ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ

Katedra aplikované elektroniky a telekomunikací

DIPLOMOVÁ PRÁCE

Elektronický zabezpečovací systém pro rodinný dům

Bc. Jan Giebl **2019**

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2018/2019**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Jan GIEBL**
Osobní číslo:	**E16N0039P**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Elektronika a aplikovaná informatika**
Název tématu:	**Elektronický zabezpečovací systém pro rodinný dům**
Zadávající katedra:	**Katedra aplikované elektroniky a telekomunikací**

Zásady pro vypracování:

1. Analyzujte trendy v zabezpečení budov se zaměřením na využití IoT.
2. Navrhněte a realizujte kompletní zabezpečení budovy, vyberte senzory, mikropočítač hlavní ústředny a periferie sloužící ke komunikaci mezi jednotlivými subsystémy.
3. Navrhněte a realizujte ovládací panel obsahující vlastní procesor, displej a klávesnici.
4. Navrhněte a realizujte bezdrátové Low Power zařízení sloužící k monitorování stavu ústředny.
5. Navrhněte a realizujte software pro řízení všech částí systému.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **40 - 60 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Systém pro sběr technologických dat v koncepci IoT Diplomová práce ČVUT [online]. (c) 2017 [cit. 10.4.2018]. Dostupné z: https://goo.gl/wyC8jf**
2. **BURDA, Karel: Úvod do kryptografie. Vydání první. Brno: Akademické nakladatelství CERM, 2015. 108 stran. ISBN 978-80-7204-925-7.**
3. **BURDA, Karel: Základy elektronických zabezpečovacích systémů. Vydání první. Brno: Akademické nakladatelství CERM, 2017. 123 stran. ISBN 978-80-7204-967-7.**

Vedoucí diplomové práce: **Ing. Petr Kropík, Ph.D.**
Katedra teoretické elektrotechniky

Datum zadání diplomové práce: **5. října 2018**

Termín odevzdání diplomové práce: **30. května 2019**

Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan

L.S.

Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 5. října 2018

Abstrakt

Předkládaná diplomová práce se zabývá návrhem a realizací funkčního elektronického zabezpečovacího systému. Práce v úvodní části seznamuje s obecnými vlastnostmi EZS systémů a následně definuje systémové požadavky, na jejichž základě dochází k návrhu vlastního zabezpečovacího systému. Návrh zahrnuje výběr vhodných komponent, ze kterých se sestaví zabezpečovací ústředna, ovládací panel, senzory a bezdrátové monitorovací zařízení. Po zhotovení funkčních prototypů dochází k návrhu řídícího softwaru všech prvků. V závěru práce je celý navržený systém zhodnocen.

Klíčová slova

Elektronický zabezpečovací systém, EZS, PZTS, Raspberry Pi, ATMega328p, Arduino. RS485, Node-RED, MySQL, GSM, Bezdrátová komunikace, NRF24L01

Abstract

This diploma thesis focuses on designing and developing the electronic security system. In the first part, it presents basic properties of the electronic security systems. In the following practical part, system requirements are defined and based on those requirements, the security system is designed. The design contains the selection of suitable components for building all the system parts which are: the main computer, the control panel, the detectors and the wireless monitoring device. After the prototypes have been assembled, the software is being designed and written. At the end of this thesis, the whole system is reviewed.

Key words

Electronic security system, Raspberry Pi, ATMega328p, Arduino. RS485, Node-RED, MySQL, GSM, Wireless communication, NRF24L01

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

podpis

V Plzni dne 29.5.2019 Bc. Jan Giebl

Poděkování

Tímto bych rád poděkoval především vedoucímu diplomové práce panu Ing. Petrovi Kropíkovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce. Dále bych chtěl poděkovat panu Ing. Karlovi Slobodníkovi za rady při návrhu krabičky ovládacího panelu pro 3D tisk. Rád bych poděkoval také své přítelkyni Ing. Anetě Volfíkové za korekturu diplomové práce.

Obsah

Obsah ... **8**

Seznam symbolů a zkratek ... **10**

Úvod ... **11**

1 Poplachový zabezpečovací systém s prvky IoT ... **12**

1.1 Úvod do problematiky ... 12

1.1.1 Architektura PZTS ... *12*

1.1.2 Prvky PZTS ... *13*

1.1.3 Typy PZTS ... *14*

1.2 Internet věcí ... 14

1.3 Současné trendy v zabezpečení budov se zaměřením na IoT ... 16

1.3.1 Standardní zabezpečovací systémy ... *16*

1.3.2 DIY zabezpečovací systémy ... *17*

2 Návrh na úrovni celého systému ... **20**

2.1 Systémové požadavky ... 20

2.2 Struktura systému ... 21

2.3 Výběr vhodných prvků ... 22

2.3.1 Raspberry Pi 3B+ ... *22*

2.3.2 ATMega328p ... *22*

2.3.3 MAX485 ... *23*

2.3.4 Senzory ... *25*

2.3.5 Displej a klávesnice ... *25*

2.3.6 Komunikační rádio NRF24L01 ... *26*

2.3.7 GSM modul ... *28*

2.4 Pravidla komunikace pro RS485 ... 28

3 Návrh a realizace Řídící ústředny ... **31**

3.1 Návrh a sestrojení HW části ústředny ... 31

3.2 Základní nastavení Raspberry Pi ... 32

3.2.1 Instalace webového serveru a databáze MySQL ... *32*

3.2.2 Node-RED ... *33*

3.3 Návrh a vývoj SW části ústředny ... 35
3.3.1 MySQL databáze ... 36
3.3.2 Hlavní/stavový program ústředny ... 38
3.3.3 Grafické uživatelské rozhraní GUI ... 41
3.3.4 Komunikační program ústředny ... 42
3.3.5 Program pro práci s GSM modulem a komunikaci s BMZ ... 44
4 Návrh a realizace sběrnicových prvků ... 46
4.1 Smart senzory ... 46
4.1.1 Návrh a sestrojení HW části senzoru ... 46
4.1.2 Návrh a vývoj SW části senzoru ... 47
4.2 Ovládací panel ... 49
4.2.1 Návrh a sestrojení HW části ovládacího panelu ... 49
4.2.2 Návrh a vývoj SW části ovládacího panelu ... 50
5 Návrh a realizace bezdrátového monitorovacího zařízení BMZ ... 52
5.1 Návrh a sestrojení HW části BMZ ... 52
5.2 Návrh a vývoj SW části BMZ ... 53
5.3 Měření spotřeby ... 55
6 Zhodnocení vlastností systému ... 58
Závěr ... 61
Seznam literatury a informačních zdrojů ... 63
Přílohy ... 1
Příloha A – Schémata zapojení ... 1
Příloha B – Motivy plošných spojů ... 4
Příloha C – Osazování ústředny ... 6
Příloha D – Osazování Smart senzoru ... 7
Příloha E – Návrh krabičky a osazování ovládacího panelu ... 8

Seznam symbolů a zkratek

API.....*Application Programming Interface*, rozhraní pro programování aplikací

ARM.....Architektura procesorů

BMZ .....Bezdrátové monitorovací zařízení

CRC.....*Cyclic Redundancy Check*, cyklický redundantní součet

DIY.....*Do It Yourself,* označení systému, který si sestaví a nakonfiguruje sám uživatel

FSK .....*Frequency-Shift Keying*, klíčování frekvenčním posuvem

GPIO.....*General-Purpose Input/Output*, univerzální vstupní/výstupní pin

GPRS .....*General Packet Radio Service*, služba pro přenos dat

GUI.....*Graphical User Interface*, grafické uživatelské rozhraní

HTTP.....*Hypertext Transfer Protocol*, internetový protokol pro přenos dat

I2C.....*Inter-Integrated Circuit*, počítačová sériová sběrnice

IoT .....*of Things*, internet věcí

ISO/OSI.....Referenční komunikační model

LED .....*Light-Emitting Diode*

LTE.....*Long Term Evolution*, technologie určená pro internet v mobilních sítích

LPWAN.....*Low-Power Wide-Area Network*, typ IoT sítí

LXDE .....*Lightweight X11 Desktop Environment*, desktopové prostředí

MCU.....Mikrokontrolér

MQTT.....*Message Queuing Telemetry Transport*, komunikační standard

MW.....*Microwave*, mikrovlnný

PCO .....Pult centrální ochrany

PHP.....*Hypertext Preprocessor*, skriptovací programovací jazyk

PIR.....*Passive Infrared Sensor*, pasivní infračervené čidlo

SCL SDA.....*Synchronous Clock, Synchronous Data*, signály I2C sběrnice

SoC .....*System on Chip*, systém na čipu

SPI .....*Serial Peripheral Interface*, typ sériové sběrnice

SQL .....*Structured Query Language*, strukturovaný dotazovací jazyk

SSL .....*Secure Sockets Layer*, protokol pro zabezpečení komunikace

TTL.....*Transistor-Transistor-Logic*, tranzistorová logika

TXD RXD .....*Data transmitter, data receiver*, vysílač, přijímač

UART .....*Universal Asynchronous Receiver and Transmitter*, sériová komunikace

WDT.....*Watchdog timer*, elektronický časovač

WiFi.....Standard bezdrátové komunikace

Úvod

Cílem diplomové práce je kompletní návrh a realizace elektronického zabezpečovacího systému pro rodinný dům. Motivací k výběru tohoto tématu byla touha navrhnout si vlastní funkční systém a zároveň si zabezpečit domácnost. Výběrem vhodných komponent se také otevře možnost rozšířit v budoucnu systém o prvky domácí automatizace.

V první kapitole je sepsán teoretický úvod do EZS systémů, do Internetu věcí a vytvořen přehled současných trendů na poli zabezpečení s využitím Internetu věcí.

V praktické části nejprve dochází k sepsání systémových požadavků a výběru vhodných komponent pro jednotlivé části systému, kterými jsou zabezpečovací ústředna, ovládací panel, Smart senzory a bezdrátové monitorovací zařízení. V této části se dále řeší i propojení ústředny s ostatními prvky a způsob komunikace.

Významná část práce je věnována návrhu ústředny. Dojde k výběru vhodného počítače, na kterém bude implementován webový a databázový server, hlavní řídící program, grafické uživatelské rozhraní a komunikační program. Do databáze se budou ukládat přijatá data od jednotlivých prvků systému a hlavní program ústředny s nimi bude dále pracovat. Webový server bude sloužit pro vzdálenou správu a řízení systému. Ústředna bude obsahovat také vybraný GSM modul pro odesílání SMS zpráv.

Následuje návrh a realizace jednak sběrnicových prvků, kterými jsou ovládací panel a Smart senzor, jednak bezdrátového monitorovacího zařízení. Dojde k výběru společného mikrokontroléru a čidel. Při návrhu bezdrátového monitorovacího zařízení bude kladen důraz na nízkou spotřebu.

V závěru práce bude systém zhodnocen a porovnán oproti vytyčeným požadavkům. Dojde také k návrhu dalších možných rozšíření systému.

1 Poplachový zabezpečovací systém s prvky IoT

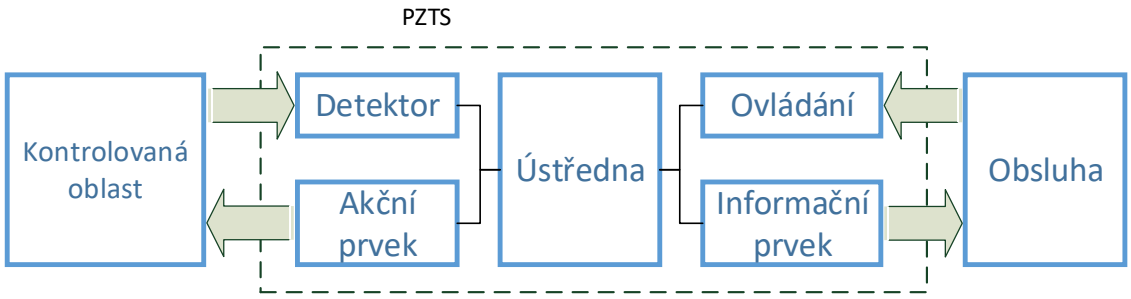
Diplomová práce se zabývá návrhem elektronického zabezpečovacího systému (dále také EZS) pro rodinný dům. Zkratka EZS je běžně užívaným pojmem označující poplachový zabezpečovací a tísňový systém (PZTS). V následující kapitole dojde k vysvětlení hlavních částí celého systému a závěr kapitoly bude věnován novým trendům v zabezpečení budov a propojení s Internetem věcí.

1.1 Úvod do problematiky

Poplachový zabezpečovací a tísňový systém lze definovat jako elektronický systém určený k detekci a signalizaci vzniku nežádoucích událostí ve sledovaném prostoru. Sledovaným prostorem je oblast, ve které se nachází střežený majetek, a kterou má majitel systému pod svou kontrolou. V současnosti jsou pro PZTS nabízeny i detektory požáru, úniku vody, či nebezpečných plynů. Další skupinou jsou tísňové události. Těmi se rozumí zdravotní potíže osob, či násilí na těchto osobách. Komerční PZTS jsou navrhovány v souladu s normami ČSN EN 50131-1 až ČSN EN 50131-7. [1]

1.1.1 Architektura PZTS

Obecnou architekturu ilustruje obrázek 1. Jádrem systému je ústředna, která celý systém řídí. Detektory slouží k detekci incidentů ve sledované oblasti. Informační zařízení mají za úkol informovat obsluhu nebo majitele o vzniku incidentu. Interakce uživatele se systémem je skrz ovládací prvek. Někdy se také v systému mohou objevit akční prvky, které mají za úkol vyvolat nějakou činnost, například zamlžení oblasti. Prvky systému mezi sebou mohou komunikovat po kabelu či bezdrátově. [1]



Obrázek 1: Obecné schéma PZTS [1]

1.1.2 Prvky PZTS

Detektor

Detektory jsou elektrická zařízení, která zpravidla obsahují čidlo. Jejich úkolem je detekovat vznik incidentu ve sledované oblasti. Detektory lze dělit na několik skupin.

- **Předmětové detektory** slouží k detekci manipulace s předměty. Nejčastěji jsou použity tzv. tíhové detektory.
- **Překážkové detektory** slouží k detekci proniknutí útočníka přes určitou překážku, ať už se jedná o hranici pozemku (ploty), pláště budov (okna, dveře), či pláště úložišť (trezor). Mohou sem patřit například otřesové detektory, jazýčkové magnetické kontakty, či detektory tříštění skla.
- **Objemové detektory** slouží k detekci pohybu útočníka v prostoru. Charakteristickou vlastností je objemový tvar detekčního diagramu. Do této kategorie patří pasivní infračervené detektory PIR, mikrovlnné senzory MW nebo duální detektory, které obsahují PIR i MW v jednom pouzdře.
- **Hraniční detektory** podobně jako objemové detektory slouží k detekci pohybu útočníka v prostoru, rozdílem je jiný tvar detekčního diagramu. Typicky se jedná buď o linii nebo plochu. Detektor tak vytváří v hlídané oblasti virtuální hranice. Do této kategorie patří například úsekový detektor se štěrbinovými kabely, liniový MW detektor, liniový PIR detektor, či lidarový detektor. [1]

Ústředna

Ústředna je centrálním prvkem každého zabezpečovacího systému. Slouží k řízení komunikace s ostatními prvky systému a k vyhodnocování přijatých informací ze senzorů. Ústředna rozděluje oblast do sekcí. Může obsahovat připojení k internetu, GSM modul pro odesílání SMS zpráv, protokoly pro komunikaci s pultem centrální ochrany, záložní bateriový zdroj a další moduly. [1]

Ovládání a Informační prvek

Ovládací panel standardně obsahuje klávesnici, displej a informační LED diody. Umožňuje uživateli komunikovat s ústřednou a ovládat její režimy. V současnosti se pro ovládání často používají dotykové panely či internetové nebo mobilní aplikace. [1]

1.1.3 Typy PZTS

Podle typu spojů lze PZTS rozdělit na kabelový systém, rádiový systém nebo hybridní systém. Hybridní systém spojuje kabelové propojení a rádiové propojení do jednoho celku. Kabelový PZTS se dá dále rozdělit podle typu propojení na smyčkovou, sběrnicovou nebo kombinovanou topologii.

Smyčková topologie je z historického pohledu nejstarší používanou topologií. Obvykle se jedná o smyčku proudovou. Smyčkou je připojeno k ústředně pouze jedno zařízení a celý systém má pak podobu hvězdy. Jedná se o jednosměrný přenos informace, kdy komunikace s detektory probíhá ve směru k ústředně a komunikace se signalizačními a akčními prvky probíhá ve směru od ústředny. Výhodou proudové smyčky je jednoduchost a robustnost, nevýhodou je omezený počet příkazů a cena za nákladný kabelový rozvod.

V současnosti se více používá sběrnicová topologie. Nejčastěji se využívá master/slave komunikace, ve které má ústředna roli mastera a v určitých cyklech dotazuje podřízené prvky. Jednou z podmínek je použití inteligentních prvků, které obsahují vlastní mikrokontrolér. Výhodou sběrnicové topologie je obousměrný přenos informace a jednodušší kabelový rozvod. Při kombinaci smyčkové a sběrnicové topologie se hovoří o kombinované topologii.

Dalším používaným řešením je rádiový či hybridní systém, ve kterém ústředna komunikuje s bezdrátovými prvky. Bezdrátové prvky jsou navrhovány nízkoodběrově, a často jsou schopny fungovat bez výměny baterie i několik let. Rádiový systém nepotřebuje kabelový rozvod, na druhou stranu bezdrátová komunikace nemusí být tolik spolehlivá a může snadněji dojít k jejímu rušení. [1]

1.2 Internet věcí

Internet věcí nebo také zkráceně IoT (z anglického *Internet of Things*) je v posledních letech stále více uváděný pojem. Tímto názvem se označuje zjednodušeně řečeno jakékoliv zařízení připojené k síti pracující samostatně bez nutnosti lidského zásahu. Například chytrý dům, auto nebo také hodinky měřící tepovou frekvenci sportovce, či informační tabule na autobusové zastávce informující o aktuální poloze konkrétních autobusů. Většinou se jedná o přenos bezdrátový. Nutno dodat, že původní myšlenka IoT není novinkou, ale je známá už několik dekád. [2]

Standardy používané pro přenos dat v IoT

Mezi standardní a běžně známé zástupce patří WiFi a Bluetooth komunikace, ale vzhledem k tomu, že je často IoT zařízení napájeno bateriově a je v zájmu přenášet data jednoduše a energeticky nenáročně na velké vzdálenosti, vyvinuly se nové LPWAN technologie. LPWAN je zkratkou slov *Low-Power Wide-Area Network*, v českém přikladu Nízkoodběrová širokopásmová síť. [3]

Mezi hlavní zástupce LPWAN patří SigFox, LoRa, IQRF nebo NB-IoT. Technologie většinou komunikují ve vyhrazeném pásmu, v Evropě je to 868 MHz. Komunikační dosah bývá maximálně desítky kilometrů, a proto je v zájmu všech společností vybudovat sítě s velkým pokrytím. Při využití zmíněného rádiového pásma jsou navíc všechny technologie omezeny připojením necelých 15 minut denně s výkonem do 25 mW. Hlavní vlastnosti technologií jsou následující. [3]

- **IQRF** vyvinula česká společnost MICRORISC již v roce 2004. Komunikace prvků v síti má topologii MESH. V této topologii komunikuje každý s každým. Srdcem komunikace je brána, ke které se může připojit až 239 dalších zařízení. Uvnitř topologie probíhá komunikace na vzdálenosti stovek metrů. Brána nasbíraná data odesílá například přes klasické Wifi do IQRF cloudu. [4]
- **LoRa** vyvinula francouzská společnost Cycleo. Komunikace v síti má hvězdicovou topologii. Pro tvorbu sítě si aliance LoRa vybírá vhodné partnery. Například v České republice jsou to České Radiokomunikace. Fyzická vrstva LoRa využívá modulaci s rozprostřeným spektrem, díky čemuž je schopná uvnitř sítě komunikovat s pohybujícími se zařízeními. Vzdálenost komunikace jsou jednotky kilometrů. [5]
- **Sigfox** bylo vyvinuto stejnojmennou firmou v roce 2009 ve Francii. Má obdobně jako LoRa hvězdicovou topologii a partnery pro tvorbu sítě jsou telefonní operátoři. V České republice je to T-mobile. Pro odesílání se využívá tzv. ultra úzké pásmo, díky čemuž je síť odolná vůči rušení. [6]
- **NB – IoT** je dalším zástupcem využívající podobně jako Sigfox spolupráci s mobilními operátory. V České republice je to Vodafone a O2. Tato technologie na rozdíl od předchozích používá GSM a LTE pásmo. [7]

1.3 Současné trendy v zabezpečení budov se zaměřením na IoT

1.3.1 Standardní zabezpečovací systémy

Standardními zabezpečovacími systémy se myslí systémy s vlastnostmi popisovanými v úvodu kapitoly. Kromě toho jsou tyto systémy charakteristické také potřebou odborné instalace systému či nekompatibilitou se zařízeními jiných dodavatelů. Hlavní společností, dodávající EZS systémy do České republiky, je především Jablotron, ale najdou se i další společnosti, například kanadský Paradox.

Jablotron 100

Řada Jablotron 100 je v České republice vlajkovou lodí na poli zabezpečovacích systémů. Systém vyvinula Česká společnost Jablotron a na trh jej uvedla v roce 2015. Oproti předchozí generaci systémů Jablotron úplně upustil od smyčkových topologií a soustřeďuje se výhradně na hybridní systémy kombinující sběrnicové a rádiové připojení.

Nejzákladnější ústředna JA-101K má následující parametry. [8]

- až 50 bezdrátových nebo sběrnicových periferií
- až 50 uživatelských kódů
- až 6 sekcí (režimů zabezpečení)
- až 8 programovatelných výstupů PG
- 20 vzájemně nezávislých kalendářů
- SMS reporty ze systému až 8 uživatelům
- 8 uživatelských SMS a hlasových reportů (vestavěný komunikátor GSM/GPRS)
- 5 nastavitelných PCO (připojení na pult centrální ochrany)
- 5 volitelných protokolů pro PCO

Požadované nastavení a velikost systému se programují prostřednictvím softwaru F-link. Pro změnu uživatelských parametrů slouží program J-Link.

Rádiová komunikace uvnitř systému probíhá v pásmu 868 MHz s použitím Jablotron komunikačního protokolu. Výrobce udává, že výdrž baterie bývá u různých bezdrátových prvků 2 až 3 roky. [8]

S příchodem řady Jablotron 100 společnost začala propojovat EZS systémy s domácí automatizací. Například je možné k EZS systému přidat další prvky starající se o ovládání topení, rolet, osvětlení, garážových vrat nebo zavlažování zahrady. [11]

Na oficiálních stránkách Jablotronu se pojem IoT vidí velmi zřídka. Přeci jen ale ve svých řadách disponují cloudovou aplikací MyJABLOTRON, ve které uživatel vidí správu celého systému. Do systému MyJABLOTRON se uživatel přihlašuje přes webový prohlížeč nebo přes mobilní aplikaci, která existuje ve verzi jak pro Android, tak i pro iPhone. Z principu se tak jedná o IoT, neboť se data z různých senzorů ukládají na cloudové úložiště a uživatel si pak prostřednictvím mobilní aplikace může data vyčíst, či systém řídit.

Na základě článků [9], [10] na internetu se zdá, že společnost Jablotron také vyvíjí nějakou další aktivitu směrem k IoT. Nicméně Jablotron se nevěnuje pouze EZS, má širší portfolio. Nedá se tak s určitostí říct, že se budoucí vývoj IoT promítne i na potenciální další generaci EZS.

1.3.2 DIY zabezpečovací systémy

DIY je zkratka anglických slov *do-it-yourself*, což ve volném překladu znamená, udělej si sám. Jak tedy z názvu vyplývá, společnou vlastností DIY systémů je velmi snadná instalace a konfigurace, kterou zvládne uživatel sám bez pomoci odborníka. Tato kategorie zabezpečovacích systémů se dá považovat za novou generaci EZS systémů pro rodinné domy. DIY systémů existuje velké množství od levných čínských řešení po technologicky vyspělé systémy. Tato kapitola věnuje pozornost výhradně vyspělejším systémům, zaměřujících se na propojení zabezpečovacích systémů, domácí automatizace a IoT.

Dalším společným rysem DIY systémů jsou: použití rádiové komunikace uvnitř systému, vysoká výdrž baterie bezdrátových prvků, připojení ústředny k internetu, využití cloudové databáze, používání mobilní aplikace k řízení systému a pěkné designové zpracování. [12]

Hlavními zástupci jsou systémy Abode, LifeShield, SimpliSafe, SmartThings ADT, Wink, Ring, Front Point, Angee, Honeywell a další. Nutno dodat, že v České republice jsou k dostání prozatím pouze nepříliš kvalitní varianty DIY systémů neznámých výrobců, které technologicky končí u posílání SMS zpráv přes GSM modul. V tomto ohledu má pravděpodobně Jablotron volnou cestu k vytvoření vlastního sofistikovaného DIY systému pro český trh. [12], [13]

Abode

Zástupcem DIY systému pro podrobnější popis byl vybrán Abode, protože bývá v recenzích dobře hodnocen. Abode druhé generace byl uveden na trh začátkem roku 2019 a je na obrázku 2. [14]

Obrázek 2: Abode EZS systém [14]

Základem celého systému je ústředna, která je připojená do internetové sítě přes Ethernet. Systém může obsahovat až 160 zařízení. Ústředna komunikuje s připojenými prvky v pásmu 433 MHz s použitím AbodeRF protokolu. Pro zajištění kompatibility s prvky od jiných výrobců ústředna obsahuje také ZigBee a Z-wave komunikaci.

Ostatní prvky jako jsou PIR detektory či dveřní a okenní čidla se jednoduše přilepí na požadované místo. Rozložení systému si uživatel nakonfiguruje pomocí mobilní aplikace, přes kterou systém i řídí. Ústředna získaná data od senzorů posílá do cloud databáze. Systém umožňuje funkci automatického zastřežení a odstřežení. Například na základě propojení s mobilním telefonem, či při používání originální klíčenky.

Ústředna podporuje propojení s dalšími chytrými zařízeními jako jsou například: hlasově ovládaní asistenti Amazon Alexa, Google assistant či prvky inteligentní domácnosti Google Nest, Ecobee a další. [14]

Angee

Systém Angee je zajímavým startupovým projektem, za kterým stojí čeští vývojáři. Startup je pojem označující začínající projekt, postavený na funkčním prototypu. Je-li prototyp úspěšný, snaží se vývojáři sehnat peníze na sériový vývoj. V případě Angee se vývojáři pokusili sehnat dostatek finančních prostředků na KickStarter se zaměřením na americký trh. Na Kickstarter se jim podařilo vybrat 531 tisíc dolarů a vývoj sériového zařízení tak mohl začít. Nyní už je projekt dokončený a na trh již pronikla první verze systému. Angee je na obrázku 3. [16]

Obrázek 3: Angee autonomní zabezpečovací systém [15]

Angee patří do podkategorie DIY systémů nazývané *All-in-one*, což v českém překladu znamená vše v jednom. Systém je tvořen kamerou a ústřednou v jedné podobě. Kamera je otočná o 360° a obsahuje navíc 6 PIR čidel tak, aby jí neunikl žádný pohyb uvnitř místnosti. K plné funkčnosti systému se používají ještě dveřní a okenní čidla, která s kamerou komunikují bezdrátově. Systém je označován jako autonomní, a to především z toho důvodu, že pokud je propojen s chytrým telefonem, tak aktivace a deaktivace probíhá automaticky při příchodu či odchodu. Díky okenním a dveřním senzorům systém umí střežit i pokud jsou uvnitř bytu lidé. Angee má implementované rozpoznávání řeči, a proto je možné systém deaktivovat hlasem. Díky pokročilému zpracování zvuku slouží i jako detektor tříštění skla. Využívá cloudové databáze a komunikaci s uživatelem přes mobilní aplikaci. Podporuje i další komfortní služby.

Systém je připojen do sítě přes WiFi. Uvnitř systému s okenními senzory ústředna komunikuje přes Bluetooth 4.1 a podporuje i komunikaci přes ISM 433 MHz. Podobné systémy nabízí například společnost Amarrylo nebo Honeywell. [15]

Zhodnocení situace

Na českém trhu v podstatě řídí trendy společnost Jablotron. Je to způsobené pravděpodobně dlouholetou tradicí a velmi rozšířenou sítí firem, které instalují systémy výhradně od Jablotronu. Nejnovější zabezpečovací systém Jablotron 100 umí určitým způsobem propojit EZS s inteligentní domácností a díky aplikaci MyJABLOTRON je podporováno také IoT. DIY EZS systémy mají na českém trhu velmi omezenou a nepříliš kvalitní nabídku. Směrem k propojení zabezpečovacích systémů s IoT mají nejblíže DIY zabezpečovací systémy, které se nabízejí na americkém trhu.

2 Návrh na úrovni celého systému

Během návrhu elektronického zabezpečovacího systému bylo zapotřebí sepsat systémové požadavky a na základě nich definovat strukturu celého systému. Ve chvíli, kdy dojde k vytvoření struktury systému, přejde se na výběr vhodných komponent a prvků pro jednotlivé subsystémy. Všemi výše zmíněnými kroky se zabývá následující kapitola.

2.1 Systémové požadavky

Před vlastním vývojem praktické části diplomové práce byl sestaven seznam požadavků. Seznam se nachází v tabulce 1.

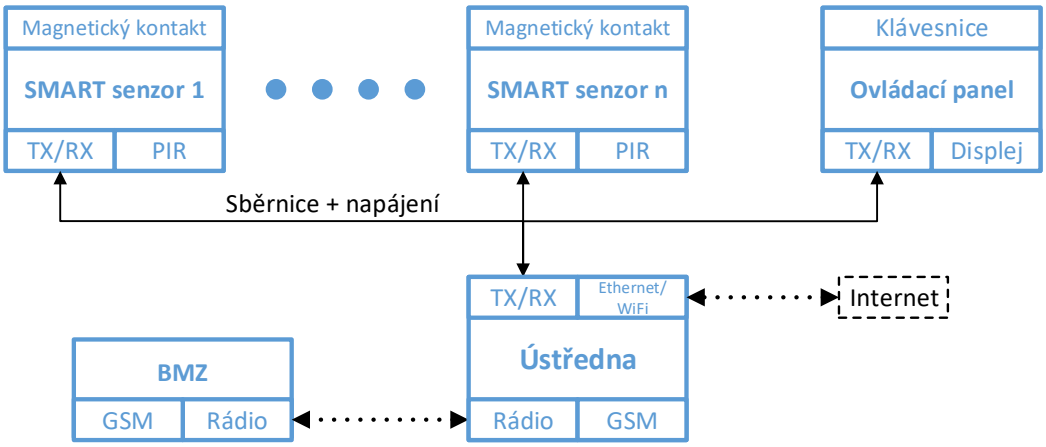
Tabulka 1: Systémové požadavky

Systémové požadavky		
Priorita	ID	Popis
1	1.1	EZS musí zahrnovat ústřednu, ovládací panel, senzory a bezdrátové monitorovací zařízení.
1	1.2	Zastřežení/odstřežení systému bude probíhat přes ovládací panel, který musí obsahovat klávesnici, displej a vlastní mikrokontrolér.
1	1.3	Systém bude obsahovat Smart senzory s PIR čidly a magnetickými kontakty.
1	1.4	Bude sestrojeno monitorovací zařízení, které bude s ústřednou komunikovat bezdrátově – toto zařízení v případě, že dojde k přerušení přívodu energie, může za určitých podmínek také vyvolat poplach.
2	2.1	Systém musí mít sběrnicovou topologii.
2	2.2	Senzory ani ovládací panel nebudou mít vlastní napájení. Napájení bude přivedeno společně se sběrnicí.
2	2.3	Po spuštění poplachu obdrží uživatel SMS a email.
2	2.4	Ústředna bude připojena k internetu.
2	2.5	Ústředna bude ukládat logy na paměťové médium (SD karta, Flash) a do databáze.
3	3.1	Systém bude mít záložní bateriový zdroj.
3	3.2	Ovládací panel bude kromě klávesnice obsahovat i čtečku otisků prstů.
3	3.3	Bude vytvořen uživatelský program, který bude pracovat s daty uloženými do databáze - uživateli nabídne celkový přehled systému a umožní mu vzdáleně zastřežit systém.

V tabulce jsou požadavky seřazeny podle priorit. Priorita 1 znamená, že požadavek úzce souvisí s některým bodem zadání diplomové práce, a tudíž je pro úspěšné dokončení klíčový. Požadavky označené prioritou 2 jsou požadavky, kterými se diplomová práce zabývá a je snaha je splnit, ale přímo nesouvisí se zadáním. Požadavky s prioritou 3 patří do kategorie požadavků, které nejsou pro diplomovou práci zásadní a jedná se spíše o bonusové vlastnosti. V tabulce je každému požadavku přiřazeno také identifikační číslo, pomocí něhož se na požadavky v textu odkazuje.

2.2 Struktura systému

Definice struktury systému byla vytvořena na základě požadavků s prioritou 1 a 2 z tabulky 1. Struktura systému je na obrázku 4.



Obrázek 4: Struktura systému

Z obrázku je patrné, že má být sestrojena zabezpečovací ústředna, která bude s ostatními prvky v systému komunikovat po sběrnici. Společně se sběrnicí bude k ostatním prvkům přivedeno i napájení. Ústředna musí obsahovat Ethernet nebo WiFi modul pro připojení k internetu, GSM modul pro odesílání SMS zpráv a rádio pro bezdrátovou komunikaci s monitorovacím zařízením. Ústředna bude mít implementovaný webový server a databázi pro sběr dat z podřízených prvků. Podřízenými prvku budou senzory a ovládací panel. Ovládací panel musí obsahovat displej a klávesnici. Senzory musí obsahovat PIR čidla a magnetické kontakty. S ústřednou bude komunikovat bezdrátové monitorovací zařízení (dále také BMZ), které sleduje bezchybný stav ústředny. V případě, že ústředna přestane odpovídat, BMZ odešle uživateli přes GSM modul zprávu o poruše.

2.3 Výběr vhodných prvků

2.3.1 Raspberry Pi 3B+

Jako mikropočítač zabezpečovací ústředny bylo vybráno Raspberry Pi 3B+. Jádrem systému Raspberry Pi je multimediální procesor BCM2837 typu SoC neboli *System on Chip*. Tento procesor se ovšem liší od procesorů, kterými jsou vybaveny notebooky či stolní počítače nejen návrhem SoC, ale i použitím ARM instrukční sady. Architektura ARM ve srovnání s počítačovou architekturou x86 vyniká nízkou spotřebou. V důsledku to však znamená, že Raspberry Pi není kompatibilní s běžným softwarem pro počítače. Raspberry Pi tak nepodporuje operační systémy Windows ani Apple OS X, ale počítá s použitím GNU/Linux, který patří do kategorie *open source* systémů. Raspberry Pi se může ovládat pomocí příkazové řádky, ale také pomocí desktopového LXDE prostředí. Raspberry Pi 3B+ je v současnosti nejvýkonnější verze počítače. Raspberry Pi byl vyvinut vzdělávací nadací Raspberry a je určen pro využití v podobných prototypových automatizačních projektech. [17], [18]

Hlavní vlastnosti Raspberry Pi 3B+

- architektura ARMv8-A (64/32-bit)
- system on chip BCM2837B0 obsahující čtyřjádrový procesor 1,4 GHz 64-bit ARM Cortex-A53
- 2,4 GHz a 5 GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, Gigabit Ethernet
- 40 pinový GPIO (UART, I2C, SPI), HDMI konektor, 4 USB 2.0 porty
- kamera port, displej port, Stereo výstup
- micro SD port pro nahrávání operačního systému a ukládání dat
- napájení 5 V/2,5 A DC [17]

S přihlédnutím na seznam požadavků z tabulky 1 je Raspberry Pi také vhodným kandidátem, protože automaticky splňuje požadavek 2.4 na připojení k internetu, a kromě toho má dobré předpoklady pro splnění požadavků 2.3, 2.5 a 3.3.

2.3.2 ATMega328p

Mikrokontrolér ATMega328p je především známý díky svému použití v *open source* platformě Arduino. Při použití Arduino zavaděče neboli bootloaderu se s pomocí knihoven jazyka Wiring stává velmi účinným a snadno programovatelným řídícím prvkem. Tento mikrokontrolér byl vybrán pro řízení všech podsystémů kromě ústředny. ATMega328p má

i velmi dobré odběrové vlastnosti. Podporuje *Power down* režim, při kterém je proudový odběr v řádech μA. V aktivním režimu při napájení 3,3 V a frekvenci 8 MHz by proudový odběr neměl přesáhnout hranici 3 mA. Při napájení 5 V a frekvenci 16 MHz už by ale spotřeba mohla být klidně 9 mA. ATMega328p kromě klasických vstupních/výstupních pinů má I2C, SPI nebo UART periferie a podporuje dva externí vstupy přerušení. [19]

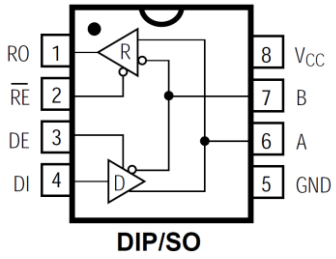
2.3.3 MAX485

Požadavky 1.2, 1.3 a 2.1 z tabulky 1 odkazují na použití komunikační sběrnice. Nabízí se celá řada sběrnic, ale bez většího váhání byl zvolen standard sériové komunikace RS485, a to především z důvodu ceny a poměrně jednoduché realizace fyzické i aplikační vrstvy ISO/OSI modelu. Tento standard je ze svého principu master/slave, v základní verzi umožňuje připojení až 32 zařízení a dovoluje zapojení na vzdálenosti stovek metrů.

Zvolený počítač Raspberry Pi ani mikrokontrolér ATMega328p nepodporují RS485, ale mají GPIO porty, které podporují sériovou komunikaci. Na základě tohoto faktu byly použity převodníky TTL úrovně na RS485 úrovně. Pro potřeby testování byly zakoupeny dva typy modulů obsahující RS485 převodník MAX485 od firmy Maxim Integrated. [20] Tento obvod slouží jako poloduplexní vysílač a přijímač RS485. Umí tedy generovat a přijímat signály A a B typické pro RS485.

Vlastnosti obvodu MAX485

- 8 pinové pouzdro obsahující jeden vysílač a jeden přijímač viz obrázek 5
- piny /RE a DE sloužící k přepínání mezi příjmem a vysíláním
- maximální napětí VCC 12 V
- přenosová rychlost 2,5 Mb/s
- klidový proudový odběr v závislosti na zapojení pinu DE buď 300 μA (DE = 0 V) nebo 500 μA (DE = VCC)
- proudový odběr při vysílání 35 mA až 250 mA
- proudový odběr při příjmu 7 mA až 95 mA

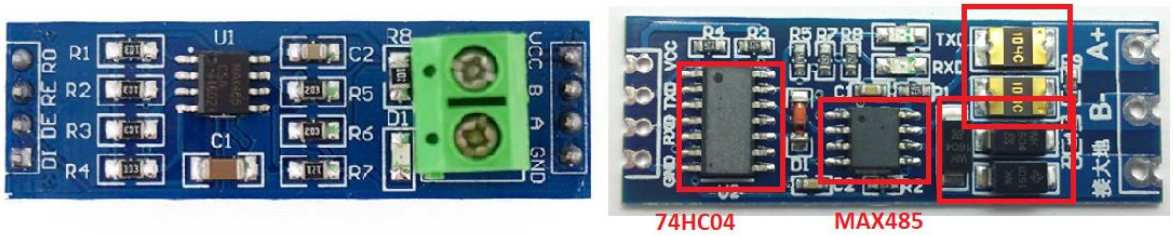


Obrázek 5: Vnitřní schéma MAX485 [20]

Obvody MAX481/483/487 podporují i nízkoodběrový mód, při kterém je proudový odběr pouhých 10 μA. Tyto čipy bohužel nejsou součástí hotového převodníku a jejích cena je přibližně trojnásobná oproti MAX485. [20]

V předchozím textu bylo zmíněno, že byly nakoupeny dva typy modulů obsahující převodník MAX485 viz obrázek 6. Bohužel dokumentace k těmto modulům není buď vůbec žádná a pokud je, tak velmi stručná. Levnější modul používá MAX485 CBA, který by měl pracovat v rozmezí teplot od 0 °C do 70 °C. Dražší modul používá MAX485 ESA, který je schopný pracovat v rozsahu teplot -40 °C až 80 °C.

Dalším rozdílem je použití obvodu 74HC04 u dražšího modulu. Tento obvod obsahuje 6 invertorů, 1 napájecí pin a 1 zemnící pin. Použití tohoto obvodu má za následek hned dvě věci. Zaprvé se stará automaticky o přepínání vysílače a přijímače přivedením správných napěťových úrovní na piny /RE a DE. Ve výchozím stavu je aktivní přijímač, ale když přijdou data na vstup TXD, aktivuje se vysílání. Druhou zásadní vlastností je omezení napětí na pinu RO hodnotou přivedeného napětí VCC. Například pokud se na pin VCC obvodu 74HC04 přivede napětí 3,3 V, je zaručeno, že na výstupech invertorů bude maximální napětí 3,3 V, i když signály A a B mají napěťové úrovně vyšší. Touto vlastností je zajištěna vzájemná kompatibilita, neboť Raspberry Pi má na pinech GPIO povolené vstupní napětí 3,3 V, zatímco například ATMega328p umístěná v ovládacím panelu bude pracovat s napětím 5 V. Dále je z obrázku patrné, že modul vpravo obsahuje další ochranné prvky. Na základě tohoto porovnání byla použita dražší verze modulu. Komunikaci se dále věnuje kapitola 2.4. [21]



Obrázek 6: Moduly s převodníky MAX485 - TTL na RS485

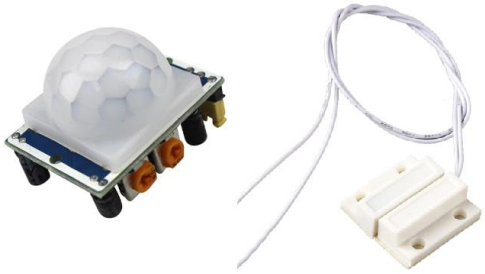
2.3.4 Senzory

PIR senzor

Jako PIR senzory byly použity moduly HC-SR501, které obsahují PIR čidlo, PIR kontrolér BISS0001, Fresnelovu čočku a potenciometry, kterými je možné nastavit citlivost obvodu a dobu sepnutí. Podle mechanického propojení jumperu se dodatečně nastavuje opakované sepnutí připojením pinu na napájení, a neopakované sepnutí obvodu připojením pinu na zem. PIR senzor patří do kategorie objemových detektorů a slouží pro detekci pohybu útočníka v prostorách uvnitř budovy. Použitý PIR senzor je na obrázku 7. [22]

Magnetický kontakt

Jak již bylo zmíněno v úvodní kapitole o PZTS, magnetický kontakt slouží k detekci otevřených dveří nebo oken. Jedná se tedy o překážkový detektor. Použitý kontakt je na obrázku 7. [23]



Obrázek 7: PIR senzor a magnetický kontakt [22], [23]

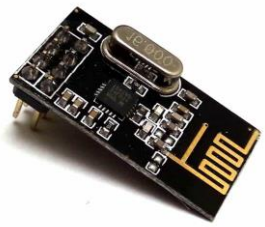
2.3.5 Displej a klávesnice

Z požadavku 1.2 vyplývá, že ovládací panel musí obsahovat displej a klávesnici pro možné interakce uživatele se zabezpečovací ústřednou. Jako klávesnice byla použita šestnáctiznaková klávesnice, která má osm datových vodičů. Jako displej byl vybrán dvouřádkový LCD displej, který se standardně používá v Arduino aplikacích. Displej musí být k mikrokontroléru připojen šesti datovými vodiči a vzhledem k omezenému počtu výstupů mikrokontroléru a nutnosti připojení klávesnice a převodníku MAX485, bylo záhodno k displeji použít obvod PCF8574, který funguje jako převodník na I2C sběrnici. Díky tomu došlo k ušetření 4 datových vodičů, neboť I2C potřebuje pouze signály SDA a SCL. Použitá klávesnice a displej jsou na obrázku 8. [24], [25]

Obrázek 8: použitý LCD displej a klávesnice [24], [25]

2.3.6 Komunikační rádio NRF24L01

Podle požadavku 1.4 by mělo dojít k sestrojení monitorovacího zařízení, které bude s ústřednou komunikovat bezdrátově. Pro tyto účely bylo vybráno rádio NRF24L01 viz obrázek 9. Jedná se o jednočipové rádio pracující na frekvenci 2,4 GHz se širokopásmovým paketovým protokolem *Enhanced ShockBurst™*. Modul je navržen pro nízkoodběrové aplikace. S Modulem lze komunikovat pomocí SPI sběrnice. Rádio používá FSK modulaci. Je možné nastavit amplitudu signálu, přenosovou rychlost a v malém rozsahu i frekvenci kanálu. [26]



Obrázek 9: NRF24L01 [26]

Základní vlastnosti rádia NRF24L01

- pracovní frekvence 2,4 GHz ISM, přenosová rychlost až 2 Mbps
- 2 nízkoodběrové módy, 900 nA při *Power down* a 22 μA při *Standby-I*
- 11,3mA odběr při 0 dBm výstupním zesílení ve stavu vysílání a 12,3 mA při přenosové rychlosti 2 Mbps ve stavu příjmu
- vestavěný napěťový regulátor
- napájecí napětí 1,9 V až 3,6 V a 5 V tolerantní vstupy
- protokol *Enhanced ShockBurst™* zahrnující automatické paketování a 6 data pipe *MultiCeiver™*
- ±60 ppm 16 MHz krystal
- pracovní teplota v rozsahu -40 °C až 85 °C

Pracovní režimy rádia

Vzhledem k tomu, že rádio má být použito v případě bateriově napájeného monitorovacího zařízení, je velmi důležitá nízká spotřeba. Následuje výčet čtyř hlavních režimů rádia NRF24L01.

- **Power down mode** je mód, ve kterém je maximální spotřeba 900 nA. Všechny registry jsou připraveny pro zápis a čtení přes SPI rozhraní. Do tohoto módu se přechází nastavením PWR_UP bitu do 0 nebo automaticky po resetu.
- **Standby mode** se dá ještě dělit na další dva módy *Standby-1* a *Standby-2*. *Standby-1* má maximální spotřebu 22 μA. Využívá se po ukončení odesílání, příjmu shozením CE bitu do nuly nebo po ukončení *Power down* módu kvůli snížení spotřeby. *Standby-2* je dočasný stav do kterého se přechází v případě, že se požaduje přechod do vysílacího *TX mode*, ale prozatím je prázdný zásobník pro odesílání. Je úspornější než *TX mode*. Odběr *Standby-2* režimu je 320 μA.
- **RX mode** demoduluje přijatý signál z rádiového kanálu a předává data širokopásmovému protokolu *Enhanced ShockBurst™*. Tento protokol se snaží najít správný paket. Pokud ho najde (sedí adresa a CRC), jsou data uložena do zásobníku pro příjem.
- **TX mode** je určen k odesílání dat. NRF24L01 zůstane v tomto módu, dokud neodešle aktuální paket. Poté, pokud se shodí vysílací bit CE, přejde do *Standby-1*. Pokud je ale vysílací bit CE pořád 1, tak v případě, že jsou další data v zásobníku pro odeslání, zahájí odesílání následujícího paketu, nebo když je zásobník prázdný, přejde do *Standby-2* módu.

Komunikační protokol Enhanced ShockBurst™

Tento protokol je založen na spojové vrstvě ISO/OSI modelu. Slouží k automatickému složení paketů, kontrole, časování, automatickému odpovídání, a k opakovanému posílání paketů. Na obrázku 10 je formát paketu/zprávy.

Obrázek 10: Formát Enhanced ShockBurst™ paketu

Preamble je střídavá bitová sekvence sloužící k synchronizaci vysílače s přijímačem. Preamble je buď 01010101 nebo 10101010, záleží na tom, jestli první bit adresy je 0 nebo

1. Adresa může být velká 3 až 5 bytů. Kontrolní pole paketu je velké 9 bitů. Obsahuje informaci o tom, jak je zpráva dlouhá, zda je paket nový či opětovně posílaný a NO_ACK bit, který říká, zda se používá funkce automatické odpovědi. Následuje vlastní zpráva, která může být dlouhá 0 až 32 bytů. Paket je zakončen Cyklickým redundantním součtem (CRC). Kontrolní součet slouží k odhalení chyby během přenosu dat. [26]

2.3.7 GSM modul

Podle požadavku 2.3 z tabulky 1 musí systém obsahovat GSM GPRS modul pro možné odesílání SMS zpráv. Pro tento účel byl vybrán modul SIM800L viz obrázek 11. Modul by měl být umístěn uvnitř zabezpečovací ústředny, ale další záložní modul by měl být součástí monitorovacího zařízení. GSM modul komunikuje přes UART piny, má jeden slot pro SIM kartu a dá se k němu připojit anténa. Modul je opatřen diagnostickou LED diodou, která pomalým nebo rychlým blikáním signalizuje, zda modul má či nemá signál. Napájecí napětí se pohybuje v rozmezí 3,4 V až 4,4 V. [27]

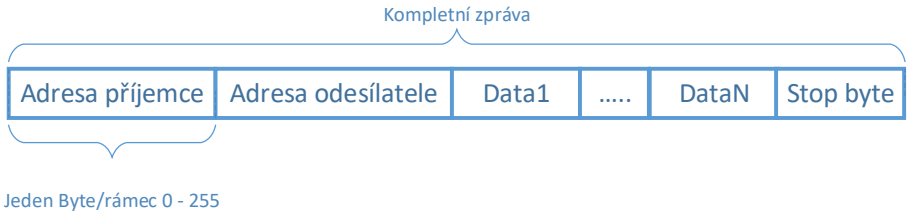
Obrázek 11: SIM800L [27]

2.4 Pravidla komunikace pro RS485

Jak již bylo zmíněno v předchozím textu, pro komunikaci mezi jednotlivými subsystémy slouží standard RS485. Výběr vhodného převodníku MAX485 byl již popsán v kapitole 2.3.3. Nyní je ale potřeba definovat si formát zpráv a pravidla komunikace. Tato kapitola se zabývá pouze obecnými pravidly. Konkrétní implementace aplikační vrstvy na úrovni subsystémů jsou podrobněji popsány v kapitolách 3.3.4 a 4.1.2.

Obecná pravidla

Délka jednoho odeslaného rámce přes RS485 je 8 bitů. Formát zprávy je na obrázku 12. Na první pozici ve zprávě se nachází rámec nesoucí informaci o adrese příjemce. Na druhé pozici je adresa odesílatele. Dále ve zprávě může figurovat různý počet datových rámců. Celá zpráva je zakončena stop bytem.



Obrázek 12: Formát zprávy

RS485 je master/slave standard, a proto má ústředna roli mastera a v určitém intervalu posílá dotazy podřízeným prvkům. Ty pak posílají odpovědi zpátky ústředně. V následujících kapitolách je definován význam datových rámců podle toho, jaký subsystém zprávu posílá.

Data odesílaná ústřednou

Ústředna posílá zprávu dlouhou celkem 5 bytů, z toho posílá dva datové rámce. Pokud ústředna posílá data ovládacímu panelu, je význam jednotlivých rámců následující:

- **Data1**: informace o stavu ústředny (podrobné vysvětlení všech stavů se nachází v kapitole 3.3.2)
- **Data2**: čas pro odpočet (synchronizování času odpočtu na straně ústředny i panelu)

Pokud data posílá ústředna senzorům, jsou datové rámce nulové, není totiž potřeba, aby ústředna předávala senzoru nějaké konkrétní informace. Zpráva slouží pouze k vybídnutí senzoru, aby poslal svá data.

Data odesílaná ovládacím panelem

Ovládací panel odesílá data ústředně vždy jako součást odpovědi na přijatou zprávu. Zpráva obsahuje 7 rámců, z čehož 4 jsou datové. Význam datových rámců je následující:

- **Data1**: 1. polovina hesla (v případě požadavku ze strany panelu) nebo 0 (v případě tzv. *heart-beat* zprávy)
- **Data2**: 2. polovina hesla (v případě požadavku ze strany panelu) nebo 0 (v případě *heart-beat* zprávy)
- **Data3**: požadavek ovládacího panelu (podrobněji definováno v kapitole 4.2.2)
- **Data4**: chybový status (podrobněji kapitola Chybové statusy sběrnicových prvků)

Data odesílaná senzorem

Senzor obdobně jako panel posílá data v odpovědi na zprávu od ústředny. Zpráva je dlouhá 5 rámců, z toho 2 rámce jsou datové.

- **Data1**: informace o senzoru, pokud je nulový nedošlo k narušení, pokud ne – 0. bit sepnutí PIR senzoru, 1. bit – rozepnutí magnetického kontaktu
- **Data2**: chybový status (podrobněji kapitola Chybové statusy sběrnicových prvků)

Chybové statusy sběrnicových prvků

Na straně ovládacího panelu a senzoru probíhá určitá kontrola komunikace. Informaci o chybách pak příslušný prvek odešle jako odpověď na následující správně přijatou zprávu od ústředny. Význam jednotlivých bitů je následující.

- 0. bit: RS485 *Serial overflow*, přetečení zásobníku 64 B
- 1. bit: delší zpráva, než je programově omezená velikost zásobníku pro příjem
- 2. bit: kontrola prázdného pole před příjmem zprávy vrátila chybu, problém s pamětí
- 3. bit: zpráva má správnou adresu příjemce, ale nesedí adresa odesílatele
- 4. bit: zpráva má správné adresy odesílatele i příjemce, ale nesedí stop byte
- 5. bit: zpráva má správné adresy odesílatele i příjemce, stop byte, ale je delší
- nulový status: žádná chyba

Chybové statusy zpráv na straně ústředny

Z důvodu kontroly odesílání a přijmu zpráv byl vytvořen i na straně ústředny status, který je společně s odpovídající zprávou ukládán do databáze pod názvem *Status_zpravy*. Význam jednotlivých bitů je následující.

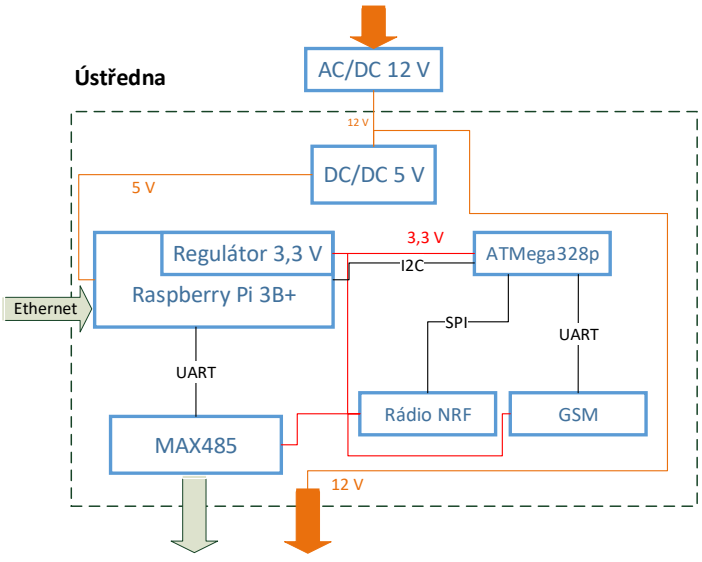
- 0. bit: time out při čekání na odpověď
- 1.bit: nevyčtená data v zásobníku pro příjem (zpráva delší, než je očekávaná délka zprávy)
- 2. bit: krátká zpráva
- 3. bit: neodpovídající stop byte
- 4. bit: chyba při odeslání dat do RS485 převodníku metodou *ser.write()*
- nulový status: žádná chyba

3 Návrh a realizace Řídící ústředny

Následující kapitola popisuje podrobný návrh ústředny od hardwarové realizace, propojení jednotlivých prvků uvnitř ústředny, až po vývoj softwaru.

3.1 Návrh a sestrojení HW části ústředny

Výběr vhodných komponent byl popsán již v kapitole 2.3. Na obrázku 13 je navržené blokové schéma ústředny. Jediným vnějším blokem je napěťový zdroj, který dodává na výstupu stejnosměrné napětí 12 V/60 W. Toto napětí je dále přivedeno do ústředny, a také je, společně s datovými signály, distribuováno dál pro napájení všech prvků systému tak, aby byl splněn požadavek 2.2 z tabulky 1. V ústředně je umístěn DC/DC měnič LM2596, jehož úkolem je snížit napětí na 5 V pro napájení Raspberry Pi. Vzhledem k tomu, že vnitřní logika na portech GPIO Raspberry Pi je 3,3 V, tak jsou i zbylé prvky uvnitř ústředny napájeny 3,3 V. Toto napětí dodává regulátor NCP1117, integrovaný v počítači Raspberry Pi. Pokud by došlo k přivedení 5 V na GPIO vstupy Raspberry Pi, mohlo by dojít ke zničení počítače.

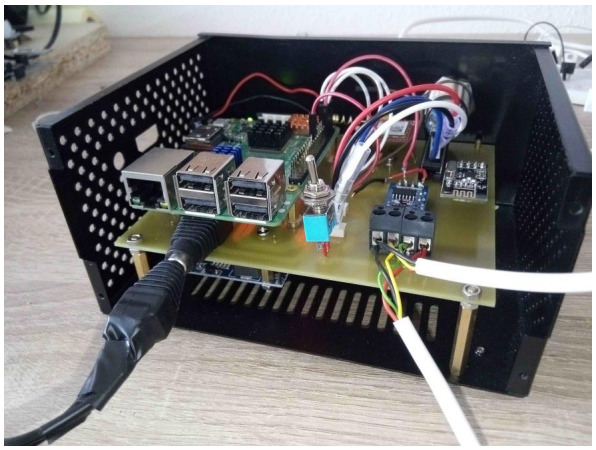


Obrázek 13: Blokové schéma ústředny

Jak již bylo zmíněno, jako počítač ústředny bylo vybráno Raspberry Pi. V Raspberry Pi bude implementována databáze pro sběr dat ze senzorů, hlavní řídící program, grafické rozhraní a hlavní komunikační program pro vyčítání dat ze senzorů. V hotové aplikaci bude zajištěno internetové připojení přes Ethernet, nicméně Raspberry Pi 3B+ má i Wifi, které bylo během vývoje využíváno častěji. Raspberry Pi je přes UART propojeno s převodníkem MAX485 a přes I2C sběrnici s mikrokontrolérem ATMega328p. Mikrokontroléer ATMega328p má na starost odesílání SMS zpráv přes GSM modul a komunikaci

s monitorovacím zařízením přes rádio NRF24L01. K mikrokontroléru ATMega328p je připojen 8 MHz oscilátor.

Na obrázku 14 je osazená ústředna. Schéma zapojení, návrh DPS a fotografická dokumentace z osazování ústředny jsou v příloze.



Obrázek 14: Osazená ústředna

3.2 Základní nastavení Raspberry Pi

Před prvním spuštěním počítače bylo potřeba připojit k Raspberry Pi monitor, myš a klávesnici a nahrát operační systém na SD kartu. Pro Raspberry Pi existuje více Linuxových distribucí, ale pro účely této práce byla vybrána distribuce Raspbian. Tato distribuce vychází z Debian a je určená přímo pro Raspberry Pi.

Následně, po instalaci systému na SD kartu a prvním spuštění, bylo třeba nastavit počítač tak, aby se automaticky připojoval k internetu. Připojení k Wifi se definuje v souboru */etc/wpa_supplicant/wpa_supplicant.conf*. Dále se nastavila připojení přes vzdálenou plochu. [18]

3.2.1 Instalace webového serveru a databáze MySQL

MySQL je *open source* databáze, která pro práci s daty používá standardizovaný strukturovaný dotazovací jazyk SQL. Do databáze se budou ukládat data ze senzorů a data potřebná pro fungování celého systému. Před instalací samotné databáze se musí nainstalovat HTTP webový server Apache2 a PHP jazyk, se kterými MySQL pracuje, příkazy:

```
sudo apt-get install -t stretch apache2
sudo apt-get install -t stretch php7.0
```

Následujícím příkazem se nainstaluje MySQL server a client, a také podpora MySQL pro PHP:

sudo apt-get install mysql-server mysql-client php7.0-mysql

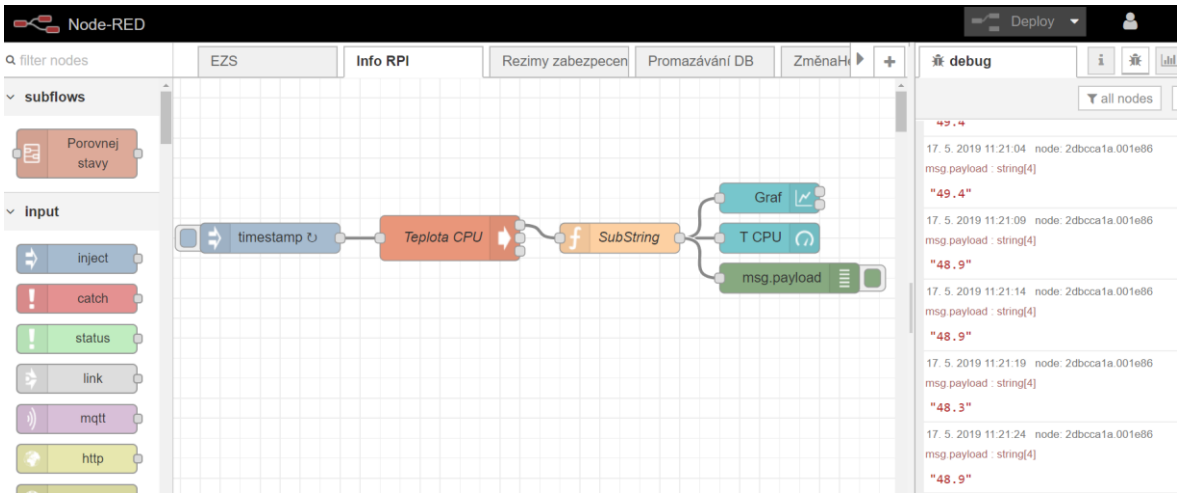
Posledním, ale neméně důležitým balíkem je phpMyAdmin. Jedná se o nástroj napsaný v jazyce PHP umožňující správu obsahu databáze MySQL. K instalaci slouží příkaz:

sudo apt-get install phpmyadmin

Během instalace se musí phpMyAdmin nakonfigurovat. Jako web server se vybere apache2 a nastaví se heslo pro přihlašování. Dále bylo potřeba přidat všechna práva uživateli root. [28]

3.2.2 Node-RED

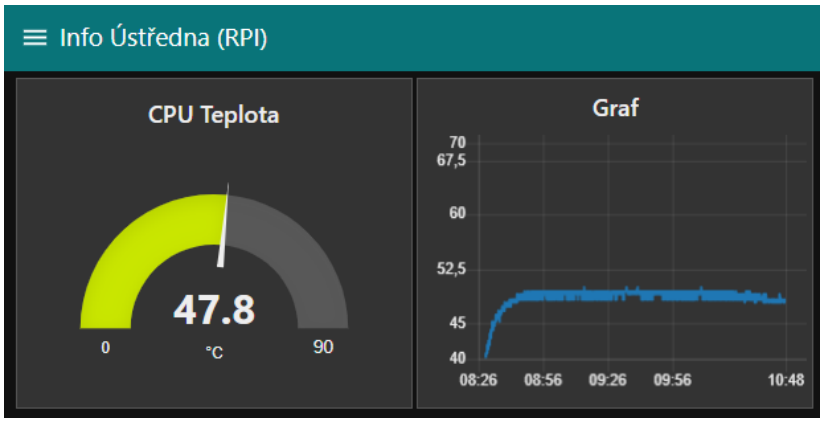
Node-RED je programovací nástroj vytvořený speciálně pro Internet Věcí a slouží k propojování zařízení, rozhraní pro programování zařízení (APIs) a online služeb. Node-RED byl již součástí nainstalovaných balíků, a tudíž nemusel být dodatečně instalován. Editor programu je na obrázku 15. a je k zobrazení v prohlížeči na adrese *lokální_adresa:1880*. V Node-RED byl naprogramován hlavní stavový automat řídící ústředny a také uživatelské grafické rozhraní. [29]



Obrázek 15: Node-RED editor

V editoru na paletě vlevo jsou tzv. *node* typy neboli typy uzlů, ze kterých se skládají jednotlivé sekvence kódu. V prostřední části obrázku je zobrazen list neboli *flow* obsahující jednu sekvenci, která má za úkol v určitém časovém intervalu vyčítat teplotu procesoru Raspberry Pi a vykreslovat ji do grafického prostředí a do *debug* okna. Celkem intuitivně je z obrázku patrný význam jednotlivých uzlů. První uzel typu *inject* má za úkol v určitém

cyklu spouštět sekvenci. Druhý uzel typu *exec* slouží k zadání příkazu do Linuxového terminálu, který vrátí informaci o teplotě jádra. Třetí uzel je typu *function* a slouží k napsání libovolné JavaScript funkce. V tomto případě má za úkol upravit textový vstup z předchozího uzlu na číslo. Číselná hodnota se dále větví do tří dalších uzlů. Zelený uzel je typu *debug* a má za úkol vypsat teplotu do *debug* okna. Modré uzly jsou z balíčku *dashboard* a mají za úkol vypsat hodnotu do grafického rozhraní. V rámci jedné sekvence se data mezi uzly předávají v základní proměnné typu *message object* (msg), Tato proměnná existuje pouze v rámci konkrétní sekvence. Grafické prostředí se obvykle nachází na adrese *lokální_adresa:1880/ui,* ale v případě této diplomové práce byl název rozhraní změněn na *lokální_adresa:1880/ezs*. Ukázka grafického rozhraní je na obrázku 16.



Obrázek 16: Node-RED grafické uživatelské rozhraní

Balíčky uzlů

Node-RED v základu obsahuje omezený počet typů uzlů, ale přes *Manage pallete* nebo zadáním příkazu *npm i název-balíčku* do terminálu v uživatelském Node-RED adresáři (obvykle *~/.node-red*) se dají doinstalovat další balíčky. Zmiňovaná sekvence pro zobrazení teploty obsahuje uzly sloužící pro vykreslování informací do grafického prostředí. Tyto uzly byly přidány až doinstalováním balíčku *dashboard*. Dále byl doinstalován balíček *contrib-i2c* pro komunikaci se zařízeními přes I2C rozhraní, a také velmi důležitý balíček *mysql* pro práci s MySQL databází. [29]

Automatické spouštění

Vzhledem k využití systému je žádoucí, aby byl hlavní program v Node-RED automaticky spuštěn po jakémkoliv restartu ústředny. Pro tento účel slouží příkaz:

```
sudo systemctl enable nodered.service
```

Zabezpečení Node-RED

Node-RED v základu nemá nijak zabezpečený přístup do editoru ani do grafického uživatelského rozhraní. Vzhledem k tomu, že přístup bez autorizace je nežádoucí, musel být Node-RED zabezpečen přístupovým heslem. [29]

Pro nastavení přihlašovacích údajů se musí upravit soubor *settings.js*, který se nachází v uživatelském Node-RED adresáři. Přidáním *adminAuth* dojde k nastavení přihlašovacích údajů pro editor. Přidáním *httpNodeAuth* se nastaví přihlašování do grafického rozhraní. Úprava souboru *settings.js* je na obrázku 17. Po instalaci balíčku *admin* byla zavolána kryptografická hashovací funkce, která vytvoří hash zadaného hesla. Hashovaná verze hesla byla následně přidána do *adminAuth* i *httpNodeAuth*. Příkaz pro vytvoření hashované verze hesla: *sudo node-red-admin hash-pw*. Z obsahu souboru *settings.js* díky tomu není patrné originální nemaskované heslo, které se ve skutečnosti používá pro přihlášení do systému.

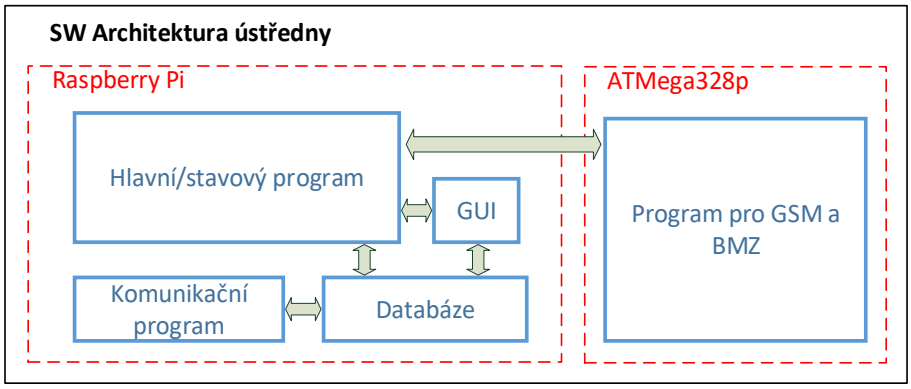
```
adminAuth: {
    type: "credentials",
    users: [{
        username: "admin",
        password: "$2a$08$uDU6JlvUalWLhMRIkmEzb.eCzcFwhKr0uSzietQ0BlnTxRuItzXLu",
        permissions: "*"
    }]
},

// To password protect the node-defined HTTP endpoints (httpNodeRoot), or
// the static content (httpStatic), the following properties can be used.
// The pass field is a bcrypt hash of the password.
// See http://nodered.org/docs/security.html#generating-the-password-hash
httpNodeAuth: {user:"user",pass:"$2a$08$uDU6JlvUalWLhMRIkmEzb.eCzcFwhKr0uSzietQ0BlnTxRuItzXLu"},
//httpStaticAuth: {user:"user",pass:"$2a$08$zZWtXTja0fB1pzD4sHCMyOCMYz2Z6dNbM6tl8sJogENOMcxWV9DN."},
```

Obrázek 17: Nastavení hesel Node-RED editoru a grafického rozhraní

3.3 Návrh a vývoj SW části ústředny

Použitím počítače Raspberry Pi a ATMega328p v jednom celku se nabízí celkem široké spektrum softwarových nástrojů k docílení komplexní funkce ústředny. Na obrázku 18 je zobrazena softwarová architektura řídící ústředny, která se skládá z pěti hlavních komponent.



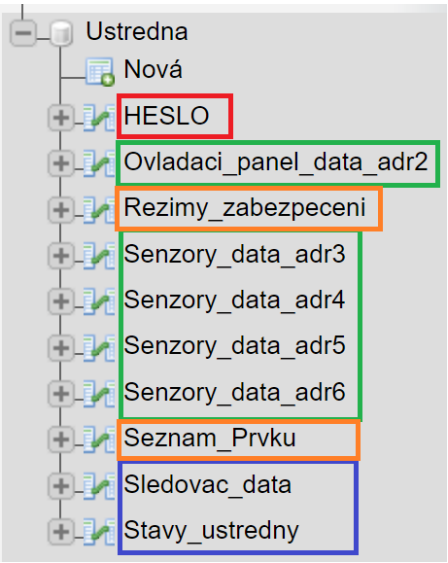
Obrázek 18: SW architektura zabezpečovací ústředny

- **Hlavní/stavový program** se především stará o přechody mezi jednotlivými stavy ústředny. Využívá při tom informace z databáze, a také do ní zapisuje. Posílá a přijímá data do/z grafického rozhraní. Předává si data s komponentou starající se o odesílání zpráv přes GSM a komunikaci s BMZ. Hlavní program byl napsán v Node-RED a je podrobně popsán v kapitole 3.3.2.
- **Databáze** má za úkol ukládání dat ze všech senzorů a ovládacích panelů, a také se do ní zapisují informace o stavech ústředny, uživatelská nastavení z GUI a informace z BMZ. Podrobněji se této problematice věnuje kapitola 3.3.1.
- **GUI** neboli grafické uživatelské rozhraní je komponenta, která nabízí uživateli podrobnější informace o systému než ovládací panel. Zároveň dává uživateli možnost řídit systém a měnit některé parametry systému. GUI je vytvořeno pomocí *dashboard* uzlů v Node-RED. Podrobněji se tvorbou grafického rozhraní zabývá kapitola 3.3.3.
- **Komunikační program** implementuje aplikační vrstvu pro standard RS485, která zajišťuje komunikaci s externími prvky a interaguje při tom s databází. Program byl napsán v Pythonu a podrobněji se mu věnuje kapitola 3.3.4.
- **Program pro GSM a BMZ** je jedinou softwarovou komponentou, která není spouštěna na Raspberry Pi, ale na mikrokontroléru ATMega328P. Jejím úkolem je odesílat zprávy přes GSM modul a komunikovat bezdrátově s BMZ. Tento program byl napsán v jazyku Wiring a podrobněji se mu věnuje kapitola 3.3.5.

3.3.1 MySQL databáze

V úvodní kapitole 3.3 bylo popsáno pět hlavních komponent, ze kterých se skládá kompletní software ústředny. Databáze je jednou z těchto komponent a je popisována jako první, protože s ní zbylé komponenty, ať už přímo či nepřímo, pracují. Navíc je klíčová pro pochopení filozofie ostatních komponent. Použitím databáze je částečně splněn požadavek 2.5. z tabulky 1.

Pro účely diplomové práce byla vytvořena databáze *Ustredna*. Tabulky uložené v databázi by se daly rozdělit do čtyř hlavních kategorií. Na obrázku 19 jsou tyto kategorie barevně rozlišeny.



Obrázek 19: Struktura vytvořené MySQL databáze

Do první kategorie spadá pouze tabulka *HESLO*, která obsahuje hashovanou verzi hesla pro zastřežení či odstřežení systému.

Tabulky zvýrazněné oranžovou barvou určují vlastnosti, či rozložení systému. Konkrétně tabulka *Seznam_Prvku* obsahuje informace o všech prvcích, se kterými ústředna komunikuje. Včetně adresy a informace, zda se jedná o senzor, ovládací panel, či hlásič. S informacemi z této tabulky na straně ústředny pracuje skript, sloužící ke komunikaci s ostatními prvky i hlavní program v Node-RED. Důležité je, aby každý prvek měl unikátní identifikátor. Tabulka *Seznam_Prvku* je na obrázku 20. Práce s tabulkou je dále zmíněna v kapitolách o hlavním programu a o komunikačním programu 3.3.2 a 3.3.4.

ID	Ovladaci_panel	Hlasic	Senzor_PIR	Senzor_MG	Popis	Tabulka 1	Maska
2	1	0	0	0	Ovladaci panel	Ovladaci_panel_data_adr2	
3	0	0	1	1	Univerzalni senzor ADR 3	Senzory_data_adr3	ADR3
4	0	0	1	1	Univerzalni senzor ADR4	Senzory_data_adr4	ADR4
5	0	0	1	1	Univerzalni senzor ADR5	Senzory_data_adr5	ADR5
6	0	0	1	0	PIR senzor ADR6	Senzory_data_adr6	ADR6

Obrázek 20: Struktura tabulky Seznam_Prvku

Do stejné kategorie patří i tabulka *Rezimy_zabezpeceni*. Tabulka se dá měnit uživatelem skrz Node-Red grafické uživatelské prostředí a jsou v ní definovány jednotlivé režimy zabezpečení. Například pokud dojde k zastřežení oblasti 0 (uživatelsky se jedná o režim noc), berou se v potaz pouze senzory, u kterých je v tabulce hodnota 1. Z obrázku 21 vyplývá, že se v tomto konkrétním nastavení jedná o senzory s adresou 5 a 6, zatímco senzory s adresou 3 a 4 jsou ignorovány. Práce s tabulkou je dále zmíněna v kapitolách o hlavním programu a o grafickém rozhraní 3.3.2 a 3.3.3.

Nazev_Rezimu	ADR3	ADR4	ADR5	ADR6
Oblast 0	0	0	1	1
Oblast 1	1	1	1	1
Oblast 2	0	0	0	0

Obrázek 21: Struktura tabulky Rezimy_Zabezpeceni

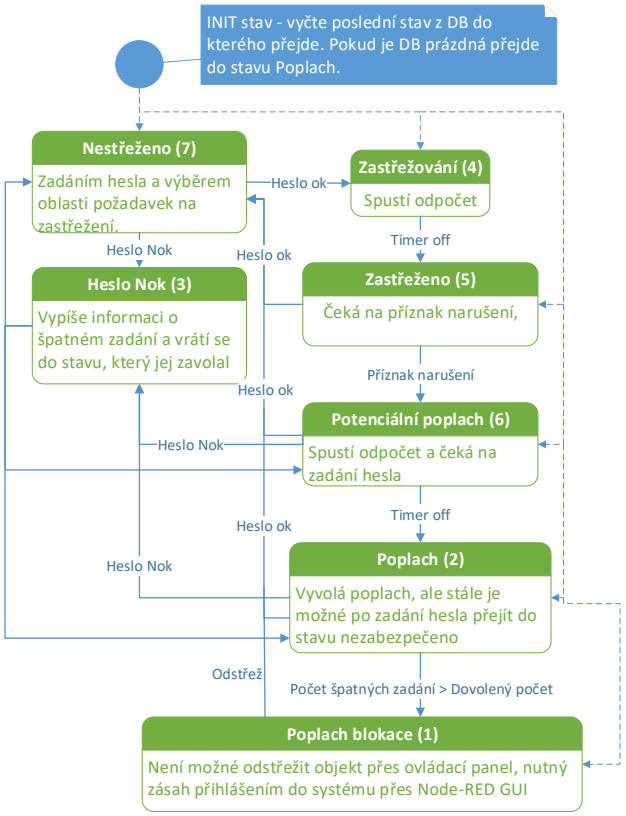
Kategorie označená zeleně čítá nejvíce tabulek. Každá z tabulek je vytvořena komunikačním skriptem a název tabulky je definován sloupcem *Tabulka* v tabulce *Seznam_Prvku* z MySQL databáze *Ustredna* viz obrázek 20. Do těchto tabulek se ukládají informace z jednotlivých prvků. Do tabulek zapisuje komunikační skript a data vyčítá Node-RED. Zápis je podrobněji popsán v kapitole 3.3.4. Práce s daty pak v kapitole 3.3.2.

Modře označená kategorie je velmi podobná zelené kategorii, neboť se do ní také ukládají určitá data v čase. Rozdíl je v zapisovateli. Zatímco do tabulek zelené kategorie jsou zapisována data komunikačním skriptem, do tabulek modré kategorie zapisuje hlavní program. V tabulce *Stavy_Ustredny* jsou zapsány stavy, ve kterých se ústředna nacházela, včetně stavu aktuálního. V tabulce *Sledovac_data* je pouze informace o komunikaci s monitorovacím zařízením.

3.3.2 Hlavní/stavový program ústředny

Stavový automat

V počáteční vývojové fázi došlo nejprve k definování stavového automatu ústředny, a až poté se přešlo na samotné programování v Node-RED. Na obrázku 22 je zjednodušený stavový automat ústředny, jehož cílem je ukázat především reakce na podněty od senzorů, ovládacího panelu a následné přechody mezi vnitřními stavy. Celkem bylo definováno sedm hlavních stavů a jeden inicializační. Během Inicializace se vyčítá z databáze z tabulky *Stavy_Ustredny* poslední známý stav ústředny, do kterého se přejde. Pokud je databáze prázdná nebo se nepodaří vyčíst poslední stav, přechází se z bezpečnostních důvodů do stavu poplach.



Obrázek 22: Stavový automat ústředny

Node-RED

Úvod do používání Node-RED byl již popsán v kapitole 3.2.2. Nyní bude popsán hlavní smysl důležitých sekvencí.

Inicializační sekvence tvoří první pomyslnou kategorii. Patří do ní sekvence pro vyčítání seznamu prvků z databáze z tabulky *Seznam_Prvku*, sekvence pro zjištění hesla z tabulky *HESLO* nebo sekvence pro zjištění posledního známého stavu ústředny z tabulky *Stavy_Ustredny*. Poslední zmiňovaná sekvence je na obrázku 23.

Obrázek 23: Node-RED inicializační sekvence stavu ústředny

Uzel pojmenovaný *Získej stav querry* připraví SQL dotaz, který je následně v uzlu *Ustredna INIT* odeslán do databáze. Návratovou hodnotou z databáze je poslední řádek tabulky *Stavy_Ustredny*. Uzel Získej poslední stav mimo jiné zpracuje data z databáze a vytvoří přes metodu *flow.set()* proměnou, ve které je uložena číselná informace o stavu ústředny. Číselný stav ústředny koresponduje s čísly jednotlivých stavů z obrázku 22.

Flow proměnné fungují tak, že přes metody *flow.get()* a *flow.set()* k nim má přístup jakýkoliv uzel (*node*) z libovolné sekvence v rámci jednoho programového listu (*flow*). Dále existují ještě proměnné typu global, které jsou přístupné pro jakýkoliv uzel v jakémkoliv listu, ale tyto proměnné nebyly při programování použity. Zmiňované inicializační sekvence definují přes metodu *flow.set()* všechny flow proměnné, se kterými následně pracují cyklické sekvence.

Druhou velkou skupinou sekvencí jsou cyklické sekvence, které by se daly dělit do dalších podskupin. Jednou významnou podskupinou jsou sekvence, které vyčítají informace z tabulek, do kterých se zapisují přijatá data ze senzorů a ovládacích panelů. Tyto sekvence jsou poměrně rozsáhlé. Mají na starost například kontrolu, že do databáze stále přicházejí nové zprávy. Kontrolují statusy zpráv. V případě ovládacího panelu zpracovávají požadavky a mohou zajistit přechod do jiného vnitřního stavu automatu. V případě senzorů vytváří příznaky sepnutí PIR, Magnetického kontaktu či špatného statusu atp.

Důležitou cyklickou sekvencí je sekvence pro zjištění změny stavu. Tato sekvence každých 500 milisekund kontroluje, zda došlo ke změně stavu. Pokud ano, zapíše informaci o změně do databáze do tabulky *Stavy_Ustredny*. V případě, že došlo k vyvolání poplachu, zajistí odeslání emailu, čímž je částečně splněn požadavek 2.3. z tabulky 1.

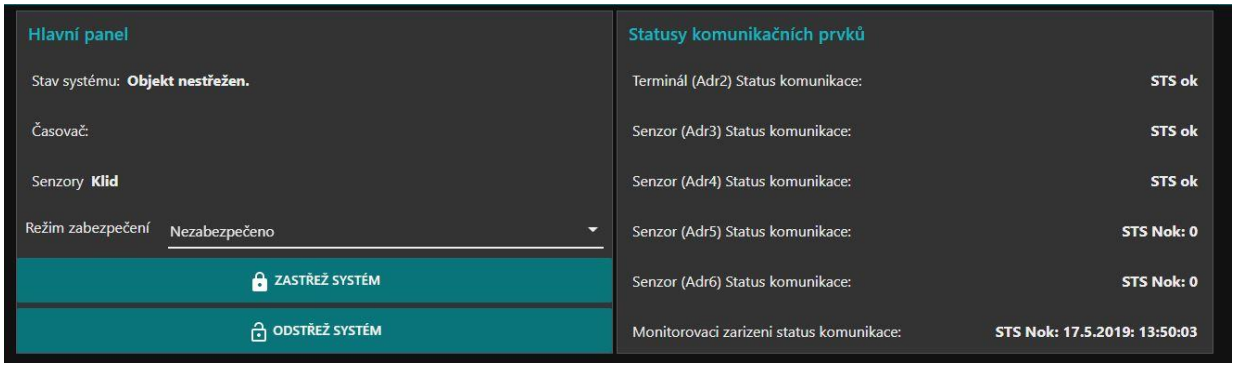
Další nezbytnou sekvencí je reakce na příznaky sepnutí senzorů ve stavu zabezpečeno. Sekvence má při splnění několika podmínek za úkol přejít do stavu potenciálního poplachu. První podmínkou je, že systém musí být ve stavu zabezpečeno. Dále se z databáze vyčítá tabulka *Rezimy_Zabezpeceni*, která, jak již bylo popsáno v kapitole 3.3.1, obsahuje informace o relevanci senzorů pro konkrétní režim. Následně se zjistí režim zabezpečení, ve kterém se systém nachází. V posledním uzlu je napsána funkce, která musí nejprve odfiltrovat irelevantní prvky podle aktuálního režimu zabezpečení, dále musí odfiltrovat prvky, které nejsou v tabulce *Seznam_Prvku* označeny jako senzor (například ovládací panel). Na závěr tento node zkontroluje, zda byly nastaveny příznaky sepnutí relevantních senzorů, na základě čehož může změnit stav ústředny na potenciální poplach.

Mezi další cyklické sekvence patří čítač pro odpočítávání, který se spouští ve stavu potenciálního poplachu či při zastřežování, anebo sekvence starající se o promazávání databáze. Poslední cyklická sekvence má na starost I2C komunikaci s komponentou, starající se o posílání zpráv přes GSM modul a komunikaci s BMZ. Tato sekvence následně ukládá informaci o BMZ do databáze do tabulky *Sledovac_data*.

3.3.3 Grafické uživatelské rozhraní GUI

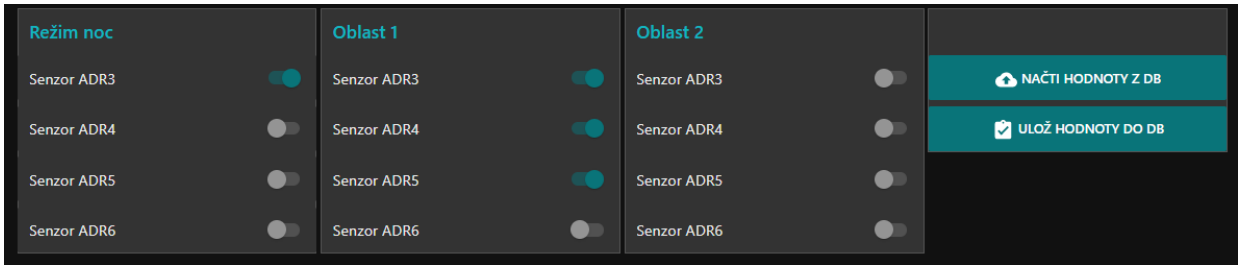
Na základě požadavku 3.3 z tabulky 1 byla vytvořena grafická nadstavba v Node-RED. Veškeré uzly zajišťující vypisování do grafického prostředí pochází z balíčku *dashboard*. Podobně jako u hlavního programu, byly i zde vytvořeny cyklické sekvence, které se starají o vykreslování, anebo byly uzly typu *dashboard* přímo implementovány v některé dříve zmíněné sekvenci v hlavním programu. Pro účely diplomové práce byly vytvořeny tři grafické záložky *EZS*, *Režimy zabezpečení* a *Historie*.

Na obrázku 24 je zobrazena první záložka *EZS*, která obsahuje základní informace o stavu systému, umožňuje uživateli zastřežit systém ve vybraném režimu a ukazuje statusy komunikačních prvků. Statusy říkají, zda je komunikace s daným prvkem v pořádku. Například zde je situace taková, že jsou do systému připojeny pouze dva senzory a jeden ovládací panel, proto jsou zbylé statusy senzorů ve stavu STS Nok:0. Nula znamená, že nikdy nepřišla odpověď od zmíněného prvku. Pokud je za statusem STS Nok uveden datum a čas, informuje to uživatele o tom, kdy přišla poslední odpověď od daného zařízení.



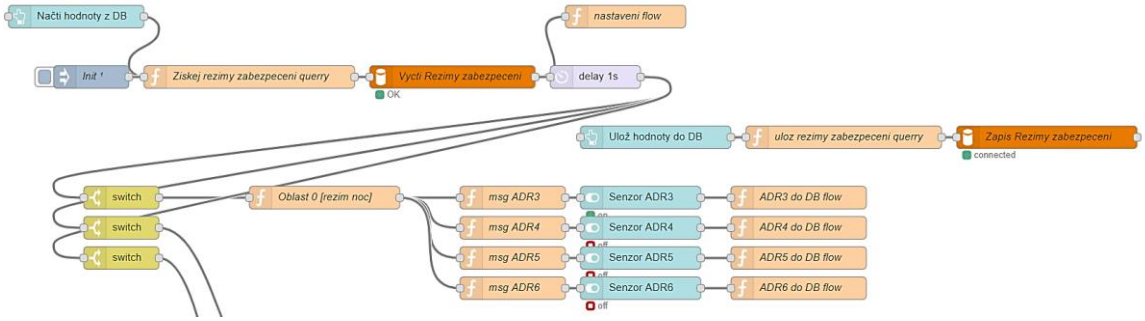
Obrázek 24: GUI, karta EZS

Další karta se jmenuje *Režimy zabezpečení* viz obrázek 25. Na této kartě si může uživatel sám nastavit, které senzory se berou v potaz při konkrétním režimu zabezpečení. Provedenou změnu konfigurace musí uživatel potvrdit tlačítkem *Ulož hodnoty do DB*, které zajistí přepsání hodnot v tabulce *Rezimy_Zabezpeceni*.



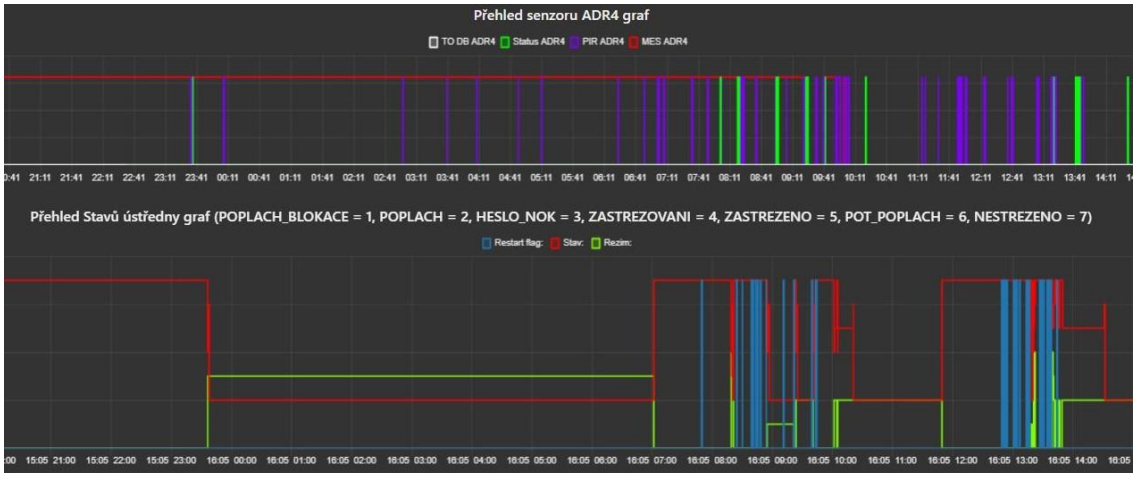
Obrázek 25: GUI, karta Režimy zabezpečení

Na obrázku 26 je ukázka sekvence, která implementuje zmíněnou funkcionalitu pro jeden konkrétní režim. Obdobně vypadá část sekvence pro ostatní oblasti. Primárně není možné změnit režim *Celý dům*, tento režim automaticky bere v potaz všechny senzory, které jsou součástí tabulky *Seznam_Prvku*.



Obrázek 26: Ukázka sekvence starající se o nastavení režimů zabezpečení

Poslední karta s názvem *Historie* je na obrázku 27. Nahoře ukazuje historii vybraného senzoru. Konkrétně informaci o sepnutí PIR, sepnutí Magnetického kontaktu (MES), statusu komunikace a statusu TO DB, který je nahozen v případě, že do databáze nepřicházejí nové zprávy (problém s databází, či pokud není zapnutý komunikační program). Dole je k vidění historie stavů ústředny. Tento graf uživateli dává informaci o minulých stavech ústředny, o vybraných režimech zabezpečení a o restartování Node-RED programu (tento příznak je nahozen i při přeložení programu).

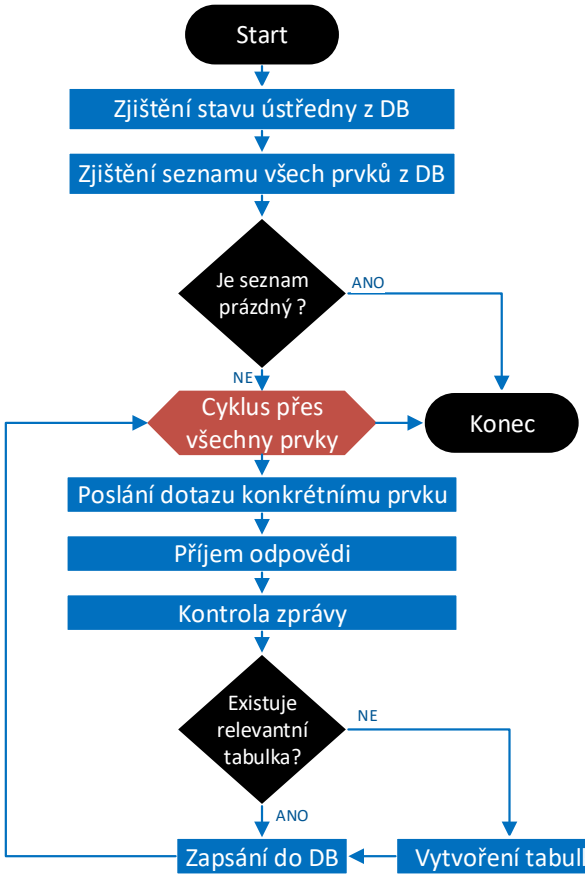


Obrázek 27: GUI, karta Historie

3.3.4 Komunikační program ústředny

V předchozích kapitolách byly již popsány komponenty, které pracují s konkrétními daty jednotlivých prvků uložených v MySQL databázi. V této kapitole je vysvětleno, jakým způsobem se tato data získávají a zapisují do databáze.

Komunikační software byl napsán v jazyce Python a Raspberry Pi ho spouští na pozadí. Na obrázku 28 je nakreslen vývojový diagram skriptu. Celý skript je součástí Přiloženého CD.



Obrázek 28: Vývojový diagram komunikačního programu

Ještě před spuštěním hlavní smyčky dochází k importu potřebných knihoven *time*, *serial* a *pymysql*. Dále se definují konstanty, globální proměnné a inicializuje se sériová komunikace.

V hlavní smyčce se nejprve z databáze *Ustredna* postupným zadáváním SQL dotazů vyčte z tabulky *Stavy_Ustredny* aktuální stav ústředny. Posléze se vyčte celý obsah tabulky *Seznam_Prvku*, která obsahuje všechny podstatné informace o jednotlivých prvcích (viz obrázek 20). Získaná data se uloží do polí. V obou případech je kód umístěn v *try – except* blocích, které zareagují na jakoukoliv výjimku při práci s databází. Následuje kontrola, zda je seznam prvků prázdný. Tato situace může nastat, pokud opravdu v příslušné tabulce není žádný prvek, ale také když je při práci s databázi vyvolána výjimka. Je-li tato podmínka splněna, program dojde na konec hlavní smyčky. Pokud seznam prvků není prázdný, pokračuje se dál.

Následuje cyklus, který má zajistit postupnou komunikaci se všemi zařízeními. Nejprve se pošle zpráva danému prvku podle toho, o jaký prvek se jedná v souladu s pravidly komunikace definovanými v odstavci Data odesílaná ústřednou v kapitole 2.4. Následně se přechází do příjmu a čeká se na odpověď od dotazovaného prvku. Na závěr se provede kontrola přijatých dat, anebo se rovnou nahodí *time out*, když nepřijde žádná odpověď. Tato kontrola používá status, který se v posledním kroku zapíše do databáze. Podoba statusu je definována v odstavci Chybové statusy zpráv na straně ústředny.

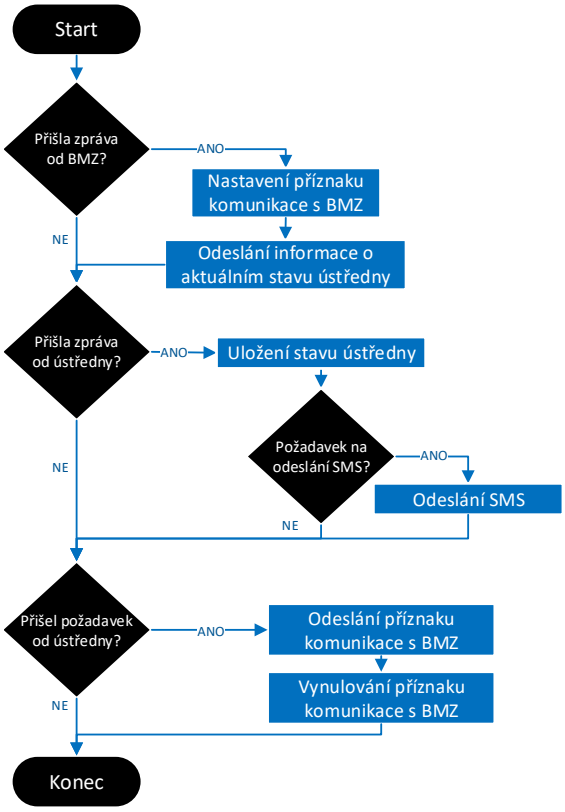
Na konci cyklu se před zápisem do databáze kontroluje, zda příslušná tabulka v databázi vůbec existuje. Správný název databáze je znám, neboť je součástí tabulky *Seznam_Prvku.* Pokud je tato tabulka v databázi již vytvořena, zapíše se do ní. Pokud neexistuje, dojde k jejímu vytvoření a zápisu. Veškerá práce s databází je opět umístěna v *try – except* blocích.

Z předchozích odstavců mimo jiné vyplývá, že aktuální verze komunikačního programu je schopná automaticky reagovat na změnu počtu prvků v systému, a to pouze na základě přidání či odebrání prvků v tabulce *Seznam_Prvku*.

Podobně jako hlavní program, i komunikační program musí být automaticky spouštěn při zapnutí systému. Existuje mnoho způsobů, jak toho docílit. Skript se dá volat během bootovací sekvence například přidáním příkazu do souboru *rc.local*. Pro účely diplomové práce byl ale zvolen jednodušší způsob, a to spouštění skriptu během inicializační fáze Node-RED. Byla napsána inicializační sekvence obsahující uzel typu *exec*, který zadá příkaz na spuštění python skriptu do terminálu.

3.3.5 Program pro práci s GSM modulem a komunikaci s BMZ

Poslední softwarovou komponentou je program starající se o obsluhu GSM modulu a o komunikaci s bezdrátovým monitorovacím zařízením. Program byl napsaný v jazyce Wiring a je nahraný v mikrokontroléru ATMega328p, který je s Raspberry Pi propojený přes I2C sběrnici. Tento program obsluhou GSM modulu pro odesílání SMS částečně splňuje požadavek 2.3 z tabulky 1. Na obrázku 29 je vývojový diagram programu.



Obrázek 29: Vývojový diagram programu pro práci s GSM modulem a komunikaci s BMZ

Před spuštěním inicializační funkce a hlavní smyčky dochází k importu potřebných knihoven *Wire* pro komunikaci přes I2C sběrnici, *SPI* pro komunikaci s rádiem NRF a *SoftwareSerial* pro ovládání GSM modulu. V *setup()* funkci dochází k inicializaci komunikace s GSM modulem, nastavení I2C adresy a nastavení režimu rádia NRF pro příjem.

V hlavní smyčce se nejprve kontroluje, zda má rádio NRF v zásobníku přijatou zprávu od BMZ. Pokud ano, nastaví se příznak komunikace s BMZ a v odpovědi se přes rádio odešle informace o stavu ústředny.

Dále se sleduje, zda přišla zpráva od hlavního programu přes I2C sběrnici. Je-li tomu tak, uloží se stav ústředny, a pokud přišel požadavek na odeslání SMS, dojde k odeslání zprávy. Tento požadavek přijde v případě, že byl vyvolán poplach, nebo když došlo k restartu ústředny.

Na závěr se kontroluje, zda si hlavní program přes I2C sběrnici žádá zprávu. Pokud ano, odešle se informace o příznaku komunikace s BMZ zařízením. Když se stane, že není v pravidelném cyklu navázána komunikace s BMZ, je tento příznak nulový a hlavní program pak tuto informaci vyhodnotí jako *time out*.

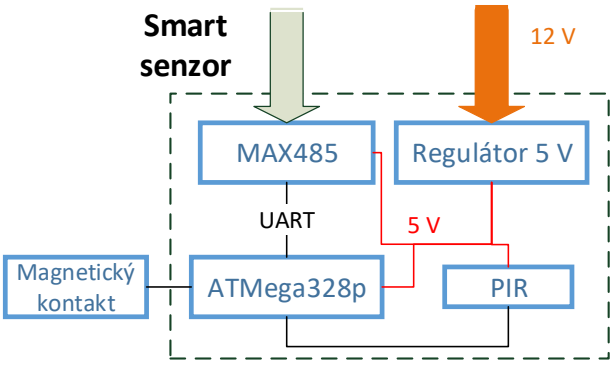
4 Návrh a realizace sběrnicových prvků

4.1 Smart senzory

Smart senzor je zařízení, které v jediném celku obsahuje čidlo, obvody pro převod a úpravu signálu, řízení a komunikaci s dalšími zařízeními. Sestavení Smart senzoru je dáno požadavkem 1.3 z tabulky 1. Následující kapitoly popisují návrh a sestrojení HW části senzoru a vývoj SW.

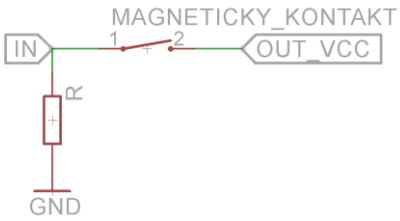
4.1.1 Návrh a sestrojení HW části senzoru

Blokové schéma zapojení senzoru je na obrázku 30. Jako čidla jsou použity PIR modul a magnetický kontakt. Mikrokontrolérem senzoru je ATMega328p. Pro komunikaci se sběrnicí se používá převodník MAX485. Výběrem a vlastnostem zmíněných komponent se zabývá kapitola 2.3.



Obrázek 30: Blokové schéma senzoru

Signál z PIR modulu je přiveden na pin 4 (PCINT18) mikrokontroléru ATMega328p. Tento pin umožňuje režim externího přerušení, které se používá pro detekci sepnutí PIR. Magnetický kontakt je připojen na dva porty mikrokontroléru podle obrázku 31.



Obrázek 31: Zapojení magnetického kontaktu

Jeden port je nastaven jako výstupní OUT a je na něj trvale přivedeno napájecí napětí VCC, které je 5 V. Druhý port je vstupní IN, a kromě magnetického kontaktu je připojen přes odpor na zem. Klidový stav nastane, pokud je kontakt sepnutý a na vstupu IN je napětí

VCC. Poplachový stav se vyvolá, pokud je kontakt rozepnutý a vstup IN je uzemněn. V případě, že dojde k přerušení vodičů magnetického kontaktu, vyvolá se automaticky poplach.

Napětí 12 V je do senzoru přivedeno společně s datovými signály. O snížení napětí na 5 V se stará regulátor LM7805. K mikrokontroléru ATMega328p je připojen 16 MHz oscilátor. Na obrázku 32 je ukázka osazených senzorů. Při návrhu DPS byly zohledněny rozměry krabičky, do které je následně senzor umístěn. Veškerá dokumentace týkající se senzoru je v příloze.

Obrázek 32: Osazené senzory

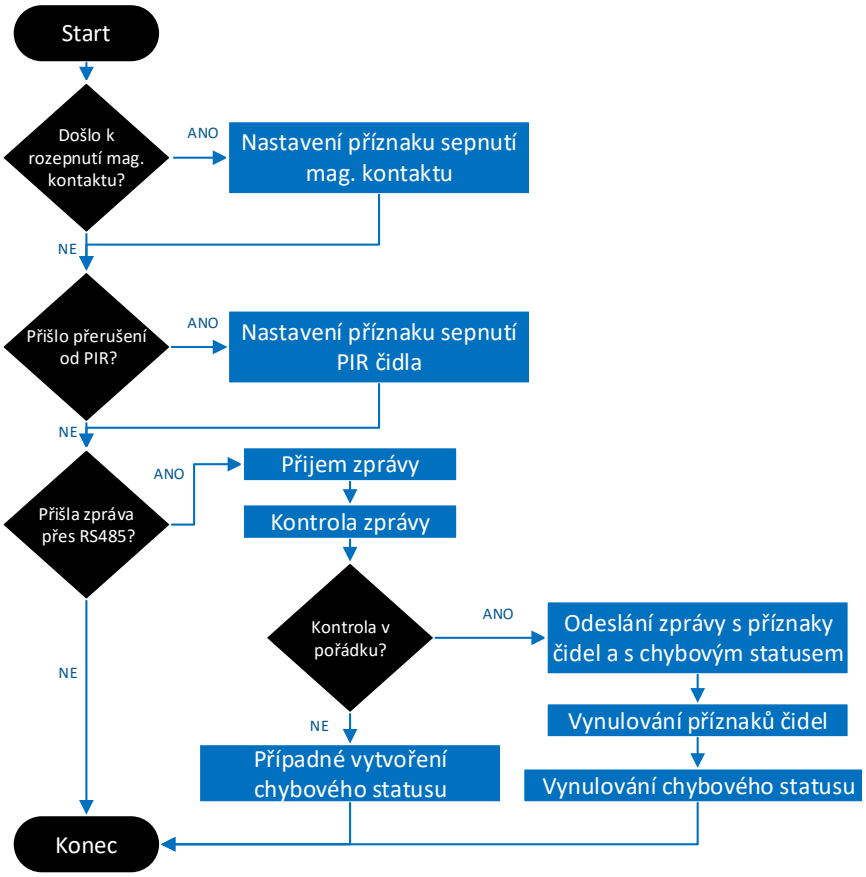
4.1.2 Návrh a vývoj SW části senzoru

Do mikrokontroléru ATMega328p byl nahrán Arduino zavaděč, díky čemuž mohl být program napsán v jazyku Wiring. Wiring je Framework jazyka C++ a usnadňuje programátorovi práci s nastavováním hardwaru. Zároveň umožňuje i implementaci zápisů v jazyce C++, neboť se při kompilaci softwaru používá C++ překladač. [30]

Program senzoru by se dal rozdělit na dvě hlavní částí. První má na starosti zpracování informací z čidel a druhá se stará o komunikaci s ústřednou. Vývojový diagram je na obrázku 33.

Zpracování informací z čidel

Z vývojového diagramu je patrné, že první část programu, která zpracovává informace ze senzorů, má za úkol nastavit příznak sepnutí konkrétního čidla. Toto vyhodnocení je velmi jednoduché, neboť v případě magnetického kontaktu se pouze kontroluje napěťová úroveň na příslušném portu a v případě PIR čidla se funkce pro nastavení příznaku zavolá externím přerušením. Příznaky sepnutí jsou uloženy v globální proměnné a nulují se až ve chvíli, kdy je informace o sepnutí odeslána ústředně.



Obrázek 33: Vývojový diagram programu Smart senzoru

Komunikační část

Jedná se o komplexnější část programu, která byla napsána s ohledem na pravidla komunikace definovaná v kapitole 2.4. Komunikační část je naprogramována univerzálně, aby mohla být použita změnou několika parametrů v programu ovládacího panelu a případně i v jakémkoliv dalším zařízení.

Nejprve je nutné si uvědomit, že u všech prvků dochází k příjmu veškerých zpráv, které se na sběrnici posílají, a až následně se odfiltrují zprávy, které mají jiného adresáta. Prakticky je to v programu řešeno tak, že se nejdřív vyčtou všechny přijaté byty ze sériového portu. Uloží se do bytového pole a následně se zavolá funkce, která provede obecnou kontrolu přijaté zprávy. Obecná kontrola provádí postupnou podmíněnou kontrolu, která zjišťuje, zda je správný adresát, odesílatel, stop byte a velikost zprávy. Podmíněná je v tom smyslu, že pokud není splněna kterákoliv nadřazená podmínka, tak už se dál nic nekontroluje. Například když není očekávaný adresát, vyskočí se z funkce a zpráva se už dál nijak nezpracovává, neboť je určena jinému prvku. Pokud se stane, že je správný adresát, ale není splněna některá z dalších kontrol, dojde k nastavení chybového statusu v souladu s odstavcem Chybové statusy sběrnicových prvků. V případě, že se v tomto kroku nesplní

žádná z chybových podmínek, je kontrola zprávy v pořádku a zavolá se funkce pro odeslání odpovědi ústředně.

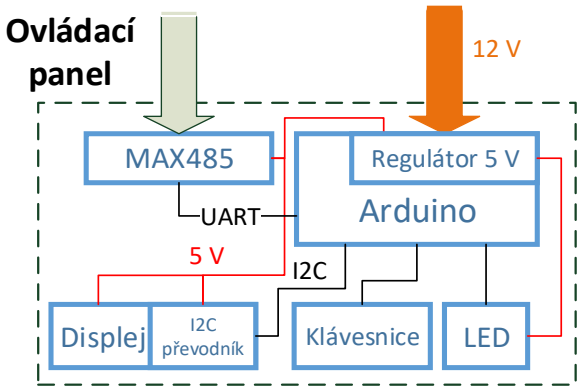
Funkce pro odeslání odpovědi má na starost sestavit zprávu do podoby, kterou očekává ústředna. Správná podoba zprávy od senzoru obsahuje postupně následující byty: adresát, odesílatel, příznaky sepnutí čidel, chybový status a stop byte. Příznaky sepnutí čidel jsou bitové hodnoty uložené v jednom bytu. 0. bit informuje o sepnutí PIR senzoru a 1. bit o sepnutí magnetického kontaktu. Po odeslání zprávy dojde k vynulování příznaků sepnutí čidel a chybového statusu.

4.2 Ovládací panel

Ovládací panel má úlohu základního informačního a ovládacího prvku. Panel pomocí displeje a informačních LED diod informuje uživatele o aktuálním stavu systému. Uživatel může vybrat režim zabezpečení a zadáním pinu na klávesnici zastřežit či odstřežit systém. Následující kapitoly popisují návrh a sestrojení HW části panelu a vývoj SW.

4.2.1 Návrh a sestrojení HW části ovládacího panelu

Blokové schéma zapojení ovládacího panelu je na obrázku 34. Byla použita klávesnice 4×4. Displej obsahující I2C převodník. Převodník MAX485. Arduino UNO a indikační LED diody.



Obrázek 34: Blokové schéma ovládacího panelu

Vzhledem k tomu, že ovládací panel má větší krabičku, bylo do ní uloženo celé Arduino. Napájecí napětí 5 V je k ostatním komponentám přivedeno z napěťového regulátoru umístěného na Arduino desce. Klávesnice a LED diody jsou připojeny na klasické vstupně výstupní porty. Displej je s Arduinem propojen přes I2C sběrnici. Převodník MAX485 je propojen standardně přes sériový port. Všechny komponenty byly umístěny do plastové krabičky, která byla navržena v on-line programu OnShape a následně vyrobena na 3D

tiskárně. První osazená verze krabičky je na obrázku 35. Veškerá další fotodokumentace a návrhy v programu Onshape jsou v příloze.

Obrázek 35: Osazený Ovládací panel

4.2.2 Návrh a vývoj SW části ovládacího panelu

Program ovládacího panelu je složitější než program obyčejného senzoru. Podobně jako v hlavním programu ústředny i zde totiž musel být naprogramován stavový automat. Program je opět psán v jazyku Wiring, a celkově by se dal rozdělit na dvě hlavní části. První část se stará o komunikaci a druhá část je stavový automat.

Komunikační část

Jak již bylo zmíněno u senzoru, tato část programu je v hlavních bodech, kterými jsou příjem zprávy a kontrola formátu zprávy, stejná. Hlavní principy tak byly popsány v předchozí kapitole. Nicméně ovládací panel, na rozdíl od senzoru, dále zpracovává a používá konkrétní data od ústředny. Jedná se o informace o stavu ústředny a o nastaveném času pro odpočet. Získané informace se ukládají do globálních proměnných a následně s nimi pracuje stavový automat.

Jiný je také formát odpovědi. Obecná pravidla zůstávají, ale mění se odeslaná data. Pokud totiž uživatel chce odeslat z ovládacího panelu nějaký požadavek do ústředny, autorizuje se čtyřmístným pinem. Čtyřmístný pin musí být rozdělen a odeslán ve dvou bytech ústředně, která provede kontrolu, zda došlo k zadání správného pinu. Společně s pinem se odešle také konkrétní číslo požadavku. V současné verzi programu byly definovány následující číselné požadavky:

- 1 – 9: požadavek na zastřežení systému, číslo určuje konkrétní režim (1 – celý dům, 2 – režim noc, 3 – oblast 1, 4 – oblast 2), zbylé hodnoty jsou rezervovány pro další možné režimy, ale nepoužívají se
- 10: informace o stisknuté klávesnici v režimu zastřeženo (má za úkol vyvolat potenciální poplach)
- 11: požadavek na odstřežení ve stavu potenciální poplach
- 12: požadavek na odstřežení ve stavu poplach

Stavový automat

Implementovaný stavový automat ovládacího panelu v principu kopíruje sedm základních stavů zabezpečovací ústředny, které byly definovány na obrázku 22. Znamená to, že pokud přijde od ústředny informace o stavu a program vyhodnotí, že se jedná o stav nový, pak stavový automat ovládacího panelu zajistí přechod do odpovídajícího vnitřního stavu. Tato reakce nastává vždy pouze při změně stavu ústředny.

Stavový automat pak dále definuje další vnitřní stavy, do kterých se může dostat ovládáním klávesnice nebo vypršením časovače. Z uživatelského pohledu má panel nejvíce vnitřních stavů, je-li ústředna ve stavu nestřeženo. V tom případě ovládací panel nabízí dvouúrovňové menu. V menu se listuje doprava a doleva tlačítky B a C, potvrzuje se tlačítkem D a krok zpět se udělá tlačítkem A. Uživatel si tak může vybrat jeden ze čtyř režimů zabezpečení, a následným zadáním pinu a potvrzením klávesou D odeslat požadavek na zastřežení. V programu je ošetřena chyba vstupu. Například pokud uživatel zadá delší nebo kratší pin, je o tom informován a musí pin zadat znovu. Když z nějakého důvodu nefunguje komunikace s ústřednou, přechází automat do speciálního stavu *time out* a informuje o tom uživatele.

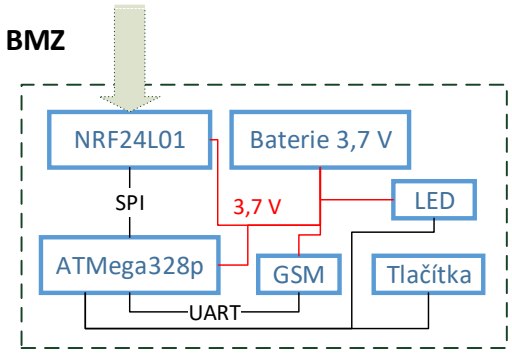
Z programového hlediska je kód celkem rozsáhlý. Při změně stavů ústředny se zavolá funkce *ZmenaStavu()*, která obsahuje *switch – case* konstrukci. Ta zajišťuje přechod do konkrétního vnitřního stavu, volání funkcí pro vykreslení informace na displej a rozsvícení příslušných LED diod. Ovládaní panelu pomocí klávesnice je obslouženo další *switch – case* konstrukcí, která definuje možné reakce na konkrétní klávesy v závislosti na vnitřním stavu automatu. Dále jsou zde implementované další funkce. Například funkce starající se o odpočet na základě přijaté informace o časovači od ústředny. Zmíněné kontroly vstupů. Funkce pro vykreslování na displej a další. Program je součástí přiloženého CD.

5 Návrh a realizace bezdrátového monitorovacího zařízení BMZ

Z požadavku 1.4. z tabulky 1 vyplývá, že muselo být sestrojeno bezdrátové monitorovací zařízení, které v případě výpadku napájení ústředny odešle uživateli SMS zprávu o posledním známém stavu, či vyvolá poplach. Následuje návrh HW a SW s ohledem na nízkoodběrové režimy. Na závěr kapitoly dojde ke zhodnocení naměřené spotřeby zařízení.

5.1 Návrh a sestrojení HW části BMZ

Základem BMZ jsou mikrokontrolér (dále také MCU) ATMega328p a rádio NRF24L01. Obě komponenty podporují nízkoodběrové módy, a jsou proto vhodnými adepty pro sestavení nízkoodběrového zařízení. Dalším nezbytným prvkem je GSM modul SIM800L pro odesílání SMS zpráv. Blokové schéma zapojení je na obrázku 36.



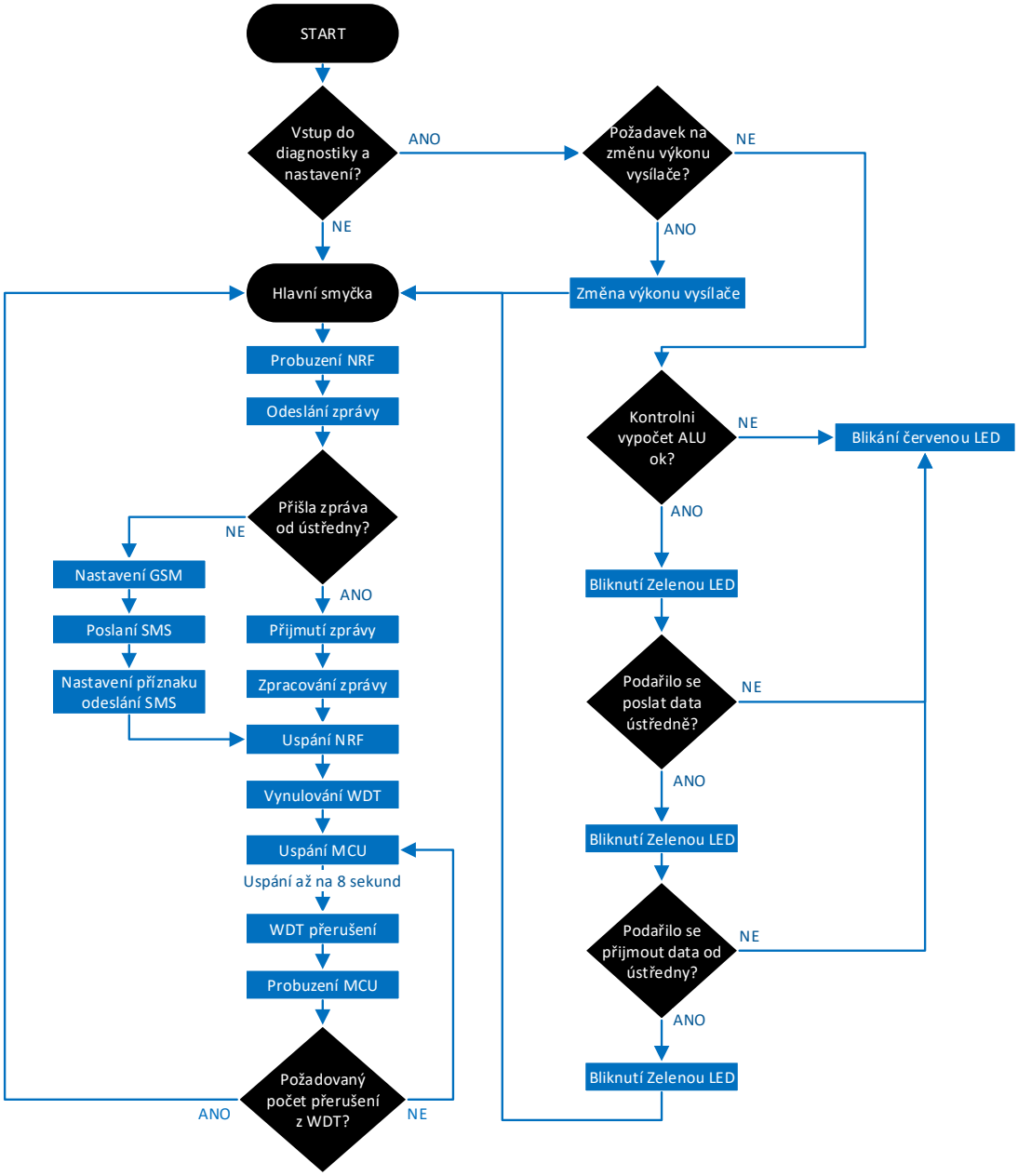
Obrázek 36: Blokové schéma BMZ

Všechny použité součásti byly vybrány tak, aby byly schopny provozu při napájecím napětí 3,7 V, v případě použití standardní Li-Ion baterie. K zajištění správné funkčnosti je k mikrokontroléru připojen 8MHz krystal. MCU komunikuje s rádiem NRF24L01 přes SPI rozhraní a s GSM modulem přes UART. Součástí prvního prototypového zařízení jsou také diagnostické LED diody a tlačítka. Prototyp je na obrázku 37.

Obrázek 37: Prototyp BMZ

5.2 Návrh a vývoj SW části BMZ

Program BMZ je psán opět ve Wiringu. Kromě knihoven pro práci s rádiem, SPI sběrnicí a UART byly připojeny další knihovny. Jedná se o knihovny *avr/sleep*, *avr/power* a *avr/wdt*. Tyto knihovny umožňují používat funkce, kterými se snadněji přechází do nízkoodběrových stavů mikrokontroléru ATMega328p. Nicméně i tak muselo být někdy v kódu použito klasické nastavování přes registry. Například nastavení watch dog časovače (dále jen WDT). Celkově se program dá rozdělit do dvou hlavních částí. Nastavovací – diagnostická část a hlavní program. Vývojový diagram celého programu je na obrázku 38.



Obrázek 38: Vývojový diagram programu BMZ

Nastavovací a diagnostická část

Jak je patrné z vývojového diagramu, vstup do nastavovací a diagnostické části je možný pouze po spuštění programu. Zařízení o této možnosti informuje rozsvícením červené a zelené LED zároveň. Pokud uživatel chce vstoupit do ovládacího režimu, musí přidržet obě tlačítka, dokud nedojde ke zhasnutí zelené LED diody. Pokud tak nastane, uživatel si může tlačítkem S3 navolit dodatečná nastavení týkající se výkonu vysílače NRF24L01, která jsou:

- Jedno stisknutí: výkon TX vysílače na MIN (zesílení -18 dBm, odběr 7 mA)
- Dvě stisknutí: výkon TX vysílače na LOW (zesílení -12 dBm, odběr 7,5 mA)
- Tři stisknutí: výkon TX vysílače na HIGH (zesílení -6 dBm, odběr 9 mA)
- Čtyři stisknutí: výkon TX vysílače na MAX (zesílení 0 dBm, odběr 11,3 mA)
- Žádné stisknutí: vstup do diagnostického režimu

Nastavená volba se potvrdí stisknutím tlačítka S2, poté dojde k příslušné akci a program přejde do hlavní smyčky. Pokud se tedy nejedná o volbu diagnostickou.

V diagnostické funkci se postupně vykonají tři testy. Nejprve se otestuje vnitřní aritmeticko – logická jednotka (ALU) a paměť. Počítá se kontrolní výpočet v celých číslech a v desetinných číslech. Porovnání obou výsledků si musí odpovídat a musí se rovnat číslu uloženému v paměti. V dalším kroku se testuje odesílání dat přes rádio. V posledním kroku se testuje navázání komunikace s protistranou. Ve chvíli, kdy konkrétní test projde, je o tom uživatel informován bliknutím zelenou LED diodou a přejde se na další test. Pokud všechny testy projdou, dojde ke zhasnutí červené LED a program přechází do hlavní smyčky. V tento okamžik má uživatel jistotu, že komunikace BMZ s ústřednou je funkční.

Hlavní program

Hlavní program je také zobrazen na obrázku 38. Nejprve dochází k probuzení rádia NRF24L01. Komunikace s ústřednou je postavená na dotazovací formě. BMZ vyšle dotaz a čeká na odpověď od ústředny.

V případě, že ústředna odpoví, dochází ke zpracování přijatých dat a uloží se informace o aktuálním stavu ústředny. Následně rádio přechází do nejúspornějšího *Power down* režimu. V dalším kroku se inicializuje WDT a dochází k uspání mikrokontroléru. ATMega328p používá nejúspornější *Power down* mód, ve kterém je spuštěn pouze hardwarový WDT, který zajistí opětovné probuzení MCU. Časovač je nastaven na

maximální počet čítání, které přeteče přibližně po 8 sekundách. Po přetečení WDT se vyvolá přerušení, které probudí mikrokontrolér. Mikrokontrolér se následně dostane do standardního režimu. Vzhledem k tomu, že WDT probudí MCU již po 8 sekundách je v SW implementován interní čítač, aby došlo k umělému prodloužení této doby. Programový čítač počítá, kolik přišlo přerušení od WDT. Pokud jich přišlo dostatečné množství, tak se program vrátí na začátek hlavní smyčky, pokud ne, dojde k opětovnému uspání MCU.

V případě, že ústředna nekomunikuje, přejde se do funkce pro odeslání SMS zprávy uživateli. V této funkci se nejprve inicializuje GSM modul a následně se odešle zpráva, která obsahuje informaci o posledním známém stavu ústředny. Aby nedocházelo k cyklickému odesílání SMS, nastaví se příznak odeslání, který je kontrolován vždy před zavoláním funkce. Příznak odeslání je vynulován až během úspěšné komunikace s ústřednou.

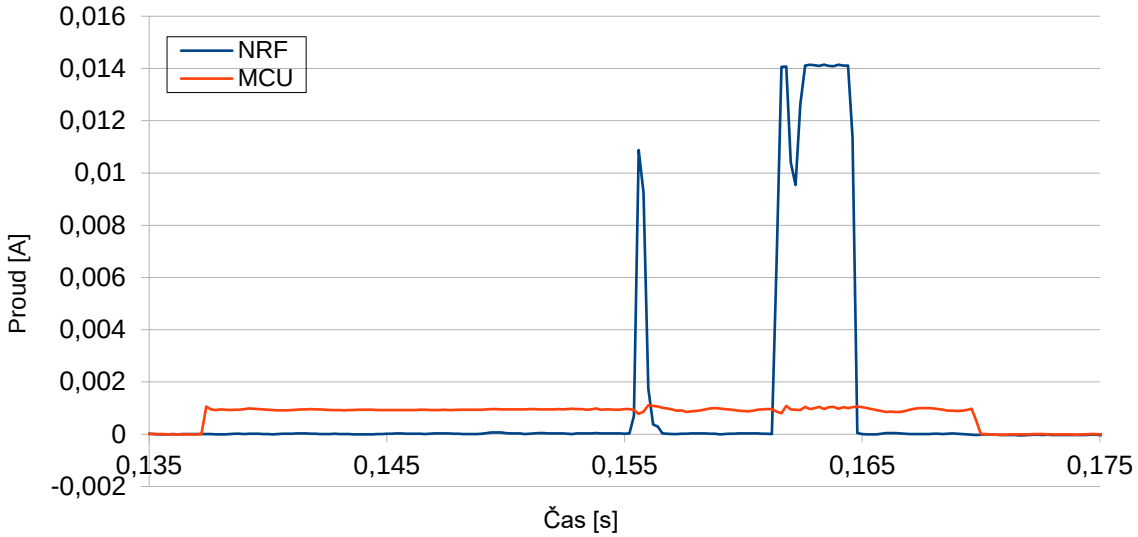
Formát paketu, který odesílá rádio, je dán komunikačním protokolem *Enhanced ShockBurst*™. Program pouze zavolá funkci pro odeslání požadavku a přijímá informaci o stavu ústředny. O správný formát zprávy se postará zmíněný protokol. Podrobnější informace o formátu zprávy a dalších vlastnostech rádia NRF24L01 byly popsány v kapitole 2.3.6.

5.3 Měření spotřeby

Vzhledem k tomu, že BMZ je nízkoodběrové zařízení, došlo k měření skutečných odběrů. Metodika měření proudové spotřeby spočívá v měření úbytku napětí na rezistoru vloženého do zemní cesty. Na tento rezistor je připojena osciloskopická sonda a průběh napětí je zobrazen na osciloskopu. V tomto případě byl zařazen rezistor 44 Ω do zemní cesty rádia a 37 Ω do zemní cesty mikrokontroléru. Na odporech vznikne úbytek napětí dostatečně velký k měření spotřeby, a zároveň dostatečně malý na to, aby negativně neovlivnil funkčnost zařízení. Měření bylo prováděno během funkční komunikace s ústřednou, a není zde uvažován vliv GSM modulu, který se nespouští cyklicky. BMZ byl během měření napájen ze dvou tužkových baterií, které měly v součtu napětí 3,2 V.

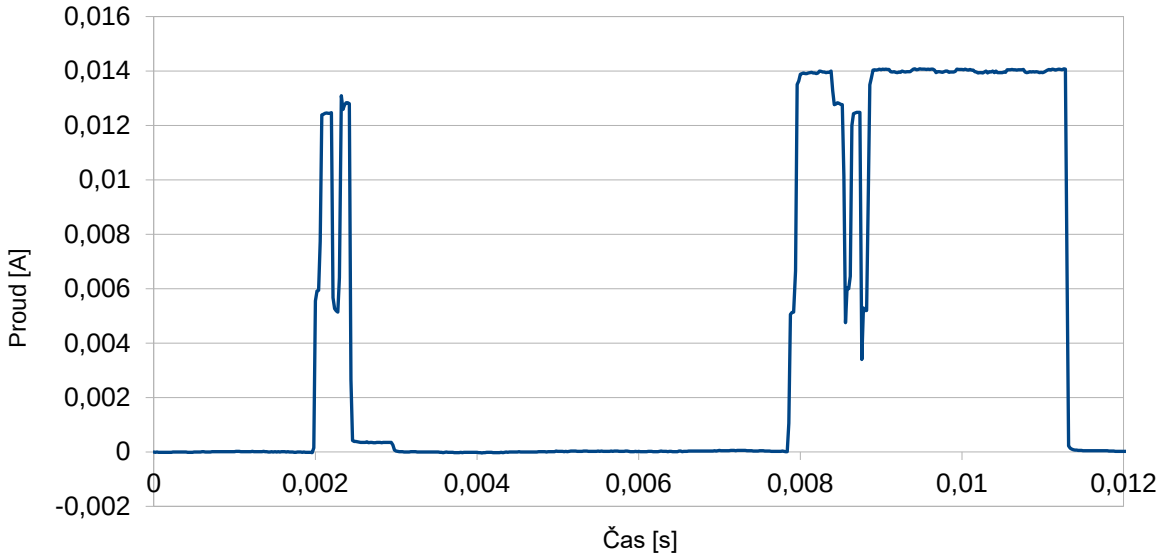
První naměřený průběh je na obrázku 39. Z průběhu je patrný celý cyklus BMZ od probuzení MCU, probuzení rádia, až po komunikaci s ústřednou a závěrečné opětovné uspání komponent. MCU je v aktivním stavu po dobu přibližně 32 ms a naměřené proudové odběry jsou přibližně 1 mA v aktivním stavu a 1 μA v *Power down* režimu. Pozoruhodný je odběr v aktivním stavu, neboť podle katalogového listu mikrokontroléru (viz [19] strana 379) by měl být proudový odběr 3 mA při napětí 3,2 V a frekvenci 8 MHz. Jedná se tak

o trojnásobně vyšší hodnotu, než byla naměřena osciloskopem. Samozřejmě je zde možný vliv nepřesnosti měření.



Obrázek 39: Průběh spotřeby BMZ

Naopak měření spotřeby rádia NRF24L01 dopadlo, až na drobnosti, podle katalogových předpokladů. Detail měření spotřeby rádia je na obrázku 40.



Obrázek 40: Průběh spotřeby rádia NRF24L01

Z naměřeného průběhu jsou patrné všechny režimy rádia. Vysílání trvá 440 μs a proudová spotřeba je přibližně 13 mA. Příjem trvá 3,44 ms a spotřeba je přibližně 14 mA. Po ukončení vysílání je na průběhu malý schodek, který má spotřebu přibližně 400 μA, což proudově i logicky odpovídá *Standby-2* režimu. Po ukončení *Standby-2* režimu a před začátkem příjmu následuje *Standby-1* režim. Tento režim má pro zvolenou metodou měření

již velmi nízkou hodnotu odběru. Podle katalogového listu by se mělo jednat o 22 μA. Posledním režimem je *Power down* režim, který je aktivní v době úplného uspání rádia. Proudový odběr *Power down* režimu je ještě nižší než odběr *Standby-1*. Podle katalogového listu je jeho hodnota 900 nA. [26]

Z naměřených výsledků se dá udělat celkový odhad spotřeby. Pokud by interval monitorovacího zařízení byl 30 sekund a spotřeby rádia a MCU by odpovídaly naměřeným hodnotám, byla by průměrná celková spotřeba zařízení 0,192 mAh. V případě napájení baterií s kapacitou 3000 mAh by se baterie vybila přibližně za 650 dní, což je přijatelný výsledek. Vypočet je ale třeba brát s nadhledem. Není zde započítána spotřeba GSM modulu, která má velmi vysoké okamžité proudové odběry během připojování do telefonní sítě a během odesílání SMS. Zároveň je počítáno s naměřenými hodnotami mikrokontroléru, které vyšly nižší než katalogové. [19], [26]

6 Zhodnocení vlastností systému

V kapitole 2.1 došlo k definici systémových požadavků. V závěru práce je proto vhodné zhodnotit celý vývoj a porovnat dosažené výsledky s těmi předem definovanými a případně navrhnout další kroky pro vylepšení vlastností systému.

Kontrola splnění definovaných požadavků

Systémové požadavky se nacházejí v tabulce 1. Požadavky s prioritou 1 úzce souvisí se zadáním diplomové práce a byly všechny splněny. Požadavek 1.1 stanovil konkrétní prvky navrhovaného EZS, kterými jsou ústředna, ovládací panel, senzory a bezdrátové monitorovací zařízení. Požadavek 1.2 formuloval hlavní komponenty ovládacího panelu, konkrétně mikrokontrolér ATmega328p, displej a klávesnici. Požadavek 1.3 definoval použití Smart senzorů, a tudíž i nepřímo odkazoval na sestavení sběrnicového systému. Dále stanovil použití čidel PIR a magnetických kontaktů. Poslední požadavek z kategorie s nejvyšší prioritou 1.4 určil hlavní vlastnost bezdrátového monitorovacího zařízení, kterými jsou rádiová komunikace s ústřednou a vyvolání poplachu v případě, že dojde k přerušení komunikace. Rádiová komunikace byla docílena použitím nízkoodběrových modulů NRF24L01. Vyvolání poplachu je docíleno posíláním SMS zprávy z GSM modulu SIM800L.

Požadavky s prioritou 2 měly za úkol určitým způsobem doplnit některé zásadní vlastnosti systému, ale nesouvisely přímo se zadáním diplomové práce. Požadavek 2.1 určil sběrnicovou topologii systému, čehož bylo docíleno použitím standardu RS485 díky modulu obsahující převodník napěťových úrovní TTL na RS485 s názvem MAX485 a implementováním vlastní komunikační vrstvy na softwarové úrovni. Na straně ústředny se o komunikaci stará skript napsaný v jazyce Python, který přijatá data zapisuje do databáze. Na straně ostatních prvků je to kód napsaný v jazyce Wiring. Požadavek 2.2 definoval, že jednotlivé prvky připojené na sběrnici nebudou mít vlastní napájecí zdroj, ale napájení k nim bude přivedeno z ústředny. Tohoto požadavku bylo docíleno použitím napěťového zdroje 12 V/60 W, který má dostatečný výkon pro napájení všech prvků včetně ústředny. Toto napětí je z ústředny rozvedeno k ostatním prvkům společně se sběrnicí, a na straně jednotlivých prvků jsou použity stabilizátory LM7805 pro snížení napětí na 5 V. Požadavek 2.3 určil způsoby vyvolání poplachu pomocí emailu a SMS zprávy. K odesílání emailu byla vytvořena speciální sekvence v Node-RED, která umožňuje práci s gmail adresami. Pro tento účel byla vytvořena emailová adresa dpPZTS@gmail.com, ze které se email odesílá

na konkrétní adresy určené v Node-RED. Pro odesílání SMS zpráv slouží GSM modul SIM800L umístěný v ústředně. Požadavek 2.4 odkazoval na nezbytné připojení k internetu, čehož je docíleno jednoduše díky použití počítače Raspberry Pi 3B+. Počítač podporuje jak Ethernet, tak WiFi. Poslední požadavek 2.5 poukazoval na vytváření logů ústředny a jejich ukládání na SD kartu a do databáze. Vzhledem k tomu, že má ústředna implementovanou MySQL databázi, dochází k ukládání právě do ní. Z databáze se tak dají vyčíst nejen veškerá získaná data od jednotlivých prvků systému, ale také historie ústředny.

Poslední kategorie požadavků s prioritou 3 obsahovala bonusové vlastnosti systému, které nebyly hlavním předmětem a nesouvisely ani se zadáním diplomové práce. Požadavek 3.1 poukazoval na použití záložního bateriového zdroje. Tento požadavek nakonec nebyl v práci řešen, a to především z toho důvodu, že záložní funkci plní bezdrátové monitorovací zařízení, které má vlastní baterii a při výpadku elektrické energie odešle SMS zprávu. Nicméně tak nejsou zachovány kompletní funkce systému. Proto použití záložního bateriového zdroje bude otázkou budoucího vývoje. Požadavek 3.2 definoval použití snímače otisků prstů u ovládacího panelu. Na tento požadavek nebyl prostor a nebude s největší pravděpodobností řešen ani v budoucnu. Celkově poslední požadavek 3.3 stanovil vytvoření grafického uživatelského rozhraní. Tento požadavek byl implementován za pomoci Node-RED dashboard uzlů. Uživateli nabízí základní přehled systému, dává mu možnost na zastřežení a odstřežení systému, a také mu umožňuje změnit režimy zabezpečení. Do GUI rozhraní se uživatel dostane přes webový prohlížeč po zadání hesla.

Hlavní systémové požadavky byly splněny a prototypový systém je plně funkční a chová se podle očekávání. Samozřejmě se jedná o první verzi a již nyní se dá vytyčit hned několik dalších úprav pro vylepšení vlastností systému.

Náměty na rozšíření vlastností systému

Dvě hlavní oblasti, ve kterých má systém určité rezervy, jsou zabezpečení a použití záložního bateriového zdroje.

Systém má implementováno určité základní zabezpečení v podobě přihlašovacích hesel do databáze, GUI a Node-RED editoru. Navíc všechna hesla pro přihlášení jsou na straně ústředny uložená v hashované podobě, tak aby případný útočník nezjistil jejich pravou podobu. Nicméně v současnosti nemá připojení přes webový server zabezpečenou komunikaci. Dá se k tomu využít například protokol SSL, který zajišťuje zabezpečení komunikace šifrováním a autentizací komunikujících stran, při použití v kombinaci

s komunikačními protokoly HTTP či MQTT. Pro uskutečnění této možnosti bude potřeba nainstalovat certifikát na Raspberry Pi a nastavit jeho využití pro Node-RED.

Další oblastí je použití záložního napěťového zdroje. Bude se muset vybrat vhodný záložní zdroj, a zároveň bude nutné zjistit spotřebu celého systému a případně ji nějakým způsobem redukovat tak, aby se docílilo co nejdelší možné výdrže akumulátoru. Nejvýznamnější spotřeba bude nejspíš na straně ústředny, a to hlavně kvůli použití počítače Raspberry Pi 3B+. Určitá redukce odběru Raspberry Pi by měla být docílena vypnutím nepoužívaných periferií, vypnutím desktopového LXDE prostředí, případně snížením výkonu počítače. Bude-li potřeba snížit také odebírané napětí ostatních prvků, dá se toho docílit snížením napájecího napětí z 5 V na nižší hodnotu, a také snížením frekvence mikrokontroléru použitím například 8 MHz krystalu namísto 16 MHz. Další úspora by mohla být docílena i vhodným použitím *Power down* režimů mikrokontroléru ATMega328p podobně, jako tomu bylo při návrhu BMZ.

Závěr

Cílem této diplomové práce byl návrh a realizace elektronického zabezpečovacího systému pro rodinný dům obsahující ústřednu, ovládací panel, senzory a bezdrátové monitorovací zařízení.

V teoretické části byly popsány hlavní vlastnosti poplachových zabezpečovacích a tísňových systémů. Dále byly popsány hlavní vlastnosti Internetu věcí. Na závěr teoretické části došlo ke shrnutí aktuální situace na trhu s elektronickými zabezpečovacími systémy s důrazem na Internet věcí.

V úvodu praktické části došlo k návrhu na úrovni celého systému. Tento návrh obsahoval sepsání systémových požadavků, definování struktury systému, návrh pravidel pro komunikaci mezi jednotlivými prvky a výběr vhodných komponent. Konkrétně se jedná o Raspberry Pi 3B+ jako počítač ústředny, ATMega328p jako mikrokontrolér všech ostatních prvků, převodník MAX485, který má na starost převod TTL výstupů na sběrnicový standard RS485, PIR senzor HC-SR501 a magnetický kontakt pro Smart senzor, displej a klávesnice pro ovládací panel, rádio NRF24L01 pro komunikaci s bezdrátovým monitorovacím zařízením a GSM modul SIM800L pro odesílání SMS zpráv.

Po návrhu na úrovni celého systému se přešlo k vývoji konkrétních prvků. Nejprve došlo k návrhu a realizaci řídící ústředny, která obsahuje zmíněné Raspberry Pi, mikrokontrolér ATMega328p a další potřebné prvky jako jsou DC/DC měnič, převodník MAX485, GSM modul a rádio NRF24L01. Byla vymyšlena softwarová architektura ústředny, která se skládá z pěti hlavních komponent. Raspberry Pi má implementované čtyři softwarové komponenty, kterými jsou MySQL databáze, řídící komunikační program pro RS485 psaný v Pythonu, hlavní stavový program psaný v Node-RED a grafické uživatelské rozhraní psané také v Node-RED. Na mikrokontroléru je implementována komponenta starající se o ovládání GSM modulu a komunikaci přes rádio NRF24L01 s bezdrátovým monitorovacím zařízením psaná v jazyku Wiring.

Následoval vývoj ovládacího panelu a Smart senzoru. Byl použit společný mikrokontrolér ATMega328p, PIR senzor, magnetický kontakt, klávesnice a displej. Po sestavení prvků došlo k naprogramování mikrokontroléru v jazyce Wiring. Účel obou zařízení je zřejmý. Smart senzor slouží k detekci nežádoucích událostí a ovládací panel dává uživateli možnost ovládat systém.

Posledním prvkem systému je bezdrátové monitorovací zařízení BMZ. Účel tohoto zařízení je rozpoznat, že došlo k vypnutí či poruše ústředny a případně vyvolat poplach odesláním SMS zprávy. BMZ obsahuje mikrokontrolér ATMega328p, rádio NRF24L01 a GSM modul. Mikrokontrolér a rádio byly vybrány s přihlédnutím na nízkoodběrové vlastnosti, neboť u bateriově napájeného zařízení je důležitá dlouhá výdrž baterie. Software byl psán tak, aby docházelo k uspání mikrokontroléru i rádia a maximálním způsobem se snižovala spotřeba zařízení. Došlo také k naměření proudových odběrů BMZ pomocí osciloskopu.

V poslední kapitole došlo ke zhodnocení vlastností systému, porovnání realizace oproti systémovým požadavkům a návrhu dalšího možného rozšíření systému například o záložní bateriový zdroj ústředny či o důkladnější zabezpečení webového serveru.

Z celkového pohledu došlo ke splnění všech bodů zadání a systém je plně funkční. Systém bude použitý pro zabezpečení rodinného domu a je možné ho do budoucna rozšířit například i o další prvky domácí automatizace.

Seznam literatury a informačních zdrojů

[1] BURDA, Karel. *Základy elektronických zabezpečovacích systémů*. Brno: Akademické nakladatelství CERM, 2017. ISBN 978-80-7204-967-7.

[2] BUYYA, Rajkumar. *Internet of Things: Principles and Paradigms*. Cambridge, 2016. ISBN 978-0-12-805395-9.

[3] *Základní úvod do oblasti internetu věcí (IoT)* [online]. 2016 [cit. 2019-04-25]. Dostupné z: https://automatizace.hw.cz/zakladni-uvod-do-oblasti-internetu-veci-iot.html

[4] PECH, Jiří. *IOT TECHNOLOGIE: IQRF* [online]. 2019 [cit. 2019-04-20]. Dostupné z: https://www.eman.cz/blog/iot-technologie-iqrf-2-5/

[5] PECH, Jiří. *IOT TECHNOLOGIE: LORA A LORAWAN* [online]. 2019 [cit. 2019-04-20]. Dostupné z: https://www.eman.cz/blog/iot-technologie-iqrf-2-5/

[6] PECH, Jiří. *IOT TECHNOLOGIE: SIGFOX* [online]. 2019 [cit. 2019-05-10]. Dostupné z: https://www.eman.cz/blog/iot-technologie-sigfox-4-5/

[7] *NarrowBand IoT* [online]. 2016 [cit. 2019-04-23]. Dostupné z: https://www.iot-portal.cz/2016/04/30/narrowband-iot/

[8] *JABLOTRON. JA-101K [online]. [cit. 2019-04-23]. Dostupné z: https://www.jablotron.com/cz/produkt/ustredna-s-vestavenym-gsm-gprs-komunikatorem-209/*

[9] *SEDLÁK, Jan. Jablotron investuje desítky milionů do vývojářů eMan, rozjedou internet věcí [online]. 2016 [cit. 2019-04-23]. Dostupné z: https://www.lupa.cz/clanky/jablotron-investuje-desitky-milionu-do-vyvojaru-eman-rozjedou-internet-veci/*

[10] *Jablotron a CZ.NIC spouští spolupráci v IoT* [online]. 2015 [cit. 2019-04-23]. Dostupné z: https://www.iot-portal.cz/2015/06/16/jablotron-a-cz-nic-spousti-spolupraci-v-iot/

[11] HERWIG, Bohumil. *Zabezpečovací zařízení Jablotron 100: co nabízí a co umí?* [online]. 2015 [cit. 2019-04-23]. Dostupné z: https://www.lupa.cz/clanky/zabezpecovaci-zarizeni-jablotron-100-co-nabizi-a-co-umi/

[12] DELANEY, John. *The Best DIY Smart Home Security Systems for 2019* [online]. 2019 [cit. 2019-05-15]. Dostupné z: https://www.pcmag.com/roundup/367402/the-best-diy-smart-home-security-systems

[13] LACOMA, Tyler. *The best DIY home security systems for 2019* [online]. 2019 [cit. 2019-05-15]. Dostupné z: https://www.digitaltrends.com/home/best-diy-home-security-systems/

[14] *Abode* [online]. [cit. 2019-05-15]. Dostupné z: https://goabode.com/

[15] *Angee* [online]. [cit. 2019-05-15]. Dostupné z: https://www.meetangee.com/

[16] MACHO, Daniel. *Angee: České ručičky zazářily, autonomní bezpečnostní systém jde na KickStarter* [online]. 2015 [cit. 2019-05-15]. Dostupné z: https://www.svetandroida.cz/angee-zabezpeceni-kickstarter/

[17] *Official website Raspberry Pi* [online]. [cit. 2019-04-20]. Dostupné z: www.raspberrypi.org

[18] UPTON, Eben a Gareth HALFACREE. *Raspberry Pi: uživatelská příručka*. Brno: Computer Press, 2013. ISBN 978-80-251-4116-8.

[19] *Datasheet ATMega328p* [online]. Atmel [cit. 2019-04-20]. Dostupné z: https://cdn.sparkfun.com/assets/c/a/8/e/4/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

[20] *Datasheet MAX* [online]. Maxim Integrated [cit. 2019-04-20]. Dostupné z: https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf

[21] *Datasheet 74HC04* [online]. Texas Instruments [cit. 2019-04-20]. Dostupné z: http://www.ti.com/lit/ds/symlink/sn74hc04.pdf

[22] DECHRISTÉ, Jean-Matthieu. *PIR sensors: HC-SR501* [online]. 2017 [cit. 2019-05-28]. Dostupné z: https://www.iot-experiments.com/pir-sensors-hc-sr501/

[23] *Magnetic contacts on Aliexpress* [online]. [cit. 2019-05-28]. Dostupné z: https://www.aliexpress.com/wholesale?catId=0&initiative_id=SB_20190527233240&SearchText=magnetic+contact

[24] M, Luboš. *LCD Displej* [online]. 2016 [cit. 2019-05-28]. Dostupné z: https://navody.arduino-shop.cz/zaciname-s-arduinem/lcd-displej.html

[25] *4x4 Matrix Keyboard [online]. [cit. 2019-05-28]. Dostupné z: https://www.aliexpress.com/wholesale?catId=0&initiative_id=SB_20190527234309&SearchText=Keyboard+4x4+Plastic+Keys*

[26] *Datasheet NRF24L01 [online]. Nordic Semiconduct. [cit. 2019-04-29]. Dostupné z: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_PreliminarP_Product_Specification_v1_0.pdf*

[27] *Soubor článku o GSM modulu SIM800L [online]. 2015 [cit. 2019-04-30]. Dostupné z: https://www.arduinotech.cz/rubrika/gsm/*

[28] *MYSLIVEC, Vojtěch. Webový server s Raspberry Pi [online]. 2013 [cit. 2019-04-29]. Dostupné z: http://vojtech.myslivec.net/webovy-server*

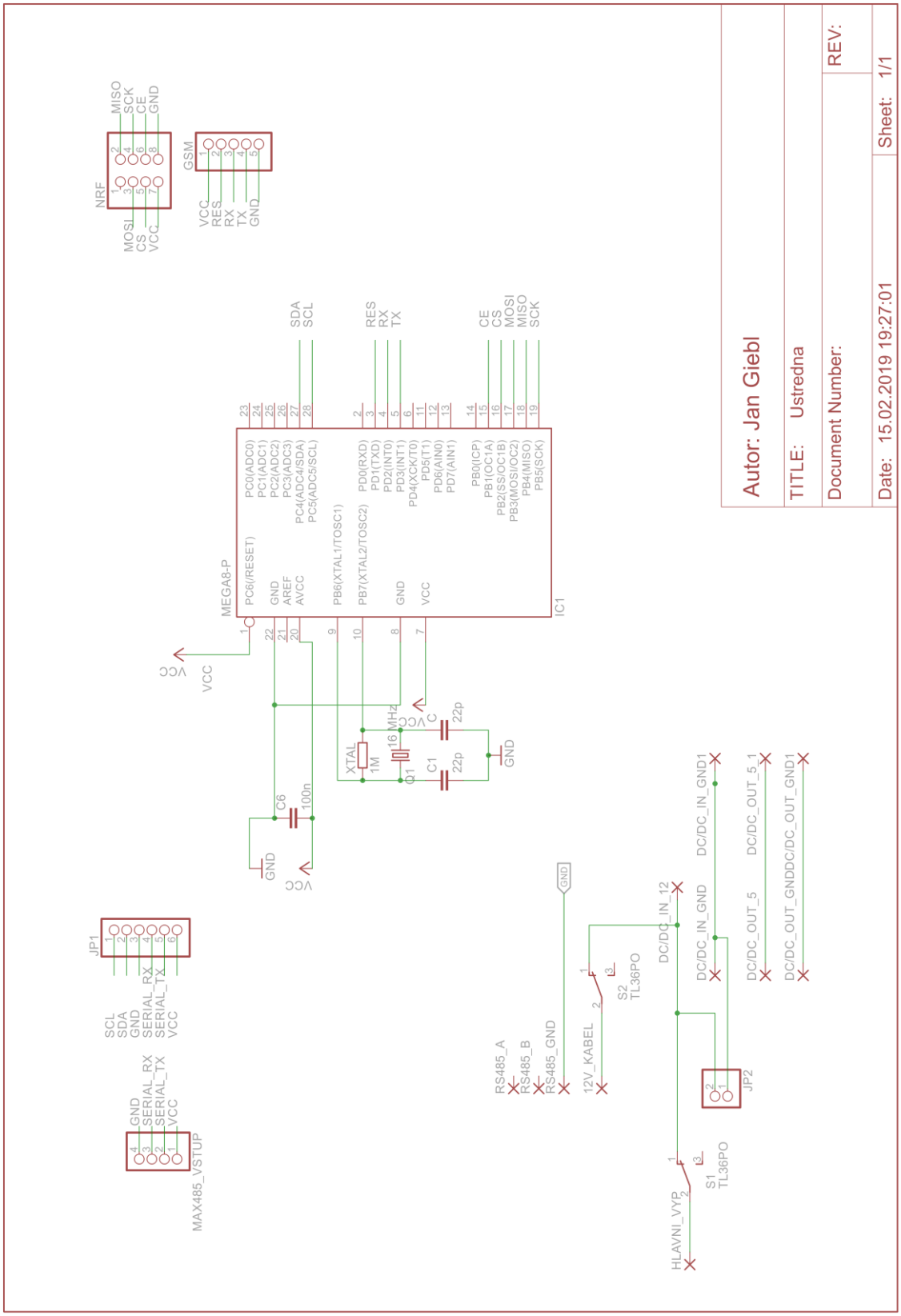
[29] *Node-RED official website [online]. [cit. 2019-04-29]. Dostupné z: https://nodered.org/*

[30] VODA, Zbyšek & tým HW Kitchen. Pr*ůvodce světem Arduina.* 1. vyd. Bučovice: Nakladatelství Martin Stříž, 2015. 240 s. ISBN 978-80-87106-90-7

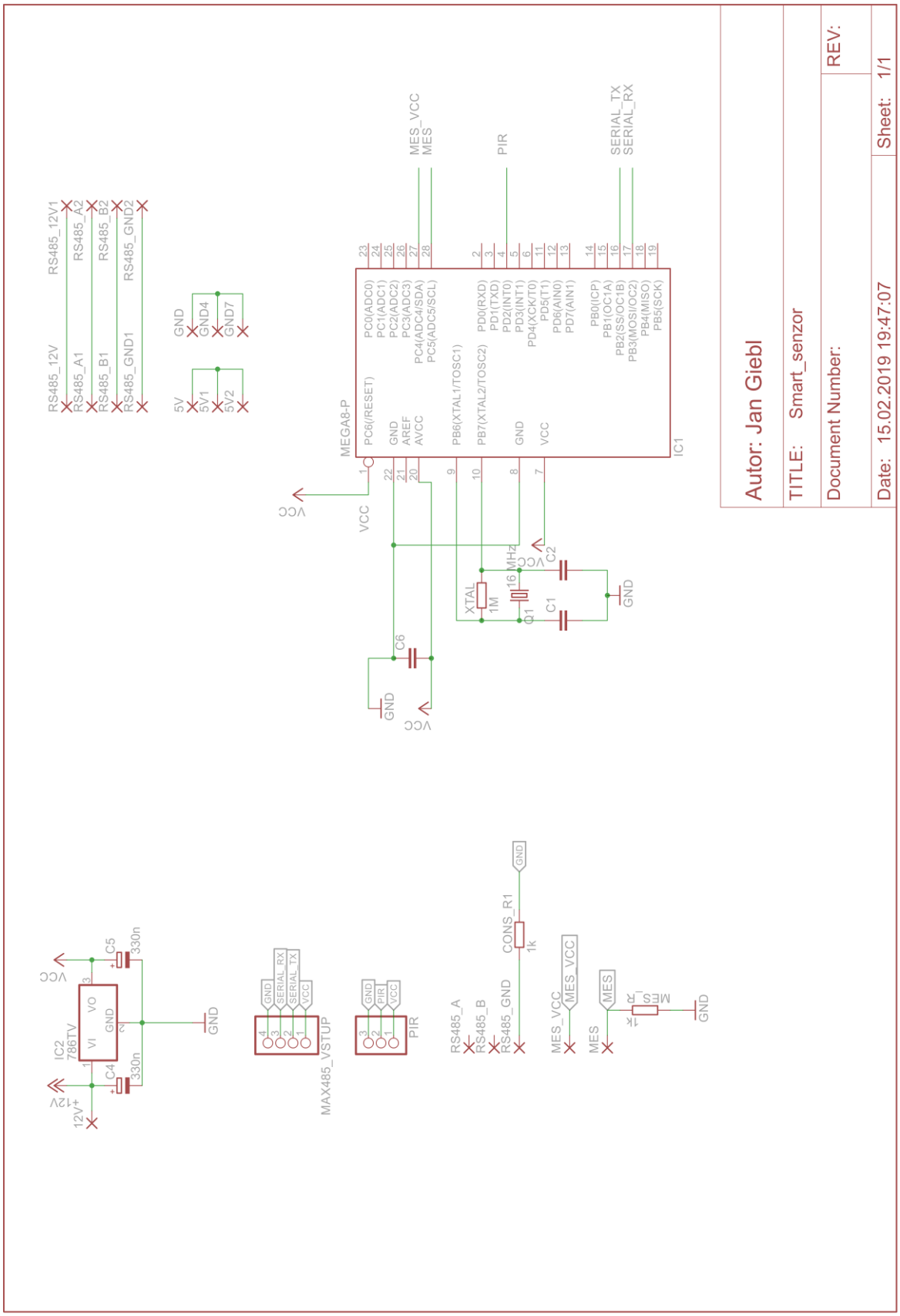
Přílohy

Příloha A – Schémata zapojení

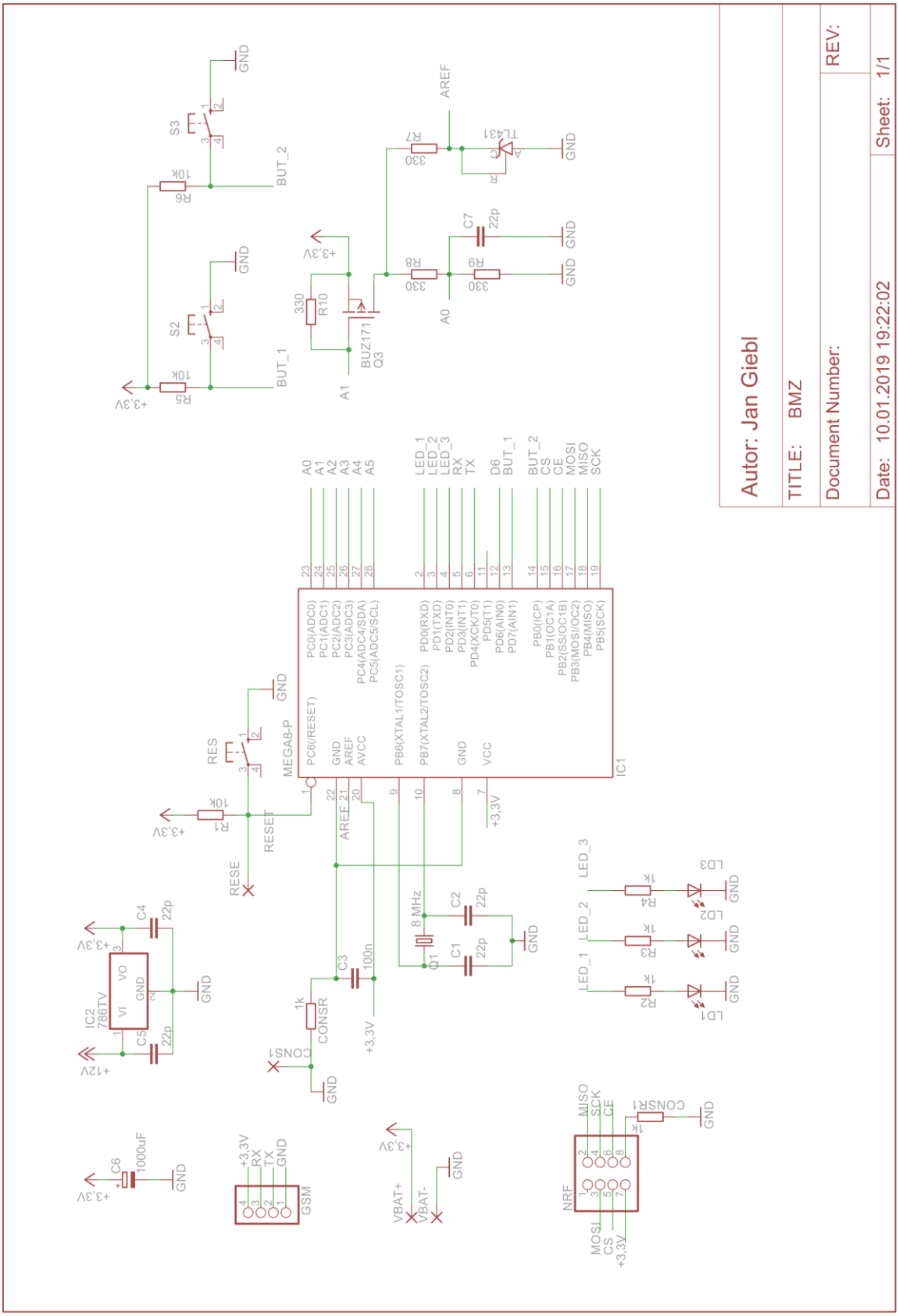
Příloha A1: Schéma zapojení desky plošných spojů pro ústřednu



Příloha A2: Schéma zapojení desky plošných spojů pro Smart senzor

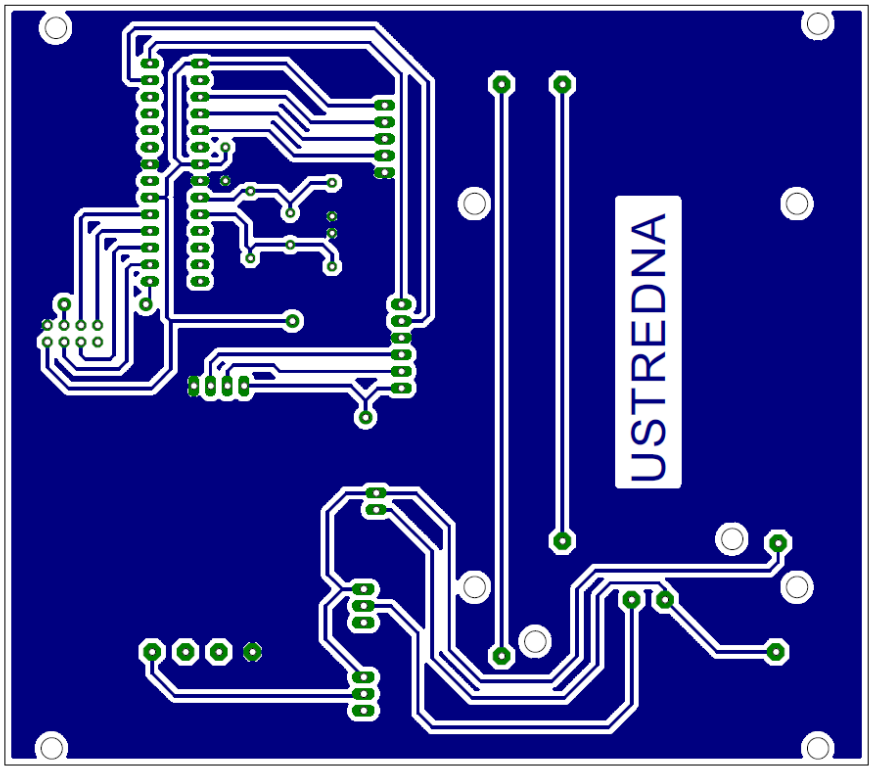


Příloha A3: Schéma zapojení desky plošných spojů BMZ

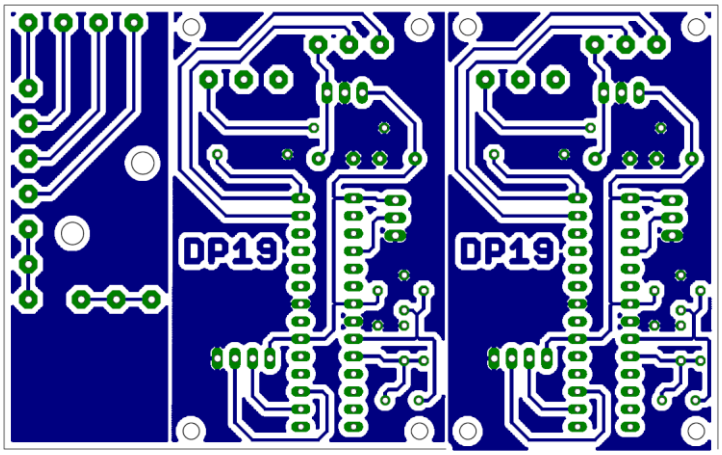


Příloha B – Motivy plošných spojů

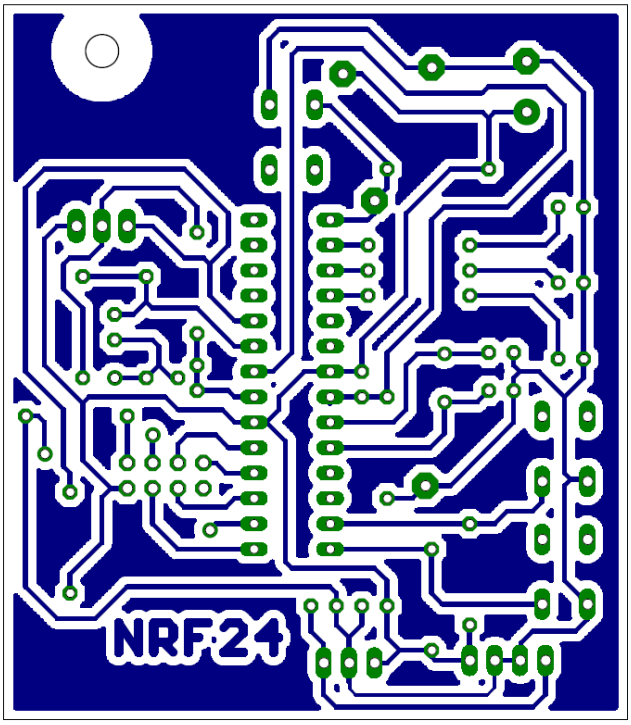
Příloha B1: Plošný spoj pro ústřednu



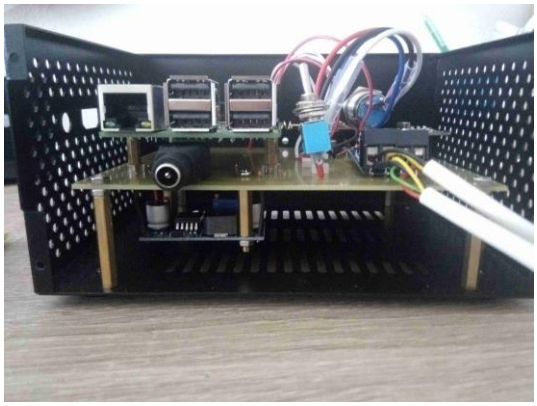
Příloha B2: Plošný spoj pro dva Smart senzory a vlevo menší plošný spoj sloužící k propojení signálů uvnitř ovládacího panelu

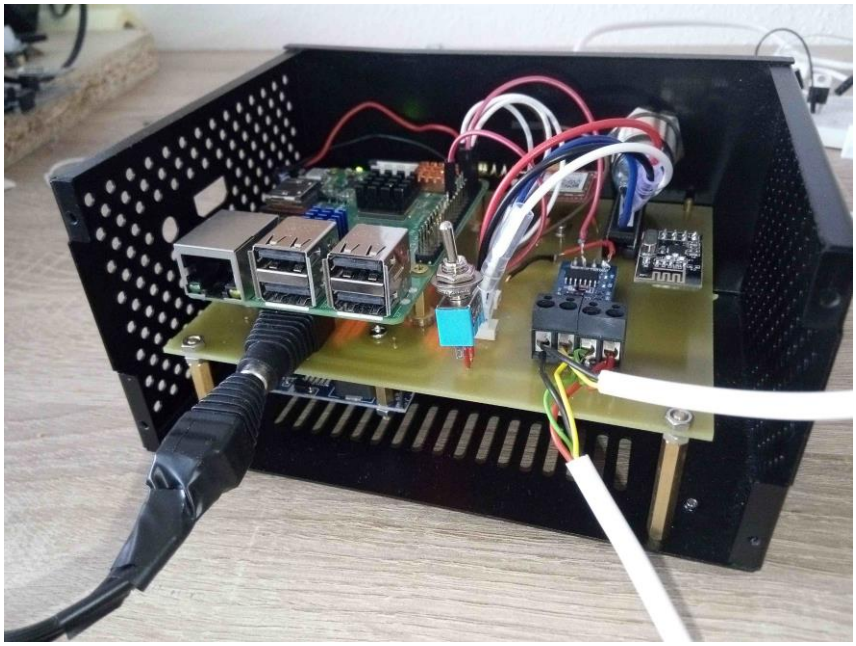


Příloha B3: Plošný spoj pro bezdrátové monitorovací zařízení

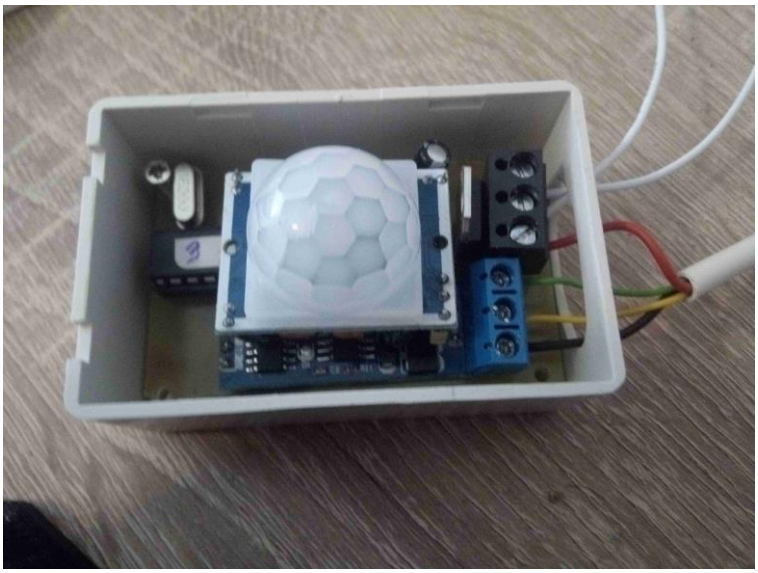


Příloha C – Osazování ústředny



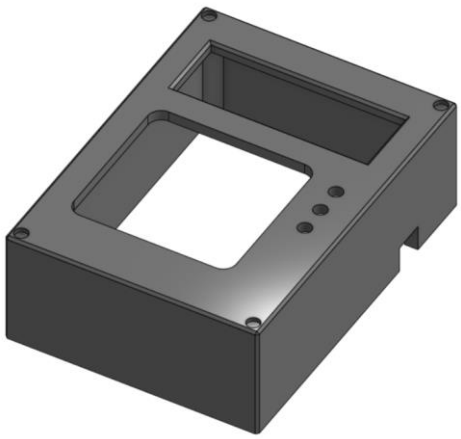


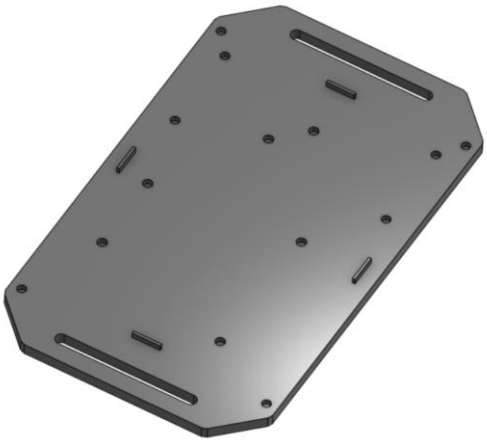
Příloha D – Osazování Smart senzoru



Příloha E – Návrh krabičky a osazování ovládacího panelu

Příloha E1: Návrh krabičky v programu Onshape





Příloha E2: Osazování ovládacího panelu