Lua, v ktorom je možné vytvárať vlastné skripty na spracovanie dát ktoré prechádzajú cez LogicMachine a posielat ich do ďalších zariadení. Naprogramované skripty je možné využívať na základe udalostí, neustále ako rezidentné skripty, plánované skripty podľa času, kedy sa majú spustiť alebo ich uložiť do užívateľskej knižnice a využívať ich po časťach tam, kde to je potrebné. LogicMachine je zariadenie ktoré zjedocuje rozdielne systémy do jedného, ktorý dokáže spolupracovať so svojimi časťami. [14]

## 2.4 Možnosti komunikácie

Na to, aby sme mohli k inteligentným objektom pristupovať, potrebujeme v takomto prostredí mať určené, akým spôsobom budeme s týmito objektmi komunikovať. Teda ako získať a odosielat dátu z a do našej aplikácie, resp. nášho inteligentného objektu. Na to nám poslúžia komunikačné protokoly. Stručne sa budeme snažiť popísať tie najdôležitejšie z nich.



Obrázok 2: Architektúra technológií Internet of Things

### 2.4.1 Model OSI/ISO

Pred popisom jednotlivých typov a protokolov komunikácie potrebujeme ozrejmiť aj referenčný model OSI/ISO. *Open Systems Interconnection Reference Model* je abstraktný model ktorý charakterizuje a štandardizuje, akým spôsobom funguje komunikácia počítačových a telekomunikačných systémov. Tento model je rozdelený na sedem vrstiev. Cieľom tohto modelu je určiť, akým spôsobom majú komunikovať systémy v určitej vrstve.

**Fyzická vrstva** Do tejto vrstvy patria zariadenia ako huby a repeatre, ktoré pracujú s bitmi. Ich úlohou je komunikáciu začať a ukončiť. Rozhoduje sa či dátu budú odosielané ako digitálne alebo analógové.

**Linková vrstva** Konektivita na tejto vrstve je poskytovaná iba zariadeniam v lokálnych uzloch siete. V tejto vrstve sa prenášajú rámce medzi dvomi priamo spojenými uzlami. Skladá sa z MAC *Media Access Control* vrstvy, ktorá je zodpovedná za fyzické adresovanie, riadenie prístupu k médiu a LLC *Logical Link Control* vrstvy, poskytujúcej multiplexovanie protokolov, riadenie toku dát a obmedzenie, resp. zabezpečenie proti chybovosti.

**Sieťová vrstva** Poskytuje možnosti odosielania paketov s premenlivou dĺžkou medzi uzlami v tej istej sieti. Má za úlohu preklad logických adries na fyzické adresy zariadení. Každé zariadenie v tejto vrstve má svoju unikátnu adresu. Router je zariadenie pracujúce an tejto vrstve.

**Transportná vrstva** Účelom tejto vrstvy je poskytovanie transparentného prenosu dát medzi koncovými zariadeniami. Tieto dátu smerujú do sieťovej vrstvy, ktorá vie smerovať pakety z jedného zariadenia do zariadenia, ktorému sú tieto pakety adresované. V tejto vrstve sa odosielajú segmenty.

**Relačná vrstva** Kontroluje komunikáciu dvoch zariadení. Zriaďuje, ukončuje a spravuje túto komunikáciu. V tejto vrstve sa nadväzujú a ukončujú TCP/IP spojenia, sokyty.

**Prezentačná vrstva** Táto vrstva sa stará o kódovanie a reprezentáciu dát. Poskytuje sa tu prevod týchto dát do zrozumiteľnej reprezentácie pre zariadenie, ktorému sú dátu určené a zároveň jeho zabezpečenie. Taktiež zabezpečuje šifrovanie, dešifrovanie. Týmto odlaďuje aplikačnú vrstvu od spomenutých úkonov.

**Aplikačná vrstva** Je vrstva, ktorá je najbližšie ku koncovému užívateľovi, teda komunikuje priamo so softvérovou aplikáciou, ktorú užívateľ ovláda. Definuje akým spôsobom

sobom a v akom formátovaní majú byť dátá zaslané a prijímané. V tejto vrstve užívateľ konkrétnie vidí výsledok, ktorý sa transformoval prostredníctvom všetkých vrstiev modelu OSI/ISO.

Podľa definícií jednotlivých vrstiev tohto modelu si môžme lepšie predstaviť ako funguje komunikácia prostredníctvom rôznych typov pripojení a protokolov, ktoré budú priblížené v nasledujúcej časti.

#### 2.4.2 Lokálne pripojenie

Týmto pripojením máme na mysli pripojenie inteligentných objektov na agregátor s riadiacou jednotkou, ktorým budú tieť dátá. Je viacero rozhraní ktorými môžu takéto objekty komunikovať. Každé z nich má svoje špecifické vlastnosti. Pokúsime sa priblížiť niektoré z nich.

**KNX** Štandardizovaný komunikačný protokol pre automatizáciu budov. KNX umožňuje integráciu množstva zariadení od rôznych výrobcov. Tento štandard má aj podporu programovania týchto zariadení. Tento štandard je rozšírený celosvetovo u produktov týkajúcich sa automatizácie. Sieť KNX zariadení je možné ovládať počítačom ako aj miktoovládačom 8-bitovej architektúry. Prostredníctvom KNX zbernice je pripojené aj naše zariadenie, ktorého nastaveniu a programovaniu sa budeme venovať v praktickej časti. [11, 12, 13]

**DALI** *Digital Addressable Lighting Interface* definuje štandardizované rozhranie pre predradníky<sup>3</sup>, ktoré zaistuje zameniteľnosť a kompatibilnosť zariadení od rôznych výrobcov. DALI komponentmi je možné vytvoriť energeticky efektívne a flexibilné systémy osvetľovania. Nevýhodou DALI rozhrania je že dokáže ovládať iba systémy osvetľovania. [4]

**RS-485** V súčasnosti tento typ pripojenia je vhodný a využívaný na riešenie prenosu dát na dlhé vzdialenosťi v zašumenejom prostredí. RS-485 dokáže pracovať vo vyvážených viacbodových systémoch. Prenos dát dokáže sprostredkovať v rýchlosťi 35 megabitov za sekundu do vzdialenosťi 10 metrov. Vzdialenosťou rapídne klesá jeho prenosová rýchlosť, čo nie je prekážkou v inteligentných domoch. [24]

**1-wire** Základom tejto technológie je použitie jedinej dátovej linky a uzemnenia za použitia sériového protokolu. 1-Wire *master* zariadenie kontroluje a inicializuje komunikáciu s ďalšími 1-Wire *slave* zariadeniami, ktoré sa pripájajú na 1-Wire zbernicu.

---

<sup>3</sup>Predradník je zariadenie, ktoré reguluje množstvo prúdu v elektrickom obvode (napr. stmievané svetlá)

Každé takto pripojené zariadenie je jedinečne identifikované pomocou továrenského 64-bitového identifikačného čísla (ID). Prvých 8 bitov tohto identifikátora napovedá o funkcií a typu zariadenia. Prostredníctvom 1-Wire sa dá komunikovať obojsmerne, polovičným duplexom.<sup>4</sup> [20]

**EnOcean** Je bezdrôtová, energiu šetriaca technológia prenosu informácií, využívaná v automatizácii budov, inteligentných domoch aj v iných odvetviach. Princípom EnOcean je komunikácia medzi zariadeniami s nízkou spotrebou energie. Ich technológia pracuje s minimálnym množstvom energie, ktorú prijíma zo svojho okolitého prostredia a konvertuje ju do použiteľnej energie na vysielanie signálov aj zo zariadení, ktoré nemajú zdroj energie ako batériu. Na prenesenie signálu na vzdialenosť 30 metrov stačí iba 50 mikrowattov. Zároveň podporuje bezdrôtové štandardy s nízkym odberom energie.[21, 22]

**X10** Protokol na automatizáciu, väčšinou využíva elektrické vedenie na signalizáciu a ovládanie, kde signály predstavujú krátke rádiovreckvenčné impulzy s frekvenciou 120 kHz transformované na digitálnu informáciu. X10 je citlivý na šumy v okolí kábla a prenos signálu trvá pol až tri štvrti sekundy.[29]

**Z-Wave** Novší protokol, ktorý používa na komunikáciu bezdrôtové pripojenie. Je navrhnutý na bezdrôtové ovládanie svetiel, stmievačov, bezpečnostných systémov a ďalších inteligentných objektov. Špecifikom tohto protokolu je komunikácia prostredníctvom nízkonapäťových rádio frekvencií v rámci sieťovej topológie *mesh*<sup>5</sup>. Pracuje v 1GHz pásmi a poskytuje rýchlosť komunikácie od 40 do 100 kbps. Systém využívajúci Z-Wave môže byť ovládaný vzdialene prostredníctvom siete internet a Z-Wave brány alebo riadiacej jednotky ktorá slúži aj ako ovládač.[1]

**ZigBee** ZigBee je protokol definovaný podľa špecifikácie IEEE 802.15.4 pre vysokoúrovňové komunikačné protokoly, ktoré sú využívané na vytváranie *Personal Area Network* (PAN) sietí v ktorých sa nachádzajú nizkospotrebové digitálne zariadenia, ktoré je možné ovládať. Dokážu prenášať dátu na diaľku 10 - 20 metrov. Rýchlosť prenosu je definovaná na 250kbit/s takže sa hodí na prenos dát z a do senzorov alebo vstupných zariadení, prenos je šifrovaný 128bitovou symetrickým šifrovacím kľúčom.[31]

**Modbus** Je sériový komunikačný protokol využívaný v PLC ovládačoch. Bežne sa používa na pripájanie priemyselných elektronických zariadení. Je jednoduchý na

<sup>4</sup>Obidve strany pripojenia sú schopné prijímať aj odosielat informácie avšak nie naraz.

<sup>5</sup>Je to sieťová topológia, kde môžu byť uzly prepojené s viac ako jedným zariadením

zavedenie a údržbu, má viacero verzií, pre účely inteligentných budov je najvhodnejšia verzia Modbus TCP/IP.[19]

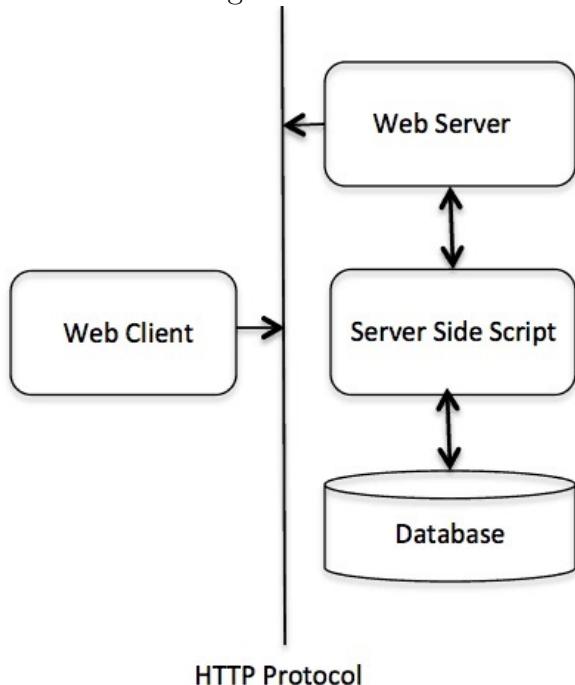
**Bluetooth** Otvorený komunikačný štandard pre bezdrôtové zariadenia, ktorý dokáže zpripojiť viacero zariadení naraz. Pôvodne bol ako náhrada za drôtové rozhranie RS-232, dnes predstavuje využitie v mobilných telefónoch, elektronike a aj v intelligentných budovách. Každé zariadenie v sieti Bluetooth je identifikované pomocou *Bluetooth Device Address*, čo môžme pripojiť k identifikácii zariadení Ethernet prostredníctvom MAC adresy. Prostredníctvom Bluetooth signálu môžeme prenášať streamy zvuku, hudby, videá (mobilný telefón prepojený na handsfree slúchadlá) Technológia je štandardizovaná podľa IEEE 802.11, Bluetooth signál dosahuje do diaľky 100 metrov v ideálnych podmienkach, teda spadá do kategórie Personal Area Network (PAN). Na komunikáciu využíva 2.4 GHz pásmo, kde dosahuje teoretickú prenosovú rýchlosť až 24 Mbit/s. Nevýhodou Bluetooth komunikácie je, že signál šírený cez stenu alebo iné prekážky rapídne klesá. Bluetooth komunikácia patrí do Aplikačnej vrstvy štandardu ISO/OSI. [26, 3]

**WiFi** Je to štandard (IEEE 802.11) pre bezdrôtové lokálne siete. Podobne ako Bluetooth využíva 2.4 GHz pásmo aj 5 GHz pásmo. WiFi pripojenie je v súčasnosti najvyužívanejšie pre prístup na internet pre notebooky, smartfóny. Nevýhodou WiFi sietí je, že pásmo 2.4 GHz je kvôli jeho rozšírenosti často rušené inými signálmi hlavne vo veľkých mestách, bytoch, kde má každá domácnosť vlastný WiFi router. V intelligentných budovách sa využíva taktiež často, pretože je to cenovo priateľná možnosť ako sa pripojiť so svojim zariadením do siete a ovládať tak objekty v budove. Pre potreby IoT je rýchlosť WiFi pripojenia viac ako postačujúca. Kedže je tento druh pripojenia rozšírený, v záujme bezpečnosti je vhodné toto pripojenie zabezpečiť heslom. WiFi siet pracuje na fyzickej a linkovej vrstve modelu ISO/OSI.[30]

**6LoWPAN** Skratka znamená *IPv6 over Low power Wireless Personal Area Networks*. Je to internetový štandard, ktorý využíva IPv6 protokol aj pre zariadenia s nízkou spotrebou. Takéto zariadenia majú limitovanú schopnosť spracovávania dát. 6LoWPAN majú definovaný mechanizmus na zapuzdrenie a kompresiu hlavičkových dát tak, aby mohli IPv6 pakety byť odoslané a prijímané prostredníctvom *low-rate wireless personal area network* (LR-WPAN).

**HTTP** Hypertext Transfer Protocol alebo HTTP protokol je aplikačný protokol pre distribuované informačné systémy, ktoré obsahujú grafické, zvukové, video a tex-

Obrázok 3: Diagram HTTP komunikácie



tové odkazy. Je to základom dátového prenosu a komunikácie na sieti World Wide Web (WWW). Využíva sa na výmenu dokumentov vo formáte Hypertext Markup Language (HTML).[36] Vývoj štandardov HTTP protokolu je riadený organizáciou *World Wide Web Consortium* (W3C) a *Internet Engineering Task Force* (IETF). Tieto štandardy a ich aktualizácie sú obsiahnuté v sérii dokumentov *Request for Comments* (RFC). Posledná verzia HTTP/2 bola štandardizovaná v roku 2015.

Základom HTTP protokolu je odosielanie požiadaviek a čakanie na odpoveď. Komunikácia prebieha medzi klientom a serverom. Príkladom nám môže byť aplikácia, ktorá odošle na inteligentné zariadenie požiadavku o zaslanie hodnoty, inteligentné zariadenie ju príjme, načíta hodnotu a odošle ju ako správu s odpovedou. [36]

HTTP protokol má tri základné vlastnosti, ktoré ho robia jednoduchým. Tieto vlastnosti takisto platia aj pre HTTPS protokol, ktorý je zabezpečený pomocou SSL.

**HTTP vytvára jednorázové pripojenie** HTTP klient vytvorí požiadavku, hned po tom, ako sa pripojí na server, odošle požiadavku, ktorá sa spracuje na strane servera. Týmto pripojením sa odošle zo serveru odpoveď na požiadavku, akonáhle sa odpoveď dostane ku klientovi, toto pripojenie sa zruší. Pri ďalšom kontaktovaní klienta na ten istý server sa musí vytvoriť nové pripo-

jenie. Táto vlastnosť HTTP pripojenia zabraňuje zahľteniu servera jediným klientom. Takto vybavuje server požiadavky postupne podľa času pripojenia klienta.

**HTTP je nezávislé od typu dát** Môžme posielat akékoľvek dát prostredníctvom HTTP protokolu, či už je to video, obrázok, hudba, čistý text alebo iné, pokiaľ klient aj server vedia ako tento dátový obsah spracovať. Či ho už priamo prehrať alebo zobraziť v prehliadači, alebo ho stiahnuť. Pri HTTP protokole sa vyzaduje špecifikovanie odosielaných dát pomocou *Media type* (MIME type). Napríklad pri odosielaní HTML stránky je to *text/html*.

**HTTP je bezstavový protokol** To, že HTTP protokol nezachováva stav vyplýva z toho, že vytvára jednorázové pripojenia. Každé pripojenie klienta na server je ako pre serverovskú tak aj pre klientsku stranu nové, kedže po každom odoslaní odpovede server zruší pripojenie a neukladá si jeho referenciu. Vďaka tejto vlastnosti HTTP protokolu nemôže ani klient ani server uchovávať alebo prenášať informácie medzi rôznymi požiadavkami z rôznych zdrojov.

Samozrejme v súčasnosti tieto vlastnosti HTTP protokolu sú modifikované rôznymi technológiami, napríklad Cookies ukladajú dátu pri pripojení pomocou HTTP protokolu, tieto dátu sa ďalej vedia použiť pri ďalšom pripojení na server, kde sa pošlú detaľy uložené u klienta serveru, vďaka čomu server skôr rozpozná klienta, resp. nebude od neho vyžadovať napr. dátu o prihlásení, kedže ich klient pošle vo svojej požiadavke na server.

**TCP Transmission Control Protocol** je zložený z viacerých protokolov poskytujúcich rôzne sieťové služby. Je to spojovo orientovaný protokol umožňujúci ovládanie a doručovanie dátovej služby

Môžme povedať, že HTTP protokol je najrozšírenejší na prehliadanie stránok, no chceme upriamíť pozornosť aj na možnosť jeho využitia v inteligentných budovách.[23]

### 2.4.3 Pripojenie do siete Internet

Pre inteligentné riešenie budov, ale aj iných sfér Internet of Things je dôležité pripojenie do siete Internet. Je potrebné hlavne pre zariadenia, ktoré ukladajú dátu do clouдовého úložiska mimo lokálnej siete. Z tohto dôvodu považujeme za vhodné popísať možnosti tohto typu pripojenia. Do siete internet je možné pripojiť sa dvomi spôsobmi, bezdrôtovo

a za použitia konektora, káblu. Poznáme viacero druhov pripojení, ktoré si špecifikujeme nižšie.

**Optický kábel** Táto metóda spočíva v prenášaní svetelných signálov prostredníctvom optického vlákna. Toto vlákno musí byť extrémne priehľadné, jeho hrúbka sa dá porovnať s ľudským vlasom. Svetelné signály sa pohybujú v jadre. Aby sa tam svetelné signály udržali, jadro je obklopené pláštom, ktorý odráža svetlo naspäť do jadra. Táto technika sa nazýva aj *total internal reflection*, voľne preložené ako totálny odraz. Kábel je obalený ochrannou vrstvou, ktorá chráni vlákno pred mechanickým poškodením a vlhkostou. Pre pripojenie do siete internet sa používa vlákno s hrúbkov 50 alebo 62.5 mikrometrov pri dĺžke vlny 850 a 1300 nanometrov. Toto vlákno dokáže prenášať viacero lúčov svetla, takéto vlákno sa nazýva aj *multimode fiber*. Rýchlosť internetového pripojenia cez optický kábel dosahuje rýchlosť až 1000 megabitov za sekundu.[5]

**ADSL** *Asymmetric Digital Subscriber Line* je jeden z typov pripojenia DSL, v ktorom rýchlosť stahovania, teda prenosu dát k užívateľovi je vyššia ako rýchlosť odosielania dát od užívateľa do siete. Je to širokopásma technológia, ktorá sprístupňuje možnosť komunikovať v sieti internet a súčasne fungovať na telefonickej linke prostredníctvom štandardnej medenej linky. Pásma pre internetové služby je delené na pásmo pre odosielanie dát (25.875 kHz až 138 kHz) a pásmo, ktorým je možné dátá prijímať (138 kHz až 1104 kHz). Maximálna rýchlosť ADSL pripojenia je pre stahovanie 24 Mbit/s a pre odosielanie 3.3 Mbit/s.[2, 46]

**Dial-Up** Dial-Up pripojenie nazývame aj vytáčané pripojenie, ktoré funguje za pomocí telefónnej linky a modemu. Počas pripojenia do internetu nesmie byť telefónna linka používaná na hovory. Pripájanie trvá niekoľko sekúnd. Prvé Dial-Up pripojenie bolo uskutočnené už v roku 1965. V súčasnosti sa tento druh pripojenia do siete internet využíva už iba zriedka. [28]

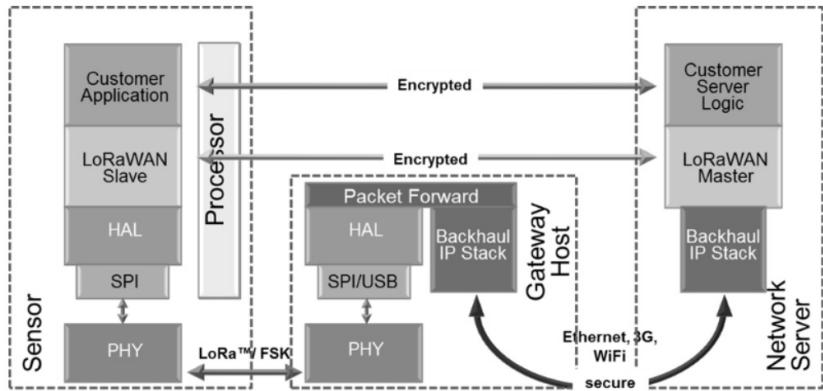
**GPRS** *General Packet Radio Service* je dátová služba pre mobilné telefóny operujúce v sieti GSM, teda funguje bezdôtovo. Mobilný telefón pripojený do tejto siete je schopný stahovať dátá rýchlosťou 40 kilobitov za sekundu a odosielať ich rýchlosťou 14 kilobitov za sekundu. Túto technológiu nahradzajú novšie, rýchlejšie. [8]

**HSDPA** *High-Speed Downlink Packet Access* podobne ako GPRS je služba ktorá poskytuje bezdrôtové pripojenie do siete internet. HSDPA je delené do viacerých

tried, najvyššiu prenosovú rýchlosť má HSDPA triedy 10 a to približne 14 megabitov za sekundu. [9]

**LTE Long Term Evolution** je na Slovensku najrýchlejším štandardom komunikácie. Vývoj LTE bol podmienený rýchlym nárastom internetovej komunikácie, využívaním online televízie, streamov, online hier. Jeho hlavným cieľom je poskytnúť čo najvyššiu prenosovú rýchlosť pri nízkej latencii signálu pri nasadení v rôznej šírke pásma. Ďalším dôvodom jeho vývoja bola podpora bezproblémovej prevádzky v mobilných zariadeniach. Najvyššia prenosová rýchlosť pre stahovanie je 100 megabitov za sekundu a pre odosielanie až 50 megabitov za sekundu. V súčasnosti má takmer celosvetové pokrytie. [17]

**LoraWAN Low Power Wide Area Network**, je navrhnutá na komunikáciu zariadení v *Internet of Things, Machine-to-Machine*. Určenie tejto siete je pre zariadenia s nízkou spotrebou elektrickej energie, siete s milionmi zariadení. Topológia siete je založená na hviezdicovej topológií, kde centrálny uzol je pripojený na ďalšie uzly, ktoré majú ďalšie koncové zariadenia, alebo uzly. Centrálny uzol prijíma a odosiela správy správy ďalšiemu uzlu, ktorý funguje ako brána a preposiela správy do aj od koncového zariadenia. Brány (*gateways*) sú pripojené do siete prostredníctvom štandardného IP pripojenia. Komunikácia koncových zariadení je obojsmerná. Komunikácia medzi koncovými zariadeniami a bránami je rozložená v rôznych frekvenčných pásmach, z tohto dôvodu rýchlosť prenosu dút týmito virtuálnymi kanálmi je od 0.3 kilobitov za sekundu do 50 kiobitov za sekundu. Zabezpečenie siete je riešené šifrovaním na sietovej vrstve, aplikačnej vrstve aj pri komunikácii koncových zariadení. V súčasnosti sa táto technológia ešte len dostáva do povedomia ľudí.[15]



Obrázok 4: Schéma LoraWAN komunikácie

## 2.5 Middleware

V posledných rokoch bolo navrhnutých niekoľko middlewareov pre smart objekty. Tento middleware, ktorý je konvenčne používaný u distribuovaných systémov, je základným nástrojom na implementáciu a dizajnovanie smart objektov, ako aj *Smart environment* aplikácií. Zároveň poskytuje abstrakciu objektov a aplikácií nad nimi, prostredníctvom ktorej môžu byť tieto objekty jednoducho vytvorené. Pod touto abstrakciou si môžeme predstaviť napríklad výpočtový model pre objekty, komunikácia medzi objektami, rozhranie senzorov a akčných členov, *discovery service*, *knowledge management*.[37]

### 2.5.1 Požiadavky

Pre inteligentné prostredie môžme zadefinovať nasledujúcich päť vlastností, ktoré by mal spliňať dobrý middleware. Podľa týchto vlastností zhodnotíme aj middleware nášho zariadenia LogicMachine.

**SE\_Req1 "Abstrakcia prostredníctvom heterogénnych vstupných a výstupných hardvérových zariadení".**

Vstupno-výstupné zariadenia sú zvyčajne heterogénne, z čoho vyplýva, že je náročné až nemožné ich spárovať alebo prinútiť ich k akejkoľvek interakcii. Tu je potrebná abstrakcia k virtualizácii týchto zariadení a ich následné použitie ako homogénne systémy podľa vzoru *plug-and-play*.

**SE\_Req2 "Abstrakcia prostredníctvom softvérových a hardvérových rozhraní".**

Hardvérové a softvérové rozhrania sú tiež heterogénne, teda je potrebná ich štandardizácia pomocou vyšších mechanizmov tak, aby bolo ich použitie jednoduché. Takéto hardvérové a softvérové komponenty založené na vysokoúrovňových rozhraniach budú schopné bezproblémovej komunikácie.

#### **SE\_Req3 "Abstrakcia prostredníctvom dátových prúdov (spojité alebo samostatné údaje alebo udalosti) a dátových typov".**

Rozdielne hardvérové a softvérové komponenty, napríklad senzory, zariadenia, mobilné aplikácie, prezentujú dátá rôznymi spôsobmi, v iných formátoch a dátových typoch. Tu je potrebná abstrakcia na formalizáciu dátových prúdov ktoré sú generované jednotlivými komponentami. Všetky tieto toky dát, či už spojité, samostatné, alebo občasné udalosti majú byť definované pod jednou spoločnou štruktúrou, teda *frameworkom*. Taktiež musí byť štandardizovaná reprezentácia dátových typov, vďaka ktorej by bola umožnená výmena dát u heterogénnych komponentov.

#### **SE\_Req4 "Abstrakcia prostredníctvom fyzickosti (umiestnenia, kontextu)".**

Objekty v inteligentnom prostredí ako aj inteligentné prostredie samé má svoju polohu, teda objekty sú definované statickou alebo dynamickou polohou a vzťahujú sa na jeden alebo viacero kontextov počas ich životného cyklu. Tu je potrebná abstrakcia tohto prostredia tak, aby sme zachytili správanie týchto objektov v prostredí v jednotlivých fázach ich životných cyklov, aby sme ich vedeli využiť v návrhu a implementácii aplikácie tohto inteligentného prostredia.

#### **SE\_Req5 "Abstrakcia prostredníctvom vývojového procesu".**

Na analýzu, dizajn a implementáciu inteligentného prostredia, musia byť definované vhodné metódy a nástroje. Tie musia byť schopné účinne modelovať inteligentné prostredie pomocou vysokoúrovňovej abstrakcie a zároveň plne podporovať ich implementáciu a nasadenie.

Pre smart objekty máme definované taktiež štyri špecifické požiadavky na middleware a jeho použitie. Vďaka týmto požiadavkám vieme presnejšie určiť ako by mal takýto middleware vyzerat.

#### **SO\_Req1 "Heterogenita a vývoj aplikácií".**

Vývoj aplikácie využívajúcej inteligentné objekty by nemal byť závislý na inteligentných objektoch, ich type alebo ich výrobcovi. Teda aplikácia by mala vedieť obsluhovať, resp. byť schopná ovládať objekty od akéhokoľvek výrobcu, prípadne využiť aj objekty, ktoré budú zabudované neskôr v budúcnosti. To znamená, že na vývoj je potrebné použiť štandardný prístup, prípadne využívať softvér pomocou adaptačných techník vrstvenia (dynamicky) medzi aplikáciou a úrovňami inteligentného objektu. Takýto prístup je žiadúcejší.

#### **SO\_Req2 "Zväčšenie variácií intelligentných objektov".**

Intelligentné objekty poskytujú množstvo služieb, ktoré sa môžu lísiť v počte a type služby medzi týmito intelligentnými objektmi, či už rozdielnymi, alebo podobnými. To znamená, že dva podobné objekty môžu poskytovať odlišné služby, no na druhej strane dva úplne odlišné objekty môžu poskytovať rovnakú službu. To znamená, že objekty nemôžme rozdeľovať iba podľa ich typu, na základe toho môžeme povedať, že u takýchto objektoch je problematické využívať štandardné rozhranie. Táto požiadavka je dôležitá, pretože definuje, ako sa môžu meniť intelligentné objekty tým, že poskytujú rôzne služby, ktré sa menia počas životného cyklu intelligentného objektu. Tým pádom nemôžme iba definovať metódy na dynamické úpravy intelligentného objektu, ale aj spôsob, ako budú tieto metódy obsluhované a skonštruované.

#### **SO\_Req3 "Manažment intelligentných objektov".**

Efektívny manažment intelligentných objektov je rozhodujúci v IoT aplikáciách, kde tieto objekty môžu komunikovať každý s každým alebo ak sú použité na splnenie nejakého konečného cieľa. Aplikácie a objekty musia byť preto schopné sa dynamicky prispôsobovať, keďže tieto objekty sa môžu postupne meniť pre rôzny účel, napríklad pre účel mobility, kvôli poruche a tak ďalej. Takže požiadavky aplikácie sa musia zhodovať so službou ktorú poskytuje intelligentný objekt počas behu. Pre tento účel sú strategické vyhľadávacie služby, ktoré dokážu v dynamickom kontexte vyhľadávať a získavať intelligentné objekty na základe ich dynamických a statických vlastností.

#### **SO\_Req4 "Evolúcia systémov intelligentného objektu".**

Aplikácie a intelligentné objekty by mali byť jednoducho, rýchlo modelované a vylepšované pomocou príslušnej programovacej abstrakcie. Vývoj môže byť vedený programovaním, učením alebo obom. To sa týka najmä intelligentných komponentov,

ktoré sú zvyčajne založené na samostatnom vývoji. Takéto komponenty sú schopné riadiť svoj vývoj na základe nejakého modelu učenia, ktorý majú k dispozícii. Ako príklad môžme uviesť softvérového agenta.

[37]

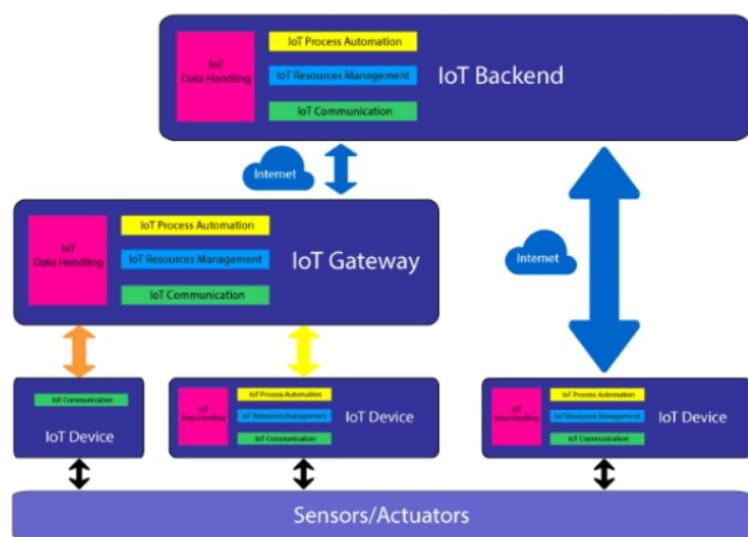
## 2.6 Architektúry

Prepojenie a komunikácia zariadení, spracovávanie a uchovávanie dát je možné robiť viacerými spôsobmi. Popíšeme si dva základné typy architektúr pre Internet of Things.

**Lokálna architektúra** jej fungovanie je založené na komunikácii v rámci inteligentného prostredia. Je vhodná pre naše riešenie, kde budú objekty komunikovať s lokálnym zariadením LogicMachine, v ktorom sa nachádza kompletná logika fungovania. Dáta sa nebudú odosielat nikam inam, avšak je možné rozšíriť túto komunikáciu aj na clouдовý systém, ak to je potrebné. Príklad takejto architektúry nájdete na obrázku 5 vľavo.

**Cloud-based architektúra** spracováva všetky dáta v nejakom dátovom úložisku v cloude. Príkladom takejto architektúry je systém fungovania aplikácie Waze pre mobilné zariadenia, ktorý nemá iný gateway medzi pripojením na cloud. Táto aplikácia musí byť pripojená do siete internet, zaznamenáva polohu a rýchlosť auta, dáta odosiela do cloutu, odkiaľ sa preposiela do ostatných zariadení. Na základe dát dokáže vyhodnotiť zápcu a mnohé iné vlastnosti cestnej premávky. Príklad tejto architektúry nájdete na obrázku 5 vpravo.

Okrem popísaných architektúr je možné kombinovať pripojenia. Pri viacerých bránoch je vhodné architektúru rozšíriť o nadradenú bránu, ktorá ich bude spravovať.



Obrázok 5: Schéma troch rôznych architektúr

# 3 Nástroje na praktickú realizáciu

Našim cieľom v tejto časti práce je aplikovať poznatky z oblasti Internet of Things na realizácii automatizácie inteligentných budov pomocou zariadenia LogicMachine a prototypovacej platformy CommandFusion, ktorá poskytuje možnosti vytvorenia grafického rozhrania pre iOS a Android zariadenia. Na strane CommandFusion platformy sa pomocou JavaScriptu pokúsime vytvoriť komunikáciu so zariadením LogicMachine, kde budeme odosielať a spracovávať dátu odoslané z mobilnej aplikácie a zároveň ich posielat naspäť do LogicMachine. Pre tento cieľ je potrebné si predstaviť prostredie CommandFusion a LogicMachine, ich možnosti realizácie komunikácie medzi nimi.

## 3.1 CommandFusion

Spoločnosť CommandFusion sa zaobrá automatizáciou a ovládaním budov. Poskytuje hardverové aj softvérové riešenia pre inteligentné budovy. Naše riešenie využíva softvér guiDesigner na dizajnovanie a programovanie Android a iOS aplikácie do mobilného zariadenia.

### 3.1.1 guiDesigner

Aplikácia guiDesigner poskytuje *drag and drop*<sup>6</sup> prostredie na vývoj užívateľského rozhrania pre aplikáciu iViewer. Skladá sa z množstva nástrojov na úpravu a tvorbu vlastných objektov a súčasťí grafického rozhrania, ktoré sú jednoducho implementovateľné. Úprava objektov je možná nielen v grafickom rozhraní, ale aj pomocou skriptov písaných v jazyku JavaScript, ktoré si jednoduchým spôsobom vieme vytvoriť a prispôsobiť vlastným potrebám.

**Grafické objekty** Ako sme už spomenuli, grafické rozhranie sa skladá z mnohých nástrojov na vytváranie, úpravu objektov. Pokúsime sa priblížiť najdôležitejšie súčasti z nich, bez ktorých by sme sa pri realizovaní praktickej časti tejto práce nezaobišli.

**Pages** Základnou časťou aplikácie je stránka, do ktorej vieme pridávať ďalšie objekty grafického rozhrania. Každý projekt musí obsahovať minimálne jednu stránku alebo viacero stránok medzi ktorými sa môžme pohybovať. Ďalším riešením je tvorba podstránok.

**Subpages** Podstránky aplikácie sú väčšinou menšie a združujú skupinu spolu súvisiacich ovládacích a zobrazovacích prvkov grafického rozhrania. Pri tvorbe rozsiahlejšieho projektu je vhodné vytvoriť viacero podstránok s obsahom, ktorý akýmsi

---

<sup>6</sup>Jednoduchá technika grafického prostredia na premiestňovanie a úpravu objektov pomocou myši

spôsobom je medzi sebou previazaný. Napríklad ovládanie jednej miestnosti, objekty zabezpečenia a podobne.

**Buttons** Tieto objekty nám dokážu zobrazovať stav alebo hodnoty prijaté spätnou väzbou a zároveň ovládať ďalšie objekty podľa joinu, ktorý majú pridelený. Nadobúdajú stavy aktívny a neaktívny. Tieto stavy vieme rozlísiť na základe obrázku, textu alebo zmenou farby tlačidla, zároveň ich vieme ovplyvňovať na základe stlačenia tlačidla alebo pomocou spätnej väzby (feedbacku) z externého zariadenia.

**Slider** Posuvník (slider) je objekt, ktorý zobrazuje analógovú hodnotu, napríklad nastavenie hlasitosti, intenzity stmievaneho svetla a podobne. Zároveň je možné tieto hodnoty meniť od užívateľa jednoduchým posunutím prstom.

**Pictures** Objekt obrázku môžme dynamicky meniť ak je uložený v priečinku s projektom alebo máme pripojenie na internet a adresu URL na tento obrázok.

**Input fields, Text** Objekt vstupného poľa dáva možnosť užívateľovi zadať text, ktorý môže byť na základe joinu ďalej spracovávaný. Textové polia majú možnosť zobrazovať texty ktoré môžu byť statické alebo dynamické. Pri obidvoch textových objektoch je možné nastaviť font, veľkosť písma.

**Webpages** Objekt webstránok umožňuje užívateľovi plnohodnotne prehliadať stránku, ktorej cestu zadá do jej nastavenia. Pomocou tlačidiel a digitálnych joinov je možné sa na takejto stránke pohybovať obnovovať ju alebo zastaviť načítavanie. Sériové joiny zase pomôžu k zmene URL cesty.

**Theme Manager** V manažérovi tém má užívateľ možnosť využiť existujúce návrhy pozadí, tlačidiel, posuvníkov a textových polí do svojho projektu. Manažér obsahuje aj možnosť vytvárania nových tém pre tieto prvky.

**Komunikačné objekty** Kedže máme zadefinované objekty grafického rozhrania v programe guiDesigner, potrebujeme sa oboznámiť aj s možnosťami komunikácie, ktorú objekty dokážu ovládať. Základnými prvkami sú joiny, tvorba príkazov a makier a manažér systému.

**Joins** Dynamiku každého projektu vytvárame pomocou prepojení alebo joinov. V CommandFusion guiDesigner rozlišujeme tri typy dát - digitálne, analógové a sériové. Dáta všetkých troch typov je možné uchovávať v dátovom úložisku typu kľúč-hodnota nazývaný *token* avšak nie je možné takto uložené dátá zobrazovať v grafickom

rozhraní. Dáta uložené v tokenoch môžme dalej spracovať a ak ich potrebujeme zoobraziť, uložíme ich do hodnoty prepojenia. Každý dátový typ má svoju skupinu joinov, ktoré môžu uložiť dané dátu. Joiny sú špecifikované číslami 0 - 999999999.

**Digitálne** nadobúdajú hodnoty 0 a 1,

**Analógové** môžu mať hodnoty od 0 do 65535,

**Sériové** určené na hodnoty textové, definovanie cesty k obrázkom alebo nastavovanie URL<sup>7</sup>

**Commands and macros** Pomocou príkazov, ktoré vieme implementovať do objektov guiDesigner vieme odosielat správy do ovládacieho systému. Tieto príkazy môžu byť ako odozva na akciu užívateľa (stlačenie tlačidla, posunutie posuvníka) alebo aj ako výsledok spracovania spätnej väzby prijatej z ovládacieho systému. Tieto príkazy môžme zoskupiť do makier, ktoré dokážu odosielat jednotlivé príkazy do rôznych ovládacích systémov a zároveň nastaviť časové rozmedzie, kedy sa tieto príkazy majú spustiť a vykonať.

**Feedback** Spätná väzba (feedback) vyhodnocuje prijaté dátu z ovládacieho systému. Vo feedbacku môžme definovať pravidlá podľa ktorých sa budú tieto dátu spracovať. Na takéto spracovanie dát sa využívajú regulárne výrazy (regex expressions). Ak prijatá správa zo systému nekorešponduje s regex výrazom, bude ignorovaná, ak bude zhodná, bude sa ďalej spracovať na základe pravidiel, ktoré si určíme v nastavení feedbacku. Regulárne výrazy vieme rozdeliť do skupín, na základe ktorých môžme identifikovať časti prijatej správy, ktoré budeme následne spracovať podľa skupiny, do ktorej patria. Pri zhode ďalej vieme určiť, aký príkaz sa má ďalej vykonať, alebo aká hodnota má byť pridelená skupine dát z regex výrazu.

**System Manager** Správca systému je mozgom celého projektu, umožňuje vytváranie ovládacieho systému v ktorom následne môžme vytvoriť a spravovať príkazy, feedbacky, makrá. Je možné vytvoriť systémy TCP klient alebo server, HTTP klient, UDP socket alebo systém s iViewer protokolom, ktorý je určený pre zariadenia od výrobcu CommandFusion.

---

<sup>7</sup>Uniform Resource Locator je formát na označenie internetového zdroja

### **3.1.2 iViewer**

iViewer je aplikácia pre platformu Android alebo iOS, do ktorej je možné nahrať vytvorené grafické rozhranie. Nahrávanie aplikácie prebieha priamo z počítača, na ktorom je uložený projekt, podmienkou nahratia je byť pripojený so zariadením s aplikáciou iViewer aj s počítačom v jednej sieti, o ostatné sa postará guiDesigner. Pred začatím nahrávania je potrebné v nastaveniach iViewera nastaviť IP adresu počítača a port, z ktorého počítač grafické rozhranie nahráva. Po spustení sa aplikácia automaticky pripojí na počítač, ktorý nahrá celé grafické rozhranie do mobilného zariadenia. Po nahratí sa v aplikácii zobrazí nadizajnované rozhranie a je priravené na používanie. Pri vývoji aplikácie je dôležitý debugger, ktorý je možné zapnúť v nastaveniach aplikácie. Debugging prebieha v prehliadači, do ktorého treba zadať adresu mobilného zariadenia s portom, cez ktorý bude aplikácia komunikovať s prehliadačom v počítači. iViewer má aj ďalšie nastavenia týkajúce sa načítavania grafického rozhrania, ukladanie predošlého grafického rozhrania, nastavenie zvukov a podobne.

### **3.1.3 JavaScript API**

Programovacie rozhranie pre iViewer poskytuje množstvo funkcií pomocou ktorých je možné vykonávať základné úlohy. Poskytuje implementácie listenerov, ktoré pri zmene prijatých parametrov spustia určitú akciu, a funkcií na rôzne operácie. Je možné pridávať funkcie podľa potreby užívateľa.

## **3.2 LogicMachine**

Zariadenie LogicMachine sme si popísali v podkapitole 2.3.1, pre prácu s ním budeme využívať senzory a akčné členy pripojené prostredníctvom KNX zbernice. K takto pripojeným objektom môžme pristupovať pomocou skupinovej adresy, ktorá jednoznačne definuje každý zo senzorov a akčných členov. Pre potreby tejto práce je potrebné ozrejmíť množinu skupinových adries, ich dátové typy a možnosti programovania v jazyku Lua.

### **3.2.1 Objekty**

Na to, aby sme mohli ovládať zariadenia pripojené na LogicMachine, je potrebné mať v nej uložené alebo vytvorené objekty. Tieto objekty sú jednoznačne definované názvom, skupinovou adresou a dátovým typom. Ďalšie parametre, ktoré môžme pridať každému objektu sú tagy na zoskupovanie objektov určitého typu, prípony alebo predpony na určenie jednotiek (napr. Celziov stupeň).

**Názov** Názov objektu môžme voliť ľubovoľne. Je vhodné, aby vystihoval zariadenie, ktoré sa pod týmto názvom bude skrývať, prípadne druh zariadenia, jeho umiestne-

nie, alebo identifikačné číslo.

**Skupinová adresa** je zložená z troch zložiek, prvá má 4 bity, druhá 1 bit a tretia 32 bitov, teda rozmedzie adres je od 0/0/1 - 31/7/255 z čoho vyplýva, že je možné do LogicMachine uložiť viac ako 65 000 objektov. Je nutné poznamenať, že tieto adresy sú unikátne pre každý objekt.

**Dátové typy** Pre potreby definovania nadobúdaných hodôt objektov si môžme vybrať dátový typ, do akého sa budú ukladať informácie. Prostredie LogicMachine poskytuje až 16 dátových typov a ďalšie ich podtypy.

- |                          |                         |                                  |
|--------------------------|-------------------------|----------------------------------|
| • 1 bit (boolean)        | signed/signed integer   | • 4 byte signed/unsigned integer |
| • 2 bit (1 bit ovládaný) | un-                     | • 4 byte floating point          |
| • 4 bit (3 bity ovlá-    | signed/signed integer   | • 4 byte access control          |
| dané)                    | dané)                   | • 14 byte ASCII string           |
| • 1 byte ASCII           | • 2 byte floating point |                                  |
| • 1 byte un-             | • 3 byte date/time      |                                  |
|                          | • 3 byte date           |                                  |

Práve vďaka vlastnostiam firmvéru LogicMachine, teda možnosti abstrahovať senzory a akčné členy pomocou skupinových adres a ich hodnôt, môžme povedať, že splňa požiadavky SE\_Req1, SE\_Req2, SE\_Req3, SE\_Req5 pre inteligentné prostredie a SO\_Req1 a čiastočne SO\_Req3 pre inteligentné objekty, ktoré sme si definovali v časti 2.5.1. Požiadavka SE\_Req4 nie je nevyhnutná pre potreby súčasnej automatizácie budovy takisto ako aj požiadavky SO\_Req2 a SO\_Req4.

### 3.2.2 Lua script

Lua je programovací jazyk ktorý vznikol v roku 1993. V LogicMachine je možné v tomto jazyku programovať skripty a funkcie pre end-user aplikáciu. Pomocou skriptov dokážeme upravovať hodnoty na adresách, ukladať ich, alebo spracovávať dátu. Skripty v zariadení LogicMachine sa delia na rezidentné, udalostné, plánované a zároveň vieme vytvorené funkcie ukladať do užívateľskej knižnice.

**Rezidentné** U týchto skriptov vieme nastaviť interval v akom sa má daný skript spúšťať. Skripty sa spúšťajú nepretržite na základe tohto intervalu.

**Udalostné** Tieto skripty sa spúšťajú na základe udalosti a definujú sa pre konkrétnu skupinovú adresu alebo skupinu objektov s rovnakým tagom. Spúštanie tohto

skriptu je podmienené zmenou hodnoty nastavenej skupinovej adresy alebo skupinových adries.

**Plánované** Ako z názvu vyplýva, plánujú sa na konkrétny čas, teda pri ich pridávaní je potrebné nastaviť čas spúšťania, dátum, deň v týždni kedy sa má spúštať tento skript. Tieto skripty môžu slúžiť napríklad na automatické zasvetenie svetla v určitom čase počas celého leta.

**Užívateľská knižnica** Do tejto knižnice môžme pridávať rôzne skripty, ktoré dalej je možné využívať pri ostatných druhoch skriptov, napríklad procedúra na odoslanie dát, transformovanie dát a podobne.

### 3.2.3 Grafické rozhranie LogicMachine

Ovládanie a spravovanie v zariadení LogicMachine je sprístupnené pomocou IP adresy zariadenia, ktorú stačí zadať do prehliadača a otvorí sa nám webová aplikácia LogicMachine. Po prihlásení má užívateľ otvorené všetky možnosti ovládania, programovania a vytvárania objektov a funkcionálít.

**Možnosti webovej aplikácie** Vymenujeme a popíšeme si najdôležitejšie možnosti, ktoré ponúka grafické rozhranie LogicMachine.

**Scripting** V tejto časti sa nachádzajú možnosti pridávania, editovania zálohovania skriptov. Skripty sú rozdelené do kategórií, ktoré sme si predstavili vyššie, v časti 3.2.2. Okrem týchto kategórií máme možnosť upravovať inicializačný skript, ktorý sa spustí po zapnutí LogicMachine, taktiež nám toto rozhranie poskytuje prehľad a príklady skriptov.

**Objects a Object logs** Tu máme prehľad všetkých objektov, ktoré komunikujú alebo boli vytvorené v zariadení LogicMachine. V tomto okne máme možnosť pridávať, mazať a upravovať existujúce objekty podľa potreby, taktiež vyhľadávať podľa adresy, tagu alebo mena. V Object logs zase nájdeme záznamy o aktivite objektov, zmene hodnôt.

**Schedulers** Služby plánovačov sú určené na nastavenie pravidelného správania sa. Napríklad na zapnutie osvetlenia na chodníku vo večerných hodinách, jeho vypnutie v skorých ranných hodinách.

**Trend logs** V logoch sa ukladajú informácie o hodnotách, ktoré následne vieme spracovať a zobraziť napríklad v grafe.

**Visualization, Vis. structure a Vis. graphics** Tieto dve časti poskytujú informácie o vytvorennej vizualizácii, ktorú je možno upravovať priamo v LogicMachine, Vis. graphics poskytuje návrh objektov, ktoré do takejto vizualizácie môžme vkladať a zobrazovať ju napr. vo webovom prehliadači.

**Utilities** Utilities obsahujú nástroje na správu systému, aktualizovanie, resetovanie, zálohovanie a rôzne ďalšie konfigurácie vrátane zmien hesla, užívateľských práv, nastavenia KNX pripojenia.

**User access** Zobrazuje informácie o prihlásenom užívateľovi, jeho právach a mene.

**Dali, Modbus, Enocean, 1-wire** Každá z týchto kariet umožňuje zobrazenie pripojených zariadení pomocou rovnomených pripájacích rozhraní.

**Alerts, Logs a Error log** Tieto karty sú vhodné na debugging alebo analyzovanie odosielania dát. v karte Alerts uvidíme všetky výpisy, ktoré máme nastavené v Lua scriptoch a v Logs zase nájdeme informácie o premenných a dátach, ktoré sa v nich ukrývajú. Error log nám sprostredkuje informácie o chybách v programoch.

**Help** Na poslednej karte nájdeme pomocné informácie k skriptovaniu, návody, popisy jednotlivých funkcia skriptov a podobne.



Obrázok 6: Vzhľad hlavného panelu

# 4 Implementácia

Implementovanie nášho riešenia bude popísané v tejto kapitole. Postupne si rozprišeme celý postup, akým sme overovali prototypovaciu platformu CommandFusion, zariadenie LogicMachine a možnosti komunikácie medzi nimi. Pozrieme sa na problémy, ktoré vznikli a otázky, ktoré sa naskytli. Ukážeme si, ako funguje naše riešenie v praxi, jeho možné vylepšenia a víziu využitia do budúcnosti.

## 4.1 Spôsoby komunikácie

CommandFusion nám poskytuje možnosti komunikovať prostredníctvom HTTP, TCP alebo iViewer protokolu. UDP protokol sme vylúčili hned na začiatku, keďže ide o protokol, u ktorého nemáme záruku doručenia dátového balíka. Takisto sme vylúčili iViewer protokol, ktorý nie je podporovaný zariadením LogicMachine. Sústredili sme sa teda na protokoly komunikácie HTTP a TCP, ktoré sme popísali v časti 2.4.2.

### 4.1.1 Tvorba HTTP requestu

Na vytvorenie HTTP requestu na ovládanie alebo zmenu hodnoty objektu v LogicMachine sme potrebovali zistiť, ako vyzerajú dátá, ktoré je treba poslat. K tomu nám pomohol prehliadač Google Chrome<sup>8</sup> a jeho možnosť preskúmať webové rozhranie LogicMachine pomocou DevTools<sup>9</sup>. V LogicMachine sme si vybrali voliteľný objekt a zmenili sme jeho hodnotu. V DevTools sme ihneď videli, aký HTTP request sa vykonal a preskúmali sme jeho časti. Zistili sme akú požiadavku máme odosielat a zároveň dátá, ktoré sú v nej potrebné. Na realizáciu sme potrebovali vytvoriť takýto request v programe CommandFusion a otestovať, či takto odoslaná požiadavka bude fungovať. Pre odosielanie takéhoto HTTP requestu nám poslúžila funkcia CommandFusion API, **CF.request()**. Túto funkciu je v našom prípade potrebné odosielat s parametrami URL, metódou odosielania, hlavičkou, telom a callback funkciou, ktorá nám zistí, či boli dátá správne odosланé. Ukážeme si, ako celý request vyzerá.

Odoslanie requestu sa nám prejavilo v zmene stavu svetla, teda dokázali sme ho na diaľku zapnúť a dostali sme aj stav odoslania tohto requestu naspäť prostredníctvom debuggera. Pre vypnutie svetla sme použili hodnotu nula. Komunikácia medzi LogicMachine a CommandFusion prostredníctvom HTTP spojenia funguje. Po otestovaní sme zistili, že reakcia na žiadosť o vypnutie je o trochu pomalšia ako sme očakávali, teda pokúsili sme sa vytvoriť aj TCP spojenie a vyskúšať jeho rýchlosť.

---

<sup>8</sup>Jeden z najpoužívanejších internetových prehliadačov na svete

<sup>9</sup>Je to skupina nástrojov na debugging priamo v Google Chrome

---

**Algoritmus 1** Ukážka HTTP requestu

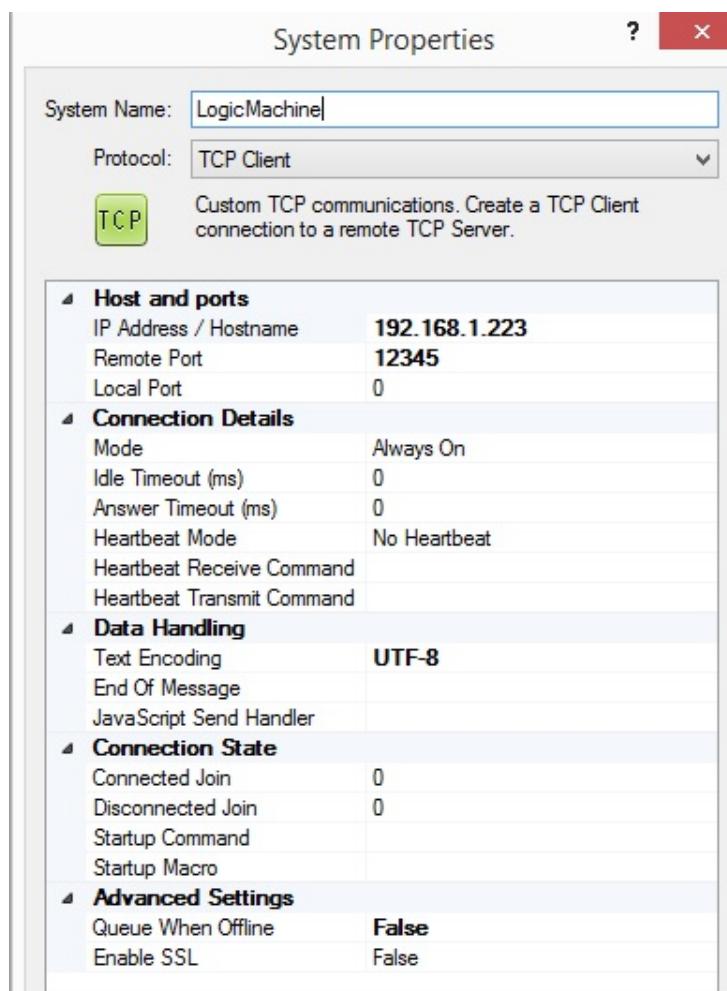
---

```
1
2  function onSendHTTPRequest() {
3
4  var data =
5  "{"address":"1/1/5","datatype":1,"value":1,"type":"text",
6  "update":false}";
7
8  var page =
9  "http://192.168.1.223/cgi-bin/scada-vis/general/objects-setvalue.cgi";
10
11 //Posielame na adresu pomocou metody POST s hlavickou autorizacie
12     data ulozene vyssie a spustame spatne volanie
13
14 CF.request(page, "POST", {"Authorization": "Basic YWRtaW46YWRtaW4xMjUy"}, 
15 data, testWebCallback);
16
17 }
```

---

#### 4.1.2 Tvorba TCP pripojenia

Pre vytvorenie TCP spojenia sme potrebovali vytvoriť v programovacom jazyku Lua v zariadení LogicMachine skript na manažovanie takýchto pripojení. Vytvorili sme TCP okruh, ktorý nám prijímal spojenia, tým pádom nám umožnil ďalšie spracovanie dát odosielaných z iVieweru a zároveň ich posielanie späť. Na strane CommandFusion sme využili funkciu CF.send, ktorá dokáže odosielat dátu do vopred vytvoreného cieľového systému. Cieľovým systémom môžeme rozumieť LogicMachine, konkrétnie jej adresu a port ktorý má otvorený. V guiDesigner sme vytvorili systém TCP klient, ktorý dokáže spracovať dátu odosielané z iVieweru a zároveň ich aplikuje na objekty.



Obrázok 7: Nastavenie TCP klienta v CommandFusion

Po vytvorení tohto klienta sme sa pokúsili odoslať retazec dát do zariadenia LogicMachine, na ktorý sme prostredníctvom TCP socketu odoslali odpoved. Odpoveď sme spracovali pomocou feedbacku, kde regulárny výraz akceptoval akékoľvek dátu. Tieto

dáta sme dali vypísať do textového poľa vytvoreného v guiDesigneri. Pre potvrdenie, že do LogicMachine prichádzajú správne dáta sme dali všetku prichádzajúcú komunikáciu vypisovať medzi *Alerty*. Komunikácia sa nám podarila nadviazať, tým pádom sme zistili, že je možné takýmto spôsobom ovládať zariadenie LogicMachine prostredníctvom aplikácie iViewer. Zároveň, po dlhšom testovaní sme usúdili, že rýchlosť odozvy na tento typ komunikácie je viditeľne rýchlejšia, teda ďalej sme sa venovali už iba komunikácii prostredníctvom TCP protokolu, ktorý nám prišiel ako ideálne riešenie nášho problému.

#### 4.1.3 Listener

Ako nás ďalší cieľ sme si určili, že chceme odosielat aktualizované hodnoty objektov z LogicMachine do iVieweru. Na to sme potrebovali vytvoriť listener, ktorý akýmsi spôsobom bude vedieť, že sa aktualizoval objekt, o ktorom cheme byť informovaní. V LogicMachine sme pridali všetkým objektom, ktoré si zaslúžia našu pozornosť tag "CommandFusion", podľa ktorého dokážeme v Lua skripte tieto objekty zoskupiť a sledovať ich zmenu. Príklad tohto skriptu nájdete v prílohe.

#### 4.1.4 Zariadenia ako klient a server

Pre plnú funkčnosť a realizáciu našej práce sme potrebovali vytvoriť architektúru komunikácie medzi CommandFusion a LogicMachine tak, aby nám LogicMachine sama dokázala odosielat nové hodnoty do iVieweru na príkaz listenera, kde sa pomocou feed-backu spracujú a zároveň, aby aplikácia iViewer dokázala ovládať pomocou TCP spojenia objekty reprezentované skupinovými adresami v LogicMachine.

**Inicializačná komunikácia** Na začiatku komunikácie sme potrebovali do mobilného zariadenia dostať informácie o všetkých objektoch, ktoré máme označené tagom "CommandFusion". TCP ring nám zabezpečil uchovanie TCP socketu, ten sme odoslali do nami vytvorenej funkcie užívateľskej knižnice, ktorá prejde všetky objekty a odošle dáta existujúcim socketom do iVieweru. Príklad algoritmu nájdete v prílohe C.1. Takýmto postupom sme už mali všetky inicializačné dáta v iVieweri.

**Komunikácia počas behu** Kedže zmeniť stav zariadenia, napríklad svetla v miestnosti, vieme ako z mobilnej aplikácie, tak aj vypínačom na stene, potrebovali sme vytvoriť v LogicMachine klienta, ktorý bude odosielat na pripojené zariadenie listenerom odovzdané dáta. Pre tieto potreby sme vytvorili v užívateľskej knižnici ďalšiu funkciu, ktorej parametrom boli iba id a udalosť obsahujúci adresu a hodnotu, ktorú nadobudli objekty. V ukážke 2 vidíme akým spôsobom sme vytvorili socket a odosielali hodnotu.

---

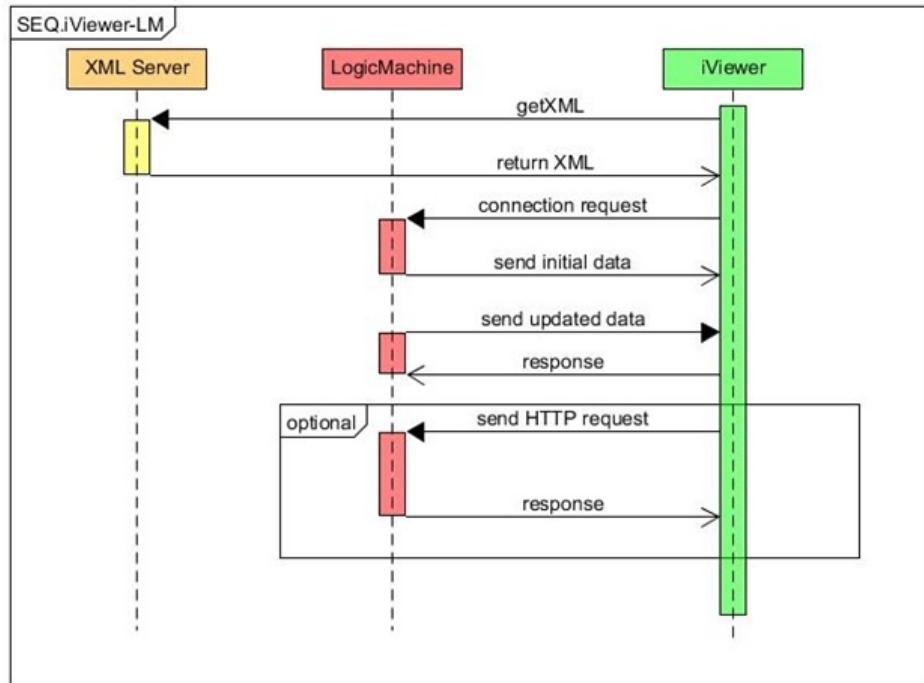
## Algoritmus 2 Vytvorenie socketu

---

```
1 //Vytvorenie socketu
2
3 local host , port = myIP , 10207
4 local socket = require("socket")
5 local tcp = assert(socket.tcp())
6
7 tcp:connect(host , port);
8
9 tcp:send("sendedValue");
```

---

Na obrázku 8 môžeme vidieť návrh realizovaného spôsobu komunikácie medzi iViewerom a LogicMachine. XML server je počítač, z ktorého sa nahráva grafické rozhranie.



Obrázok 8: Sekvenčný diagram nami navrhnutej komunikácie

Teraz, keď už máme definované spôsoby výmeny dát a komunikácie, môžme pristúpiť k logike systému.

## 4.2 Aplikačná logika

Na správnu výmenu informácií medzi zariadeniami potrebujeme stanoviť pravidlá, ako budú vyžerať dátá, ktoré budeme odosielat a ako ich transformujeme do prezentovateľnej polohy. Prvým problémom je transformovať skupinovú adresu z LogicMachine na join v CommandFusion na to, aby sme mohli jednoznačne meniť hodnoty tlačidiel a zobrazovacích prvkov v grafickom rozhraní. Ďalej budeme potrebovať aj opačný postup, teda transformovať join na skupinovú adresu. To nám bude nápomocné na vytvorenie jednotného spôsobu na odosielanie dát pre všetky skupinové adresy a dátové typy, ktoré definujeme ako druhú problémovú skupinu. Dátové typy z LogicMachine budeme musieť akýmsi spôsobom namapovať na typy v CommandFusion, teda určiť ktorý dátový typ sa má zobraziť na akom zobrazovacom objekte.

### 4.2.1 Problém join - skupinová adresa

Tento problém sa nám podarilo vyriešiť aj vďaka vysokému číslu možných hodnôt joinov. Vytvorili sme univerzálny systém na premapovanie joinov na skupinové adresy a opačne, ktorý sme implementovali ako do LogicMachine, tak aj do JavaScriptu na strane CommandFusion. Vieme, že skupinová adresa sa skladá z troch častí, prvá nadobúda hodnoty 0-31, druhá 0-7 a tretia 1-255. Na základe toho sme join vytvorili ako 6-miestne číslo. Algoritmy pre LogicMachine aj pre CommandFusion nájdete v prílohe D.1 a A.1. Rozoberieme si to po častiach.

**Prvá časť** Teda hodnota od 0-31. Kedže prvá časť skupinovej adresy môže nadobudnúť dvojciferný tvar, aj prvá časť joinu sa bude skladáť z dvoch cifier. V prípade dvojcifernej skupinovej adresy budú prvé dve číslice joinu totožné s prvými dvomi číslicami skupinovej adresy. Ak bude skupinová adresa v rozmedzí 0-9, v join-e sa pred ňu pridá číslica 9.

**Druhá časť** Riešenie druhej časti je jednoduché, jednocierná časť adresy sa bude rovnať jednociernému číslu.

**Tretia časť** Tu môže skupinová adresa nadobúdať až trojciferné číslo, teda v prípade jednocifernej časti skupinovej adresy budeme pridávať dve nuly, v prípade dvojcifernej časti jednu nulu a trojcifernej sa bude rovnať hodnota joinu hodnote skupinovej adresy.

Ako príklad uvedieme adresu 1/2/3 konvertujeme na join s hodnotou 912003 alebo adresa 11/0/24 sa ako join reprezentuje 110024.

#### 4.2.2 Problém dátových typov

Pre správne zobrazovanie hodnôt sme potrebovali vymyslieť, akým spôsobom budeme rozlišovať dátové typy. Tento problém sme riešili na strane LogicMachine, ktorá nám poskytuje vstavané funkcie na detekciu dátových typov. Túto funkciu sme využili na ich určenie. Museli sme sa vysporiadať ešte s jedným problémom. Pre hodnoty stmievacích svetiel sme určili dátový typ 1 byte unsigned integer, teda hodnoty od 0-255. Aby sme tieto hodnoty vedeli preniesť do slidera v mobilnej aplikácii, potrebovali sme ich malým výpočtom upraviť, keďže sme zistili, že hodnoty na každom slideri sú v rozmedzí 0 - 65536. Krátkym výpočtom sa dostávame k hodnote 257, ktorou je potrebné stále pred odoslaním dát do iViewera pôvodnú hodnotu prenásobiť. Problém dátových typov sme však nevyriešili úplne, keďže v LogicMachine sme mali pripojený aj teplotný senzor, ktorého dátový typ bol totožný s dátovým typom stmievacích svetiel, avšak nebolo ju potrebné nijakým spôsobom násobiť. Riešenie sme našli v rozdelení skupinových adres podľa účelu. Je viacero spôsobov ako odlišiť typ zariadenia, alebo dát. My sme to rozlíšili na základe strednej hodnoty skupinovej adresy. Kedže máme 8 možností zadania tejto časti skupinovej adresy, rozhodli sme sa určiť hodnotu strednej časti adresy pre dátový typ *boolean* 1, pre stmievacie svetlá 2 a pre teploty 3.

### 4.3 Zhodnotenie praktickej časti

Cieľ, našej práce sa nám podarilo naplniť. Dokázali sme vytvoriť systém komunikácie medzi LogicMachine a CommandFusion prostredníctvom TCP protokolu. Vytvorili sme architektúru komunikácie a systém konverzie skupinových adres na joiny a opačne, ktorý pomôže pri ďalšom rozvíjaní a pokračovaní v tomto projekte. Realizáciu tohto projektu sa nám podarilo aplikovať na rodinný dom, ktorý momentálne využíva popísanú komunikáciu aj rozhranie na ovládanie jeho jednotlivých prvkov ako sú svetlá, žalúzie, garážové brány, prostredníctvom tejto komunikácie sme dokázali pripojiť taktiež teplotné senzory a zobrazovať teploty v izbách na obrazovke mobilného zariadenia. Do budúcnosti je možné tento systém rozšíriť o kontrolu teplôt, ovládanie televízie, zvukových zariadení.

**Zabezpečenie** Systém, ktorý sme vytvorili je potrebné do budúcnosti zabezpečiť šifrovaným spojením. Našou snahou bolo zašifrovať komunikáciu pomocou SSL protokolu<sup>10</sup>, avšak nepodarilo sa nám to z dôvodu firmvérového obmedzenia na strane LogicMachine. Vývojári LogicMachine nám prisľúbili že s novým firmvérom príde aj podpora SSL komunikácie a ďalšie nové možnosti úprav.

---

<sup>10</sup>Security Socket Layer - poskytuje asymetrické šifrovanie na základe dvojice šifrovacích kľúčov - verejný a súkromný

# Záver

Cieľom našej práce bolo ozrejmiť metodológiu Internet of Things, jej rozšírenie a aplikovateľnosť v rôznych odvetviach. Zároveň sme sa snažili priblížiť históriu vzniku tohto pojmu. Práca sa sústredila aj na definovanie jednotlivých súčastí ako sú senzory, akčné členy, agregátory, popísanie ich dôležitosti a úlohy pri riešení problematiky Internet of Things. Popísali sme možnosti komunikácie medzi zariadeniami, vlastnosti týchto komunikácií, ako aj konkrétnie zariadenia schopné komunikácie prostredníctvom definovaných rozhraní. Priblížili sme možnosti architektúr, snažili sa popísať vhodný middleware. Pri praktickej realizácii sme priblížili zariadenia a technológie s ktorými sme pracovali. Predstavili sme zariadenie LogicMachine, možnosti jeho grafického rozhrania, systém skupinových adries, hodnôt ktoré nadobúdajú ako aj možnosti programovania v jazyku Lua. Na druhej strane sme predstavili platformu CommandFusion, v ktorej sme definovali grafické objekty, spôsoby komunikácie a prenosu dát. Riešili sme pripojenie pomocou HTTP protokolu ako aj TCP. Môžme prehlásiť, že sa nám podarilo vytvoriť komunikáciu medzi týmito dvomi zariadeniami, čo vyústilo do využitia v praxi.

Do budúcnosti si vieme predstaviť širšie využitie nášho riešenia, ktoré bude potrebné zabezpečiť.

# Zoznam použitej literatúry

- [1] About z-wave technology - z-wave alliance. [http://z-wavealliance.org/about\\_z-wave\\_technology/](http://z-wavealliance.org/about_z-wave_technology/). (Accessed on 05/17/2016).
- [2] Adsl interface overview - technical documentation. [http://www.juniper.net/documentation/en\\_US/junos12.3x48/topics/concept/security-adsl-interface-overview.html](http://www.juniper.net/documentation/en_US/junos12.3x48/topics/concept/security-adsl-interface-overview.html). (Accessed on 05/17/2016).
- [3] Bluetooth radio interface, modulation & channels :: Radio-electronics.com. <http://www.radio-electronics.com/info/wireless/bluetooth/radio-interface-modulation.php>. (Accessed on 05/12/2016).
- [4] Dali\_manual\_engl.pdf. [http://www.dali-ag.org/fileadmin/user\\_upload/pdf/news-service/brochures/DALI\\_Manual\\_engl.pdf](http://www.dali-ag.org/fileadmin/user_upload/pdf/news-service/brochures/DALI_Manual_engl.pdf). (Accessed on 05/12/2016).
- [5] The foa reference for fiber optics - optical fiber. <http://www.thefoa.org/tech/ref/basic/fiber.html>. (Accessed on 05/11/2016).
- [6] Gira for the home. <http://download.gira.de/data2/19111901011.pdf>. (Accessed on 05/17/2016).
- [7] Gira homeserver. <http://www.gira.com/en/gebaeudetechnik/systeme/knx-eib-system/knx-produkte/server/homeserver.html>. (Accessed on 05/17/2016).
- [8] Gprs and edge. <http://www.3gpp.org/technologies/keywords-acronyms/102-gprs-edge>. (Accessed on 05/11/2016).
- [9] Hsdpa. <http://www.3gpp.org/technologies/keywords-acronyms/99-hspa>. (Accessed on 05/11/2016).
- [10] Jung - facility pilot server knx system technology. <http://www.jung.de/en/1359/products/technology/knx-system/facility-pilot-server/>. (Accessed on 05/17/2016).
- [11] Knx - an overview. [http://www.knxuk.org/index.php?option=com\\_content&view=article&id=53&Itemid=219](http://www.knxuk.org/index.php?option=com_content&view=article&id=53&Itemid=219). (Accessed on 05/13/2016).
- [12] Knx for specifiers. [http://www.knxuk.org/index.php?option=com\\_content&view=article&id=3&Itemid=108](http://www.knxuk.org/index.php?option=com_content&view=article&id=3&Itemid=108). (Accessed on 05/13/2016).

- [13] Knx the standard. [http://www.knxuk.org/index.php?option=com\\_content&view=article&id=6&Itemid=111](http://www.knxuk.org/index.php?option=com_content&view=article&id=6&Itemid=111). (Accessed on 05/13/2016).
- [14] Logicmachine re:actor v3 | logic machine platform for knx/eib, enocean, modbus, dali, bacnet. <http://openrb.com/logicmachine-reactor-v3/>. (Accessed on 05/17/2016).
- [15] Lora technology. <https://www.lora-alliance.org/What-Is-LoRa/Technology>. (Accessed on 05/11/2016).
- [16] Loxone | miniserver setup - documentation. <http://www.loxone.com/enen/service/documentation/miniserver/miniserver-setup.html>. (Accessed on 05/17/2016).
- [17] Lte system. <https://sites.google.com/site/lteencyclopedia/home>. (Accessed on 05/11/2016).
- [18] M2m/iot sector map :: Beecham research. <http://www.beechamresearch.com/article.aspx?id=4>. (Accessed on 05/17/2016).
- [19] Modbus tcp/ip overview. <http://www.rtautomation.com/technologies/modbus-tcpip/>. (Accessed on 05/17/2016).
- [20] Overview of 1-wire technology and its use - tutorial - maxim. <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1796>. (Accessed on 05/12/2016).
- [21] Partners using energy harvesting wireless sensor solutions | enocean - technology. <https://www.enocean.com/en/technology/>. (Accessed on 05/12/2016).
- [22] Radio technology (rf) from enocean for energy harvesting wireless sensor solutions | enocean technology. <https://www.enocean.com/en/technology/radio-technology/>. (Accessed on 05/12/2016).
- [23] Rfc 7230 - hypertext transfer protocol (http/1.1): Message syntax and routing. <https://tools.ietf.org/html/rfc7230>. (Accessed on 05/12/2016).
- [24] Rs-422 and rs-485 standards overview and system configurations. <http://www.ti.com/lit/an/slla070d/slla070d.pdf>. (Accessed on 05/13/2016).
- [25] Samsung smartthings hub. <https://shop.smartthings.com/#!products/samsung-smartthings-hub>. (Accessed on 05/17/2016).

- [26] The story behind bluetooth technology | bluetooth technology website. <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth>. (Accessed on 05/12/2016).
- [27] Vera3 - advanced smart home controller. <http://getvera.com/controllers/vera3/>. (Accessed on 05/17/2016).
- [28] What is dial-up? <http://www.computerhope.com/jargon/d/dialup.htm>. (Accessed on 05/11/2016).
- [29] What is x10? <http://www.smarthome.com/sc-what-is-x10-home-automation>. (Accessed on 05/17/2016).
- [30] Wi-fi and the osi model | control engineering. <http://www.controleng.com/single-article/wi-fi-and-the-osi-model/8b71b0494b6b7fd5291856d02e104eb4.html>. (Accessed on 05/15/2016).
- [31] Zigbee/ieee 802.15.4 overview. [http://www.csie.nuk.edu.tw/~lhyen/wn/zigbee\\_802\\_15\\_4.pdf](http://www.csie.nuk.edu.tw/~lhyen/wn/zigbee_802_15_4.pdf). (Accessed on 05/17/2016).
- [32] BELL, J. guidesigner basics [commandfusion wiki]. <http://www.commandfusion.com/wiki2/software/gui-designer/gui-designer-basics>. (Accessed on 05/17/2016).
- [33] COMPUTER SCIENCE DEPARTMENT, T. C. M. U. A smarter grid with the Internet of Things. [https://www.cs.cmu.edu/~coke/history\\_long.txt/](https://www.cs.cmu.edu/~coke/history_long.txt/), 2014. [Online; accessed 28-April-2016].
- [34] ERLICH, Y. A vision for ubiquitous sequencing. <http://genome.cshlp.org/content/25/10/1411/>, 2015. [Online; accessed 26-April-2016].
- [35] EVANS, D. The Internet of Things - How the Next Evolution of the Internet is Changing Everything. [http://www.cisco.com/c/dam/en\\_us/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf/](http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf/). [Online; accessed 26-April-2016].
- [36] FIELDING, R. T., GETTYS, J., MOGUL, J. C., NIELSEN, H. F., MASINTER, L., LEACH, P. J., AND BERNERS-LEE, T. Rfc 2616 - hypertext transfer protocol – http/1.1. <https://tools.ietf.org/html/rfc2616>, 6 1999. (Accessed on 05/01/2016).
- [37] FORTINO, G., AND TRUNFIO, P. Internet of Things based on Smart Objects , 2014.

- [38] FRIEDEMANN, M., AND FLOERKEMEIER, C. From the Internet of Computers to the Internet of Things . <http://www.vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf/>, 2010. [Online; accessed 28-April-2016].
- [39] HÖLLER, J., TSIATSIS, V., MULLIGAN, C., KARNOUSKOS, S., AVESAND, S., AND BOYLE, D. Internet of Things - Converging Technologies for Smart Environments and Integrated Ecosystems. [http://www.internet-of-things-research.eu/pdf/Converging\\_Technologies\\_for\\_Smart\\_Environments\\_and\\_Integrated\\_Ecosystems\\_IERC\\_Book\\_Open\\_Access\\_2013.pdf/](http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf/). [Online; accessed 26-April-2016].
- [40] LA DIEGA, GUIDO, N., AND IAN, W. Contracting for the ‘Internet of Things’: Looking into the Nest. <http://ssrn.com/abstract=2725913/>, 2016. [Online; accessed 28-April-2016].
- [41] MONNIER, O. A smarter grid with the Internet of Things. [http://e2e.ti.com/blogs\\_/b/smartgrid/archive/2014/05/08/a-smarter-grid-with-the-internet-of-things/](http://e2e.ti.com/blogs_/b/smartgrid/archive/2014/05/08/a-smarter-grid-with-the-internet-of-things/), 2014. [Online; accessed 26-April-2016].
- [42] PONTIN, J. ETC: Bill Joy’s Six Webs . <https://www.technologyreview.com/s/404694/etc-bill-joys-six-webs/>, 2005. [Online; accessed 28-April-2016].
- [43] RAJI, R. Smart networks for control. [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=284793&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs\\_all.jsp%3Farnumber%3D284793/](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=284793&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D284793/), 1994. [Online; accessed 28-April-2016].
- [44] ROUSE, M. Internet of Things (IoT). <http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT/>, 2014. [Online; accessed 26-April-2016].
- [45] SKELETON, A. Fibaro z-wave home center 2 mega review. <http://www.automatedhome.co.uk/reviews/fibaro-z-wave-home-center-2-mega-review.html>, 9 2012. (Accessed on 05/17/2016).
- [46] SOBAN, Z. Adsl overview. <http://hw-server.com/adsl-overview>, 3 2004. (Accessed on 05/17/2016).
- [47] UNKNOWN. Internet of Things Global Standards Initiative. <http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/>. [Online; accessed 26-April-2016].

- [48] VERMESAN, O., AND FRIESS, P. Internet of Things - Converging Technologies for Smart Enviroments and Integrateed Ecosystems. [http://www.internet-of-things-research.eu/pdf/Converging\\_Technologies\\_for\\_Smart\\_Environments\\_and\\_Integrated\\_Ecosystems\\_IERC\\_Book\\_Open\\_Access\\_2013.pdf/](http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf/). [Online; accessed 26-April-2016].
- [49] WEISER, M. From the Internet of Computers to the Internet of Things . <http://web.media.mit.edu/~anjchang/ti01/weiser-sciam91-ubicomp.pdf/>, 1991. [Online; accessed 28-April-2016].
- [50] Wood, Alex. *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press.

# Prílohy

A	Skupinová adresa na join . . . . .	II
B	Listener v LogicMachine . . . . .	III
C	Funkcia odosielajúca stavy tagovaných zariadení . . . . .	IV
D	Funkcia na prevod joinu na skupinovú adresu . . . . .	V

# A Skupinová adresa na join

---

## Algoritmus A.1 Skupinová adresa na join

---

```
1 --Kód v jazyku Lua v zariadení LogicMachine , ako vstupný argument
2 --prichádza skupinová adresa
3 function addrToJoin (gaddr)
4     tmp1 = string.sub(gaddr, 1, 2)
5     tmp1 = string.gsub(tmp1, "/", "", 1)
6     if string.len(tmp1) == 1 then
7         if (tmp1 == "0") then
8             tmp1 = "90"
9         else
10            tmp1 = "9" .. tmp1
11        end
12    end
13    tmp2 = string.sub(gaddr, 3, 4)
14    tmp2 = string.gsub(tmp2, "/", "", 1)
15    tmp3 = string.sub(gaddr, 5)
16    tmp3 = string.gsub(tmp3, "/", "", 1)
17    if string.len(tmp3) == 2 then
18        tmp3 = "0" .. tmp3
19    elseif string.len(tmp3) == 1 then
20        tmp3 = "00" .. tmp3
21    end
22    join = tmp1 .. tmp2 .. tmp3
23    return join;
24 end
```

---

## B Listener v LogicMachine

---

### Algorithmus B.1 Listener v LogicMachine

---

```
1  if not client then
2      require('genohm-scada.eibdgm')
3      require('user.update')
4      ids = {}
5      objects = grp.tag('CommandF')
6
7      for _, object in ipairs(objects) do
8          ids[object.id] = true
9      end
10
11     function handler(event)
12         local id = event.dstraw
13
14         if ids[id] then
15             sendsock(id, event)
16         end
17     end
18
19     client = eibdgm:new()
20     client:sethandler('groupread', handler)
21     client:sethandler('groupwrite', handler)
22     client:sethandler('groupresponse', handler)
23 end
24 client:step()
```

---

# C Funkcia odosielajúca stavy tagovaných zariadení

---

**Algoritmus C.1** Funkcia odosielajúca stavy tagovaných zariadení

---

```
1  function CFmain(socket)
2  -- myIP = adresa zariadenia na ktoré posielame data
3  myIP = storage.get('myIP', 0)
4      objects = grp.tag('CommandF')
5  socket:send(myIP)
6
7  for k,v in pairs(objects) do
8
9      address = objects[k].address;
10     address = addrToJoin(address);
11     values = objects[k].data;
12     datatype = objects[k].datatype;
13
14     value = string.format("%os", values)
15     --odosielaná hodnota je v tvare groupaddress#hodnota
16     hodnota = address .. "#" .. value .. '\r\n';
17     end
18
19     socket:send(hodnota)
20   end
21 end
```

---

# D Funkcia na prevod joinu na skupinovú adresu

---

**Algoritmus D.1** Funkcia na prevod joinu na skupinovú adresu

---

```
1 // JavaScript v CommandFusion na transformaciu joinu na skupinovú adresu
2 function joinToGaddr(join) {
3     tmp3 = join.substring(3, 6);
4     tmp3 = tmp3.replace(/\^0+/, '')
5     tmp2 = join.substring(2, 3);
6     tmp1 = join.substring(0, 2);
7     if(tmp1.substring(0,1) == '9')
8         tmp1 = tmp1.substring(1,2)
9
10    total = tmp1.concat(' ', tmp2, ' ', tmp3)
11
12    return total
13 }
```

---