TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Inteligentní domácnost a zabezpečení domu pomocí IoT

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Jaroslav Vondrák**
Vedoucí práce: Ing. Tomáš Martinec, Ph.D.

Liberec 2018

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Intelligent home and home security using IoT

Bachelor thesis

Study programme: B2646 – Information Technology
Study branch: 1802R007 – Information Technology

Author: **Jaroslav Vondrák**
Supervisor: Ing. Tomáš Martinec, Ph.D.

Liberec 2019

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Zadání bakalářské práce

Inteligentní domácnost a zabezpečení domu pomocí IoT

Jméno a příjmení: **Jaroslav Vondrák**
Osobní číslo: M16000064
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Zadávající katedra: Ústav mechatroniky a technické informatiky
Akademický rok: **2018/2019**

Zásady pro vypracování:

1. Seznamte se s principy inteligentní domácnosti a navrhněte univerzální platformu pro realizaci inteligentní domácnosti s použitím prvků IoT.
2. Realizujte vybrané klíčové prvky (bezpečnostní i komfortní) navrženého systému tak, aby bylo možné testovat jejich funkčnost a připravte i vhodnou centrální platformu pro řízení systému.
3. Vytvořte pro tyto prvky i pro řídící platformu software tak, aby bylo možné testovat a demonstrovat funkce inteligentní domácnosti a aby bylo možné systém v budoucnosti dále doplňovat o další prvky.
4. Navrhněte způsob programování automatizovaných akcí tak, aby si mohl domácnost automatizovat i laický uživatel.

Rozsah grafických prací:	dle potřeby dokumentace
Rozsah pracovní zprávy:	30–40 stran
Forma zpracování práce:	tištěná/elektronická

Seznam odborné literatury:

[1] Inteligentní řízení objektů ? zabezpečení, provoz, efektivita. Dostupné z: http://www.stavebnictvi3000.cz/clanky/inteligentni-rizeni-objektu-zabezpeceni-provoz-efektivita/.

[2] ZANDL, Patrick. Mesh sítě ? P2P architektura v bezdrátových sítích. Marigold [online]. 2003 [cit. 2018-02-25]. Dostupné z: https://www.marigold.cz/item/mesh-site-p2p-architektura-v-bezdratovych-sitich.

[3] Mesh wifi routery. In: Chip.cz [online]. 2018 [cit. 2018-03-13]. Dostupné z: https://www.chip.cz/casopis-chip/11-2017/mesh-wi-fi-routery/.

[4] Eleven internet of things iot protocols you need to know about. In: Rs-online [online]. [cit. 2018- 03-13]. Dostupné z: https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about.

Vedoucí práce:	Ing. Tomáš Martinec, Ph.D. Ústav mechatroniky a technické informatiky
Datum zadání práce:	10. října 2018
Předpokládaný termín odevzdání:	30. dubna 2019

L. S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci 10. října 2018

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Rád bych poděkoval Ing. Tomášovi Martincovi Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování této práce.

Anotace

Tato práce se zabývá především realizací univerzální platformy IoT systému, který je určen pro inteligentní domácnost. Systém je lehce rozšiřitelný pomocí zásuvných modulů. Je možné, např. přidávat nové aktivní prvky, komunikační moduly nebo komunikační protokoly do již existující infrastruktury. V teoretické části práce jsou rozebrány nejznámější komunikační protokoly, standardy a moduly. Dále je v teoretické části podrobně rozepsán návrh platformy. V praktické části jsou realizované vybrané klíčové prvky a k nim napsán ovládací software včetně řídicího systému. Nechybí ani návrh automatizovaných funkcí, které si uživatel může lehce nastavit přes ovládací panel.

Klíčová slova

Internet věcí, zásuvný modul, bezpečnost, inteligentní domácnost, automatizace

Annotation

This thesis deals with the implementation of the universal platform of the IoT system, which is created for the intelligent home. The system is easily extensible with plug-ins. For example, it is possible to add new active elements, communication modules or communication protocols to an existing infrastructure. In the theoretical part of the thesis, the most known communication protocols, standards and modules are discussed and the platform design is described in detail. In the practical part, the selected key elements are implemented and the control software is written together with the control system. There is also a design of automated functions that the user can easily adjust through the control panel.

Key words

Internet of Things, plug-in, security, smart home, automation

Obsah

Seznam ilustrací

Seznam tabulek

Seznam použitých zkratek a symbolů

IoT – Internet of Things, Internet věcí
PLC – PowerLine Communication, úzkopásmový a širokopásmový přenos zpráv po el. síti
FFD – Full Functional Device
RFD – Reduced Functionality Device
PAN – Personal Area Network, osobní síť
AES – Advanced Encryption Standard, standard pokročilého šifrování
LPWA – low-power wide-area network
I/O – Input/Outpus, vstupně-výstupní
OTA – Over-the-Air programming
RPi – Raspberry Pi
NAS – Network Attached Storage, datové úložiště na síti
GUI – Graphical User Interface, grafické uživatelské rozhraní
OP – Ovládací panel
ŘS – Řídicí systém

Pokud zkratka nemá připsaný český překlad, znamená to, že oficiální český překlad neexistuje nebo je překlad pro danou zkratku nepřesný a nevystihuje tak její přesný účel nebo funkci.

1. Úvod

Internet věcí, známý jako IoT (Internet of Things), je v dnešním světě častým tématem, o kterém se hovoří v mnoha oborech po celém světě. Počínaje inteligentním domem přes inteligentní křižovatky a konče inteligentním městem, které je řízeno autonomně pomocí strojového učení a umělé inteligence.

Chytrá zařízení můžeme vidět všude kolem nás. Mohou to být telefony, inteligentní zásuvky a světla, automatické otevírání oken, nejrůznější čidla a mnohem více. Smyslem IoT je vzít všechna tato zařízení a spojit je v jeden celek řízený jedním protokolem. Využití takové infrastruktury může být nejen pro naše pohodlí, ale i pro naší bezpečnost. V dnešní době již existují chytré domy, které sledují pravidelné činnosti jednotlivých obyvatel (např. rodiny). K takovým činnostem patří např. to, kdy jdou spát, kdy vstávají, kdy jedí, kdy se sprchují atd. Tímto sledováním můžeme omezit např. náklady na vyhřívání vody. Pokud mezi 8 až 14 hodinou nebude ani jeden z obyvatelů doma, není potřeba např. vyhřívat vodu. Systém ovšem ví, že po 14 hodině se již obyvatelé vrací domů, a tak (např. půl hodiny předem) automaticky zapne vyhřívání vody. A to je jen špička ledovce toho, co takový systém může obstarávat.

Pokud se přesuneme do nějaké větší infrastruktury, např. k dopravě, má IoT úplně jiné využití. Tam se může jednat o řízení dopravy. Díky inteligentním křižovatkám pak nemusí docházet k ucpávání silnic (doprava může být např. odkloněna a přesměrována přes jiné silnice a křižovatky).

Můžeme se také přesunout do inteligentních měst, kde systém může optimalizovat využití energetických zdrojů, a tím omezit znečišťování životního prostředí.

Jak lze vidět, IoT má potenciál v mnoha oborech, např. zemědělství, správa energie, monitoring, lékařství, doprava, letectví, média, bezpečnost.

Pojďme se tedy vrátit k inteligentnímu domu. Pokud se ponoříme trochu hlouběji do této problematiky, zjistíme, že nejde o nic jiného než spoustu senzorů, obvodů, řídicích systémů a softwaru, který zpracovává všechna data a na základně řídící logiky se pak rozhoduje, jak s nimi naložit a co se stane dál. Hlavní částí takového systému je jeho, jak už bylo naznačeno, řídící logika, tedy software, který má vše na starost a který zajišťuje veškeré funkce inteligentní domácnosti za pomocí různých zařízení. Nezapomeňme také na zabezpečení systému. Je potřeba chránit citlivá data, která si systém uchovává, ale také chránit systém před vnějšími vlivy, jako je voda, oheň anebo výpadek energie.

Tato práce se zabývá právě touto problematikou a bude zde rozebrán návrh a poté i realizace systému IoT pod názvem „IoT - HS“ (Internet of Thinks – Home System), který zajistí komfort, úsporu energií a především bezpečnost pro všechny obyvatele inteligentního domu.

2. Inteligentní domácnost

Inteligentní dům je především takový dům, který nám dává pocit komfortu a bezpečí. Inteligentní dům ovšem necílí pouze na zpříjemnění bydlení, ale také na to, aby zajistil menší energetickou spotřebu a tím snížil náklady na provoz a byl tedy přívětivější k životnímu prostředí.

Mozkem inteligentního domu můžeme označit řídicí systém, kde je prováděna všechna řídící logika díky jednotlivým aktivním prvkům rozmístěným po celém domě. Každý z těchto prvků např. senzor, světlo, zásuvka a další, je napojen k řídicímu systému díky různým technologiím. Buď pomocí vodičů, nebo bezdrátově rádiovými vlnami. Nejčastější a nejméně problematická cesta ke spojení aktivních prvků s řídicím systémem je právě pomocí bezdrátových technologií. Na trhu najdeme mnoho možností, jak komunikaci zprostředkovat. Nejvíce rozšířenou technologií je Bluetooth a Wi-Fi. Touto technologií disponuje nespočet různých zařízení jako např. telefony, notebooky, tablety a mnoho dalších. Není tedy divu, že uživatelé si chtějí koupit zařízení, které mohou ihned ovládat pomocí svého telefonu.

Tato práce ovšem směřuje trochu výše. Bude v ní využito hned několik technologií a komunikačních modulů zároveň. Pokud se zaměříme pouze na komunikaci přes Wi-Fi, je celý systém velice náchylný na vnější útok (rušičky, nabourání do sítě a další). Pokud použijeme více komunikačních prostředků, je možné, že když jeden selže, nahradí ho jiný a systém bude stále provozuschopný. Před návrhem samotného systému je potřeba se seznámit se součástmi inteligentního domu a technologiemi, které se používají pro přenos dat mezi čidly a řídicím systémem.

2.1. Řídicí systém

Řídicí systém je mozkem celého domu. Právě řídicí systém rozhoduje o tom, kdy se má ohřívat voda, jaké světlo se v domě zapne, kdy se má zapnout klimatizace, která ventilace se má otevřít a mnohé další. Pokud řídicímu systému přidáme o něco vyšší inteligenci, může být velice dobrým společníkem v domácnosti. Ovládání hlasem, dáváním různých příkazů anebo jen zeptáním se „jaké je dnes počasí?“ nám dává jistý pocit bezpečí a komfortu. Díky vyšší inteligenci systém může identifikovat různé členy domácnosti a tak rozpoznat, že se v domácnosti pohybuje někdo neznámý, nebo že je v domácnosti neobvyklá aktivita.

Systém nám pak pomocí různých prostředků může dát najevo, co se děje. Dokonce nám může nabídnou možnosti, jak problém vyřešit.

Komunikaci a nastavení řídicího systémů můžeme provést hned dvěma způsoby. Jeden je, jak už bylo zmíněno, díky hlasovému ovládání. Druhým může být panel umístěný na zdi, kde si uživatel může nastavit systém dle své potřeby a sledovat stav systému.

Otázkou je, na čem takový řídicí systém bude fungovat, jaký bude mít výkon, co vše bude potřebovat ke svému provozu atd. Pokud se bude jednat o menší síť s několika málo zařízeními, je možné použít jednodeskový počítač. Nejlepším příkladem je

Raspberry Pi (dále jako RPi), který se využívá v mnoha malých systémech, a to hlavně díky jeho malým rozměrům a nízké pořizovací ceně. RPi nabízí skoro stejné funkce jako slabší stolní počítač. Je možné k němu připojit USB zařízení, monitor, reproduktory, ethernet rozhraní a také nechybí Wi-Fi nebo Bluetooth. Je možné na něj nainstalovat plně funkční operační systémy, jako Raspbian, Ubuntu mate nebo jinou distribuci Linuxu. Výhodou RPi jsou GPIO piny, díky kterým je možno k počítači připojit různá zařízení. Pořizovací cena se pohybuje kolem 1200 Kč. Pokud se bude jednat o méně komplikovaný systém, může se používat např. počítač Arduino. Pokud bude systém vyžadovat vyšší hardwarové nároky, je zapotřebí již zapojení plně funkčních serverů, které budou zpracovávat požadavky.

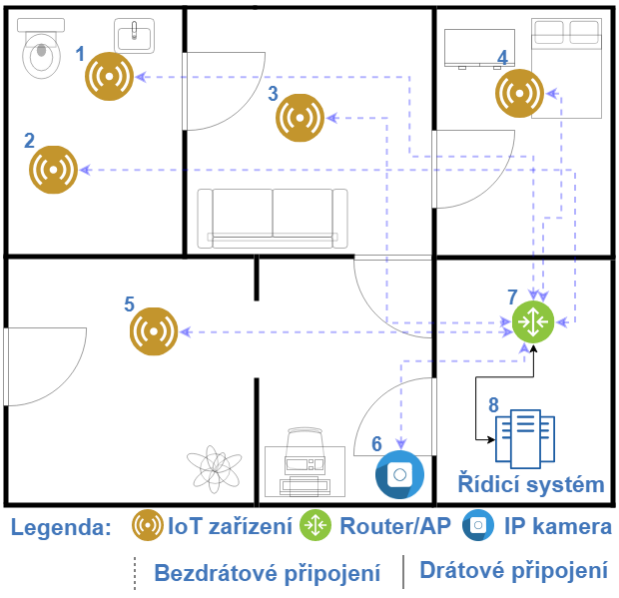
Je potřeba si také uvědomit, kolik dat bude systém ukládat. RPi si svá data ukládá na SD kartu, která má omezený počet přepisování. Pokud tedy systém bude sbírat velká množství dat, je potřeba ukládání dat obstarat jiným způsobem.

2.2. Infrastruktura

Nejspolehlivější cestou propojení všech aktivních prvků s centrálním systémem je pomocí kabeláže. Tato cesta ale není vždy optimální. Zejména kvůli vysoké ceně a časové náročnosti. Kabeláž má ale tu výhodu, že je spolehlivá a signál nelze ničím rušit (pouze přestřižením kabelu).

Nejjednodušším způsobem je bezdrátová komunikace. Ta ovšem má také jisté nevýhody. Rušení rádiových signálů, rádiový šum a především různé překážky, přes které signál musí projít. Existují různé topologie, které lze použít pro bezdrátovou komunikaci. Nejčastěji se používá stromová topologie, kdy je k jednomu rozbočovači připojeno několik zařízení. Pokud existuje více rozbočovačů, jsou propojeny mezi sebou (pokud existuje jen jeden rozbočovač, jedná se o topologii hvězdicovou).

Na obrázku 1 můžeme vidět jednoduchou infrastrukturu na principu hvězdicové topologie. Všechny aktivní prvky jsou spojeny s rozbočovačem pomocí bezdrátové technologie. Rozbočovač poté předá data určenému adresátovi. Na první pohled vidíme, že použití této topologie by nebylo ideální, jelikož signál k aktivnímu prvku č. 2 a 1 musí překonat tři zdi. Připojení tedy nebude zcela stabilní a spolehlivé.



Obrázek 1- Hvězdicová topologie

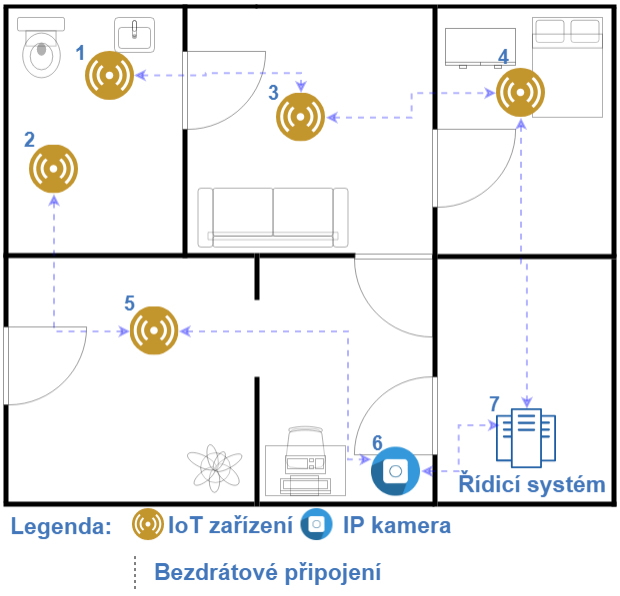
Jedním z řešení by mohlo být přidání rozbočovače do levé dolní místnosti. Rozbočovač poté pomocí PLC (power line communication), nebo jinou technikou, propojit s řídicím systémem. Touto technikou nám vznikne topologie stromová. V případě, že je problém s komunikací, nebo je signál slabý, připojíme jednoduše nový rozbočovač a tím by se měl problém vyřešit. Ani jedna z topologií, ať už hvězdicová či stromová, není pro rozsáhlou infrastrukturu vhodná. Nejlepší technikou, jak propojit zařízení mezi sebou, je pomocí topologie typu mesh.

- **Mesh topologie**

V mesh topologii se z aktivních prvků stávají tzv. nody (uzly). Každý nod se chová jako opakovač signálu. To znamená, že všechna data, která příjme, odešle dál bez ohledu na to, pro koho jsou určena (pokud jsou určena jemu, data samozřejmě dále nepřeposílá). Může ovšem existovat i tzv. směrovací tabulka, ve které jsou uloženy informace o různých uzlech, a jak se k určitému uzlu dostat. Směrovací tabulka také brání zacyklení. U této topologie existuje určitá nejednoznačnost. Můžeme se setkat s názvy jako smíšená topologie, obecný graf, úplný graf apod.

I přes jistou nejednoznačnost si pro jednoduchost rozdělme tuto topologii pouze na smíšenou a topologii „každý s každým“, jinak nazýváno jako úplný graf. Se smíšenou topologií se můžeme setkat často u sítí, u kterých chceme zvýšit odolnost vůči výpadkům. Nod si nyní můžeme představit jako router nebo switch. Tato topologie se využívá v telekomunikačních sítích nebo na Internetu. Úplný graf by u těchto infrastruktur nebylo možné použít. Máme zde nespočet uzlů a propojit každý s každým, by bylo velice obtížné. Centrální prvek v těchto topologiích neexistuje. Pokud se přeruší jedna cesta, je vždy nahrazena jinou a data by vždy, za určitých podmínek, měla dojít k příjemci.

Úplný graf je velice podobný smíšené topologii, jen s tím rozdílem, že každý nod je propojený s každým. Redundance je zde obrovská a řízení provozu obtížnější. Je potřeba softwarově vyřešit problémy zacyklení a vícenásobného příjmu dat.



Obrázek 2 - Mesh topologie

Na obrázku 2 je zobrazen příklad, jak může vypadat mesh síť v případě zapojení inteligentní domácnosti. Každý prvek (tedy nod) je již jakýmsi samostatným směrovačem. Tímto řešením vznikla decentralizovaná síť a existuje více cest, jak se dostat k příjemci. V minulém řešení (obrázek 1) k selhání sítě stačilo odpojit AP.

Výhody takové sítě jsou především snadné rozšíření, decentralizace a alternativní cesty při selhání uzlů. Nevýhodou takové sítě je, že vyžaduje směrování provozu a především ochranu proti zacyklení. Jestli výhody převažují nad nevýhodami, to záleží na velikosti sítě. Pokud se v domě budou nacházet dvě žárovky a teplotní čidlo, není potřeba postavit síť na topologii typu mesh, ale postačí jedna z výše uvedených topologii (např. hvězdicová topologie).

Pokud se bavíme o topologii typu mesh, nesmíme zapomenout zmínit nejpoužívanější protokol Z-Wave a nejznámější komunikační modul ZigBee, který je přizpůsoben nejen pro tuto topologii (více v kapitole 2.3. Komunikace).

2.3. Komunikace

Výběr topologie je jedna věc, výběr nejvíce vyhovujícího komunikačního zařízení nebo technologie už je o něco těžší. Jak už bylo zmíněno, pro rozsáhlou síť v inteligentním domě je nejlepší topologie typu mesh. Musí se ale vyřešit, jak jednotlivá zařízení propojit. To lze pomocí bezdrátových technologií jako např. Bluetooth, Wifi, ZigBee. Aby inteligentní dům opravdu patřil do skupiny IoT, je potřeba jej propojit s vnějším světem pomocí internetu. K propojení se světem ovšem nemusí vždy sloužit jen internet.

Existují technologie jako SigFox, LoRaWan (Long Range Wide Area Network) nebo NarrowBand IoT. Tyto technologie fungují na principu, který známe u mobilních telefonů. Pokud se chceme spojit se vzdáleným zařízením, operátor nám díky různým branám zprostředkuje komunikaci.

2.3.1. Protokol Z-Wave

Z-Wave je bezdrátový komunikační protokol, který se používá převážně v inteligentních domech pro komunikaci a ovládání např. světel, zámků, termostatů a dalších zařízení. Tento protokol je speciálně vytvořen pro topologii mesh. Každá síť vytvořena pomocí protokolu Z-Wave je tvořena nody (např. inteligentní světla) a centrálním prvkem, který má na starost ovládání celé domácnosti. Centrální prvek také slouží jako brána (gateway) do vnějšího světa. Tento protokol lze implementovat do libovolného zařízení. Na trhu se najde celá řada zařízení s tímto protokolem jako inteligentní zásuvky, různé bezpečnostní prvky a jiná automatizační zařízení.

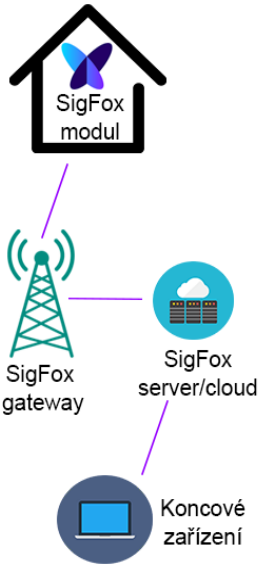
Parametry a specifikace
Frekvence: 868,40; 868,52; 869,85MHz Dosah: až 100 m Přenosová rychlost: závislá na zařízení

2.3.2. Protokol SigFox

SigFox je bezdrátová technologie vyvinutá francouzskou společností Sigfox, která se zabývá výrobou nízkoenergetických zařízení. Nízkoenergetickým zařízením rozumíme takové zařízení, které nemusí být neustále zapnuté a vysílá poměrně malé množství dat. Rozhodně se nehodí pro běžné účely chytré domácnosti, jako je bezpečnostní nebo kamerový systém. Může ovšem posloužit jako jakési diagnostické zařízení, nebo pokud se stane nějaká neobvyklá událost, může sloužit jako komunikační prostředek na delší vzdálenost.

Síť SigFox patří mezi LPWAN sítě (Low-Power Wide-Area). Vyžaduje k provozu mobilního operátora, který zajišťuje díky branám (serverům) provoz.

Všechna odeslaná data jsou uložená na SigFox serveru/cloudu, ke kterému má uživatel přístup odkudkoliv na světě. Pokud se vrátíme k příkladu s diagnostickým zařízením, může si každý člen domácnosti zkontrolovat, zda je domácnost v pořádku nebo dokonce odeslat příkaz (např. k zapnutí ohřevu vody) ze svého mobilní telefonu nebo jiného zařízení (obrázek 4). Jak vidíme, chytrá domácnost nemusí být přímo napojena na internet, a i přesto můžeme ovládat nebo sledovat stav sítě. Může ovšem nastat diskuze, jestli se stále jedná o IoT, protože domácnost není napřímo připojena přes veřejnou, nutno zmínit zabezpečenou, síť.



Obrázek 3 – SigFox

SigFox má ovšem svá omezení, a to především velikost poslaných dat. Standard podporuje až 140 odeslaných zpráv za den směrem od zařízení k serveru o velikosti 12 bajtů. Celkem jsou od serveru k zařízení za den povoleny až 4 zprávy o velikosti 8 bajtů. Díky takto nízkému množství posílaných dat je možné modul napájet až 15 let přes bateriové články. Síť funguje na frekvenci 868 MHz (v Evropě) a je vysoce odolná vůči rušení. Je také dobře chráněná proti zneužití.

K přístupu do sítě je zapotřebí zakoupit licenci, která se musí každoročně obnovovat. Je možné si samozřejmě zakoupit i víceleté licence.

Parametry a specifikace
Protokol: SigFox Frekvence: 868 nebo 915 MHz Dosah: až 50 km (ve volném krajině) Přenosová rychlost: až 100 bit/s

2.3.3. Protokol LoRaWAN

LoRa je jedno z řešení pro přenos malých dat na velké vzdálenosti, kde je kladen velký důraz na malou spotřebu. LoRaWAN se říká síti, na které LoRa pracuje. V Evropě LoRaWAN funguje v pásmu 868 MHz a rychlost přenosu je 250 bps až 50 kbps. Zařízení dokáže fungovat na bateriích 5-15 let (při přenosech malého množství dat).

K realizaci sítě potřebujeme bránu, která zprostředkovává komunikaci mezi jednotlivými zařízeními a internetem. Ta je nazývána jako „The Things GateWay“ (produkt The Things). Dalším zařízením je „The Things Uno“, což je jednodeskový počítač, ke kterému lze připojit senzory. Tento počítač sám obstará připojení do sítě LoRaWAN. Výhodou je, že je postaven na platformě Arduino (velká dostupnost knihoven a velké možnosti programování). Brána pak obstará přenos dat na cloud, kde uživatel může sledovat stav senzorů nebo jiných zařízení. Brána spolehlivě zprostředkuje spojení mezi zařízeními v okruhu až 5 km a síť může mít až 10 000 zařízení.

Výhodou oproti SigFoxu je, že LoRaWAN není omezena na přenesené množství dat za den, lze tedy odeslat neomezené množství za jeden den. Dále není potřeba obnovování licence do sítě.

2.3.4. Standard Wi-Fi

Nejznámější technologie pro připojení více zařízení v síti je pomocí Wi-Fi sítě. Nejznámější je proto, že tuto technologii podporují všechna zařízení jako např. telefony, chytré televize, chytré zásuvky, trouby, reproduktory, automobily a mnohá další. Je to proto ideální technologie pro vybudování levné a výkonné sítě.

Wi-Fi operuje v bezlicenčním pásmu 2,4 – 2.485 GHz. Komunikační pásmo a rychlost určují standardy, které popisují bezdrátovou komunikaci (viz tabulka 1). Wi-Fi zajišťuje komunikaci pouze na spojové vrstvě, jsou proto přenášeny pouze zapouzdřené rámce a ve vyšších vrstvách je možné využít jiné protokoly (např. pro komunikaci v počítačové síti se využívá rodina protokolů TCP/IP).

Tabulka 1- Wi-Fi standardy

Standard	Frekvence [GHz]	Max. Teoretická přenosová rychlost [Mb/s]	Průměrná skutečná rychlost [Mb/s]
802.11	2,4	2	0,9
802.11a	5	54	23
802.11b	2,4	11	4,3
802.11g	2,4	54	19
802.11n	2,4 nebo 5	600	-
802.11ac	2,4 nebo 5	1800	-

Důležitým prvkem v síti je zařízení AP (Access Point – přístupový bod), ke kterému se připojují další různá zařízení. Pokud se tedy chce zařízení připojit k jinému zařízení nebo k internetu, je potřeba veškerou komunikaci směrovat přes AP. Pokud AP přestane fungovat, komunikace již není možná (viz obrázek 1). Existuje také varianta, která se nazývá Ad-hoc. Tato varianta umožňuje spojení zařízení bez použití přístupového bodu (princip mesh sítě viz obrázek 2). Výhodou Wi-Fi je také to, že má zabudované šifrovací algoritmy. Doporučené šifrování je pomocí WPA2. Pokud je potřeba vyšší zabezpečení, je možné povolovat připojení do sítě pomocí certifikátů. Bezpečnost však může být diskutabilní. Na jednu stranu je možné použít prověřené šifrovací techniky, na druhou stranu je zabezpečení Wi-Fi známé a je tu stále možnost nabourání do sítě. Další bezpečnostní dírou může být rušení frekvence nebo podstrkování cizích paketů. Bezpečností systém by měl být tedy jednoznačně oddělen od všech dalších sítí, nebo být realizován úplně jinou technologií. Z výše uvedeného je zřejmé, že realizaci bezpečnostního systému je nejlepší provést pomocí drátů ve zdech. V tomto případě již není tak snadné narušit infrastrukturu bezpečnostního systému.

2.3.5. Standard Bluetooth

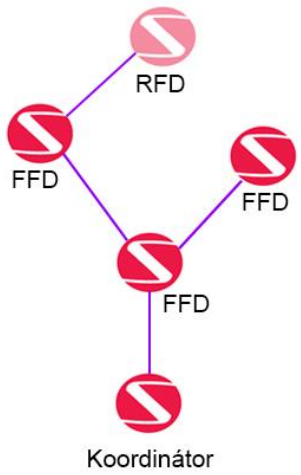
Bluetooth je dalším příkladem velice známého a rozšířeného standardu pro bezdrátovou komunikaci, která dokáže propojit dvě a více komunikačních zařízení. Tuto technologii nalezneme opět v zařízeních jako telefony, sluchátka, notebooky, hodinky a v mnoha dalších. Bluetooth je definováno standardem IEEE 802.15.1. Je děleno podle výkonnostních úrovní, kterými je definována i komunikační vzdálenost. Např. výkon 100 mW dokáže komunikovat až na vzdálenost 100 m. Této vzdálenosti je ovšem možné dosáhnout jen tehdy, pokud není mezi zařízeními žádná překážka. Pokud se mezi zařízeními vyskytuje např. zeď, signál velice rychle slábne, a tím se snižuje rychlost přenášených dat. Bluetooth v nejnovější verzi 5.0 dosahuje rychlosti přenosu dat až 255 Mbit/s.

Bluetooth pracuje v pásmu 2,4 GHz a využívá metodu FHSS, která zajišťuje lepší odolnost vůči rušení díky přeskakování mezi 79 frekvencemi. I přes mechanizmus FHSS není vhodné používat tuto technologii např. pro bezpečností systém z důvodu velké náchylnosti na překážky. U Wi-Fi bylo zmíněno, že pracuje pouze na spojové vrstvě a o nic dalšího se již starat nemusí. Bluetooth je v tomto směru trochu složitější, jelikož pracuje na více vrstvách ISO/OSI modelu. Je tedy potřeba mít speciální protokoly, které zajistí komunikaci. Tento způsob velice komplikuje SW podporu Bluetooth jako např. vývoj ovladačů na různá zařízení.

2.3.6. Komunikační modul ZigBee

ZigBee je bezdrátová technologie určena pro zařízení s nízkým výkonem. Hodí se zejména pro automatizaci v inteligentních domech. U ZigBee byl kladen důraz na nízkou spotřebu energie, a tak se hodí do zařízení, která mají periodicky se opakující činnost (např. přenos dat z pohybového čidla). Je plně podporována topologie mesh se směrováním provozu. Podporuje také zapojení do stromové nebo hvězdicové topologie. Standard ZigBee (IEEE 802.15.4) jasně definuje zapojení sítě. Standard definuje dva druhy zařízení. Prvním z nich je FFD (Full Functional Device), což je zařízení s plnou funkčností. Druhým pak je definováno zařízení RFD (Reduced Functionality Device), které je již funkčně omezeno. V síti také existuje koordinátor, který má např. za úkol přidělování adres připojeným nodům.

Jelikož RFD nezajišťuje veškeré služby, jako např. přeposílání dat nebo implementaci kompletního protokolového rámce, je nezbytné, aby byl zapojen jako koncové zařízení. RFD se používá z důvodu omezení hardwarové náročnosti. Komunikace probíhá pouze s koordinátorem. FFD již implementují kompletní protokolové rámce a zajišťují přeposílání dat (směrování), a tak mohou být zapojeny libovolně. Provoz je celkově kontrolován koordinátorem, který zjišťuje stav sítě (připojení nového prvku, start sítě). Jednotlivá zařízení jsou adresována pomocí adresy o délce 64 bitů. Lze tedy připojit až 65 535 zařízení. Koordinátor zajišťuje ještě přidělování PAN ID, které slouží k rozdělení sítí v případě, že by v blízkosti byla vytvořena jiná síť pomocí technologie ZigBee.



Obrázek 4- ZigBee

ZigBee používá šifrovací mechanismus založený na AES s klíčem o délce 128 bitů. Lze také použít ověřování pomocí MAC adresy. Šifrování probíhá na síťové vrstvě.

Parametry a specifikace
Standard: IEEE 802.15.4 Protokol: SigFox Frekvence: 868 MHz; 902-928 MHz; 2,4 GHz Dosah: 10-100 m Přenosová rychlost: až 250 kbit/s (2,4 GHz)

2.3.7. Komunikační modul IQRF

Ve vývoji komunikačních modulů pro IoT nezaostává ani Česká Republika. Jičínská společnost IQRF Tech představila v roce 2004 bezdrátový modul pod názvem IQRF. Díky protokolu IQMESH, který se stará o veškeré směrování a přenos dat, je možné připojit všechny IQRF moduly pomocí mesh topologie, což zajišťuje velké pokrytí rozlehlých ploch (jeden modul dokáže komunikovat až na vzdálenost 500 m ve volném prostoru). Modul je osazen 12 I/O piny, kterými lze ovládat různá čidla nebo jiná zařízení. Každý modul má již nahraný svůj vlastní operační systém. Díky hardwarovému profilu lze ovlivňovat funkčnost každého komunikačního modulu (např. ovládání pinů).

Každá síť postavená na těchto modulech má jeden řídicí prvek (tzv. koordinátora). Dále v síti existují podřízené prvky, které jsou řízeny koordinátorem. Pokud je potřeba poslat data na jeden z modulů, jsou data směrována (je nalezena nejkratší cesta) k danému příjemci. Pokud je do sítě zapojen nový modul, koordinátor tomuto modulu přidělí časový interval, ve kterém může vysílat svá data. Jinak by došlo ke kolizi dat a docházelo by k velkému zpoždění příjmu dat nebo jejich úplné ztrátě. Do sítě je možno připojit celkem 240 modulů. To se může zdát jako vysoké číslo, díky kterému lze dosáhnout velkého pokrytí. Vysílání paketu se ovšem pohybuje od 30 do 50 ms, a tak je potřeba zvážit časovou náročnost vysílání dat (oběma směry). Při vysílání v síti, která má celkem 100 modulů, trvá poslání paketu 5 sekund (jeden skok trvá 50 ms), než je zpráva poslána přes celou sít až k příjemci. Na odpověď se tedy pak čeká celkem 10 sekund v případě, kdy je zpráva směrována přes všech 100 nodů.

IQRF se hodí především tam, kde je potřeba přenášet malý objem dat. Klade se zde velký důraz na energetickou úsporu a především bezpečné a spolehlivé zasílání dat v síti.

Parametry a specifikace
Protokol: IQMESH Frekvence: 868; 916; 433 MHz Dosah: až 500 m Přenosová rychlost: až 20 kb/s

2.3.8. Komunikační modul NodeMcu CP2102 Lua ESP8266

NodeMcu CP2102 Lua ESP8266 (dále jen NodeMcu) je vývojová deska, která je vybavena mikročipem ESP8266, CP2102 a mikro USB, přes který lze vývojovou desku programovat. Mikročip ESP8266 umožňuje jiným mikrokontrolérům připojit se na Wi-Fi síť a vytvářet jednoduchá TCP/IP spojení. Čip CP2102 obstarává převod z USB na signály sériové linky.

NodeMcu má 4096Kb FLASH paměť, obsahuje celkem 12 pinů z toho jeden analogový, nízké nároky na spotřebu a především OTA (On The Air – technika umožňující přehrání firmwaru na dálku přes síť). Díky všem těmto uvedeným vlastnostem byla tato vývojová deska vybrána pro návrh a realizaci IoT systému. Její vlastnosti budou rozebrány v dalších kapitolách o návrhu platformy a realizaci systému.

2.4. Návrh platformy

Návrh platformy pro inteligentní dům je stejně důležitý, jako položit základy pro dům samotný. Při takovém návrhu je potřeba promyslet v čem bude např. systém programován, na jakém operačním systému bude spuštěn server, jaké hardwarové komponenty budou použity pro stavbu různých senzorů a jaká bude logika komunikace mezi různými prvky sítě. Je tedy potřeba vymyslet univerzální platformu, která bude umožňovat jednoduché a rychlé přidávání nových prvků do sítě, programování různých doplňujících funkcí (tzv. zásuvných modulů/pluginů), aktualizaci softwaru a také přidávání zcela nových prvků od jiných výrobců.

2.4.1. Řídicí systém

Už na začátku byla zmíněna nejdůležitější část inteligentního domu a to řídicí systém. Ten, jak již bylo zmíněno, by mohl být spuštěn na nějakém jednočipovém počítači. Pro tento návrh platformy byl ovšem vybrán zcela jiný hardware, a to NAS server od Synology. NAS server je síťové uložiště pro různá data a slouží primárně k zálohování. Pokročilejší servery jsou ale vybaveny moderními operačními systémy postavenými na Linuxu. Lze na nich provozovat velké množství služeb jako např. multimediální server, webový server, databázový server a mnohé další. Co je nejdůležitější, je ovšem to, že na serveru od Synology lze spouštět programy vytvořené v programovacím jazyce Java.

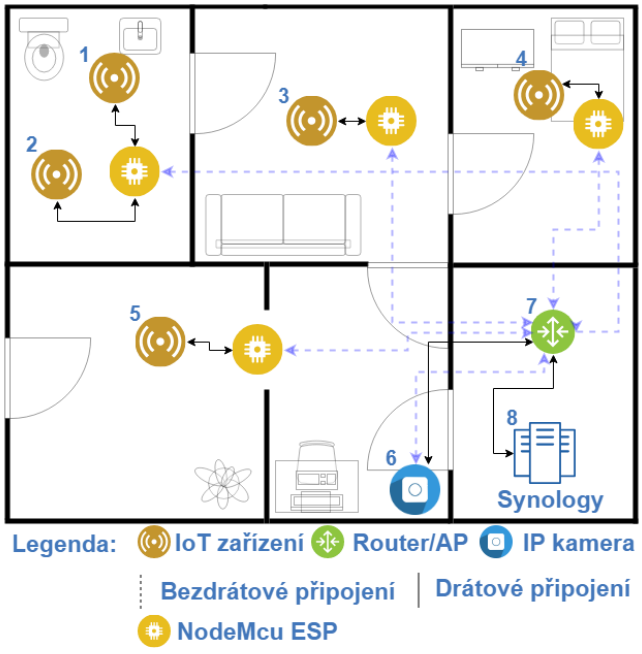
NAS server byl vybrán z důvodu velkého výpočetního výkonu, spolehlivosti a hlavně možnosti velkého výběru služeb. Pokud by bylo vybráno např. Raspberry, nedosáhneme ani poloviny výkonu a možností, které nabízí server od Synology, nebude možnost provozovat další služby jako multimediální server, zálohování dat nebo webové služby v takové míře jako s NAS serverem. NAS server podporuje připojení mnoha IP kamer, synchronizaci dat v síti, nejrůznější zabezpečení dat a především zapojení více pevných disků. Systém tedy nebude omezen na jednu SD kartu.

Pro tuto práci bude použit NAS Synology DS418 DiskStation.

2.4.2. Aktivní prvky

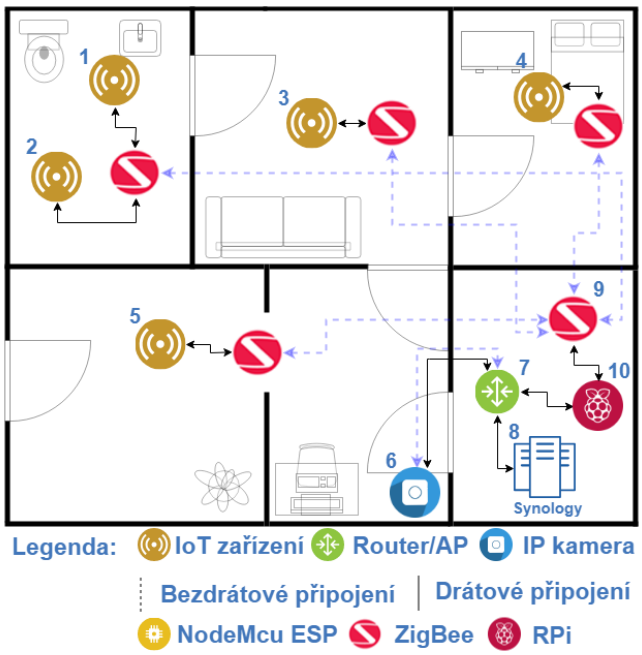
Jelikož se jedná o univerzální návrh platformy, u aktivních prvků se není třeba držet zásad o použitém hardwaru nebo dokonce softwaru, který aktivní prvek v síti bude používat. Veškerá kompatibilita bude řešena řídicím systémem a díky podpoře zásuvných modulů. Viz kapitola 2.4.5. Zásuvné moduly.

V této práci budou veškeré aktivní prvky realizovány pomocí vývojové desky NodeMcu CP2102 Lua ESP8266, díky které bude možné aktivní prvky připojit do Wi-Fi sítě a komunikovat s řídicím systémem nebo ostatními prvky. Komunikace je tedy realizována pomocí standardu Wi-Fi a protokolech TCP/IP. Wi-Fi se v současné době běžně v domácnostech používá zejména pro distribuci internetu. Pokud se zajistí přísná pravidla ve firewallech a řízení sítě, (jako např. oddělená LAN síť pouze pro prvky IoT a další síť pro zařízení jako počítače, tiskárny nebo telefony) je Wi-Fi celkem bezpečnou a nejjednodušší cestou propojení aktivních prvků chytré domácnosti. K takové komunikaci se nejlépe hodí právě NodeMcu, který podporuje standardy 802.11b/g/n a zabezpečení WPA/WPA2. Velkou výhodou je, že NodeMcu podporuje mód AP. Díky tomuto módu bude velmi snadné prvotní nastavení aktivního prvku. Pokud uživatel bude chtít připojit nové zařízení do sítě, NodeMcu založí AP, na které se uživatel připojí a pomocí speciálně určeného programu nastaví NodeMcu tak, aby se mohl připojit k domácí síti. O aktualizace aktivního prvku je postaráno díky OTA, který přes síť nahraje nový firmware, aniž by uživatel musel zapojovat prvek do notebooku a obtížně instalovat nový firmware do daného zařízení. Pokud nebude vyhovovat připojení přes Wi-Fi, je možnost připojení přes jakoukoliv jinou technologii jako např. ZigBee, SigFox a jiné (opět doinstalováním zásuvného modulu do řídicího systému).



Na obrázku 5 je zobrazena varianta s NodeMcu, který se k Wi-Fi síti připojuje pomocí čipu ESP. Touto variantou lze docílit velice rychlého připojení všech prvků do již existující sítě. NAS server je připojený k routeru. Na obrázku je také příklad zapojení IP kamery, která je připojena k NAS serveru pomocí Wi-Fi a nebo pomocí síťového kabelu.

Obrázek 5- NodeMcu ESP varianta



Na dalším obrázku je zobrazena varianta pomocí ZigBee. Pro takové připojení je ale potřeba přidat „prostředníka“, který bude komunikovat se ZigBee a předávat data řídicímu systému ve správném formátování. V uvedeném příkladu je za prostředníka vybráno RPi, které umí komunikovat v počítačové síti a je schopno se ZigBee komunikovat pomocí Rx a Tx pinů.

Obrázek 6 - ZigBee varianta

Aktivní prvky lze samozřejmě propojit i jinými technologiemi. Je ovšem potřeba, aby příslušný modul dodržel pravidla pro komunikaci s řídicím systémem (viz příloha 1 tabulka 3). Komunikace probíhá pomocí předdefinovaných příkazů (viz kapitola 3. Realizace klíčových prvků). Další důležitou věcí u aktivních prvků je záložní napájení (hlavně u zabezpečovacích). Každý prvek by měl být tedy vybaven baterií, která udrží zařízení alespoň dvě hodiny v provozu. Je mnoho způsobů jak toho dosáhnout. V této práci bylo vybráno 5V UPS s vestavěným step-up a nabíjecím modulem (8650 battery step up board UPS 3.7V Boost DC 5V). Díky této UPS lze lehce vytvořit záložní napájecí systém pro každé zařízení. Dále aktivní prvky mohou obsahovat signalizační LED diody, různé senzory (tepelné, pohybové, světelné), reproduktory a mnohé další.

2.4.3. Ovládací panel a jiné ovládací prvky

Ovládací panel, díky kterému bude mít uživatel možnost ovládat celý systém, je připojený pomocí Wi-Fi sítě a je v přímém spojení s řídicím systémem, který může ovládací panel ovládat pomocí předem definovaných příkazů. Realizace ovládacího panelu může být různá. Díky modulům lze vytvořit jakýkoliv ovládací panel pomocí jakéhokoliv zařízení (Raspberry Pi, Arduino a jiné). Je ovšem potřeba opět zachovat formát předávaných dat řídicímu systému. Díky NAS serveru lze snadno založit webový server, na kterém mohou být spuštěny webové stránky, díky kterým může uživatel ovládat domácnost ze svého počítače, notebooku, telefonu, tabletu nebo jiného chytrého zařízení. Další možností jak ovládat domácnost, může být vytvoření speciální aplikace přímo pro Android, iOS, Windows nebo jiný operační systém.

V této práci bude ovládací panel realizován pomocí Raspberry Pi 3 B+ a dotykového 7" displeje od firmy Waveshare. Dále bude ovládací panel vybaven LED diodami, reproduktorem, baterií a dalšími komponenty (více v kapitole 3. Realizace klíčových prvků).

2.4.4. Řídicí systém a ovládací panel

Jak již bylo výše zmíněno, NAS server podporuje spuštění aplikací, které jsou napsané v jazyce Java. Všechen kód pro řídicí systém i ovládací panel bude proto psán v tomto jazyce. Výhodou je, že pokud NAS server nebude v budoucnu vyhovující, lze snadno řídicí systém převést na jakýkoliv jiný systém podporující Javu. Veškerá data budou ukládána do databáze MariaDB, která je velice podobná MySQL a je plně podporována Synology NAS serverem.

Řídicí systém (dále jen ŘS) bude spuštěn jako proces na pozadí na NAS serveru. Bude ovládán vzdáleně pomocí příkazů, anebo pomocí konzole přes samotný NAS server. Stejně tak i moduly na serverové části budou ovládány pomocí příkazů. ŘS nebude řešit žádnou manipulaci s GUI nahraného modulu. GUI ovšem bude řešit software ovládacího panelu (dále jen OP). Samotný SW OP se bude starat např. o přihlášení uživatelů, zobrazení povolených modulů v dané místnosti, zobrazení oznámení a další.

Nejdůležitější částí jsou ovšem moduly. ŘS systém sám o sobě nebude vědět jak ovládat různé senzory, kdy rozsvítit světla, kdy dát uživateli zprávu o tom, že se stala nějaká podmíněná akce (výjimkou jsou integrované zásuvné moduly, které jsou součástí ŘS). To vše budou zajišťovat moduly, které poté pošlou příkaz ŘS, aby např. byl zapnut celo-systémový poplach, nebo aby se na všech panelech ukázalo poplašné, popřípadě upozorňující oznámení (všechny nejdůležitější příkazy jsou vypsány v příloze 1 až 4).

2.4.5. Zásuvné moduly

- **Princip zásuvných modulů**

Zásuvný modul je rozdělen do dvou částí. První část je serverová (která řeší co se má stát, když se stane podmíněná událost). Druhá část je uživatelská (volitelná), tedy to co může vidět nebo ovládat uživatel na svém ovládacím panelu (dá se říct, že je to klient-server v klient-serveru). Komunikace mezi panelem a serverem je opět realizována pomocí domácí sítě (buď WIFI nebo kabelové).

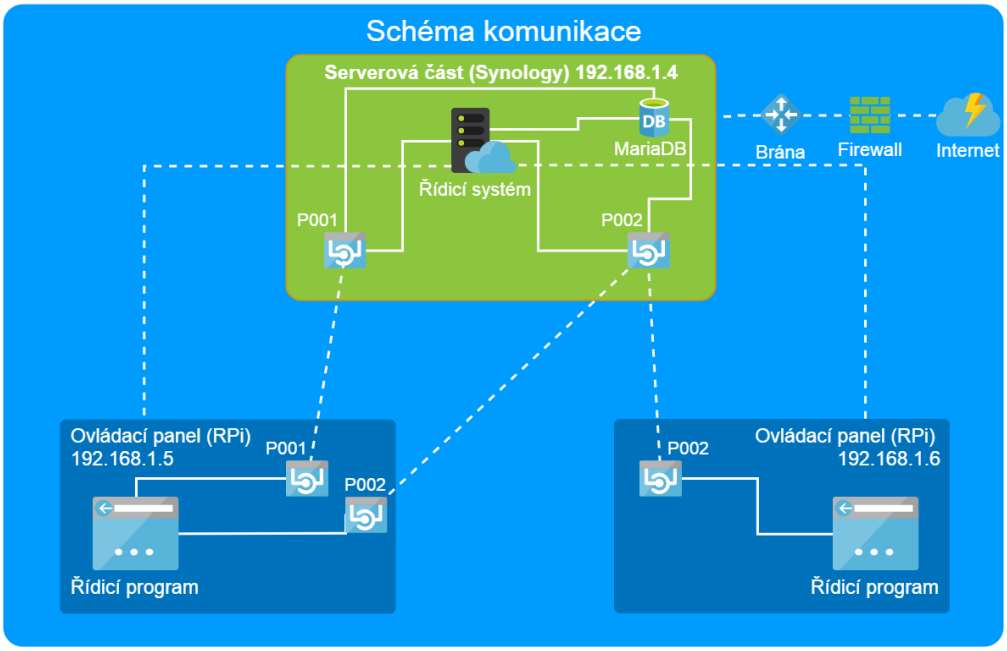
Dejme tomu, že ŘS má IP adresu 192.168.2.4. Jeden z ovládacích panelů má adresu 192.168.2.5. Po spuštění modulu je zapotřebí zajistit komunikaci mezi serverovou a uživatelskou částí. Ovšem takových modulů tam mohou být desítky. Komunikace je tedy rozdělená pomocí různých portů, kdy každý modul musí mít svůj vlastní komunikační port. Jak bylo zmíněno výše, uživatelská část je volitelná. Lze tedy

naprogramovat modul, který bude fungovat zcela automaticky bez zásahu uživatele (samozřejmě pomocí serveru lze modul ovládat - záleží na programátorovi).

Z toho vyplývá, že různé moduly lze naprogramovat tak, že mohou mezi sebou komunikovat. Pouze v konfiguraci modulu se nastaví fixní IP adresa a port. Možností kombinace je několik.

Dejme tomu, že byl na ŘS přidán modul „světla“. Tento modul bude mít své jedinečné označení (P001) z důvodu načítání modulů ze složky, ukončování modulů podle jména a hlavně proto, aby byl modul přístupný jen v povolených místnostech (respektive ovládacích panelech podle IP adresy). Tento modul bude jednoduše vypínat a zapínat světla, která byla přidělena danému modulu pomocí IP adres. Při spuštění modulu je zajištěno, pomocí ŘS, předání volného portu, po kterém bude modul s ŘS komunikovat (serverová část modulu s ŘS). Serverová část modulu komunikuje vždy s ŘS pouze v lokální síti (tedy pouze na adrese 127.0.0.1). Komunikační port mezi serverovým a uživatelským modulem se musí nastavit fixně již při vývoji modulu.

Nyní tedy máme spuštěnou serverovou část modulu, která komunikuje s ŘS na adrese 127.0.0.1:5001 a s uživatelským modulem na adrese 192.168.2.5:6001. Tímto způsobem je zajištěna komunikace všemi směry. Problém nastává v okamžiku, kdy uživatel chce mít stejný modul i v jiné místnosti a chce jím ovládat jiná světla. Řešení je takové, že se vezme ten samý modul s jiným jedinečným označením (např. P002). Tímto se modul chová jako naprosto nový modul a lze ho přiřadit do jiné místnosti s rozdílným nastavením. Samozřejmě lze zobrazit jeden a ten samý modul na více ovládacích panelech (viz obrázek 7). ŘS řeší za moduly jedinou věc a to tu, na kterých panelech se budou spouštět (respektive zobrazovat). Připojení se k ŘS (respektive k serverové části modulu) si musí uživatelský modul řešit sám. Uživatel může nastavit IP adresu a port ručně, nebo port bude fixní a IP adresa se bude načítat automaticky, jelikož modul se může dotázat OP na IP adresu ŘS.



Obrázek 7- Schéma komunikace zásuvných modulů

- **Integrované zásuvné moduly**

Server také obsahuje integrované moduly, které nelze odstranit. Je to např. modul bezpečnosti, kam se přidávají bezpečnostní prvky objektu (budovy, místnosti, skladu), modul pro zobrazení bezpečnostních kamer, kalendář, plánovač úloh, automatizace (např. automatické zakódování systému), atd.

- **Nahrávání zásuvných modulů**

Nahrávání modulů pomocí programu je povoleno pouze uživateli, který má práva administrátora. Při nahrávání modulů na server se také zvolí, na jaké klienty (ovládací panel) se modul nahraje (lze v budoucnu změnit - nahrávat na další panely nebo naopak odstranit modul z určitých panelů). Princip je takový, že nový modul se nahraje do složky „plugins“, příkazem se poté zvolí to, aby byl modul vždy nahrán při zapnutí ŘS. Dále se nastaví, na jaké ovládací panely se má modul nahrát, a kde všude se má zobrazit (modul může být nahrán na všech ovládacích panelech, ale může se zvolit, jestli se na daném panelu spustí/zobrazí).

Modulu je také zapotřebí zajistit nějaký úložný prostor, kam si bude ukládat svá data. Jak již bylo zmíněno, ŘS si všechna data ukládá do databáze MariaDB, moduly nebudou výjimkou a data by se měla ukládat právě a jen do této databáze. Je zapotřebí, aby každý modul měl ve své kořenové složce create script, ve kterém budou příkazy na vytvoření požadovaných tabulek. Při prvním spuštění si ŘS create script zkopíruje a vytvoří tabulky. Poté bude create script vymazán z bezpečnostních důvodů. Pojmenovávání tabulek je ve formátu {jedinečnéOzančeníPluginu}_{názevTabulky}. Pokud dva moduly budou chtít využívat stejné tabulky, je potřeba toto ošetřit přímo ve zdrojovém kódu modulu.

2.4.6. Řízení aktivních prvků

Pro aktivní prvky neexistuje žádné pravidlo, které by určovalo, jaký programovací jazyk se má používat. Formát dat, která se posílají ŘS (respektive modulu) také nelze jednoznačně určit, protože každému modulu se může přiřadit aktivní prvek s jedinečnými vlastnostmi. Výjimkou jsou integrované moduly, které se již řídí pravidly, která určují formát posílání dat. Od ŘS k aktivnímu prvku a od aktivního prvku k ŘS. Např. bezpečnostní senzory posílají svá data ve formátu [BEZ-{stavSenzoru}-{celkovýStav}-{stavBaterie}]. Data pak mohou vypadat takto „BEZ-0-0-100“, což znamená, že není zaznamenán žádný pohyb, nebyla nalezena žádná chyba a baterie je nabita na 100 %.

3. Realizace klíčových prvků

V této kapitole budou realizovány tři klíčové prvky podle výše uvedeného návrhu platformy. Jedná se o bezpečnostní čidlo (pohybové), zařízení ovládající LED pásy a ovládací panel, díky kterému může uživatel pohodlně ovládat celý systém.

3.1. Bezpečnostní čidlo

- **Hardwarová část**

Důležitou částí každého čidla je komunikační modul ESP8266, který dokáže čidlo připojit do již existující Wi-Fi sítě. K realizaci bezpečnostního čidla je zapotřebí dalších komponentů, kterými jsou pohybový senzor, bzučák, LED dioda, baterie, 5V UPS, napájení přes microUSB, ON/OFF přepínač a reset tlačítko.

K bezpečnostnímu čidlu byla vytvořena speciálně upravená krabička, do které lze snadno výše uvedené komponenty uchytit a krabičku umístit buď na zeď, nebo jednoduše položit na jakýkoliv nábytek. Krabička byla navrhována s ohledem na to, aby byla esteticky neutrální a aby zabírala co nejméně prostoru (3D návrhy krabičky lze nalézt na přiloženém CD ve složce „Bezpečnostní čidlo“).

- **Softwarová část**

Po zmáčknutí tlačítka RESET se čidlo přepne do režimu AP (LED dioda dvakrát pomalu zabliká a poté zhasne) a vytvoří se přístupový bod, ke kterému se mohou připojit např. notebooky, telefony a další zařízení. Zapne se také webový server, na kterém lze nastavit připojení do již existující sítě (zadání SSID, přístupového hesla a popřípadě zadání statické IP adresy). Číslo komunikačního portu je vždy 6478. Po zadání přístupových údajů se začne ESP8266 připojovat do sítě s tím, že LED dioda rychle bliká. Po připojení do sítě LED dioda svítí bez blikání a webový server je vypnutý.

Díky příkazům lze čidlo nastavovat ,nebo si vyžádat požadovaná data. Pokud ŘS pošle např. příkaz „GET STATUS“, tak čidlo pošle svůj aktuální stav ve formátu [BEZ-{stavSenzoru}-{celkovýStav}-{stavBaterie}]. Řetězec „BEZ“ značí, že čidlo patří do skupiny bezpečnostních aktivních prvků. Dále „stavSenzoru“ označuje, jestli je stav daného pinu HIGH nebo LOW. Pokud není zaznamenán žádný pohyb (PIN D3 je HIGH), je posláno „BEZ-…-1-…“. Řetězec „celkový_stav“ obsahuje informace o tom, zda je zapnuta LED dioda a bzučák. Řetězec může mít formát např. „BEZ-10-1-…“ (LED dioda je zapnutá a bzučák vypnutý). Jako poslední řetězec v odpovědi je uveden stav baterie v procentech. Výsledná odpověď může mít tvar „BEZ-10-1-95“.

Příkazů pro ovládání senzoru je celkem 7. V následující tabulce jsou vypsány všechny příkazy, kterými lze čidlo nastavovat nebo ovládat, odpověď na daný příkaz a popis příkazu.

Tabulka 2 - Seznam příkazů bezpečnostního čidla

Příkaz	Odpověď	Popis příkazu
GET STATUS	**STATUS** 'BEZ-{stav_senzoru}-{celkový_stav}-{stav_baterie}'	Čidlo pošle svůj aktuální stav včetně stavu senzorů.
GET LED	**LED** 'BEZ-{stav_LED}'	Vyžádání stavu LED diody (svítí/nesvítí).
GET BUZ	**BUZ** 'BEZ-{stav_bzucaku}'	Vyžádání stavu bzučáku.
SET LED [HIGH | LOW]		Zapnutí /vypnutí LED diody.
SET BUZ [HIGH | LOW]		Zapnutí/Vypnutí bzučáku.
RESTART		Restartování čidla. Přepnutí do AP modu a zapnutí web serveru.

Po úspěšném připojení do sítě ESP8266 vytvoří server, na který se může plugin (klient) připojit a poslat příkaz. Po odeslání odpovědi NodeMcu klienta odpojí, aby bylo možné zpracovat další požadavky od ostatních klientů. Pokud nastane situace, že se klient připojí, ale neodešle žádný příkaz, dojde k automatickému odpojení po 2 sekundách nečinnosti.

3.2. LED pás

- **Hardwarová část**

Jako další zařízení pro demonstraci funkčnosti systému byl vybrán LED pás, který slouží k příjemnému osvětlení místností nebo různých předmětů (např. obrazů, nábytku). LED pás je vybaven NodeMcu ESP8266, programovatelným RGB LED páskem, světelným senzorem, ON/OFF přepínačem a reset tlačítkem. V tomto případě je LED pásek možno ovládat pomocí jednoho digitálního pinu. Pokud by byl zapojen jiný LED pásek, který by měl RGB analogové vstupy, muselo by se připojit rozšíření analogových pinů pro ESP8266. Pokud by takové řešení nebylo dostačující, může se k modulu ESP8266 přidat mikrokontrolér nebo jiné zařízení, které by obsluhovalo daný LED pásek. Kontrolér by pak mohl komunikovat s ESP pomocí sériové linky.

LED pás je vybaven světelným senzorem, který slouží pro měření intenzity světla v dané místnosti. Když je v místnosti dostatečné světlo, je LED pásek vypnutý i za předpokladu, že byl poslán LED pásu příkaz pro zapnutí LED pásku (tuto funkci lze pomocí příkazu vypnout). Krabička pro LED pás byla opět vytvářena s ohledem na co nejmenší rozměry. V této krabičce se ovšem vyskytuje pouze ovládací zařízení LED

pásku. Samotný LED pásek je umístěn v držáku, který je speciálně určený pro LED pásek (3D návrhy krabičky lze nalézt na přiloženém CD ve složce „LED pás“).

- **Softwarová část**

Připojení LED pásu do sítě je realizováno stejným způsobem jako u bezpečnostního čidla. Kontaktování čidla se ale už poněkud liší. Pokud modul posílá LED pásu příkaz „GET STATUS“ je k tomuto příkazu ještě přiložena informace o stavu LED pásku (zapnutý/vypnutý) a o barvě (např. **GET STATUS –S '1;#4d66cc'**).

V tabulce 3 lze nalézt základní příkazy pro ovládání LED pásku.

Tabulka 3 - Seznam příkazů LED pásu

Příkaz	Odpověď	Popis příkazu
GET STATUS –S '{stav};{barva}'	**STATUS** 'LED-{stav_pásku}-{barva}'	LED pás pošle nastavenou barvu a stav LED pásku
GET LIGHT	**LIGHT –L** '{intenzita_světla}'	Poslání naměřené hodnoty světelným senzorem.
SET LIGHT [ON | OFF]		Zapnutí/vypnutí automatického řízení LED pásku podle intenzity světla
RESTART		Restartování čidla. Přepnutí do AP modu a zapnutí web serveru.

3.3. Ovládací panel

Ovládací panel je vybaven Raspberry Pi 3 model B+, 7" dotykovým displejem, bzučákem, reproduktorem + zesilovačem (12 V), modulem step-up (5 V na 12 V), baterií, UPS modulem RPi UPSPack Standard, ON/OFF přepínačem.

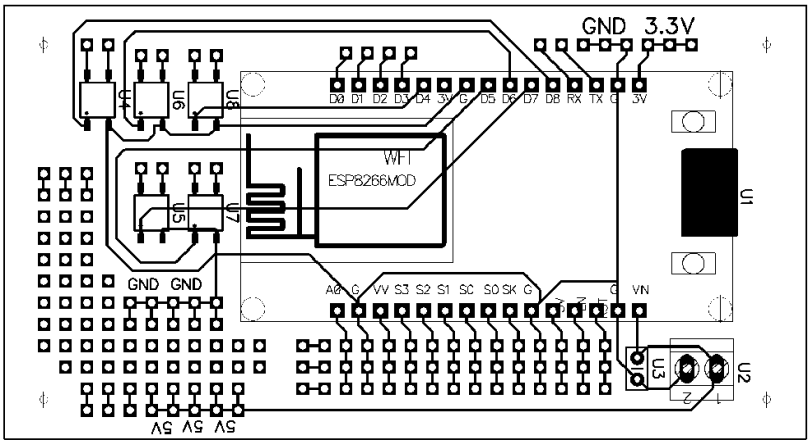
Komunikaci s řídicím systémem lze provozovat buď pomocí bezdrátové sítě Wi-Fi nebo pomocí kabelového připojení přes Ethernet rozhraní, kterým disponuje RPi. Při výpadku elektrického proudu je připravena baterie s kapacitou 2200 mAh, která se postará o dodávku energie v průměru na 5 hodin při vypnutém displeji (při odběru 300 mA). Se zapnutým displejem vydrží panel v průměru 3,5 hodiny (celkový odběr 500 mA).

O upozornění obyvatel na událost nebo nebezpečí se postará digitální, 2kanálový, 15W zesilovač, díky kterému může ovládací panel přehrávat různé zvukové nebo hlasové nahrávky. Aby bylo používání ovládacího panelu co nejjednodušší, byla navržena speciální krabička pro uchycení dotykového displeje a dalších potřebných komponentů. Ovládací panel lze tak vzít pohodlně do ruky nebo přichytit na zeď (3D návrhy lze nalézt na přiloženém CD ve složce „Ovládací panel“).

Veškeré další informace o ovládacím panelu lze nalézt v kapitole 4.2. Ovládací panel a zásuvné moduly.

3.4. Univerzální plošný spoj

Vzhledem k tomu, že skoro všechny aktivní prvky mají velice podobné technické vybavení, byl navržen univerzální plošný spoj, ke kterému se dá jednoduše připojit kterýkoliv senzor nebo LED dioda. Spoj je také vybaven mnoha piny pro připojení napájení nebo dalších potřebných součástek. Dále spoj umožňuje připojení celkem pěti relé (SMD). Relé je pak možno spínat pomocí pinů D4 až D8.



Obrázek 8 - Plošný spoj

4. Řídicí Software

Nedílnou součástí každého systému je software, který se stará o správné fungování celého systému. V této kapitole bude podrobně rozebrán software řídicího systému a ovládacího panelu včetně podrobnějšího popisu funkcí integrovaného modulu bezpečnosti (IntBez01) a komfortního, externího modulu LED pásů (ExtLED01).

Všechen software je psán v programovacím jazyce Java. Pro ukládání veškerých dat je využívána databáze MariaDB. Projekt využívá pouze standardní knihovny jazyku Java, aby byl systém plně multiplatformní. Grafické uživatelské rozhraní je realizováno pomocí knihovny JavaFX.

4.1. Řídicí systém a zásuvné moduly

Software řídicího systému se dělí na dvě části. První část, nazývána jako jádro systému, se stará o základní věci jako načtení všech zásuvných modulů, vytvoření serverů pro připojení klientů (ovládacích panelů), veškerou práci se sítí a databází. Druhá část, nazývána příkazová část systému, se stará o zpracovávání příkazů od uživatele, zásuvných modulů a ovládacích panelů. Jelikož příkazy jsou posílány přes počítačovou síť, jsou nejprve zpracovány jádrem, které pošle daný příkaz příkazové části systému. Ta určí, o jaký příkaz se jedná, provede požadované úkony a pokud je potřeba, odešle odpověď (opět přes jádro).

Obrázek 9 - Software - schéma

Na obrázku 9 je zobrazeno základní schéma komunikace mezi jádrem, moduly a příkazovou částí. Příkazová část také obsahuje mnoho funkcí, které se po identifikování daného příkazu provedou. Příkladem může být příkaz „**GET DBS URL**“, který plugin pošle jádru hned po svém zapnutí. Tímto příkazem plugin žádá o přístupové údaje k databázi.

Jádro příkaz předá příkazové části, ta zjistí, o jaký druh příkazu se jedná a co má provést. Jelikož je příkazová část v přímém spojení s jádrem, zavolá funkci jádra, která předá příkazové části požadované údaje. Ta pak pomocí příkazu „**DBS DRIVER –S** '{údaje}'“ pošle přístupové údaje modulu, který o údaje požádal. Někdy je ovšem potřeba poslat modulu příkaz (např. přidání senzoru). To znamená, že příkazová část nemůže poslat čistý příkaz ve formě „**ADD SENSOR –N** 'Sensor1'“. Příkaz je potřeba zabalit do příkazu „**SENDP –P** '{název_pluginu}' **–C** '{příkaz}'”. Tento příkaz může zadat sám uživatel přes konzoli

nebo může být poslán modulu pomocí ovládacího panelu (klientská část modulu). Příkaz je poslán přes jádro příkazové části. Ta identifikuje, že se jedná o příkaz, který obsahuje další příkaz (**-C**) a tento příkaz je určen pro jeden ze spuštěných modulů. Nyní je ale potřeba určit, jak přesně modulu tento příkaz předat. V části příkazu „**-C**“ se vyskytuje pouze název modulu. Příkazová část tedy nemůže modulu samostatně poslat zprávu, protože neví, jak se s daným modulem spojit. Jediný, kdo ví, jak se s modulem spojit, je samotné jádro. Na obrázku 9 lze vidět, že jádro se skládá ze tří důležitých částí. Jednou z částí je „komunikace s moduly“. Tato část obstarává veškerou komunikaci se spuštěnými moduly (přijímá data od modulů a předává je příkazové části, odesílá data modulům a kontroluje, zda moduly jsou stále aktivní a je možné s nimi komunikovat). Jelikož část „Komunikace s moduly“ obsahuje veškeré informace o modulech, tak nejjednodušší cestou je poslat příkaz této části a ta se již sama postará o poslání daného příkazu ke správnému modulu. Této části jádra je tedy předána část příkazu „**-P**“, která obsahuje název modulu a část „**-C**“, která obsahuje příkaz pro daný modul.

Výše uvedený postup se ale o trochu změní, když ten samý příkaz pošle ovládací panel (uživatel přes uživatelské rozhraní přidá senzor a uživatelská část modulu pošle příkaz řídicímu systému). Jelikož každý modul má spuštěný svůj vlastní server, na který se mohou klienti (uživatelská část modulu, spuštěna na ovládacích panelech) připojit a poslat příkazy, není tedy potřeba nic předávat přes jádro řídicího systému. Každý modul pak obsahuje svou vlastní příkazovou část, která vykoná požadované operace. Po vykonání operací je modul povinen odeslat odpověď o výsledku a informovat tak uživatele přes uživatelské rozhraní o výsledku. Klienta pak odpojí a čeká na další připojení. Problém nastává v případě, kdy spuštěný modul (tedy serverová část modulu) usoudí, že je potřeba informovat uživatele o nějakém nebezpečí nebo informovat o nějaké události. Serverová část modulu sama o sobě neví, kolik je spuštěných klientů (uživatelská část modulu) a na jakých ovládacích panelech je modul povolen. Všechny tyto informace má uložené jádro, které přesně ví, jaké moduly jsou spuštěny a na jakých ovládacích panelech jsou povoleny. Modul tedy jádru pošle příkaz „**SEND TO CLIENT –P '**{název_pluginu}**' –C '{**název_panelu}**' –M '**{příkaz}**'**“. Příkaz je opět zpracován příkazovou částí a jádru je poslána informace, že má poslat všem (**-C '-'**) klientům (tedy ovládacím panelům) na kterých je spuštěn modul „**-P**“ zprávu „**-M**“. V části příkazu „**-C**“ lez specifikovat, na jaké klienty (ovládací panely) se má příkaz odeslat.

Tento způsob komunikace se ovšem doporučuje pouze integrovaným modulům. Příkladem integrovaného modulu je bezpečnostní systém (IntBez01). Externí moduly by v žádném případě neměly používat jako komunikační tunel jádro řídicího systému. Externí moduly mohou využívat příkaz „**SEND TO CLIENT**“ pouze v případě, kdy je potřeba zapnout globální alarm (**SEND TO CLIENT –P '**IntBez01**' –C '-' –M 'GLOBAL ALARM ON'**). Globální alarm se vždy přepošle modulu „IntBez01“. Pokud by ovšem bylo potřeba poslat zprávu externímu modulu (uživatelské části) přes jádro, lze použít příkaz „**SEND TO CLIENT**“, ovšem již trochu komplikovanějším způsobem. V části „**–M**“ se musí zadat informace, že je potřeba příkaz poslat určitém modulu, který je spuštěn na ovládacím panelu. Příkaz by mohl vypadat např. takto:

SEND TO CLIENT –P 'P001**' –C '-' –M 'SENDP –P “**P001**“ –C “**SHUT DOWN**“'**

Vývoj externích modulů je velice flexibilní a tak je čistě na programátorovi, jaký způsob komunikace mezi serverovou a klientskou částí zvolí. Při vývoji externích modulů je ale nutno myslet na výkon jádra. Čím více budou externí moduly používat funkce jádra, tím je větší pravděpodobnost, že jádro nebude mít dostatek času vyřizovat požadavky např. od bezpečnostního modulu, který se stará o veškerou bezpečnost v budově. Rychlost vyřizování požadavků je ovšem velice závislá na výkonu zařízení, na kterém je řídicí systém spuštěn. Pokud se jedná o slabší zařízení jako např. Raspberry Pi, není doporučeno spouštět příliš mnoho modulů a využívat jádro jako komunikační tunel. Pokud bude systém spuštěn na výkonnějším počítači, je možno jádro jako komunikační tunel používat.

4.1.1. Integrovaný modul – IntBEZ01

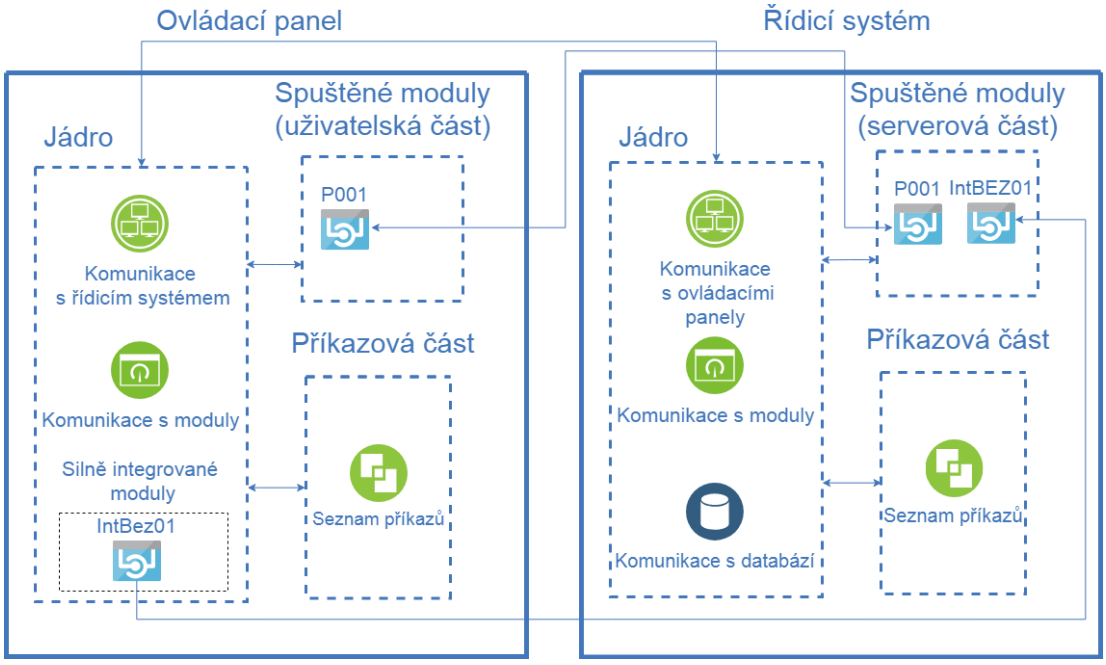
Integrovaný bezpečností modul, pod názvem IntBEZ01, je jedním z nejdůležitějších modulů v celém systému. Tento modul se stará o bezpečnost a informování obyvatel domu, že nastala nebezpečná situace (např. pohyb v zabezpečené oblasti, detekce plamene, otevření okna a jiné). Na straně řídicího sytému se tento plugin tváří stejně jako jiné externí moduly. Je stejně načítán, platí pro něj stejná pravidla jako pro externí, ale s tím rozdílem, že nelze vypnout (nelze zajistit to, aby jej řídicí systém při spuštění nenačetl).

Na rozdíl od řídicího systému je tento modul silně integrován v ovládacích panelech. Každý panel má podobnou logiku řízení jako řídicí systém. Má své jádro, spuštěné moduly a příkazovou část. IntBEZ01 se ale nevyskytuje ve spuštěných modulech, ale přímo v jádru řídicího programu ovládacího panelu (viz obrázek 10). Je zde ale možnost omezení přístupu k nastavení modulu pro určité ovládací panely. Modul ale stále zůstává aktivní z důvodu globálního alarmu a informování obyvatel o případném nebezpečí. K tomuto modulu mají přístup i ostatní externí nebo interní moduly pomocí předdefinovaných příkazů. Pokud např. pošle externí modul řídicímu systému příkaz „**GLOBAL ALARM ON**“, je tento příkaz automaticky přesměrován právě bezpečnostnímu integrovanému modulu. Ten se poté postará o kontaktování ovládacích panelů a jiných informačních zařízení, která upozorní na nebezpečí (např. sirény).

IntBEZ01 má možnost přidání senzorů, sirén a profilů. Senzory se pak dělí do různých skupin. Např. pohybové čidlo lze nastavit jako příchodové nebo základní. Dále lze nastavit, o jaký druh senzoru se jedná (pohybový senzor, senzor plamene nebo senzor otevření okna). Jako další se nastavuje IP adresa daného zařízení, port, a jestli má být zařízení aktivované nebo deaktivované. Po přidání senzorů a sirén je potřeba vytvořit profil. Ke každému profilu lze pak přidat libovolné množství senzorů a sirén. Při aktivaci profilu (lze jich aktivovat více najednou) jsou pak kontrolovány určité senzory, které byly přidány do daného profilu. Pokud např. senzor zaznamená pohyb, zapnou se pouze ty sirény, které jsou přidané do daného profilu. Senzory plamene jsou kontrolovány permanentně. Modul také disponuje funkcí, která kontroluje, zda jsou všechna zařízení připojená k síti a lze je kontaktovat. V případě, že s jedním nebo více zařízeními ztratí kontakt, je ihned aktivován globální poplach a zobrazeno oznámení

na všech ovládacích panelech. Při obnově kontaktu je zobrazeno oznámení, které informuje uživatele o stavu zařízení.

Aktivace, deaktivace profilů nebo vypnutí globálního alarmu se provádí zadáním bezpečnostního pinu pomocí ovládacího panelu nebo příkazové řádky.



Obrázek 10 - Software - schéma komunikace

Na obrázku 10 je zobrazena komunikace mezi ovládacím panelem a řídicím systémem. Jak lze vidět, integrovaný modul IntBEZ01 je součástí jádra na straně ovládacího panelu, zatímco u řídicího systému se chová stejně jako externí modul. Další integrované moduly (např. kalendář, časovač nebo jiný program usnadňující práci se systémem) lze na straně ovládacího panelu spustit v části „Spuštěné moduly“. Tím se pak nemusí provádět žádná změna v hlavním programu ovládacího panelu. Pokud je přidán nový silně integrovaný modul, je potřeba přehrát celý software na straně ovládacího panelu.

Výhodou silně integrovaných modulů je jednodušší implementace a tím i větší spolehlivost funkčnosti modulu. Takové moduly jsou také mnohem míň náročné na výkon zařízení (předpokládá se, že ovládací panel bude zařízení s nižším výkonem jako např. RPi.). Nevýhodou je již výše zmíněný problém s přehráním celého softwaru při přidání nového modulu.

4.1.2. Externí modul – ExtLED01

Modul ExtLED01 je příkladem externího modulu. Tento modul je tedy již spuštěn pouze v části spuštěných modulů, jak na straně řídicího systému, tak na straně ovládacího panelu (viz obrázek 10 – modul s názvem P001). Tento modul se zabývá především výší komfortu v nastavených místnostech. Jedná se o zapnutí LED pásů, když jsou obyvatelé v místnosti, zpříjemnění osvětlení v nočních hodinách, zapnutí LED pásů z estetických důvodů nebo vypnutí pásů, když se v místnosti nikdo nevyskytuje. Příkladem může být potřeba jednoho z obyvatel jít se v noci napít. Modul díky pohybovým senzorům naprosto přesně ví, kde se obyvatel nachází a určí tak, jaké LED pásy se mají rozsvítit. Jelikož v tomto případě se jedná o noční hodiny, modul LED pásy nastaví na příjemné, tlumené světlo a po odchodu obyvatele opět LED pásy modul vypne (více o nastavení automatizovaných akcí v kapitole 5. Automatizace).

Jelikož tento modul nepoužívá jádro jako komunikační tunel, je obtížné informovat uživatelské části modulu o změně nastavení (nastavení se může měnit na více ovládacích panelech, ale je potřeba o této informaci informovat i ostatní ovládací panely, respektive uživatelské části modulu). Tato problematika je řešena tak, že je vytvořený server na straně řídicího sytému, kam se uživatelské části modulu periodicky připojují a zjišťují nastavení, které poté, pokud je nutno, zobrazí.

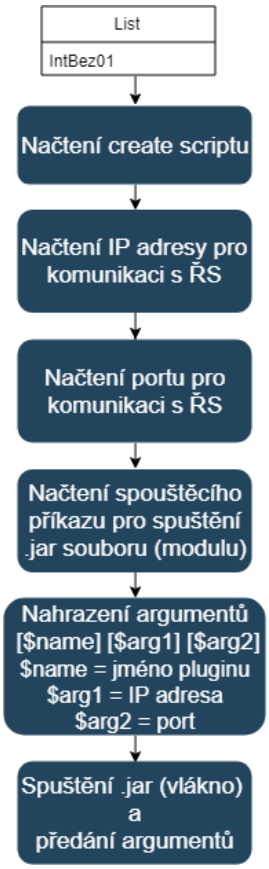
Pokud se přidává nový externí modul (nebo i integrovaný), musí se vyřešit nastavení komunikačního portu, přes který bude komunikovat uživatelská a serverová část modulu. O tuto problematiku se již řídicí systém nestará a je tak čistě na programátorovi, zda port nastaví fixně nebo nechá uživatele port nastavit (řídicí systém se stará pouze o přiřazení komunikačního portu pro komunikaci mezi jádrem a částí spuštěných modulů). IP adresu řídicího systému si může externí modul vyžádat přímo od jádra ovládacího panelu.

Předpokládejme, že řídicí systém je spuštěn na Synology pod IP adresou 192.168.1.10 a ovládací panel má přiřazenou IP adresu 192.168.1.11. Programátor zvolil možnost fixního přiřazení portu a tak externí modul na straně řídicího systému vytvoří server na adrese 192.168.1.10:5555. Uživatelská část modulu se tedy vždy připojí právě na tuto adresu (IP adresu řídicího systému si vyžádá od jádra ovládacího panelu) s fixním portem (nastaveno programátorem) a odešle příkaz, na který serverová část odpoví a čeká na připojení dalšího modulu.

4.1.3. Načítání externích/integrovaných modulů a připojení klientů

Veškeré informace o tom, jaké se mají načíst moduly nebo pro které klienty vytvořit server, jsou uloženy v databázi. Prvním krokem po spuštění systému je tedy vytvoření aktivního připojení s databází, ze které si řídicí systém načte požadované nastavení. Po úspěšném načtení jsou nejdříve načteny integrované moduly a poté jsou načteny externí moduly. Po načtení všech modulů přichází na řadu vytvoření serverů pro

klienty (ovládací panely). Pro každého klienta je vytvořen právě jeden server a klient udržuje aktivní spojení s řídicím systémem po celou dobu spuštění.



Obrázek 11- Software - spuštění

Načtení modulu je ale o něco málo složitější než načtení klientů (pro ty se pouze vytvoří server). Modulům je nutno předat více informací. Na obrázku 11 je znázorněný postup spouštění modulu (celý vývojový diagram lze nalézt na přiloženém CD). Nejprve je prověřeno, zda se ve složce vyskytuje create script. Pokud ano, řídicí systém screate script načte a pošle příkazy databázi. Tato funkce je především potřeba v případě, kdy je modul spuštěn poprvé a je nutné pro něj vytvořit databázové relace, kam si bude ukládat svá data. Create script se po načtení smaže a pokračuje se na další krok. Řídicí systém určí IP adresu a port, přes který bude s modulem komunikovat. IP adresa je většinou přidělena lokální (tedy 127.0.0.1), port se vždy přidělí dynamicky (většinou je zvolen první volný port).

Jelikož každý modul je samostatný spustitelný soubor uložený ve složce „Plugins/{název_modulu}", musí se tento soubor korektně spustit v rámci systému, na kterém je řídicí systém spuštěn. Řídicímu systém je potřeba sdělit, jak tento spustitelný soubor spustit. Proto má každý modul ve své složce speciální soubor, kam se napíše příkaz pro spuštění daného modulu. Jako příklad byl zvolen integrovaný modul IntBEZ01. Pokud se spouští tento modul, řídicí systém načte soubor „IntBEZ01.txt", kde nalezne přesný formát příkazu, díky kterému bude možné modul spustit. Příkaz může mít formát např. „**java -jar plugins\[$name]\[$name].jar [$arg1] [$arg2]"**. Nyní řídicí systém nahradí argumenty za správné hodnoty (viz obrázek 11) a vytvoří vlákno, ve kterém bude spuštěn daný modul. Pokud bude tedy nutno načíst speciální knihovny pro spuštění modulu, stačí poupravit příkaz. Stačí pouze zachovat formát argumentů, jinak je lze libovolně přemisťovat a vytvářet tak vždy jedinečný spouštěcí příkaz.

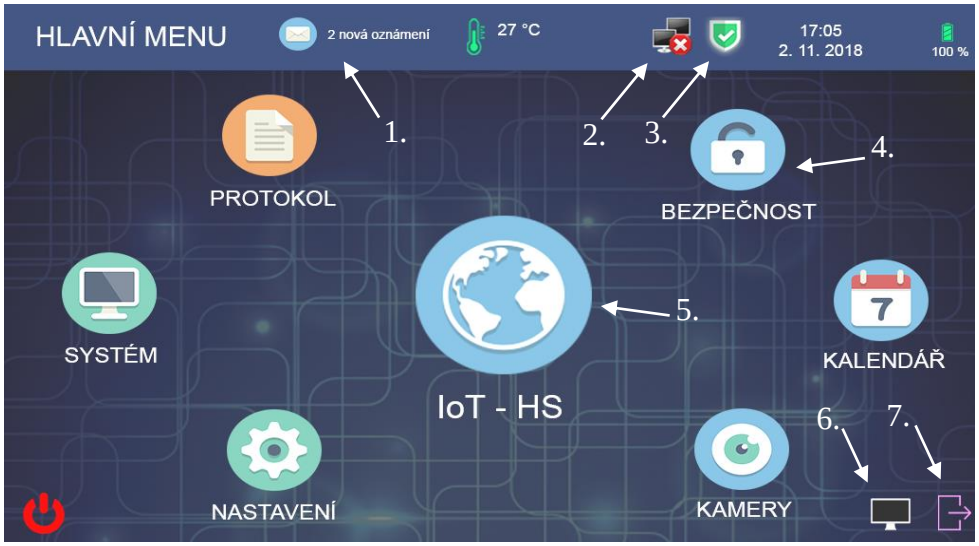
4.2. Ovládací panel a zásuvné moduly

Díky ovládacímu panelu mohou obyvatele domu pohodlně ovládat celý systém, přistupovat k různým modulům, automatizovat různé akce nebo jen sledovat stav senzorů nebo systému samotného. Z obrázku 10 je na první pohled vidět, že ovládací panel a řídicí systém jsou si velice podobné svou funkčností. Jeden z prvních rozdílů je ten, že ovládací panel nepřistupuje vůbec k databázi a tím nemají přístup k databázi ani uživatelské části modulů. Z bezpečnostních důvodů je databáze přístupná jen na lokálním zařízení (je přístupná jen řídicímu systému a serverovým částem modulů). Jediné co ovládací panel potřebuje, je mít uloženou IP adresu řídicího sytému a nastavení (např. za jak dlouho se má vypnout displej, nastavení sítě atd.). Jelikož se

jedná o malé množství dat, která si potřebuje ovládací panel ukládat, jsou všechna data ukládána do konfiguračních souborů. Protokol chyb nebo různé události jsou zpravidla posílány řídicímu systému a ten jej zaeviduje do databáze.

Hlavní výhodou ovládacího panelu je jeho grafické uživatelské rozhraní, které zjednodušuje ovládání nebo nastavení systému. Jelikož na ovládacím panelu lze nastavit i citlivé funkce řídicího sytému, je zde možnost přihlášení administrátora, který má přístup do nastavení systému, nastavení ovládacího panelu nebo zobrazení protokolu událostí. Pokud se administrátor odhlásí, jsou přístupné pouze funkce modulů, bezpečnostního systémů a jiných integrovaných modulů (pokud jsou na daném ovládacím panelu povoleny). Načtení externích nebo integrovaných modulů je stejné jako na straně řídicího sytému.

4.2.1. Grafické uživatelské rozhraní



Obrázek 12 - Software – GUI

Popis částí uživatelského rozhraní

1. Oznámení (zobrazeno na všech ovládacích panelech)
2. Nelze připojit k řídicímu systému
3. Alespoň jeden profil je zapnutý
4. Silně integrovaný modul IntBEZ01
5. Externí moduly
6. Tlačítko pro vypnutí displeje
7. Přihlášení/odhlášení administrátora

Hlavní menu uživatelského rozhraní se dělí na tři části. Jedna z částí je horní informační lišta, kde uživatel může vidět, zda je ovládací panel připojen k řídicímu systému, jestli je aktivován jeden nebo více profilů, stav baterie, teplotu v místnosti a oznámení. Oznámení je speciální funkce řídicího systému, kterou mohou využívat všech moduly. Jedná se o informační systém, díky kterému mohou moduly informovat

uživatele o jakékoliv události. Pokud chce modul poslat oznámení, musí poslat příkaz řídicímu systému ve formátu „**NFC –S '{**předmět**}' –M '{**zpráva**}'**“. Řídicí systém se poté postará o distribuci oznámení na všechny ovládací panely. Je také doplněno datum poslání oznámení, a jaký modul oznámení zaslal. Pokud je potřeba uživatele informovat okamžitě, je zapotřebí zapnout globální alarm (**GLOBAL ALARM ON**) a přeposlat zprávu příkazem „**GNFC –M '{zpráva}'“.** Tímto příkazem se zobrazí zpráva uprostřed obrazovky se zprávou (na všech ovládacích panelech). Pokud ovšem nehrozí žádné nebezpečí a jedná se jen o chybu, např. čidla teploty, posílá se oznámení nebo zobrazení zprávy pomocí příkazu „**GNFC**“. Zde je nutno dodat, že globální alarm může vypnout pouze integrovaný modul IntBEZ01 po zadání pinu, ostatní moduly mohou globální alarm pouze zapnout.

Další část rozhraní (vlevo) je přístupná pouze administrátorovi. Nastavení systému nebo nastavení panelu by při špatném nastavení mohlo mít kritické následky na bezpečnost systému. Např. bezpečnostní modul by nemusel fungovat správně. Protokol je z jisté části přístupný i bez přihlášení administrátora a uživatel si tak může zobrazit základní sdělení systému o chybách.

Poslední část je již přístupná i bez přihlášení a jedná se o silně integrované moduly (vpravo). Každý silně integrovaný modul je zobrazen jako samostatná ikonka v menu (v případě přidání více silně integrovaných modulů a nedostatku místa v hlavním menu je možné seskupovat do složek více modulů). V této poslední části se také vyskytuje „IoT – HS“, kde lze přistupovat k externím modulům. Na obrázku 13 je zobrazeno, jak snadno lze k externím modulům přistupovat.

Obrázek 13 - Software GUI 2 ExtLED01

Popis částí uživatelského rozhraní

1. Zobrazení oblíbených místností
2. Zobrazení všech místností
3. Nastavení oblíbených místností a přiřazení modulu do dané místnosti

Každý externí modul má ve své složce uloženou ikonu a konfigurační soubor, ve kterém je uloženo, pod jakým názvem má být modul zobrazován. Místnost je už určena ovládacím panelem a uživatel si může vybrat z předem definovaných obrázků a zvolit si sám libovolný název. Po vytvoření místností k nim uživatel přiřadí moduly. Kliknutím na ikonku nastavení v oblasti místností se zobrazí okno, kde lze do místnosti přidat moduly. Také je možno v tomto nastavení určit, zda je místnosti „oblíbená“ a po kliknutí na ikonu „oblib.“ se zobrazí jen místnosti vybrané uživatelem.

V nastavení modulů lze nastavit pouze oblíbené moduly. Nastavení modulu samotného řeší právě daný modul. Nastavení ovládacího panelu (vytvořené místnosti a přiřazené moduly různým místnostem) nelze distribuovat do ostatních ovládacích panelů. Je tedy nezbytné vždy v každém panelu vytvořit místnosti samostatně a znovu přidat modul do dané místnosti.

4.2.2. Načítání externích/integrovaných modulů

Při zapnutí se ovládací panel pokouší připojit k řídicímu serveru. Pokud je připojení úspěšné, zjišťuje, jaké moduly jsou na něm povoleny pomocí příkazu „**GET PLUG**“, kterým požádá řídicí systém o zaslání povolených modulů. Ten co nejdříve odpoví příkazem „**START PLUG -S '**{povolené_moduly}**'**“. Spuštění modulů je stejné jako na straně řídicího systému s tím rozdílem, že je načtena ikona a název modulu, pod kterým se má modul reprezentovat.

Dále je po spuštění ovládacího panelu poslán příkaz „**GET GLOBAL ALARM**“ pro zjištění, zda je spuštěn globální alarm a příkaz „**IS PROFILE SECURED**“, kterým ovládací panel zjistí, zda je zapnutý alespoň jeden profil (zobrazení štítu na informativní liště).

4.3. Doplňování systému o další funkce

Doplňování systému o nové funkce je díky zásuvným modulům velmi snadné. Pokud se dodrží jistá pravidla, může si programátor naprogramovat jakékoliv funkce a jednoduše je přidat do systému. Aby celá problematika přidávání zásuvných modulů byla ještě jednodušší, byla navržena šablona pro programování nových zásuvných modulů (serverová část modulu). Tato šablona obsahuje funkce jako vytvoření komunikace s řídicím systémem (balíček ShellCom) a připojení k databázi. Šablona také obsahuje balíček „Command“, kde jsou uloženy základní příkazy, díky kterým lze komunikovat s řídicím systémem. Balíček obsahuje tři třídy. „CommandsFromShell.java“ je třída, kde jsou uloženy již zmíněné příkazy pro komunikaci s řídicím systémem.

Další třída „CommandsFromPluginClientSide.java“ je určena pro komunikaci s klientskou částí modulu. Veškeré příkazy, které klientská část pošle, jsou předány právě této třídě a programátor zde může vytvářet své vlastní funkce k daným příkazům. Poslední třída je „SeperateComm.java“, která se stará o rozdělení části příkazu. Příkladem může být příkaz „**GNFC –M 'zkušební zpráva'“**. Po přijetí tohoto příkazu

je zavolána třída „SeperateComm.java“, které je v konstruktoru předán tento příkaz. Ta se postará o jeho rozdělení a pak pomocí funkce „GetMain()“ lze získat hlavní část příkazu (tedy „**GNFC**“) a funkce „GetSubComm(“-M“)“ vrátí vnořený řetězec v části „**-M**“ („zkušební zpráva“).

Tímto způsobem lze systém doplnit o jakékoliv funkce jako hlasové ovládání, ovládání zařízení od různých výrobců, přidávání jiných komunikačních zařízení nebo protokolů/standardů než je Wi-Fi.

Podobná šablona je připravená také pro klientskou část modulu. Šablony pro vytváření modulů lze nalézt na přiloženém CD.

- **Příklad přidání modulu (serverová část)**

Předpokládejme, že máme již vytvořený projekt s balíčky „Command“ a „ShellCom“ a chceme vytvořit modul, který bude ovládat LED panely (tedy jako ExtLED01). V prvním kroku je zapotřebí vytvořit nový balíček (např. Controller). V tomto balíčku se bude nacházet hlavní třída „Main“ a třída, která ovládá LED pásy „LEDcontroller“. Třída „Main“ je již předpřipravená, není tedy nutné provádět nějaké výrazné změny. V prvním kroku se vytvoří spojení s řídicím systémem „ShellCommunication.ConnectToShell(String argsIP, String argsPort)“. Po úspěšném navázání komunikace je vytvořeno spojení s databází „GetAccessDBS()“. Po těchto dvou krocích již může programátor vytvořit instanci na svou třídu „LEDcontroller“ a psát své vlastní funkce. Je také nezbytné přidat funkce do třídy „CommandsFromHS“, jelikož veškeré příkazy, které řídicí systém odešle, jsou zpracovány právě touto třídou a je potřeba zadat, jaké funkce se mají volat po přijetí určitého příkazu.

5. Automatizace

V minulých kapitolách již bylo naznačeno, jaké automatizované funkce bude mít integrovaný modul IntBEZ01 i externí modul ExtLED01. Pokud se přesuneme k bezpečnosti, bylo by jistě dobré, aby se dům sám zabezpečil, pokud nejsou obyvatelé přítomni delší dobu. Pokud v domě není nikdo přítomen, je jistě zbytečné svítit LED pásy, které mohli obyvatelé ručně zapnout a po odchodu je zapomněli vypnout. V této kapitole budou rozebrány různé způsoby automatizace modulů IntBEZ01 a ExtLED01.

- **Integrovaný zásuvný modul IntBEZ01**

Tento integrovaný modul umožňuje uživateli zvolit, kdy se mají jednotlivé profily sami zabezpečit a popřípadě, kdy se mají vypnout. Tato funkce je obzvláště výhodná v případě, kdy obyvatelé zapomněli zabezpečit dům a nebudou přítomni přes noc. Systém se v požadovaný čas sám zabezpečí a ráno opět (pokud je tak nastaveno) vypne profily. Tuto funkci lze nastavovat pouze po přihlášení administrátora.

- **Externí zásuvný modul ExtLED01**

Modul ExtLED01 již nabízí více možností automatizace. Jednou z hlavních funkcí může být automatické vypnutí a zapnutí světel. Pomocí pohybových senzorů (ty se mohou využívat např. i v integrovaném modulu IntBEZ01) lze zajistit automatické rozsvícení světel v dané místnosti.

Při přidávání LED pásu je možnost přidělení pohybového senzoru (zadání IP adresy a portu senzoru). Pokud daný pohybový senzor zaznamená pohyb, LED pás se rozsvítí. V momentě, kdy v místnosti není v požadovaný čas žádný pohyb, LED pás se opět vypne. Toto chování lze nastavit buď permanentně, nebo jen v požadovaný čas. Pokud funkce není zapnutá, LED pásy je nutno zapnout ručně pomocí ovládacího panelu nebo spínače na zdi.

Další automatickou funkcí je střídání barev v požadovaný čas. V pozdějších večerních hodinách je jistě příjemnější, když je světlo utlumené a LED pás je nastaven spíše na teplejší barvy. Naopak při práci je vhodnější, když je světlo intenzivnější a LED pás je nastaven na studenější barvu. Obyvatelé si jednoduše v nastavení LED pásu nastaví od kdy má svítit jaká barva, a pokud to LED pás podporuje, tak i intenzitu světla.

Všechny tyto funkce zajišťuje sám modul, který pak posílá LED pásu příkazy pro rozsvícení nebo zhasnutí. Samotný LED pás je ovšem ještě vybaven senzorem světla, díky kterému je schopný určit, jestli je zapotřebí LED pás rozsvítit nebo jestli je to zbytečné, jelikož je v místnosti příliš intenzivní světlo. Tato funkce je zcela nezávislá na výše uvedených funkcích. Pokud tedy bude nastaveno automatické zapnutí LED pásu od 17:00 do 8:00 dalšího dne a v 17:00 bude ještě světlo, LED pás se nerozsvítí ani v případě, kdy mu sám modul pošle příkaz na rozsvícení. Hlídání intenzity světla lze v nastavení LED pásu vypnout a LED pás se tak rozsvítí vždy.

Obrázek 14 - Nastavení ExtLED01

Na obrázku 14 je vidět, jak jednoduše lze nastavit noční hodiny pro automatické rozsvícení LED pásů pomocí senzoru pohybu. Dále lze nastavit celkem tři barevné profily, které se aktivují v nastavený čas. Pokud si uživatel nepřeje, aby LED pásy byly automaticky řízeny pohybovými senzory, stačí vymazat IP adresu pohybového senzoru.

6. Závěr

Cílem práce bylo navrhnout univerzální platformy IoT systému. Takový systém má za úkol zabezpečení domácnosti a zajištění většího komfortu pro všechny obyvatele. Při návrhu systému byl kladen důraz na jeho schopnost rozšiřování se o nové funkce a technologie. Není tedy problém v budoucnu přidat do již existující infrastruktury naprostou novou technologii, která bude zajišťovat přenášení dat nebo komunikaci mezi aktivními prvky. Systém byl navržen společně se dvěma moduly. Jeden modul pro zabezpečení domácnosti a druhý modul pro demonstraci, jak přidávat různé externí moduly a tak obohatit systém o zcela nové funkce. Pro ukázání funkčnosti modulů byly realizovány tři klíčové prvky. K bezpečnostnímu modulu byl realizován senzor pohybu a k externímu modulu LED pás.

Další klíčový prvek již slouží pro jednoduché ovládání celého systému včetně modulů. Jedná se o grafický, dotykový ovládací panel, který je umístěn v místnostech, a díky němu mohou obyvatelé ovládat různé aktivní prvky pomocí modulů. Ovládací panel obsahuje intuitivní grafické uživatelské rozhraní, přes které si může automatizovat domácnost, nebo měnit různá nastavení i laický uživatel. Na závěr se dá říct, že systém lze nasadit již do skutečného prostředí a je plně schopný ke každodennímu provozu. Testování systému proběhlo ve více etapách. Samostatně bylo testováno chování aktivních prvků (schopnost připojit prvek k již existující síti, udržení spojení bez výpadků, testování při výpadku elektrického proudu). Poté byly testy prováděny znovu s tím rozdílem, že aktivní prvky již byly obsluhovány integrovanými nebo externími moduly.

I když systém obsahuje funkci přidávání nových modulů a tím tedy zajištění nových funkcí, je přesto potřeba rozšiřovat a vylepšovat samotný řídicí systém. Pokud zapátráme v oblasti zabezpečení samotného řídicího systému, tak při aktuálním stavu řídicí systém neobsahuje žádnou obranu při útoku z vnějšího „světa“. Je zcela závislý na zabezpečení domácí sítě. V budoucnu bude tedy jistě zapotřebí přidat funkci šifrování dat na aplikační vrstvě a veškerou komunikaci mezi řídicím systémem a aktivními prvky šifrovat. Pokud se bude dbát na přísná pravidla zabezpečení domácí sítě a do sítě budou připojována jen autorizovaná zařízení, tak prozatím nehrozí žádné vážné nebezpečí.

SEZNAM POUŽITÉ LITERATURY

[1] *11 Internet of Things (IoT) Protocols You Need to Know About* [online]. 20. 4. 2015 [cit. 2019-04-18]. Dostupné z: https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about

[2] *Inteligentní řízení objektů – zabezpečení, provoz, efektivita* [online]. 20. 4. 2015 [cit. 2019-04-18]. Dostupné z: https://www.stavebnictvi3000.cz/clanky/inteligentni-rizeni-objektu-zabezpeceni-provoz-efektivita

[3] *Mesh sítě – P2P architektura v bezdrátových sítích* [online]. Dostupné z: https://www.stavebnictvi3000.cz/clanky/inteligentni-rizeni-objektu-zabezpeceni-provoz-efektivita

[4] Chapter 2 - Networks [online]. 2014. Dostupné z: https://www.sciencedirect.com/science/article/pii/B9780124077737000028

[5] *Z-Wave* [online]. 26. 2. 2016. Dostupné z: https://www.iot-portal.cz/2016/02/26/z-wave/

[6] *Sigfox Technology Overview* [online]. Dostupné z: https://www.sigfox.com/en/sigfox-iot-technology-overview

[7] *Sigfox* [online]. 26. 2. 2016. Dostupné z: https://www.iot-portal.cz/2016/02/26/sigfox/

[8] *What is the LoRaWAN™ Specification?* [online]. Dostupné z: https://lora-alliance.org/about-lorawan

[9] *802.11 Wireless Standards* [online]. Dostupné z: http://www.pearsonitcertification.com/articles/article.aspx?p=1329709&seqNum=4

[10] *Chapter 3 - ZigBee and IEEE 802.15.4 Protocol Layers* [online]. 2008. Dostupné z: https://internetofthingsagenda.techtarget.com/definition/ZigBee

[11] *About IQRF* [online]. Dostupné z: https://www.iqrf.org/iqrfabout

[12] *S IVONOU SPURNOU O IQRF ALLIANCE* [online]. Dostupné z: http://www.jaknaiot.cz/iqrf-alliance/

[13] *NodeMCU a jeho verzie: doska s Wi-Fi čipom ESP8266* [online]. Dostupné z: https://www.root.cz/clanky/nodemcu-a-jeho-verzie-doska-s-wi-fi-cipom-esp8266/

Seznam příloh

Tabulka 1 - Příkazy ŘS

Příkaz	Popis
LOG –M '{text}'	Příkaz pro zapsání události do záznamů systému. Při zápisu je přidán čas a datum události a jaký modul událost poslal (platí i pro řídicí systém nebo ovládací panel).
RP	Seznam právě běžících modulů.
RPI –P '{název_modulu}'	Zobrazení dostupných informací o modulu.
CLIENT –C '{název_klienta}'	Vypsání modulů, které jsou povoleny na panelu. Pokud není přidán argument –C, vypíše se seznam všech ovládacích panelů a k nim přiřazené moduly.
SDP –P '{název_modulu}'	Zakázání modulu.
ENP –P '{název_modulu}'	Povolení spuštění modulu.
DIP –P '{název_modulu}'	Zakázání spuštění modulu.
CEP –P '{název_modulu}'	Povolení spuštění modulu na zadaném ovládacím panelu,
CDIP –P '{název_modulu}'	Zakázání spuštění modulu na zadaném ovládacím panelu
SEND –P '{název_modulu}' –C '{příkaz}'	Poslání modulu příkaz.
STARTPL –P {název_modulu}'	Zapnutí modulu.
STOPPL –P {název_modulu}'	Vypnutí modulu.
SHUTDOWNP	Vypnutí všech modulů.
RESTART –C {název_klienta}'	Restartování ovládacího panelu.
RESTART SYSTEM	Restartování řídicího systému.
SEND TO CLIENT -P 'modul' -C 'klient' -M 'zpráva/příkaz'	Poslání příkazu zadanému modulu (jádro je využito jako komunikační tunel). Pokud není zadáno –P, je příkaz určen ovládacímu panelu, jinak je příkaz přeposlán zadanému modulu. Pokud není zadáno –C, je příkaz určen všem ovládacím panelům.

Tabulka 2 - Příkazy ŘS <- plugin

Příkaz	Popis
GET DBS DRIVER	Požádání o přístupové údaje pro připojení do databáze
GET DBS URL	
GET DBS USER	
GET DBS PASS	
NFC –M '{text}'	Poslání oznámení.
GNFC –M '{text}'	Poslání zprávy, která se zobrazí na všech ovládacích panelech.
GLOBAL ALARM ON	Zapnutí globálního alarmu.
GLOBAL ALARM OFF	Vypnutí globálního alarmu.
LOG –M '{text}'	Zapsání události do záznamů.

Tabulka 3 – Příkazy modul -> aktivní prvek

Příkaz	Popis
GET STATUS	Čidlo pošle svůj aktuální stav včetně stavu senzorů.
GET LED	Vyžádání stavu LED diody (svítí/nesvítí).
GET BUZ	Vyžádání stavu bzučáku.
SET LED [HIGH | LOW]	Zapnutí /vypnutí LED diody.
SET BUZ [HIGH | LOW]	Zapnutí/Vypnutí bzučáku
RESTART	Restartování čidla. Přepnutí do AP modu a zapnutí web serveru.
NETWORK –S '{SSID}' –P '{heslo}'	Nastavení sítě.

Tabulka 4 - Ukázka příkazů integrovaného modulu IntBEZ01

Příkaz	Popis
ADD MOTION -A "{název}, {ip_adresa}, {port}, [1 | 0]"	Přidání nového pohybového senzoru.
ADD SIREN -A "{název}, {ip_adresa}, {port}, [1 | 0]"	Přidání nové sirény.
SET -A "{název_nastaveni}" -S "{hodnota}"	Změna hodnoty v nastavení.
ADD PROFILE -A "{Profil}"	Přidání profilu.
ADD TO PROFILE MOTION -D "{název_čidla}" -F "{název_profilu}"	Přidání pohybového senzoru do existujícího profilu.
ADD TO PROFILE SIREN -D "{název_čidla}" -F "{název_profilu}"	Přidání sirény do existujícího profilu.
SET PSWSHA -P "{nove_heslo}" –O "{staré_heslo}"	Změna hesla pro aktivaci profilu
SET ALERT ON -P "{heslo}" -F "{název_profilu}"	Aktivování/deaktivování zadaného profilu. Pro aktivaci/deaktivaci je potřeba zadat heslo.

U integrovaného modulu IntBEZ01 je využito jádro jako komunikační tunel. Příkazy jsou tedy zapouzdřeny v příkazu „SEND TO CLIENT“ a je nutné nahradit znak „"“ za „'“. Pokud jsou příkazy poslány modulu přímo, musí být použit znak „"“.