

**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: **Michal Ližičiar**  
ID študenta: 5982  
Študijný program: Aplikovaná informatika  
Študijný odbor: 9.2.9 aplikovaná informatika  
Vedúci práce: Ing. Matúš Jókay, PhD.

Názov práce: **Anonymizácia internetového prístupu**

Špecifikácia zadania:

Cieľom práce je vytvoriť zásuvný modul pre internetový prehliadač, ktorý bude schopný buď náhodne alebo selektívne meniť informácie používané na identifikáciu používateľa pri jeho prístupe na cieľový server.

Úlohy:

1. Analyzujte dostupnosť a funkčnosť podobných modulov.
2. Analyzujte informácie používané na identifikáciu používateľa pri prístupe na stránku.
3. Navrhnite, implementujte a otestujte anonymizačný modul pre zvolený internetový prehliadač.

Zoznam odbornej literatúry:

1. YARDLEY, G. Better Privacy. [online]. 2012. URL: <http://nc.ddns.us/BetterPrivacy/BetterPrivacy.htm>.
2. ECKERSLEY, P. A Primer on Information Theory and Privacy. [online]. 2010. URL: <https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy>.

Riešenie zadania práce od: 24. 09. 2012

Dátum odovzdania práce: 24. 05. 2013

**Michal Ližičiar**

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
Bakalárska práca:	Využitie metodiky IoT v inteligentných budovách
Vedúci záverečnej práce:	prof. Ing. Štefan Kozák, PhD.
Miesto a rok predloženia práce:	Bratislava 2016

Tu dačo bude...

Kľúčové slová: anonymizácia, identifikácia používateľa, zásuvný modul, Mozilla Firefox, internet

# 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

There will something be...

Keywords: anonymization, identification of user, plugin, Mozilla Firefox, internet

## Vyhlásenie autora

Podpísaný(á) Marek Hrebík čestne vyhlasujem, že som bakalársku prácu Využitie metodiky IoT v inteligentných budovách vypracoval(a) na základe poznatkov získaných počas štúdia a informácií z dostupnej literatúry uvedenej v práci.

Vedúcim mojej bakalárskej práce bol prof. Ing. Štefan Kozák, PhD.

Bratislava, dňa 17.5.2016

.....

podpis autora

# Podakovanie

Chcem sa poďakovať vedúcemu záverečnej práce, ktorým bol prof. Ing. Štefan Kozák, PhD., za odborné vedenie, rady a pripomienky, ktoré mi pomohli pri vypracovaní tejto bakalárskej práce.

# Obsah

<b>Úvod</b>	<b>12</b>
<b>1 Metodológia Internet of Things</b>	<b>13</b>
1.1 Vznik pojmu Internet of Things . . . . .	13
1.2 Využitelnosť . . . . .	13
1.3 História . . . . .	14
<b>2 Zariadenia a prepojitelnosť</b>	<b>16</b>
2.1 Senzory . . . . .	16
2.2 Akčné členy . . . . .	17
2.3 Agregátory . . . . .	17
2.3.1 Riadiace aplikácie . . . . .	17
2.4 Možnosti komunikácie . . . . .	20
2.4.1 Model OSI/ISO . . . . .	20
2.4.2 Lokálne pripojenie . . . . .	22
2.4.3 Pripojenie do siete Internet . . . . .	26
2.5 Middleware . . . . .	28
2.5.1 Požiadavky . . . . .	29
2.6 Architektúry . . . . .	31
<b>3 Nástroje na praktickú realizáciu</b>	<b>33</b>
3.1 CommandFusion . . . . .	33
3.1.1 guiDesigner . . . . .	33
3.1.2 iViewer . . . . .	35
3.1.3 JavaScript API . . . . .	36
3.2 LogicMachine . . . . .	36
3.2.1 Objekty . . . . .	36
3.2.2 Lua script . . . . .	37
3.2.3 Grafické rozhranie LogicMachine . . . . .	38
<b>4 Implementácia</b>	<b>40</b>
4.1 Spôsoby komunikácie . . . . .	40
4.1.1 Tvorba HTTP requestu . . . . .	40
4.1.2 Tvorba TCP pripojenia . . . . .	42
4.1.3 Listener . . . . .	43



4.1.4	Zariadenia ako klient a server . . . . .	43
4.2	Aplikačná logika . . . . .	45
4.2.1	Problém join - skupinová adresa . . . . .	45
4.2.2	Problém dátových typov . . . . .	46
4.3	Zhodnotenie . . . . .	46
	<b>Záver</b>	<b>47</b>
	<b>Zoznam použitej literatúry</b>	<b>48</b>
	<b>Prílohy</b>	<b>I</b>
A	Štruktúra elektronického nosiča	II
B	Algoritmus	III

## Zoznam obrázkov a tabuliek

Obrázok 1	Spektrum využiteľnosti IoT . . . . .	14
Obrázok 2	Architektúra technológií Internet of Things . . . . .	20
Obrázok 3	Diagram HTTP komunikácie . . . . .	25
Obrázok 4	Schéma LoraWAN komunikácie . . . . .	28
Obrázok 5	Schéma troch rôznych architektúr . . . . .	32
Obrázok 6	Vzhľad hlavného panelu . . . . .	39
Obrázok 7	Nastavenie TCP klienta v CommandFusion . . . . .	42
Obrázok 8	Sekvenčný diagram nami navrhutej komunikácie . . . . .	44

## **Zoznam skratiek a značiek**

IoT - Internet of Things

## Zoznam algoritmov

1	Ukážka HTTP requestu . . . . .	41
2	Vytvorenie socketu . . . . .	44
A.1	TCP ring v LogicMachine . . . . .	II
B.1	Listener v LogicMachine . . . . .	IV

# Úvod

Metodológia Internet of Things sa čoraz viac dostáva do každodenného života každého človeka. Pomocou nej dokážeme šetriť 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 možnosti 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 aplikácii 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

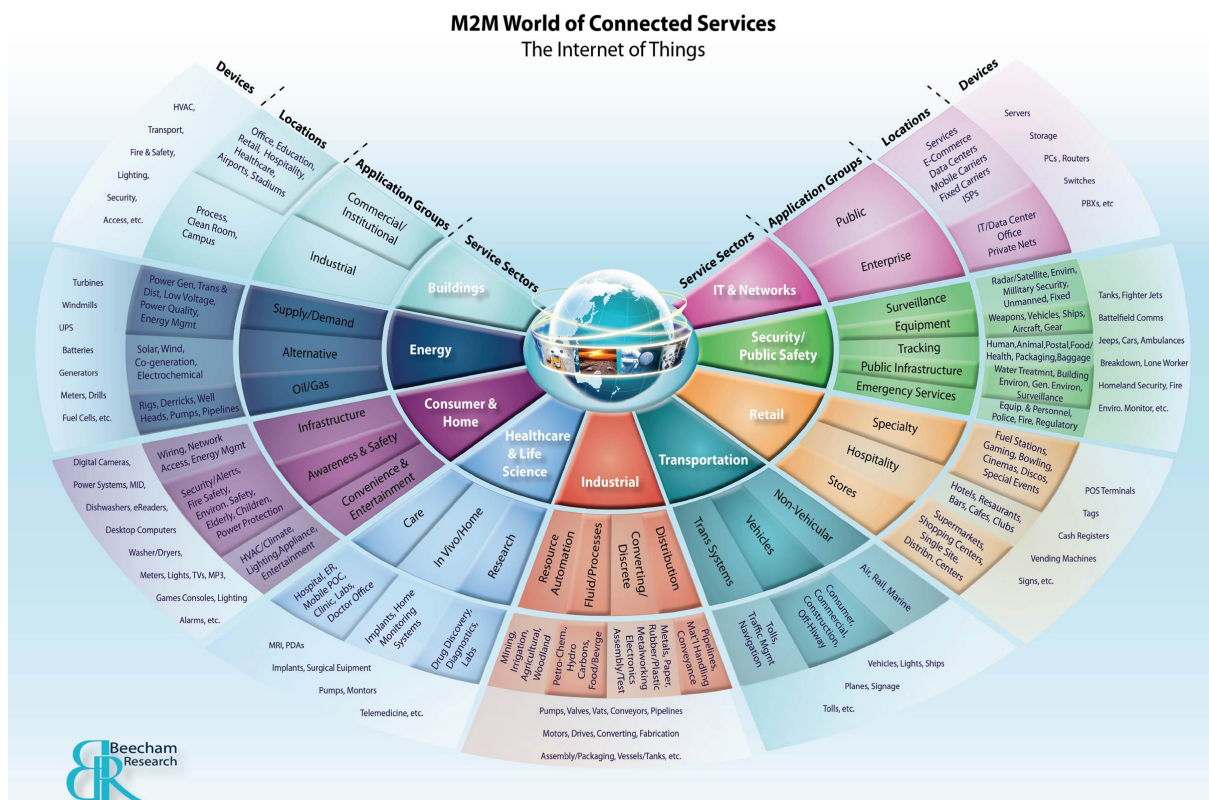
Internet of Things je sieť objektov (vozidiel, budov alebo iných zariadení), ktoré obsahujú elektroniku, softvér alebo senzory schopné vyslať a prijímať signál medzi nimi. Toto internetové pripojenie zabezpečuje zber a výmenu dát v takomto systéme.[31] IoT zabezpečuje ovládanie objektov naprieč existujúcou sieťovou infraštruktúrou. Vytvára možnosti integrovania počítačových systémov do reálneho sveta, čo sa odráža na efektívnosti ako aj ekonomických výhodách.[32] 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 internetovej infraštruktúry.[21] Touto metodológiou sa budeme zaoberať v celej práci, z tohto dôvodu je potrebné objasniť a popísať ako vznikla, popísať časti z ktorých sa skladá, možnosti prepojenia jednotlivých 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 sieť objektov prepojených pomocou rádiových frekvencií identifikácie.[34] IoT ponúka pokročilejšiu konektivitu zariadení, systémov a služieb, ktorá presahuje komunikáciu Machine to Machine a pokrýva celý rad protokolov, domén a aplikácií.[25] Prepojenie týchto vstavaných zariadení má využitie na poli automatizácie takmer vo všetkých smeroch, zároveň umožňuje aplikovanie napríklad inteligentných sietí, inteligentných budov a miest.[27]

## 1.2 Využitelnosť

Slovo "Things" v IoT odkazuje na širokú škálu zariadení využiteľných pri monitorovaní srdcových implantátov. Uplatnenie nájdú aj rôznorodé čipy na hospodárske zvieratá, elektronické senzory v moriach, zabudované senzory v automobiloch a iných objektoch, monitoring patogénov [20], prevádzka zariadení v teréne či na vojenské, alebo záchranné účely.[30] Zároveň ho môžeme definovať aj ako neoddeliteľnú zmes hardvéru, softvéru, dát a služieb. [26] Tieto zariadenia zbierajú užitočné informácie pomocou existujúcej technológie a následne dáta nezávisle posielajú do ďalších zariadení.



Obrázok 1: Spektrum využiteľnosti IoT

## 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 odosielať informáciu o svojej zásobe nápojov, ktoré malo k dispozícii ako aj informáciu o tom, či vydaný nápoj bol studený. [19] 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 a kedykoľvek. Zmienil sa aj o vízii IoT, takisto ako aj spoločnosti Percom, UbiComp. [24, 33] 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. [29] 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.[28] V tomto roku sa stal koncept IoT populárnym, a to vďaka *Auto-ID Center* na Massachussettskej Univerzite (MIT) a ich publikáciách trhovej analýzy.



## 2 Zariadenia a prepojitelnosť

Je potrebné objasnenie pojmov senzor, akčný člen, agregát, 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 konkrétne 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 dáta. Zmenu tejto veličiny senzor zaznamená a odošle ju ďalej ako výstup. Tieto hodnoty ďalej spracúvajú agregátory a posielajú informáciu do akčných členov, ktoré na základe hodnôt vykonajú predom definovanú akciu. Senzor premieňa úžitkovú energiu na elektrické dáta. 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 dáta na displeji a sprostreduje komunikáciu pomocou pripojenia na internet. V tomto prípade môžeme povedať, že aj vodič auta využívajúci službu Waze je akčný člen, ktorý usmerňuje a prispôsobuje rýchlosť vozidla podľa aplikácie, napríklad pri dopravných zápchach. Do budúcnosti si vieme predstaviť, že takáto aplikácia by ovplyvňovala nie vodiča, ale priamo vozidlo tým, že by uberala alebo pridávala plyn.

## 2.2 Akčné členy

Akčný člen, alebo *actuator* je zariadenie fyzickej vrstvy, ktoré zodpovedá za kontrolu a ovládanie senzorov vo väčšine prípadov je toto ovládanie sprostredkované 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 môžeme 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 odosielať dáta do riadiacej aplikácie prostredníctvom rôzneho middleware. Ich úlohou je zoskupiť dáta zo senzoru, spracovať ich a odoslať do akčného členu, ktorý spracuje tieto dáta a vykoná akciu. Niektoré agregátory odosielať tieto dáta 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, spracováva ich a preposiela do akčných členov. Každé zariadenie pripojené v agregátore 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 inštalácie KNX systému do počítačovej siete a siete internet. Takto pripojené zariadenie je schopné ovládať inteligentný dom a objekty v ňom, kamery, audiosystémy, svetlá a ďalšie zariadenia od iných výrobcov. Výrobca svoj produkt prezentuje ako komplexný ovládač, ktorý je energeticky nenáročný, zabezpečený a ovládateľný jedným prstom. Ich riešenie poskytuje aj aplikáciu pre iOS zariadenia, ktoré po pripojení sa do siete vedú 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.

**Jung Facility pilot** Zariadenie určené na trvalú prevádzku, montuje sa k rozvádzačom 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 vzdialenej údržby a druhé na prevádzku. Vďaka tomuto systému sa eliminujú akékoľvek ďalšie poplatky na dodatočnú kúpu ďalšieho rozhrania.

**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.

**Fibaro Home Center** Systém od poľskej spoločnosti Fibaro je mozgom autimatizovaných systémov *Fibaro systems* alebo iných systémov ktoré komunikujú prostredníctvom pripojenia Z-Wave. Ich vizualizácia má intuitívne vizuálne rozhranie. Hardvérové vybavenie sa skladá z procesoru Intel Atom s frekvenciou 1,6 GHz s 1GB RAM a 2GB SLC pevného disku. Zaujímavosťou 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. Ďalšou výhodou je prehľadnejšie menu zariadení vo vizualizácii. Zaujímavou funkcionalitou je geolokácia

zariadení s ich mobilnou aplikáciou. Ak chcete vedieť, kde sa nachádzajú ostatní členovia rodiny alebo firmy, stačí si pozrieť mapu, na ktorej je znázornená pozícia zariadenia.

**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.

**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 dáta lokálne, čo môžeme považovať za bezpečnejšie ako ukladanie do cloudových služieb.

**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>1</sup>, EnOcean, DALI, Modbus. Základné funkcionality spočívajú v IP routingu, vizualizácii, programovaní a vytváraní brán medzi zariadeniami rôznych štandardov. Má funkcionality vzdialenej diagnostiky, a ovládania IP routerov ktoré sú zapojené v odlišných inštaláciách. Rôzne funkcionality na zaslanie e-mailu pri chybe zbernice, pretečení bufferu, prípadne stavu pripojených zariadení. Obsahuje Redis in-memory databázu typu kľúč-hodnota, do ktorej je možné ukladať rôzne objekty a ich hodnoty pre ďalšie použitie. Novou funkcionality, ktorá prišla s najnovším firmvérom je tvorba vizualizácie vrátane pridávania fontov, predprogramovaných prechodov pri zmene okna vo webovom prehliadači. Tvorba vizualizácie je jednoduchá aj pre laikov, keďže systém má v sebe už zabudované šablóny, kde stačí pridávať ďalšie objekty. Užívateľ si vie sám namodelovať ako bude vyzeráť jeho ovládacia aplikácia pre jeho inteligentný dom. Programovanie

---

<sup>1</sup>Používa sa na ukladanie zálohy, databázy skriptov.

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 posielajú 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 zjednocuje rozdielne systémy do jedného, ktorý dokáže spolupracovať so svojimi časťami.

## 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 odosielať dáta 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 ozrejniť 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 dáta 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 na tejto vrstve.

**Transportná vrstva** Účelom tejto vrstvy je poskytovanie transparentného prenosu dát medzi koncovými zariadeniami. Tieto dáta 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, sockety.

**Prezentačná vrstva** Táto vrstva sa stará o kódovanie a reprezentáciu dát. Poskytuje sa tu prevod týchto dát do zrozumiteľnej reprezentácie pre zariadenie, ktorému sú dáta určené a zároveň jeho zabezpečenie. Taktiež zabezpečuje šifrovanie, dešifrovanie. Týmto odľahčuje prezentačnú vrstvu.

**Aplikačná vrstva** Je vrstva, ktorá je najbližšie ku koncovému užívateľovi, teda komunikuje priamo so softvérovou aplikáciou, ktorú užívateľ ovláda. Definuje akým spôsobom a v akom formátovaní majú byť dáta zaslané a prijímané. V tejto vrstve užívateľ konkrétne vidí výsledok, ktorý sa transformoval prostredníctvom všetkých vrstiev modelu OSI/ISO.

Podľa definícií jednotlivých vrstiev tohto modelu si môžeme 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 tiecť dáta. Je viacero rozhraní ktorými môžu takéto objekty komunikovať. Každé z nich má svoje špecifické vlastosti. 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 mikroovládačom 8-bitovej architektúry. Prostredníctvom KNX zbernice je pripojené aj naše zariadenie, ktorého nastaveniu a programovaniu sa budeme venovať v praktickej časti. [6, 7, 8]

**DALI *Digital Addressable Lighting Interface*** definuje štandardizované rozhranie pre predradníky <sup>2</sup>, ktoré zaisťuje zameniteľnosť a kompatibilitnosť 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. [2]

**RS-485** V súčasnosti tento typ pripojenia je vhodný a využívaný na riešenie prenosu dát na dlhé vzdialenosti v zašumenom prostredí. RS-485 dokáže pracovať vo vyvážených viacbodových systémoch. Prenos dát dokáže sprostredkovať v rýchlosti 35 megabitov za sekundu do vzdialenosti 10 metrov. Vzdialenosťou rapídne klesá jeho prenosová rýchlosť, čo nie je prekážkou v inteligentných domoch. [15]

**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. Každé takto pripojené zariadenie je jedinečne identifikované pomocou továrenského 64-bitového identifikačného čísla (ID). Prvých 8 bitov tohto identifikátora napovedá

---

<sup>2</sup>Predradník je zariadenie, ktoré reguluje množstvo prúdu v elektrickom obvode (napr. stmievané svetlá)

o funkcionalite a typu zariadenia. Prostredníctvom 1-Wire sa dá komunikovať obojsmerne, polovičným duplexom.<sup>3</sup> [11]

**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.[12, 13]

**X10** Protokol na automatizáciu, väčšinou využíva elektrické vedenie na signalizáciu a ovládanie, kde signály predstavujú krátke rádiové impulzy transformované na digitálnu informáciu.

**Z-Wave** Novší protokol, ktorý používa na komunikáciu bezdrôtové pripojenie. Je navrhnutý na bezdrôtové ovládanie svetiel, stmievačov, bezpečnostných systémov a ďalších inteligentných objektov. Systém využívajúci Z-Wave môže byť ovládaný vzdialene prostredníctvom siete internet a Z-Wave brány alebo riadiacej jednotky ktorá slúži aj ako ovládač.

**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 nízkoenergetické digitálne zariadenia, ktoré je možné ovládať. Dokážu prenášať dáta 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.

**Modbus** Je sériový komunikačný protokol využívaný v PLC ovládačoch. Bežne sa používa na pripájanie priemyselných elektronických zariadení. Je jednoduchý na zavedenie a údržbu, má viacero verzií, pre účely inteligentných budov je najvhodnejšia verzia Modbus TCP/IP.

**Bluetooth** Otvorený komunikačný štandard pre bezdrôtové zariadenia, ktorý dokáže zprípojiť viacero zariadení naraz. Pôvodne bol ako náhrada za drôtové rozhranie

---

<sup>3</sup>Obidve strany pripojenia sú schopné prijímať aj odosielať informácie avšak nie naraz.



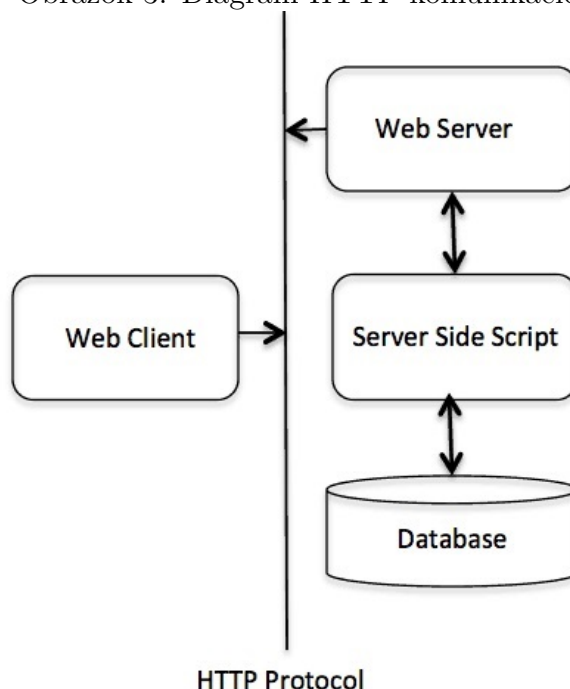
RS-232, dnes predstavuje využitie v mobilných telefónoch, elektronike a aj v inteligentných budovách. Každé zariadenie v sieti Bluetooth je identifikované pomocou *Bluetooth Device Address*, čo môžeme prirovnať k identifikácii zariadení Ethernet prostredníctvom MAC adres. Prostredníctvom Bluetooth signálu 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. [16, 1]

**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 inteligentných budovách sa využíva taktiež často, pretože je to cenovo prijateľná 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. Keďže je tento druh pripojenia rozšírený, v záujme bezpečnosti je vhodné toto pripojenie zabezpečiť heslom. WiFi sieť pracuje na fyzickej a linkovej vrstve modelu ISO/OSI.[18]

**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 spracovávanía 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 textové 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).[22] 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*

Obrázok 3: Diagram HTTP komunikácie



*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 prijme, načíta hodnotu a odošle ju ako správu s odpoveďou. [22]

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, hneď 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 pripojenie. Táto vlastnosť HTTP pripojenia zabráňuje zahŕňaniu servera jediným klientom. Takto vybavuje server požiadavky postupne podľa času pripojenia klienta.

**HTTP je nezávislé od typu dát** Môžeme posilať akékoľvek dáta 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 zobrazíť v prehliadači, alebo ho stiahnuť. Pri HTTP protokole sa vyžaduje 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, keďže 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 dáta pri pripojení pomocou HTTP protokolu, tieto dáta sa ďalej vedia použiť pri ďalšom pripojení na server, kde sa pošlú detaily uložené u klienta serveru, vďaka čomu server skôr rozpozná klienta, resp. nebude od neho vyžadovať napr. dáta o prihlásení, keďže 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

Môžeme povedať, že HTTP protokol je najrozšírenejší na prehliadanie stránok, no chceme upriamiť pozornosť aj na možnosť jeho využitia v inteligentných budovách.[14]

### 2.4.3 Pripojenie do siete Internet

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 plášťom, 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 vlhkosťou. 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.[3]

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

**Dial-Up** Dial-Up pripojenie nazývame aj vytáčané pripojenie, ktoré funguje za pomoci 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. [17]

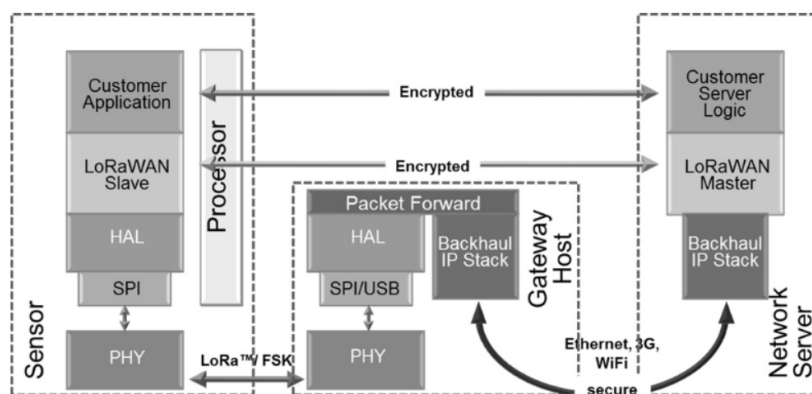
**GPRS** *General Packet Radio Service* je dátová služba pre mobilné telefóny operujúce v sieti GSM, teda funguje bezdrôtovo. Mobilný telefón pripojený do tejto siete je schopný sťahovať dáta rýchlosťou 40 kilobitov za sekundu a odosielať ich rýchlosťou 14 kilobitov za sekundu. Túto technológiu nahrádzajú novšie, rýchlejšie. [4]

**HSDPA** *High-Speed Downlink Packet Access* podobne ako GPRS je služba ktorá poskytuje bezdrôtové pripojenie do siete internet. HSDPA je rozdelené do viacerých tried, najvyššiu prenosovú rýchlosť má HSDPA triedy 10 a to približne 14 megabitov za sekundu. [5]

**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 sťahovanie je 100 megabitov

za sekundu a pre odosielanie až 50 megabitov za sekundu. V súčasnosti má takmer celosvetové pokrytie. [10]

**LoraWAN *Low Power Wide Area Network***, je navrhnutá na komunikáciu zariadení v *Internet of Things*, *Machine-2-Machine*. Určenie tejto siete je pre zariadenia s nízkou spotrebou elektrickej energie, siete s miliónmi zariadení. Topológia siete je založená na hviezdicovej topológii, kde centrálny uzol je pripojený na ďalšie uzly, ktoré majú ďalšie koncové zariadenia, alebo uzly. Centrálny uzol prijíma a odosiela 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 dát týmito virtuálnymi kanálmi je od 0.3 kilobitov za sekundu do 50 kiobitov za sekundu. Zabezpečenie siete je riešené šifrovaním na sieťovej vrstve, aplikačnej vrstve aj pri komunikácii koncových zariadení. V súčasnosti sa táto technológia ešte len dostáva do povedomia ľudí.[9]



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 objektmi,

rozhranie senzorov a akčných členov, *discovery service, knowledge management*. [23]

### 2.5.1 Požiadavky

Pre inteligentné prostredie môžeme zadať nasledujúcich päť vlastností, ktoré by mal spĺňať 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 dáta 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 vyzeráť.

#### **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 inteligentných objektov".**

Inteligentné objekty poskytujú množstvo služieb, ktoré sa môžu líšiť v počte a type služby medzi týmito inteligentnými 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 nemôžeme 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 inteligentné objekty tým, že poskytujú rôzne služby, ktoré sa menia počas životného cyklu inteligentného objektu.

Tým pádom nemôžeme iba definovať metódy na dynamické úpravy inteligentného objektu, ale aj spôsob, ako budú tieto metódy obsluhované a skonštruované.

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

Efektívny manažment inteligentných 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 prispôbovať, 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 inteligentný objekt počas behu. Pre tento účel sú strategické vyhľadávacie služby, ktoré dokážu v dynamickom kontexte vyhľadávať a získať inteligentné objekty na základe ich dynamických a statických vlastností.

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

Aplikácie a inteligentné objekty by mali byť jednoducho, rýchlo modelované a vylepšované pomocou príslušnej programovacej abstrakcie. Vývoj môže byť vedený programovaním, učením alebo obomi. To sa týka najmä inteligentných 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 môžeme uviesť softvérového agenta.

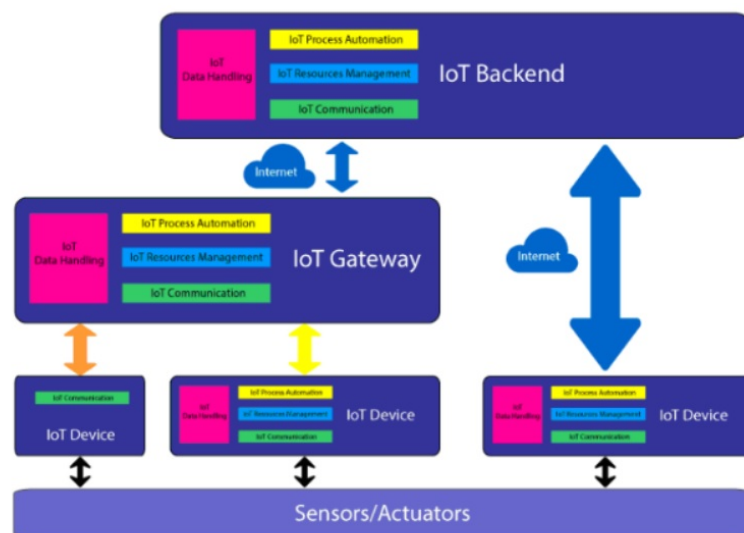
## **2.6 Architektúry**

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. Dáta sa nebudú odosielať nikam inam.

**Cloud-based architektúra** spracováva všetky dáta v nejakom dátovom úložisku v cloude. Príkladom takejto architektúry je systém fungovania aplikácie Waze pre mobilné zariadenia. Táto aplikácia musí byť pripojená do siete internet, zaznamenáva polohu a rýchlosť auta, dáta odosiela do cloudu, odkiaľ sa preposiela do ostatných zariadení. Na základe dát dokáže vyhodnotiť zápchu a mnohé iné vlastnosti cestnej premávky.





Obrázok 5: Schéma troch rôznych architektúr

## 3 Nástroje na praktickú realizáciu

Naším cieľom v tejto časti práce je aplikovať poznatky z oblasti Internet of Things na realizáciu 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 dáta 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 zaoberá 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>4</sup> prostredie na vývoj užívateľského rozhrania pre aplikáciu iViewer. Skladá sa z množstva nástrojov na úpravu a tvorbu vlastných objektov a súčastí 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 prispôbiť 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 môžeme 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>4</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 rozlíšiť 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 stmievaného svetla a podobne. Zároveň je možné tieto hodnoty meniť od užívateľa jednoduchým posunutím prstom.

**Pictures** Objekt obrázku môžeme 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 oboch 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** Keďže máme zadané 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 dáta zobrazovať v grafickom

rozhraní. Dáta uložené v tokenoch môžeme ďalej spracovať a ak ich potrebujeme zobrazíť, uložíme ich do hodnoty prepojenia. Každý dátový typ má svoju skupinu joinov, ktoré môžu uložiť dané dáta. 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>5</sup>

**Commands and macros** Pomocou príkazov, ktoré vieme implementovať do objektov guiDesigner vieme odosielať 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 môžeme zoskupiť do makier, ktoré dokážu odosielať jednotlivé príkazy do rôznych ovládacích systémov a zároveň nastaviť časové rozmedzie, kedy sa tieto príkazy majú spustiť a vykonať.

**Feedback** Spätná väzba (feedback) spracováva prijaté dáta z ovládacieho systému. Vo feedbacku môžeme definovať pravidlá podľa ktorých sa budú tieto dáta spracovávať. 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 spracovávať na základe pravidiel, ktoré si určíme v nastavení feedbacku. Regulárne výrazy vieme rozdeliť do skupín, na základe ktorých môžeme identifikovať časti prijatej správy, ktoré budeme následne spracovávať 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 môžeme 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.

### 3.1.2 iViewer

iViewer je aplikácia pre platformu Android alebo iOS, do ktorej je možné nahráť vytvorené grafické rozhranie. Nahŕavanie aplikácie prebieha priamo z počítača, na

---

<sup>5</sup>Uniform Resource Locator je formát na označenie internetového zdroja

ktorom je uložený projekt, podmienkou nahratia je byť v 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 grafick 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 poslúchaných 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 môžeme pristupovať pomocou skupinovej adresy, ktorá jednoznačne definuje každý zo senzorov a akčných členov. Pre potreby tejto práce je potrebné ozrejmiť 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, skupinovú adresou a dátovým typom. Ďalšie parametre, ktoré môžeme pridať každému objektu sú tagy na zoskupovanie objektov určitého typu, prípony alebo predpony na určenie jednotiek (napr. Celziov stupeň).

**Názov** Názov objektu môžeme voliť ľubovoľne. Je vhodné, aby vystihoval zariadenie, ktoré sa pod týmto názvom bude skrývať, prípadne druh zariadenia, jeho umiestnenie, alebo identifikačné číslo.

**Skupinová adresa** je zložená z troch zložiek, prvá má 4 bity, druhá 1 bit a tretia 32 bitov, teda rozmedzie adries 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 môžeme vybrať dátový typ, ktorý bude daný objekt pracovať. Prostredie LogicMachine poskytuje až 16 dátových typov a ďalšie ich podtypy.

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

Práve vďaka vlastnostiam firmvéru LogicMachine, teda možnosti abstrahovať senzory a akčné členy pomocou skupinových adries a ich hodnôt, môžeme povedať, že spĺňa 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 dáta. 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. Spúšťanie 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 spúšťať tento

skript. Tieto skripty môžu slúžiť napríklad na automatické zasvietenie svetla v určitom čase počas celého leta.

**Užívateľská knižnica** Do tejto knižnice môžeme pridávať rôzne skripty, ktoré ďalej 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 funkcionalít.

**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 zobrazíť napríklad v grafe.

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

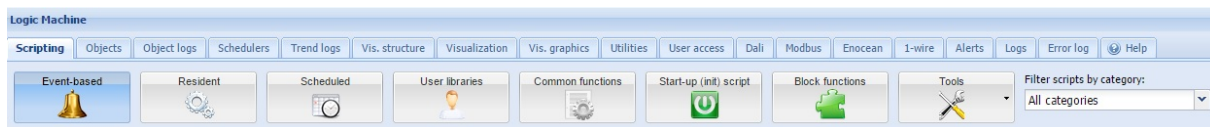
**Utilities** Utilities obsahujú nástroje na správu systému, aktualizovanie, resetovanie, zálohovanie a rôzne ďalšie konfigurácie vrátane zmien hesla, užívateľských práv, nastavenia KNX pripojenia.

**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 odosielaní dát. v karte Alerts uvidíme všetky výpisy, ktoré máme nastavené v Lua skriptoch 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 funkcionálít skriptov a podobne.



Obrázok 6: Vzhľad hlavného panelu



## 4 Implementácia

Implementovanie nášho riešenia bude popísané v tejto kapitole. Postupne si rozpíšeme 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 UDP alebo iViewer protokolu. UDP protokol sme vylúčili hneď 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 dáta, ktoré je treba poslať. K tomu nám pomohol prehliadač Google Chrome <sup>6</sup> a jeho možnosť preskúmať webové rozhranie LogicMachine pomocou DevTools <sup>7</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 odosielať a zároveň dáta, 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 odosielať s parametrami URL, metódou odosielania, hlavičkou, telom a callback funkciou, ktorá nám zistí, či boli dáta správne odoslané. 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>6</sup>Jeden z najpoužívanejších internetových prehliadačov na svete

<sup>7</sup>Je to skupina nástrojov na debugging priamo v Google Chrome

---

**Algoritmus 1** Ukážka HTTP requestu

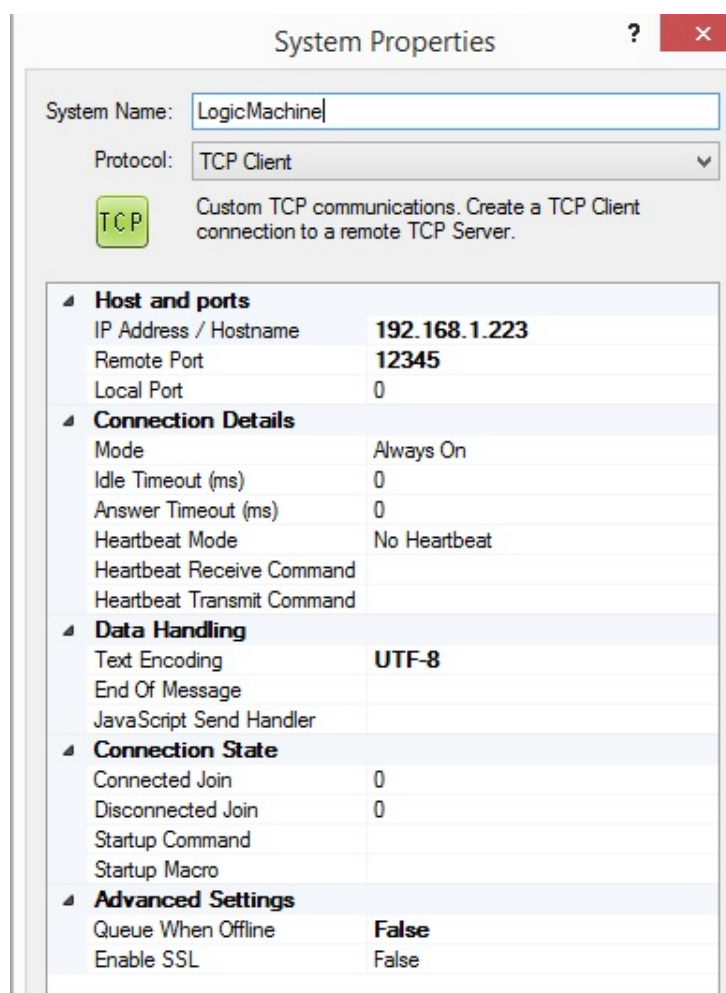
---

```
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 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 odosielať dáta do vopred vytvoreného cieľového systému. Cieľovým systémom môžeme rozumieť LogicMachine, konkrétne jej adresa a port ktorý má otvorený. V guiDesigner sme vytvorili systém TCP klient, ktorý spracováva dáta odosielané z iVieweru a zároveň ich aplikuje na objekty.



Obrázok 7: Nastavenie TCP klienta v CommandFusion

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

dáta sme dali vypísať do textového poľa vytvoreného v guiDesigneri. Pre potvrdenie, že do LogicMachine prichádzajú správne dáta sme dali všetku prichádzajúcu 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 náš ďalší cieľ sme si určili, že chceme odosielať 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 odosielať nové hodnoty do iVieweru na príkaz listenera, kde sa pomocou feedbacku spracujú a zároveň, aby aplikácia iViewer dokázala ovládať pomocou TCP spojenia objekty reprezentované skupinovými adresami v LogicMachine.

**Inicializačná komunikácia** Pri 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. Takýmto postupom sme už mali všetky inicializačné dáta v iVieweri.

**Komunikácia počas behu** Keďže 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 odosielať 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 odosieliť hodnotu.

---

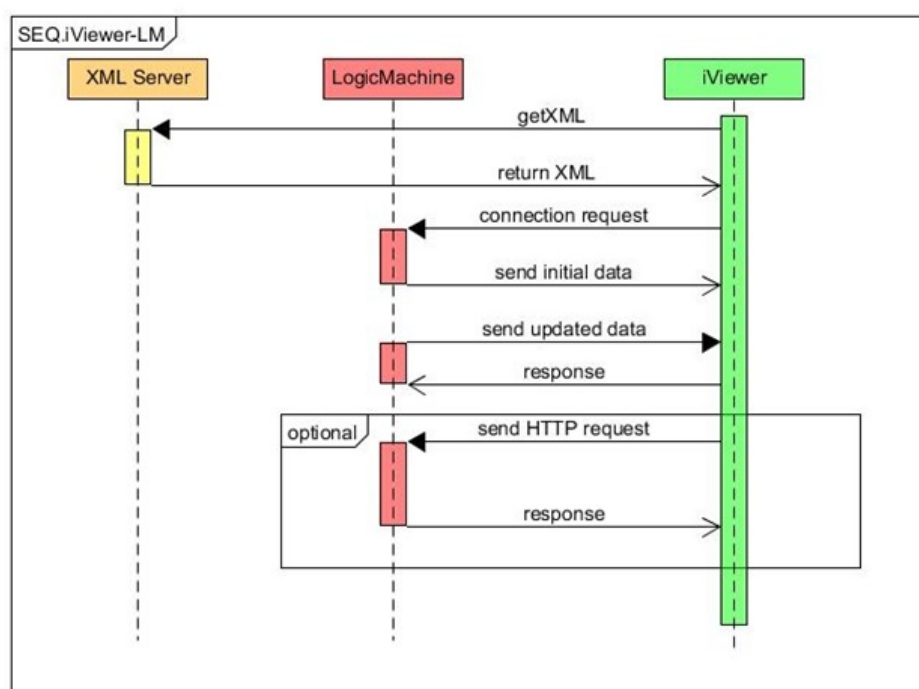
**Algoritmus 2** Vytvorenie socketu

---

```
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, môžeme pristúpiť k logike systému.

## 4.2 Aplikačná logika

Na správnu výmenu informácií medzi zariadeniami potrebujeme stanoviť pravidlá, ako budú vyzeráť dáta, ktoré budeme odosielať 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 zobrazovať 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. Rozoberieme si to po častiach.

**Prvá časť** Teda hodnota od 0-31. Keďže prvá časť skupinovej adresy môže nadobúdať dvojciferný tvar, aj prvá časť joinu sa bude skladať 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, jednociferná časť adresy sa bude rovnáť jednocifernému 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 rovnáť hodnota joinu hodnote skupinovej adresy.

Ako príklad uvidíme 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, no stretol nás ešte jeden problém. 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 adries podľa účelu. Je viacero spôsobov ako odlíšiť typ zariadenia, alebo dát. My sme to rozlíšili na základe strednej hodnoty skupinovej adresy. Keďže 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

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 adries 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 teplot, 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>8</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>8</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 konkrétne 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. Môžeme prehlásiť, že sa nám podarila spojazdniť komunikácia medzi týmito dvomi zariadeniami, čo vyústilo do využitia v praxi.

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



# Zoznam použitej literatúry

- [1] 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).
- [2] 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).
- [3] The foa reference for fiber optics - optical fiber. <http://www.thefoa.org/tech/ref/basic/fiber.html>. (Accessed on 05/11/2016).
- [4] Gprs and edge. <http://www.3gpp.org/technologies/keywords-acronyms/102-gprs-edge>. (Accessed on 05/11/2016).
- [5] Hsdpa. <http://www.3gpp.org/technologies/keywords-acronyms/99-hspa>. (Accessed on 05/11/2016).
- [6] 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).
- [7] 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).
- [8] 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).
- [9] Lora technology. <https://www.lora-alliance.org/What-Is-LoRa/Technology>. (Accessed on 05/11/2016).
- [10] Lte system. <https://sites.google.com/site/lteencyclopedia/home>. (Accessed on 05/11/2016).
- [11] 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).
- [12] Partners using energy harvesting wireless sensor solutions | enocean - technology. <https://www.enocean.com/en/technology/>. (Accessed on 05/12/2016).

- [13] 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).
- [14] Rfc 7230 - hypertext transfer protocol (http/1.1): Message syntax and routing. <https://tools.ietf.org/html/rfc7230>. (Accessed on 05/12/2016).
- [15] Rs-422 and rs-485 standards overview and system configurations. <http://www.ti.com/lit/an/s11a070d/s11a070d.pdf>. (Accessed on 05/13/2016).
- [16] The story behind bluetooth technology | bluetooth technology website. <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth>. (Accessed on 05/12/2016).
- [17] What is dial-up? <http://www.computerhope.com/jargon/d/dialup.htm>. (Accessed on 05/11/2016).
- [18] 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).
- [19] 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].
- [20] ERLICH, Y. A vision for ubiquitous sequencing. <http://genome.cshlp.org/content/25/10/1411/>, 2015. [Online; accessed 26-April-2016].
- [21] 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].
- [22] 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).
- [23] FORTINO, G., AND TRUNFIO, P. Internet of Things based on Smart Objects , 2014.

- [24] 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].
- [25] HÖLLER, J., TSIATIS, 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].
- [26] 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].
- [27] 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].
- [28] 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].
- [29] RAJI, R. Smart networks for control. [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=284793&url=http%3A%2F%2Fieeexplore.ieee.org%2Fexpls%2Fabs\\_all.jsp%3Farnumber%3D284793/](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=284793&url=http%3A%2F%2Fieeexplore.ieee.org%2Fexpls%2Fabs_all.jsp%3Farnumber%3D284793/), 1994. [Online; accessed 28-April-2016].
- [30] ROUSE, M. Internet of Things (IoT). <http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT/>, 2014. [Online; accessed 26-April-2016].
- [31] UNKNOWN. Internet of Things Global Standards Initiative. <http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/>. [Online; accessed 26-April-2016].
- [32] VERMESAN, O., AND FRIESS, P. 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].

- [33] 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].
- [34] Wood, Alex. *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press.

# Prílohy

A	Štruktúra elektronického nosiča . . . . .	II
B	Algoritmus . . . . .	III

# A Štruktúra elektronického nosiča

---

Algoritmus A.1 TCP ring v LogicMachine

---

# B    Algorithmus

---

**Algorithmus B.1** Listener v LogicMachine

---

```
1
2  if not client then
3      require('genohm-scada.eibdgm')
4      require('user.update')
5
6      ids = {}
7      objects = grp.tag('CommandF')
8
9      for _, object in ipairs(objects) do
10         ids[ object.id ] = true
11     end
12
13     function handler(event)
14         local id = event.dstraw
15
16         if ids[ id ] then
17             sendsock(id, event)
18
19         end
20     end
21
22     client = eibdgm:new()
23     client:sethandler('groupread', handler)
24     client:sethandler('groupwrite', handler)
25     client:sethandler('groupresponse', handler)
26 end
27
28 client:step()
```

---