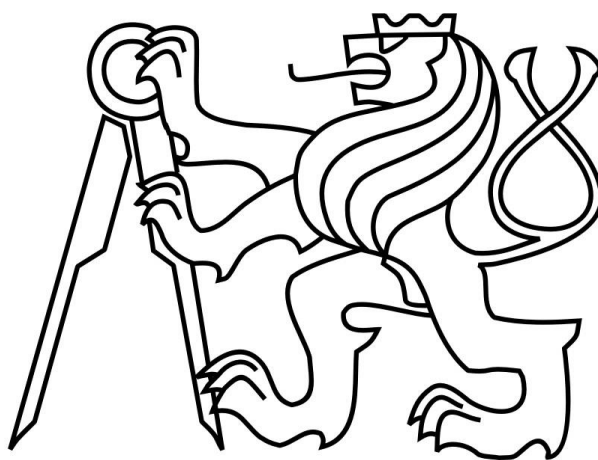


České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra mikroelektroniky



Bakalářská práce

Hybridní systém domácí automatizace

Autor: Michal Fuxa

Vedoucí práce: Ing. Vladimír Janíček, Ph.D.

2018

Čestné prohlášení

„Prohlašuji, že jsem tuto práci vypracoval samostatně s přispěním vedoucího práce a použil jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

V Praze dne 24. 5. 2018

Michal Fuxa

Poděkování

„Rád bych touto cestou poděkoval vedoucímu práce doktoru Janíčkovi za cenné rady a připomínky v průběhu práce. Dále bych rád poděkoval rodině za podporu a neskonalou trpělivost.“

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Fuxa** Jméno: **Michal** Osobní číslo: **439571**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra mikroelektroniky**
Studijní program: **Komunikace, multimédia a elektronika**
Studijní obor: **Aplikovaná elektronika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Hybridní systém domácí automatizace

Název bakalářské práce anglicky:

Hybrid system of home automation

Pokyny pro vypracování:

- 1) Prostudujte problematiku domácí automatizace a stávající nabídky produktů na trhu.
- 2) Navrhněte systém s centrální jednotkou a drátovými a bezdrátovými perifériemi.
- 3) Realizujte funkční vzorek ve formě demonstračního tabla.
- 4) Otestujte parametry produktu.
- 5) Zhodnoťte dosažené výsledky.

Seznam doporučené literatury:

- 1) Schwartz M. - Home Automation with the Raspberry Pi: Build Home Automation Systems Using The Power of The Raspberry Pi, ASIN: B00S6QL7DY
- 2) Monk S. - Raspberry Pi Cookbook: Software and Hardware Problems and Solutions, ASIN: B01FV5OTRI
- 3) Hoile C. - Raspberry Pi and AVR Projects: Augmenting the Pi's ARM with the Atmel ATmega, ICs, and Sensors (Make), ASIN: B00PCZJGWQ

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Vladimír Janíček, Ph.D., katedra mikroelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **26.09.2017**

Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **28.02.2019**

Ing. Vladimír Janíček, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Rípků, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Anotace

Bakalářská práce se zabývá problematikou inteligentních elektroinstalací a analyzuje situaci na trhu. Hlavním cílem práce je návrh a realizace vlastního řešení. V první části je vysvětlen pojem inteligentní elektroinstalace a je provedeno porovnání s klasickou elektroinstalací. Následuje analýza komerčně dostupných systémů. Druhá část se již věnuje samotnému návrhu a realizaci systému. Vývoj probíhá od stanovení klíčových parametrů přes návrh schéma zapojení, desek plošných spojů až po samotnou výrobu funkčních prototypů a oživení pomocí vlastního firmware. Poslední částí je konstrukce demonstračního modelu, který ukazuje základní principy funkce systému.

Klíčová slova: inteligentní elektroinstalace, analýza trhu, hardware, firmware

Abstract

Topic of this Bachelor thesis is problematic of intelligent wiring systems and market analysis. Main goal of this thesis is design and realization of own solution. First part is focused on explanation of term of intelligent wiring system and comparison with traditional wiring systems. The second part describes the design and implementation of the system. The development starts from the determination of the key parameters and proceeds through the design of the wiring diagram, the printed circuit boards to the production of functional prototypes and firmware development. The last part presents construction of demonstration model. Model shows the basic principles of system functionality.

Key words: intelligent wiring system, market analysis, hardware, software

Obsah

Seznam použitých zkratk a symbolů	8
Seznam tabulek	9
Seznam rovnic	9
1. Teoretický rozbor	11
1.1. Inteligentní elektroinstalace	11
1.1.1. Vysvětlení pojmu	11
1.1.2. Klasická vs. inteligentní elektroinstalace	11
1.2. Loxone	12
1.2.1. Miniserver a Extensions	12
1.2.2. Miniserver Go a Loxone Air	13
1.3. ABB	14
1.3.1. ABB Ego-n®	14
1.3.2. ABB i-bus® KNX	14
1.4. Siemens	15
1.4.1. LOGO!	15
1.5. OpenMotics	15
1.5.1. Distribuce	16
1.6. Papouch	16
1.6.1. Řada Quido	16
1.6.2. Papago	17
1.7. Philips HUE	18
2. Návrh vlastního řešení	19
2.1. Vlastnosti systému	19
2.1.1. DIY stavebnice	19
2.1.2. Arduino kompatibilní	20
2.1.3. Modulární systém	20
2.1.4. Centralizovaný systém	20
2.1.5. Hybridní systém	21
2.2. Komunikační technologie a topologie	21
2.2.1. Bezdrátová komunikace	21
2.2.2. RS485	22
2.2.3. Topologie	22
2.3. Návrh komunikačního protokolu RS485	23
2.3.1. Tvar zprávy	23
2.3.2. Zpracování zprávy	24
2.3.3. Typy zpráv (instrukce)	24
2.3.4. Kontrolní součet	25
3. Hardware	26
3.1. Řídící jednotka	26
3.1.1. Napájení	27
3.1.2. ATmega1284p (MCU)	27
3.1.3. Raspberry Pi Zero W	28
3.1.4. NRF24L01 (RF transceiver)	28
3.1.5. Obvod reálného času DS1307 (RTC)	28
3.2. IO modul	29
3.2.1. Digitální vstup	29
3.2.2. Digitální výstup	30

3.2.3.	Digitální teplotní senzor Dallas (Maxim).....	31
3.3.	Univerzální bezdrátový uzel (node)	32
3.3.1.	ATmega328p (MCU)	32
3.3.2.	Senzor teploty a vlhkosti DHT22	33
3.3.3.	Řízení spotřeby.....	33
3.3.4.	Vlastní spotřeba.....	34
3.4.	Bezdrátové relé	34
3.4.1.	Napájení	35
4.	Programová část (firmware)	36
4.1.	Implementace protokolu (RS485)	36
4.1.1.	Inicializace	36
4.1.2.	Příjem zprávy	36
4.1.3.	Kontrola zprávy.....	36
4.1.4.	Odeslání zprávy.....	37
4.2.	Bezdrátová komunikace (RF 2.4 GHz)	38
4.2.1.	Základní popis knihovny RadioHead	38
4.2.2.	RadioHead Driver	38
4.2.3.	RadioHead Reliable Datagram.....	38
4.3.	Řídící jednotka (deska s MCU)	39
4.3.1.	Popis hlavní smyčky.....	40
4.3.2.	Inicializace periférií a alarmů.....	41
4.3.3.	Zpracování zpráv protokolu RS485	41
4.3.4.	Zpracování zpráv knihovny RadioHead.....	42
4.3.5.	Externí příkazy a Raspberry Pi.....	42
4.4.	IO modul.....	43
4.5.	Univerzální bezdrátový uzel.....	44
4.5.1.	Bezdrátový vypínač.....	44
4.5.2.	Bezdrátový senzor teploty	45
4.6.	Bezdrátové relé	46
4.7.	Webové rozhraní.....	47
5.	Demonstrační model.....	48
5.1.	Základní konstrukce	48
5.1.1.	Schéma a volba komponent.....	48
5.2.	Fyzické zapojení	49
5.2.1.	Napájení modelu	49
5.2.2.	Univerzální bezdrátové uzly.....	49
5.2.3.	Bezdrátové relé.....	49
	Závěr.....	50
	Seznam použité literatury a zdrojů	51
	Seznam příloh.....	53

Seznam použitých zkratek a symbolů

AC	Alternating current
ACK	Acknowledgement
ARQ	Automatic Repeat-reQuest
ASCII	American Standard Code for Information Interchange
DC	Direct current
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPIO	General-purpose input/output
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IO	Input/Output
LED	Light-Emitting Diode
LF	Line Feed
MCU	Microcontroller unit
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
OLED	Organic light-emitting diode
OSH	Open-source hardware
PCB	Printed circuit board
RFID	Radio Frequency Identification
RTC	Real-time clock
SPI	Serial Peripheral Interface
TI	Texas Instruments
UART	Universal asynchronous receiver-transmitter
WDT	Watchdog timer
Wi-Fi	Wireless Fidelity

Seznam obrázků

Obr. 1 - Graf závislosti nákladů na výkonnosti elektroinstalace [1]	12
Obr. 2 - Instalace Miniserveru a rozšiřujících modulů Extensions [2].....	13
Obr. 3 - Miniserver Go [2]	13
Obr. 4 - Řídicí a rozšiřující modul LOGO! [4]	15
Obr. 5 - OpenMotics Gateway Module [5]	16
Obr. 6 - Quido se čtyřmi vstupy a čtyřmi výstupy s komunikací po RS485 [6]	17
Obr. 7 - Papago s ethernetem a dvěma teploměry [6]	17
Obr. 8 - Příklad sestavy s Philips HUE [7].....	18
Obr. 9 - Topologie systému.....	22
Obr. 10 - Blokové schéma hybridní řídicí jednotky	26
Obr. 11 - rozložení pinů MCU ATmega1284p [8].....	27
Obr. 12 - SMD modul s klonem NRF24L01	28
Obr. 13 - Blokové schéma IO modulu.....	29
Obr. 14 - zapojení digitálního vstupu	30
Obr. 15 - zapojení digitálního výstupu	30
Obr. 16 - Zapojení teplotního senzoru DS18B20 a) dvou vodičově a b) tří vodičově	31
Obr. 17 - Blokové schéma univerzálního bezdrátového uzlu.....	32
Obr. 18 - Rozložení pinů MCU ATmega328p [13]	33
Obr. 19 - Blokové schéma bezdrátového relé	35
Obr. 20 - Zapojení vstupních a výstupních svorek modulu.....	35
Obr. 21 - Principiální vývojový diagram firmware řídicí jednotky.....	40
Obr. 22 - Vývojový diagram firmware IO modulu	43
Obr. 23 - Vývojový diagram bezdrátového vypínače	45
Obr. 24 - Vývojový diagram bezdrátového teplotního senzoru	46
Obr. 25 - Vývojový diagram bezdrátového relé	47
Obr. 26 - Schéma zapojení demonstračního modelu.....	48

Seznam tabulek

Tab. 1 - Přehled tvaru zápravy protokolu.....	23
---	----

Seznam rovnic

(1) - Rovnice výpočtu kontrolního součtu	27
--	----

Úvod

Již od samotného počátku se snaží člověk pro sebe najít to největší pohodlí a nebojí se investovat nemalé prostředky a úsilí. V posledních letech se čím dál tím více rozšiřují tzv. Inteligentní budovy a rodinné domy. Tento fakt mě přesvědčil se o tuto problematiku začít více zajímat, a to zejména z hlediska řídicí elektroniky a inteligentních elektroinstalací. Překvapila mě především cena těchto systémů, díky které je jejich použití až na výjimky otázkou velkého luxusu. Hlavním cílem mé práce je vyvinout vlastní systém inteligentní elektroinstalace, navrhnout a vyrobit první prototypy vlastního hardware, naprogramovat prvotní verze firmware s implementovaným komunikačním protokolem a v neposlední řadě také vytvoření funkčního modelu inteligentního domu, který bude sloužit k ověření funkčnosti a propojení jednotlivých částí. Cílem není vytvořit nový komerční systém, ale pokusit se na základě analýzy trhu navrhnout alternativní přístup k této problematice.

1. Teoretický rozbor

Tato kapitola se zabývá osvětlením pojmu inteligentní elektroinstalace, a především ukazuje výběr z aktuálně dostupných systémů.

1.1. Inteligentní elektroinstalace

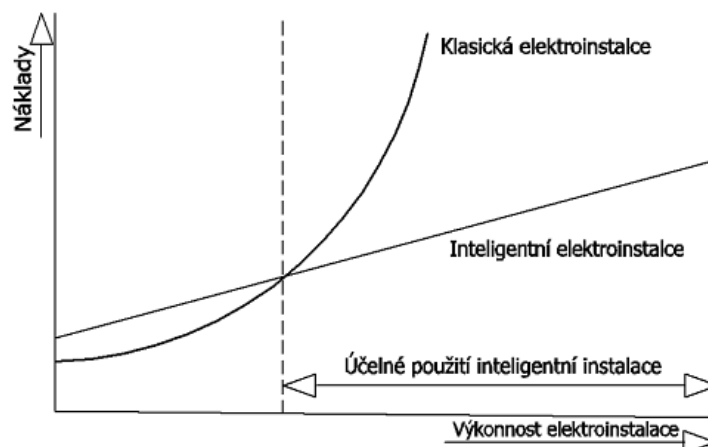
Aby bylo možné dále pracovat s pojmem inteligentní elektroinstalace, je nutné si jeho význam nejprve rozebrat.

1.1.1. Vysvětlení pojmu

Inteligentní elektroinstalace je systém, který se v budově (např. v rodinném domě) stará o ovládání a řízení zařízení a procesů. Charakteristickým rysem inteligentní elektroinstalace je komunikace mezi jednotlivými prvky systému. Kromě silových signálů se zde tedy přenáší i data. Jedná se zpravidla o modulární systém a jednotlivé prvky spolu komunikují např. po sběrnici. Výhodou inteligentní elektroinstalace je fakt, že spolu jednotlivé řízené prvky nemusí být přímo propojené, díky tomu se dají řešit i poměrně složité konstrukce. Typickými funkcemi jsou například ovládání světel, žaluzií, měření a regulace klimatických podmínek, bezpečnostní prvky apod. Toto jsou parametry, které mohou mít příznivý vliv na komfort uživatele a na provozní náklady domu. Nejmodernější systémy nabízí například také spojení s multimediálním systémem. [1]

1.1.2. Klasická vs. inteligentní elektroinstalace

Klasická elektroinstalace žádnými pokročilými funkcemi nedisponuje. Všechny prvky jsou propojeny napřímo, nenachází se zde žádná řídicí elektronika a nepřenáší se žádná data. To znamená především nižší pořizovací náklady prvků elektroinstalace. Nižší pořizovací cena však platí pouze pro jednoduché zapojení a projekty. Se složitějšími konstrukcemi začíná cena strmě stoupat. V určitou chvíli se tak můžeme do bodu, kdy se vyplatí více pořízení inteligentní elektroinstalace. Tuto situaci lze znázornit jednoduchým grafem, který je znázorněn na obr. 1. [1]



Obr. 1 - Graf závislosti nákladů na výkonnosti elektroinstalace [1]

1.2. Loxone

Rakouská firma Loxone nabízí komplexní řešení pro chytrý dům ve formě vlastní chytré elektroinstalace. I přesto, že se jedná o jedinou platformu, je schopna díky své velké všestrannosti naleznou uplatnění pro velké množství projektů. Funkcionalita systému je opravdu široká, kromě standardního ovládání světel, žaluzií, regulace teploty nabízí například systém pro úsporu energie, ovládání fotovoltaiky, zabezpečení, ovládání bazénového systému, ale také například ovládání sauny. Celkově umožňuje systém tři způsoby propojení, které mohou fungovat současně. Jedná se o asi nejrozšířenější sběrníkovou topologii, pro speciální řadu periferií je to topologie stromová (Loxone Tree) a v neposlední řadě bezdrátová technologie Loxone Air. Díky tomu se systém hodí jak pro rekonstrukce, tak pro novostavby. Loxone nabízí řešení pro rodinný dům řádově za desítky tisíc korun. [2]

1.2.1. Miniserver a Extensions

Miniserver (Obr. 2) je základním stavebním kamenem celého systému. V základním provedení se jedná o devíti-modulovou krabičku určenou pro montáž na DIN lištu. Jedná se vlastně o řídicí jednotku se základními vstupy (8 digitálních a 4 analogové) a výstupy (8 digitálních a 4 analogové). Dále obsahuje LAN konektor pro připojení do sítě. A kromě Loxone sběrnice pro komunikaci s moduly obsahuje i konektor KNX pro standardizovanou komunikaci. Loxone je modulární systém. Extensions označuje celou škálu rozšíření připojitelných na Loxone sběrnici. Výchozí možností je rozšíření počtu vstupů a výstupů. Toto rozšíření se jmenuje Extension. Nabízí dvojnásobný počet vstupů a výstupů jako základní Miniserver (plus 2 digitální vstupy navíc). Je určený všude tam, kde nestačí základní nabídka vstupů a výstupů. V podobném duchu se nesou i rozšíření o

výkonová relé (Relay Extension) a stmívače (Dimmer Extension). Loxone nabízí i celou řadu převodníků úrovní a protokolů. V nabídce nechybí převodník na protokol Modbus, řízení osvětlení DALI nebo DMX, sběrnice 1-Wire, RS232 apod. [2]



Obr. 2 - Instalace Miniserveru a rozšiřujících modulů Extensions [2]

1.2.2. Miniserver Go a Loxone Air

Miniserver Go (Obr. 3) je odlehčená varianta Miniserveru určená především pro rekonstrukce. Tato řídicí jednotka je totiž navržena pouze pro použití bezdrátových periférií. Kromě napájení přes micro USB už obsahuje pouze LAN konektor. Miniserver Go komunikuje pouze s bezdrátovými perifériemi Loxone Air. Jedná se například o chytrá kapacitní tlačítka, I/O modul zabudovatelný pod omítku, bezdrátové zásuvky, detektory kouře apod. [2]



Obr. 3 - Miniserver Go [2]

1.3. ABB

Firma ABB je na našem trhu známa především jako dodavatel elektro instalačního materiálu pro průmysl i domácnosti. V jejím portfoliu se však najdou i řady inteligentních elektroinstalací.

1.3.1. ABB Ego-n®

Ego-n® je centralizovaný sběrniceový systém liniové topologie. Umožňuje propojit až 512 zařízení. Systém se skládá ze dvou typů sběrnic. Použita je tzv. primární sběrnice, ta slouží k propojení akčních členů a snímačů (maximálně 64). Každá primární sběrnice obsahuje řídicí jednotku a jednotku napájení. V systému může být až 8 primárních sběrnic. Řídicí jednotky jednotlivých sběrnic jsou propojeny tzv. sekundární sběrnici (typicky mezi rozvaděči). Ta dále zprostředkovává komunikaci s okolím pomocí různých převodníků GSM, Wi-Fi apod.). Zařízení na primární sběrnici jsou propojena kabelem se čtyřmi vodiči. Některá zařízení na sběrnici mají možnost komunikovat s dalšími zařízeními bezdrátovou technologií. Typicky se jedná tlačítkové snímače a hlásiče kouře. Nabídka akčních členů a snímačů je velmi rozmanitá. Od základního ovládání světel, přes stmívače, regulaci teploty může systém řešit i zabezpečení objektu. Ego-n® není řešení pro plně autonomní inteligentní dům, jedná se spíše o chytřejší náhradu tradiční elektroinstalace. Cena projektu s tímto systémem se pohybuje od desítek do stovek tisíc korun. [3]

1.3.2. ABB i-bus® KNX

I-bus® je decentralizovaný systém inteligentní domácnosti založený na mezinárodním standardu pro inteligentní domácnosti KNX. Jedná se o komunikační protokol na vlastní sběrnici. Decentralizace v tomto případě znamená, že po řádné konfiguraci komunikují snímače přímo s konkrétními akčními členy. Není třeba řídicích jednotek ani PC. Všechny konfigurace (programy) jsou uloženy přímo v jednotlivých zařízeních. Sběrnice KNX umožňuje pracovat v několika liniích (odbočkách). Každá linie pojme až 64 zařízení (s opakovači až 256). Linií může být celkem 15. Systém je konstruován pro automatický provoz budovy, automatické ovládání světel, vytápění, větrání, ovládání žaluzií, vrat, dveří apod. Vše v rámci ekologických a ekonomických požadavků uživatele. Je zde řešení pro většinu oblastí. Dále se pak jedná o spolupráci se zabezpečovacím systémem, vytváření dojmu přítomnosti, protokolování a hlášení událostí. Díky svému rozsahu se hodí především pro velké budovy a komerční projekty velkého měřítka, administrativní budovy a haly. Cena projektu s tímto systémem se pohybuje ve stovkách tisíc korun. [3]

1.4. Siemens

Siemens je německý výrobce elektroniky, jeden z největších výrobců na světě. Zabývá se výrobou Domácích spotřebičů, domácí i průmyslové automatizace, výrobou motor, lokomotiv, železniční automatizace a mnoho dalšího.

1.4.1. LOGO!

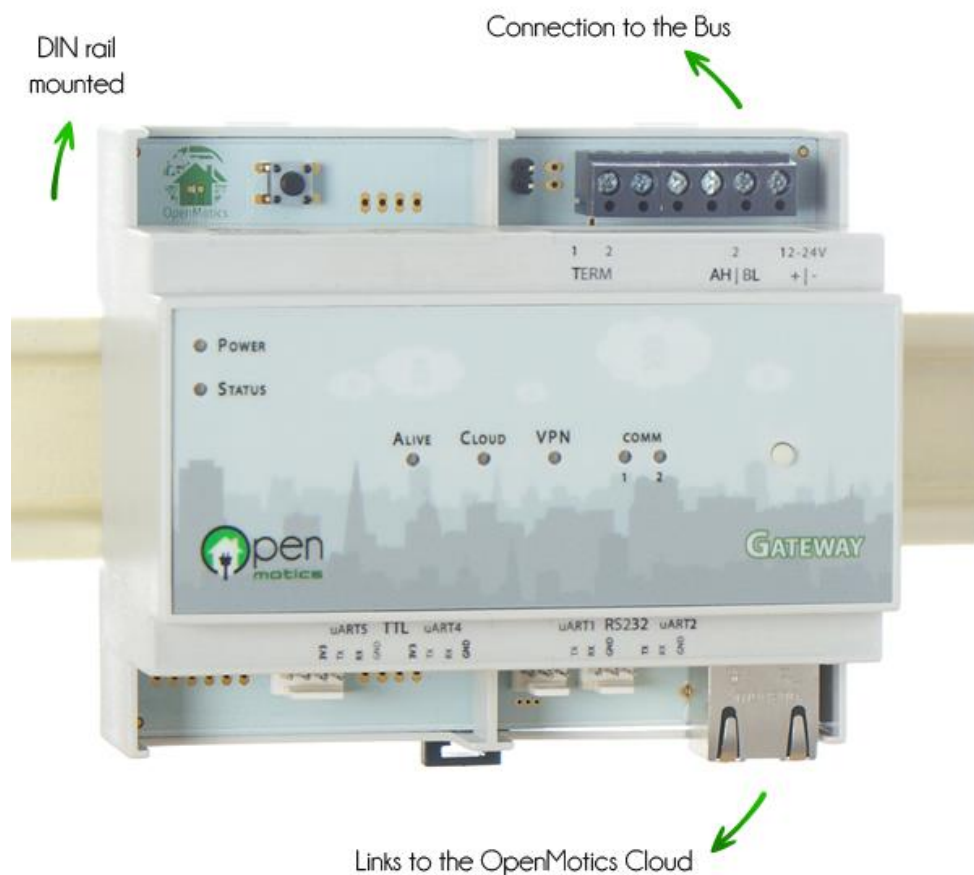
Zajímavou alternativou ke klasickým systémům domácí automatizace je logický systém Siemens LOGO! (Obr. 4). Nejedná se primárně o systém inteligentní elektroinstalace, jedná se o programovatelný logický automat (PLC). LOGO! je kompaktní systém, tzn. V jedné krabici je obsažena jak řídicí část, které se běžně říká CPU, tak programovatelné vstupy a výstupy. Kromě toho je v nabídce celá řada rozšiřujících modulů s dalšími vstupy a výstupy (i analogovými), komunikační jednotky (např. pro sběrnici KNX). Řídicí software je nutné kompletně naprogramovat v dodávaném software (Například symbolicky pomocí logických obvodů). Výhodou takového řešení je především nízká pořizovací cena a vysoká variabilita. Pořizovací cena systému je přibližně 5000 Kč za řídicí modul. Nenáročný projekt pak může vyjít zhruba na 10-20 tisíc Kč. [4]



Obr. 4 - Řídicí a rozšiřující modul LOGO! [4]

1.5. OpenMotics

OpenMotics je open source systém inteligentní elektroinstalace. Nabízí standardní funkcionalitu v podobě ovládání světel, měření teploty, vytápění nebo měření spotřeby. Jedná se o modulární sběrníkový systém (RS485) s vlastním komunikačním protokolem. Řídicí jednotka (Gateway Module - Obr. 5) obsahuje kromě master desky s MCU PIC také malý ARMový mikropočítač BeagleBone, který slouží k propojení systému s vlastním cloudem. [5]



Obr. 5 - OpenMotics Gateway Module [5]

1.5.1. Distribuce

Systém OpenMotics lze koupit, jako standardní komerční systém. Rozdíl je v tom, že jsou k dispozici všechny zdrojové podklady jako je schéma zapojení, plošný spoj, seznam součástek i kompletní firmware. Tyto podklady je možné upravovat a vyvíjet tak například vlastní kompatibilní modul nebo si upravit firmware. [5]

1.6. Papouch

Papouch je česká firma zabývající průmyslovou i domácí automatizací. Kromě zakázkového návrhu hardware nabízí i stálou nabídku hardware pro domov i průmyslové použití.

1.6.1. Řada Quido

Quido (Obr. 6) je řada modulů s digitálními vstupy a výstupy. Vyrábí se různé modely s různými počty reléových výstupů a opticky oddělených vstupů (případně další varianty). Součástí

modulů je i možnost připojení teplotního čidla. Moduly jsou určeny pro připojení k nadřazenému systému. Na výběr je z variant používajících USB, RS232/RS485 nebo Ethernet. Komunikace probíhá pomocí vlastního protokolu Spinel, případně je možno přepnout na protokol Modbus. Standardně jsou k výrobku dodávány aplikace pro řízení modulů z PC. [6]



Obr. 6 - Quido se čtyřmi vstupy a čtyřmi výstupy s komunikací po RS485 [6]

1.6.2. Papago

Řada jednoúčelových modulů se vzdáleným přístupem. Každý modul je vybaven přepojením přes ethernet (Obr. 7) nebo Wi-Fi. Jednotlivé moduly se vyrábí v několika variantách podle osazených periférií. Na výběr jsou varianty se dvěma teplotními senzory, kombinovanými senzory (teplota, vlhkost, rosný bod), digitální vstupy a výstupy, čtečka RFID apod. Ethernetová verze je navíc kompatibilní s technologií PoE. Výstup je realizován přes webové rozhraní. [6]



Obr. 7 - Papago s ethernetem a dvěma teploměry [6]

1.7. Philips HUE

HUE (Obr. 8) je jednoúčelová sada zařízení pro inteligentní osvětlení. Jedná se chytré LED žárovky, barevné osvětlení, lampy, stropní osvětlení. Srdcem systému je pak tzv. Bridge, který sjednocuje všechna světla a umožňuje připojení systému do sítě a na internet. Nevýhodou může být omezený počet prvků osvětlení, které výrobce, kromě žárovek se standardními paticemi, nabízí. [7]



Obr. 8 - Příklad sestavy s Philips HUE [7]

2. Návrh vlastního řešení

Na základě získaných informací bylo na řadě zabývat se podobou a návrhem vlastního systému. V první řadě je nutné stanovit požadavky na vlastnosti takového systému. Dalším bodem návrhu je výběr základních technologií a sestavení výchozí topologie systému.

2.1. Vlastnosti systému

Od začátku se zabývám myšlenkou vytvořit vlastní systém inteligentní elektroinstalace, který by svými funkcemi a možnostmi mohl konkurovat relativně drahým komerčním systémům. Mé vlastní požadavky na vlastnosti systému vycházejí z obecných požadavků pro tyto systémy. Zároveň by však měl systém nabídnout nové technologie nebo alespoň inovativní využití těch stávajících.

2.1.1. DIY stavebnice

Tento projekt nemá ambice vytvořit finální komerční produkt. Hlavním rysem je cílení spíše na pokročilé uživatele formou personifikovatelné stavebnice. Systém tedy nemá striktně definované funkce. Je pouze na uživateli, jaké funkce do systému implementuje, jestli použije předpřipravený firmware nebo pouze doporučené knihovny. Tento přístup je vhodný především pro uživatele, kteří mají rádi plnou kontrolu nad instalovaným systémem.

Dlouhodobým cílem přesahujícím rámec bakalářské práce je uvolnění systému v rámci open-source hardware (OSH) a open-source software. Zapojení komunity přináší nové možnosti, jako například nové nápady, odezva od dalších vývojářů a možnost návrhu vlastních modulů.

2.1.2. Arduino kompatibilní

S předchozí podkapitolou úzce souvisí zachování kompatibility s velmi oblíbenou platformou Arduino. Platforma má velkou komunitu uživatelů a téměř nekonečné množství knihoven, návodů a vzorových programů. To je pro systém je pro tento systém velmi stabilní stavební kámen. Arduino je kompletní platforma pro snadný vývoj vlastního hardware. Platforma je původně založena na vývojovém kitu s mikrokontrolerem AVR. Nejznámější Arduino deska, Arduino Uno, je osazena MCU ATmega328p. Dnes platforma Arduino podporuje celou řadu desek i s MCU mimo rodinu AVR. Spolu s ním je součástí platformy také společný jazyk založený na jazyku C/C++ (vlastně se jedná pouze o knihovnu pro tento jazyk) a vývojové prostředí Arduino IDE a bootloader, to je malý program v paměti MCU, který slouží pro zavedení firmware pomocí rozhraní UART. Arduino tedy nevyžaduje programátor. Vývojové prostředí je vlastně jednoduchý textový editor s kompilátorem, správou knihoven a kompatibilních Arduino desek.

2.1.3. Modulární systém

Jedním ze základních požadavků na budoucí hardware systému je jeho složení z jednotlivých modulů. Modulární systém je totiž velice výhodný, a to hned z několika hledisek. Již nainstalovaný systém můžeme rozšiřovat i v průběhu jeho používání. Nový modul pouze připojíme k ostatním modulům a nastavíme (naprogramujeme) jeho funkci. Další výhodou je možnost vytváření dalších typů modulů v době, kdy už je systém nějakou dobu dostupný. Můžeme tak snadno vyhovět i požadavků, které vzniknou až v době, kdy už je systém dávno na trhu nebo reagovat na vznik nových technologií. Tak můžeme snadno vyhovět různorodým požadavkům zákazníků. Každý si může svůj projekt nakonfigurovat dle svých představ. V případě poruchy můžeme vyměnit pouze vadný modul. To nám šetří peníze a hlavně čas (nemusíme instalovat znovu celý systém).

2.1.4. Centralizovaný systém

Systém by měl být koncipován jako centralizovaný. Znamená to především přítomnost řídicí jednotky jakožto nadřazeného prvku, který kontroluje a řídí chod podřízených modulů. Jedná se o klasický přístup řízení v průmyslové i domácí automatizaci. Všechny vstupy do systému jsou vyhodnoceny centrální (řídicí) jednotkou a je jim následně přiřazena odpovídající akce (odpovídající výstup). Tento přístup umožňuje použití jednoduššího firmware pro funkční moduly, klíčový je zde hlavně firmware řídicí jednotky. Toto šetří práci uživateli při vývoji nebo úpravě existujícího firmware. Zároveň to snižuje nároky na výpočetní výkon jednotlivých modulů a tím i cenu.

Nejmodernější, a především dražší systémy používají decentralizované řešení. Není zde tedy žádná řídicí jednotka a moduly spolu komunikují napřímo. Toto znamená obrovské nároky především na zajištění bezkolizního provozu a také složitější komunikační protokol. Pro systém, kde je hlavním parametrem jednoduchá úprava a vývoj firmware, je tento způsob řízení jeví jako nevhodný. [16]

2.1.5. Hybridní systém

Hlavní myšlenka celého systému je postavena kolem pojmu „hybridní“. Slovo „hybridní“ v tomto případě znamená, že základní stavební kámen celého systému (řídicí jednotka) by měl být vybaven jak bezdrátovou, tak sběrníkovou (drátovou) komunikací a zároveň by měl být osazen jednodeskovým ARM počítačem pro běh webového rozhraní a zpracování nasbíraných dat (nikoliv řízení). Hlavní výhodou takového systému spočívá v tom, že pokud uživatel jednou navrhne projekt jako čistě drátový, může v budoucnu systém rozšířit o libovolný bezdrátový modul (např. z důvodu absence bourání a zednických prací), aniž by musel dokupovat rozšiřující převodník apod. Vystačí si pouze s již nainstalovanou řídicí jednotkou.

2.2. Komunikační technologie a topologie

Při návrhu nového systému je třeba si ujasnit volbu komunikačních technologií a způsob jejich použití. To je velmi důležité, protože to má zásadní vliv na vlastnosti systému.

2.2.1. Bezdrátová komunikace

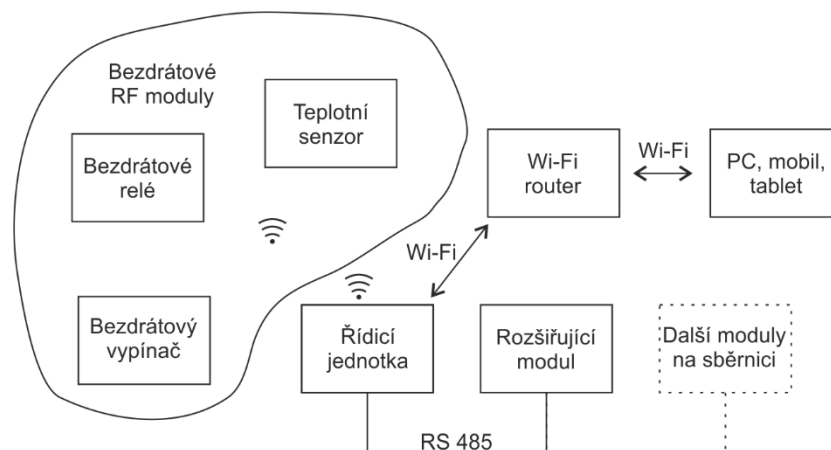
Drátová komunikace má své výhody, ale v rámci inteligentní elektroinstalace může přinášet spoustu nevýhod a problémů. Především se jedná o velké množství kabeláže (a to i té silové). A pokud se jedná například o rekonstrukci (vylepšení), nemusí být instalace nové kabeláže vůbec možná. Z tohoto důvodu pokládám možnost bezdrátové komunikace jako nutnost. Já jsem zvolil velmi levné rádiové moduly s obvodem nRF24L01 od společnosti Nordic Semiconductors. Tyto moduly pracují v pásmu 2,4 GHz. Oproti běžně využívané frekvenci 433 MHz jsou finančně dostupnější a nabízejí vyšší komunikační rychlost. To teoreticky zvyšuje maximální počet zařízení, které může řídicí jednotka obsloužit. Nevýhodou vyšší frekvence je o něco kratší maximální vzdálenost dvou zařízení. Tento problém řeší například použití opakovačů. Případně je možné použít hybridní řešení v podobě topologie typu *mesh*. To znamená bezdrátové propojení všech zařízení mezi sebou.

2.2.2. RS485

Pro sběrniceovou komunikaci byla vybrána velice robustní průmyslová sběrnice RS485. Standard sériové sběrnice mající podobný základ jako sběrnice RS232. RS485 je dvou vodičová (značené jako A a B nebo + a -) polo-duplexní (střídavě obousměrná) sběrnice, určená pro datovou a řídicí komunikaci až pro 32 zařízení (uzlů) na vzdálenost až 1,2km. Vedení by mělo být na obou stranách zakončeno zakončovacími odpory tzv. terminátory, které mají hodnotu 120Ω . Terminátory zabraňují odrazům na vedení a pomáhají omezit další rušivé elementy vznikající především na dlouhém vedení. Na rozdíl od RS232 se jedná o diferenciální vedení, logické úrovně jsou zde reprezentovány rozdílovým napětím mezi oběma datovými vodiči. Pro rozdíl napětí na A a B větší než 200mV je detekována log. 0, při rozdílu A a B menším než -200mV je detekována log. 1.

2.2.3. Topologie

Celá topologie systému se skládá ze dvou hlavních částí. První částí tvoří sběrniceová topologie RS485. Ta je určená především pro multifunkční moduly s určením do domovního rozvaděče. Jedná se například o vstupně-výstupní moduly (digitální i analogové) s více kanály. Pro jednoúčelové moduly, u kterých se počítá s přímo s umístěním do obytného prostoru, je použita bezdrátová komunikace hvězdicové topologie, každý modul tedy komunikuje napřímo s řídicí jednotkou. Pro připojení systému do domácí sítě (případně do internetu) slouží integrovaná síť Wi-Fi. Celou topologii znázorňuje obr. 9.



Obr. 9 - Topologie systému

2.3. Návrh komunikačního protokolu RS485

Komunikaci na sběrnici RS485 je třeba nějak řídit. K tomuto účelu slouží komunikační protokol, tedy vlastně soubor pravidel pro tuto komunikaci. Pro účely tohoto systému byl navržen jednoduchý ASCII protokol. Označení ASCII znamená, že jednotlivé odeslané bajty paketu odpovídají znakům z tabulky ASCII. Výhodou takového protokolu je fakt, že je obsah zpráv snadno čitelný pro lidské oko i v sériovém terminálu a díky tomu je ladění firmware s tímto protokolem velmi snadné. Použití protokolu zjednodušuje komunikaci a zvyšuje její spolehlivost.

2.3.1. Tvar zprávy

Jedním z nejdůležitějších parametrů protokolu je přesně definovaný rámec pro zprávu. Každá zpráva se skládá z předem definovaných částí, jen tak je možné její bezproblémové zpracování. Tento protokol má konkrétně sedm částí a jejich přehled znázorňuje tab. 1.

Tab. 1 - Přehled tvaru zprávy protokolu

	START	ADDR	TYPE	LEN	DATA	SUM	END
Zápis	#	03	3A	03	011	EF	LF ('\n')
Hodnota	-	3	58	3	-	239	'-

Uvozovací znak (START) – Začátek každé zprávy je označen jednotným uvozovacím znakem. Díky tomu zařízení bezpečně pozná příchod nové zprávy. Jedná se o znak '#' odeslaný jako jeden ASCII bajt.

Adresa (ADDR) – Adresa slouží k jednoznačné identifikaci, komu má být zpráva určena, případně od koho přijatá odpověď. Rozměr adresy je jeden bajt, jedná se tedy o rozsah 0-255. Samotný zápis pak probíhá pomocí dvou znaků jako číslo v hexadecimálním tvaru (00 pro adresu s číslem 0 a FF pro adresu s číslem 255). Adresa 0 je rezervována pro hromadné zprávy, adresa 1 pak pro zařízení typu *master*.

Typ instrukce (TYPE) – Instrukce určuje význam zprávy (čtení vstupů, změna výstupů atd.). Rozsah je opět 0-255, zápis pomocí dvou bajtů v hexadecimálním tvaru.

Délka dat (LEN) – Udává počet znaků části s daty. Slouží ke snadnému oddělení od zprávy, protože délka dat může být proměnná. Rozměr opět jeden bajt, reprezentovaný dvěma znaky.

Obsah zprávy (DATA) – Jedná se o samotný obsah zprávy. Formát není striktně dán a závisí na použité instrukci. Maximální délka je však omezena na 255 znaků.

Kontrolní součet (SUM) – Jednobyťový kontrolní součet pro kontrolu správnosti zprávy. Zápis pomocí dvou bajtů v hexadecimálním tvaru.

Zakončovací znak (END) – Slouží k detekci konce zprávy. Je použit znak LF (line feed '\n').

Rámec zprávy nepoužívá oddělovače. Výsledná zpráva vypadá takto „#033A03011EF“ + zakončovací znak.

2.3.2. Zpracování zprávy

Přijatou zprávu je třeba v cílovém zařízení řádně zpracovat. Tento proces probíhá v několika krocích. Prvním krokem je periodická kontrola sériového bufferu a čekání na startovací znak „#“. Po přijetí startovacího znaku je přečten (a uložen do znakového řetězce) celý obsah bufferu až po zakončovací znak LF (line feed). Následuje rozdělení řetězce na jednotlivé části. Jako první je oddělena a zkontrolována adresa, pokud souhlasí adresa, můžeme pokračovat v dalším rozdělování na typ zprávy, délku dat, samotná data a sumu. Poté porovnáme délku řetězce a vypočítáme a porovnáme sumu. Pokud oba údaje souhlasí, může následovat provedení příkazu ze zprávy. Pokud některý z údajů nesouhlasí, odesílá modul zprávu typu ARQ (Automatic Repeat reQuest).

2.3.3. Typy zpráv (instrukce)

Instrukce jednoznačně určuje význam odeslané zprávy (směr master -> slave). V opačném směru slouží jako jedna z kontrol věcné správnosti odpovědi. Návratová odpověď tedy musí mít stejný typ instrukce jako přijatý dotaz. Rozměr instrukce je jeden bajt (1 B), v protokolu je tedy prostor pro 256 různých instrukcí. Pro větší přehlednost jsou instrukce rozděleny do skupin podle určení. Například rozsah 0-F je rezervován pro servisní a chybové zprávy, rozsah 10-2F pro konfigurační zprávy, 30-3F pro zprávy týkající se digitálních vstupů a zprávy 40-4F pro digitální výstupy.

2.3.4. Kontrolní součet

Součástí zprávy je i 1B kontrolní součet (1 byte checksum). Ten slouží ke kontrole přijatých dat. Jednotlivé bajty zprávy nejsou při výpočtu váhovány. Jedná se pouze o základní kontrolu, protože výsledná suma není pro každou zprávu unikátní. Mohou se vyskytovat zprávy s identickým kontrolním součtem. Výpočet znázorňuje rovnice (1).

$$X = START + ADDR + TYPE + LEN + DATA \quad (1)$$

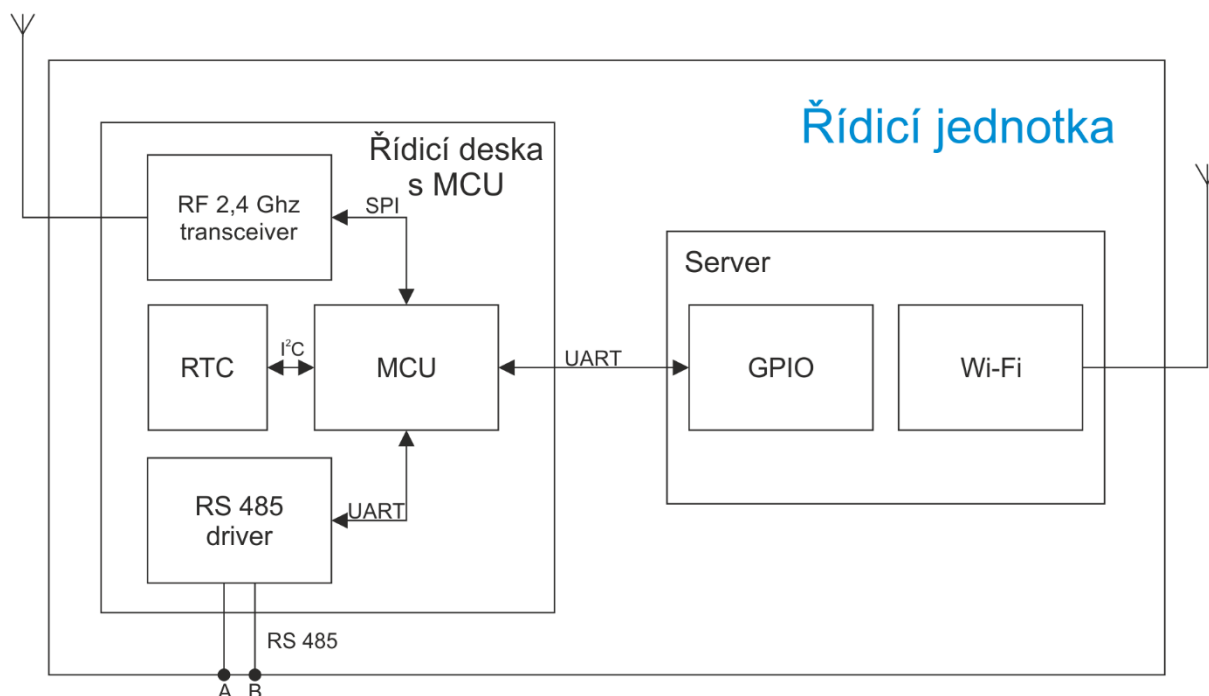
$$SUM = X - \frac{X}{256} * 256$$

3. Hardware

Tato kapitola se zabývá návrhem modulů systému z hlediska hardware. K návrhu schémat i plošných spojů byl použit program Autodesk Eagle. Všechny PCB byly vyrobeny profesionálně ve specializovaných firmách a osazeny ručně.

3.1. Řídicí jednotka

Řídicí jednotka je srdcem centralizovaného systému. Zajišťuje komunikaci mezi uživatelem a jednotlivými prvky chytré elektroinstalace. Samotný modul se skládá ze dvou hlavních částí (blokové schéma je znázorněno na obr. 10). První část je deska řídicí komunikaci s moduly. Slouží k řízení jak drátové, tak bezdrátové komunikace. Má za úkol pokládání dotazů a zpracování všech informací ze sběrnice a bezdrátové komunikace. Je to vlastně master pro obě rozhraní. Měla by rovněž zajišťovat, aby každé akci uživatele byla přiřazena odpovídající reakce v systému (tato propojení jsou součástí firmware). Deska je osazena mikrokontrolérem ATmega1284p, jehož klíčovými vlastnostmi jsou dva kanály rozhraní UART a dostatečně velká (128 kB) flash paměť pro firmware. Druhou část modulu tvoří jednodeskový mikropočítač Raspberry Pi Zero W. V modulu slouží jako server pro běh případného uživatelského rozhraní. Obě části mezi sebou komunikují pomocí sériového rozhraní UART.



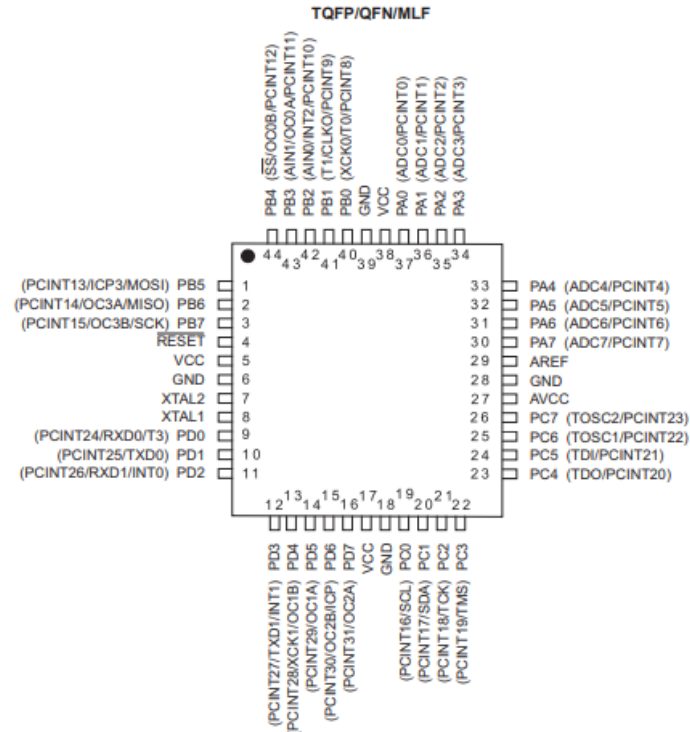
Obr. 10 - Blokové schéma hybridní řídicí jednotky

3.1.1. Napájení

Napájení pro celý modul je řešeno skrze Control board, jehož součástí je spínaný stabilizátor napětí LM2575 od TI. Jedná se o stabilizátor s pevným výstupním napětím 5 V, maximální výstupní proud tohoto obvodu (v použitém zapojení) je 1 A. Tato hodnota je dostačující pro napájení Control boardu i Raspberry Pi. Součástí řídicí desky je dále lineární stabilizátor 3,3 V v SMD provedení (SO8), ten vytváří snížené napětí pro RF transceiver.

3.1.2. ATmega1284p (MCU)

ATmega1284p (rozložení pinů na obr. 11) je 8bitový mikrokontroler z rodiny AVR od firmy Microchip (dříve Atmel). Je vybaven 32 vstupně-výstupními piny (každý pin s podporou externího přerušení), 128 kB programové flash paměti (zvlášť pro řídicí jednotku je dostatek místa pro firmware důležitý), 4 kB EEPROM pro konstanty a proměnné a 16 kB SRAM. V MCU je k dispozici rozhraní SPI, dvakrát rozhraní UART a I²C. Maximální rychlost MCU je 20 MIPS (milionu operací za sekundu) při frekvenci 20 MHz. Arduino IDE tento mikrokontrolér nativně nepodporuje, nicméně je k dispozici volně dostupný balíček MightyCore [], který podporu pro tento a další 40/44pinové AVR přidává. [8]



Obr. 11 - rozložení pinů MCU ATmega1284p [8]

3.1.3. Raspberry Pi Zero W

Raspberry Pi je označení pro jednodeskové mikropočítače s procesorem ARM a GPIO konektorem. Jsou oblíbené zejména mezi kutily a experimentátory z řad laické i odborné veřejnosti. Velkou výhodou těchto počítačů je obrovská komunita uživatelů a dostupných návodů apod. Zde je konkrétně použita nejkompaktnější verze Zero W. Osazena je jednojádrovým procesorem s taktem 1 GHz a 512 MB RAM. Označení W znamená, že tato verze podporuje také bezdrátové technologie Bluetooth a Wi-Fi. [9]

3.1.4. NRF24L01 (RF transceiver)

NRF24L01(+) je 2,4GHz radio transceiver od fy Nordic Semiconductors. Jedná se o systém on chip řešení s rychlostí až 2 Mbps. Vhodný pro ultra low power aplikace (26 μ A standby a 900 nA power-down mode). Napájení v rozsahu 1,9 až 3,6 V, datové piny jsou však tolerantní k 5V logice. [10]

Pro návrh systému byl použit tento obvod v podobě hotového modulu s SMD montáží (Obr. 12). Tyto moduly jsou velmi cenově dostupné (méně než 2 USD za kus). Ty jsou osazeny čínským klonem tohoto obvodu. Klon je však téměř 100% kompatibilní s originálem a lze ho bez problémů použít. Velkou výhodou jsou příznivé rozměry (12x18 mm).



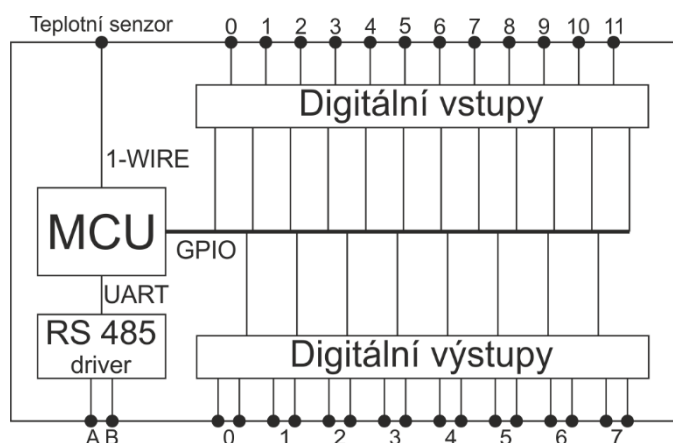
Obr. 12 - SMD modul s klonem NRF24L01

3.1.5. Obvod reálného času DS1307 (RTC)

Obvod hodin reálného času (Real-time clock – RTC) je obvod pro udržení informace o aktuálním čase. Pro řídicí jednotku byl použit obvod DS1307 od firmy Maxim. Tento obvod komunikuje po sběrnici I²C. I²C je sériová dvou vodičová (hodiny a obousměrná data) sběrnice pro pomalé periferie podporující více master zařízení najednou (multi-master). Obvod disponuje základními informacemi o čase (hodiny, minuty, sekundy) a datu (den, měsíc. Rok a den v týdnu)., kromě toho podporuje 12 i 24 hodinový formát času a disponuje obdélníkovým výstupem pro hodinové impulzy. Jako zdroj hodinového signálu je použit křemíkový krystal o frekvenci 32,768 kHz. Obvod také obsahuje přímo vstup pro záložní lithiovou baterii (3V). [11]

3.2. IO modul

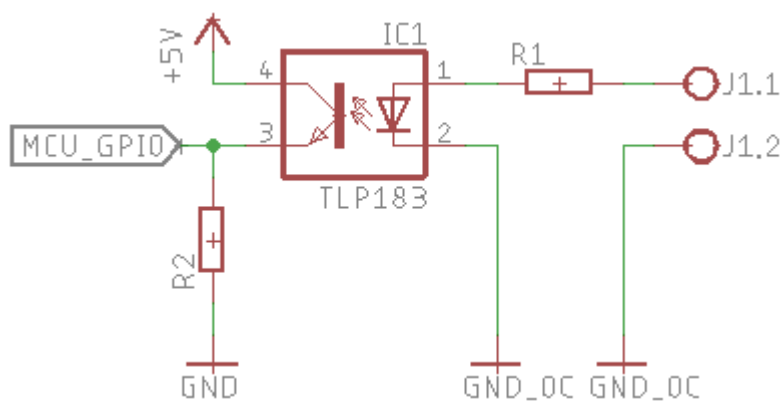
IO modul je modul s digitálními vstupy a výstupy (Inputs-Outputs). Blokové schéma modulu je znázorněno na obr. 13. Jedná se o základní rozšiřující modul (vstupy a výstupy lze považovat za výchozí funkce). Hlavním použitím modulu tedy budou především různá tlačítka, vypínače, koncové spínače. Reléové výstupy pak můžeme použít například ke spínání osvětlení, spínání zásuvek a dalších spotřebičů. Obvodově částečně vychází z řídicí jednotky. Je osazen stejným MCU ATmega1284p, který má dostatečný počet IO pinů. Dále má modul identickou napájecí část se spínaným stabilizátorem LM2575. Tento konkrétní modul je osazen 8 reléovými výstupy a 12 digitálními galvanicky oddělenými vstupy. Pro přímé spínání 230V zátěže jsou použita síťová relé s maximálním proudem kontaktem 2 A. Součástí modulu je navíc svorkovnice se pro sběrnici 1-Wire® pro připojení teplotních čidel Dallas od firmy Maxim. Některé systémy nabízejí tuto funkcionalitu v podobě samostatného modulu (např. Loxone). Zde se jedná o doplňkovou funkci jednoho ze základních modulů a je pouze na uživateli, za tuto možnost programově využije.



Obr. 13 - Blokové schéma IO modulu

3.2.1. Digitální vstup

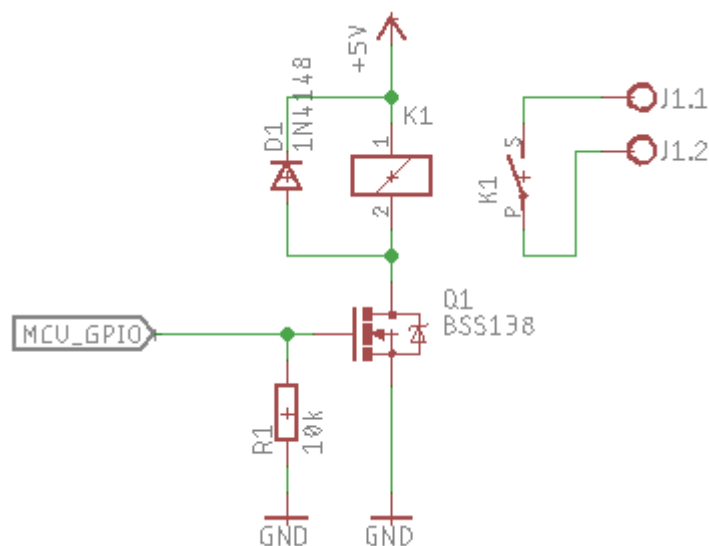
Opticky oddělené digitální vstupy jsou řešeny pomocí prvku optočlenů. Konkrétně je osazen obvod TLP183 od fy Toshiba. Jedná se o digitální optoizolátor (zdroj světla a fotodetektor v jednom pouzdře), kde je jako detektor použit bipolární fototranzistor. Zapojení digitálního vstupu ukazuje obr. 14. Proud do LED diody je nastaven rezistorem R1. Jeho hodnota určuje rozsah vstupního napětí. Pro zkušební účely se počítá s jmenovitým napětím 5V.



Obr. 14 - zapojení digitálního vstupu

3.2.2. Digitální výstup

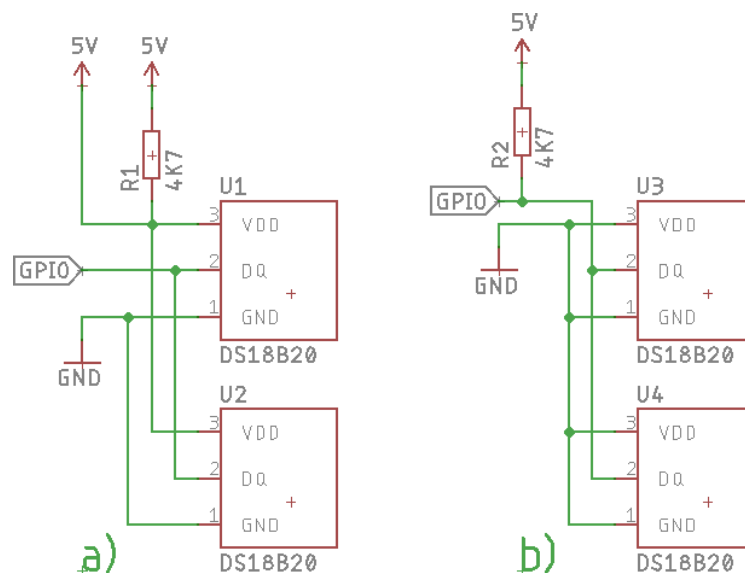
Cívky výstupních relé nelze spínat přímo výstupním pinem MCU, protože je cívka relé indukční zátěž a také protékající proud by byl příliš vysoký. Pro spínání cívky výstupních relé byl zvolen malý spínací N-MOSFET BSS138, který je vhodný pro logické aplikace. Zapojení jednoho výstupu znázorňuje obr. 15. Cívka relé je umístěna v drainu tranzistoru Q1 a napájena je napětím +5 V. Source Q1 je spojen se zemí. Kladná logická úroveň (log. 1) sepne tranzistor Q1, ten se otevře a cívkou relé prochází proud, relé je sepnuto. Změna logické úrovně na log. 0 uzavře tranzistor a kontakt relé se rozezne. Dioda D1 slouží k ochraně tranzistoru Q1 před zápornými napěťovými špičkami, vznikající v důsledku spínání indukční zátěže.



Obr. 15 - zapojení digitálního výstupu

3.2.3. Digitální teplotní senzor Dallas (Maxim)

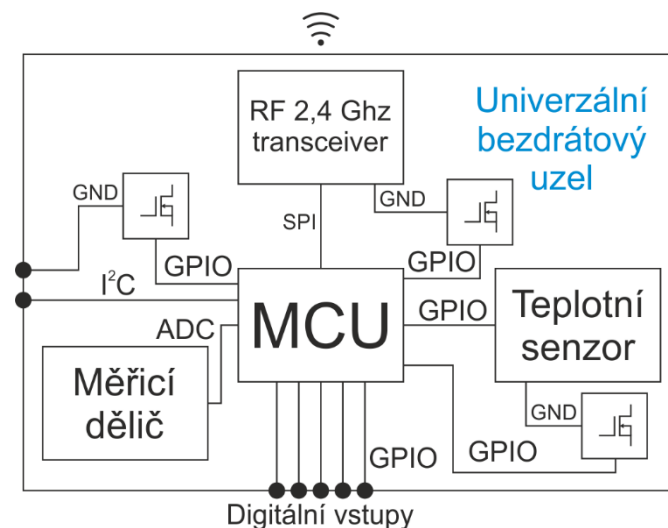
Digitální teplotní senzory Dallas (dnes od fy Maxim) jsou zavedeným pojmem nejen v oblasti automatizace. Jedná se o velmi přesné (z výroby kalibrované) teplotní senzory. Příkladem může být velmi oblíbený senzor DS18B20 s 12bitovým digitálním výstupem a přesností $\pm 0,5 \text{ }^\circ\text{C}$ v rozsahu $-10 \text{ }^\circ\text{C}$ až $+85 \text{ }^\circ\text{C}$ (plný rozsah je $-55 \text{ }^\circ\text{C}$ až $+125 \text{ }^\circ\text{C}$). Senzory komunikují po speciální sběrnici 1-Wire®, která vyžaduje pouze jeden datový vodič a napájení. Každé zařízení má svůj jedinečný 64bitový identifikační kód a na jedné sběrnici může být připojeno více zařízení. Kromě standardního zapojení se třemi vodiči, je možné senzory používat v tzv. Parasitic Power Mode, která umožňuje použití senzoru pouze se dvěma vodiči. Zařízení je pak napájenou pouze díky $4.7\text{k}\Omega$ pull-up rezistoru na datovém vodiči (na desce s MCU). Zapojení je znázorněno na obr. 16. [12]



Obr. 16 - Zapojení teplotního senzoru DS18B20 a) dvouvodičově a b) třívodičově

3.3. Univerzální bezdrátový uzel (node)

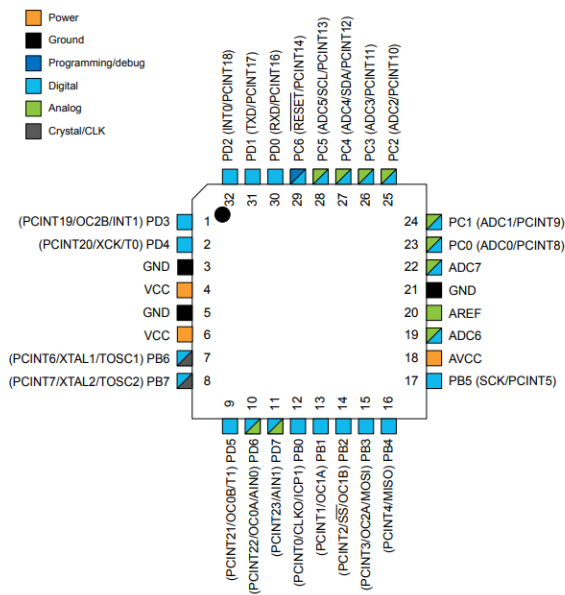
Univerzální bezdrátový modul je základní zařízení pro bezdrátovou část systému. Slouží ke konstrukci především jednoúčelových bezdrátových zařízení, jako jsou například teplotní čidla, bezdrátové vypínače apod. Blokové schéma je znázorněno na obr. 17. Srdcem zařízení je z důvodu kompatibility s Arduino Uno (případně Nano) mikrokontrolér ATmega328p. Celé zařízení je navrženo tak, aby jeho PCB bylo možné použít pro více typů bezdrátových uzlů. V základním provedení je plošný spoj osazen mikrokontrolérem a transceiverem nRF24L01. Volitelně je ho možné osadit senzorem vlhkosti a teploty DHT22 nebo DHT11, pinovou lištou pro 4 digitální vstupy/výstupy a vstup pro externí přerušení, pinovou lištu pro připojení zařízení přes I²C. I²C v tomto případě slouží především pro připojení 0,96“ OLED displeje. Zařízení je konstruováno na dvouvrstvém PCB a svými rozměry je navrženo pro instalování do běžné elektroinstalační krabičky pod omítku.



Obr. 17 - Blokové schéma univerzálního bezdrátového uzlu

3.3.1. ATmega328p (MCU)

Druhým a posledním použitým typem MCU je ATmega328p, který rovněž patří do rodiny AVR. Tento mikrokontrolér je velmi rozšířený zejména díky oblíbenému vývojovému kitu Arduino Uno (a podobným). Oproti 1284p se jedná o MCU v menším pouzdře, má tedy pouze 23 vstupně-výstupních pinů. K dispozici je 32 kB programové flash paměti, 1 kB EEPROM pro konstanty a proměnné a 2 kB SRAM. Najdeme zde rozhraní SPI, dvakrát rozhraní UART a I²C. Maximální rychlost MCU je 20 MIPS (milionu operací za sekundu) při frekvenci 20 MHz. Rozložení pinů pouzdra TQFP32 ukazuje obr. 18. [13]



Obr. 18 - Rozložení pinů MCU ATmega328p [13]

3.3.2. Senzor teploty a vlhkosti DHT22

DHT22 je kalibrovaný digitální senzor teploty a vlhkosti. Jedná se o čínský výrobek původem z Ebay. Jeho cena je přibližně 2 USD. Rozsah teplotního senzoru je -40 až 80 °C s rozlišením 0,1 °C. Vlhkost vzduchu měří na celá procenta. Komunikace s MCU probíhá po jednom digitálním pinu, jedná se o specifickou softwarovou sběrnici. [17]

3.3.3. Řízení spotřeby

Vzhledem k tomu, že jsou tyto moduly navrženy primárně pro použití při napájení z baterií, bylo nutné se u nich zaměřit také na spotřebu elektrické energie. Omezení spotřeby elektrické energie bylo provedeno jak z hlediska vhodného návrhu z hardwarové stránky, tak z hlediska firmware. Prvním krokem bylo snížení pracovní frekvence MCU ATmega328p z 16 MHz (běžných pro Arduino) na 8 MHz. Tento krok rovněž umožní provozovat MCU při nižším provozním napětí, než je 5 V. Dalším krokem bylo přidání spínacích N-MOSFET tranzistorů pro možnost ovládní napájení všech použitých periférií (transceiver, teplotní čidlo atd.). Toto umožňuje vypínat periférie v době, kdy jsou neaktivní a tím omezit jejich spotřebu.

Z hlediska firmware nabízí ATmega328p možnost přepnutí do několika režimů spánku podle hloubky usnutí (odráží se na perifériích, které budou v daném režimu k dispozici). Pro významné omezení spotřeby modulu v klidovém režimu je nutno využít nejhlubšího režimu spánku, který je označen jako Power-down mode. Z tohoto režimu je možné probudit MCU pouze za požití watchdog timeru nebo externího přerušení. V závislosti na typu modulu se hodí oba tyto způsoby probuzení.

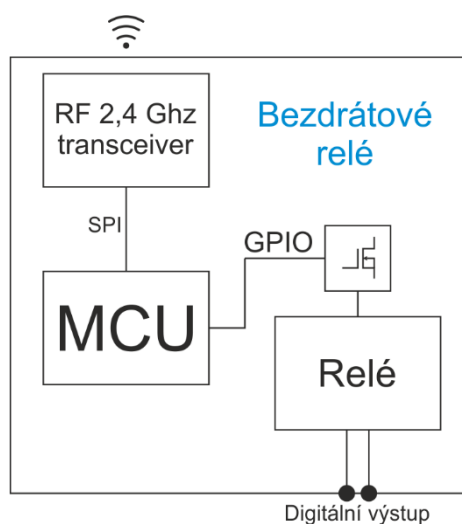
Watchdog timer se hodí například pro teplotní senzory, externí přerušení například pro vypínače nebo tlačítka. Možnou nevýhodou může být fakt, že maximální doba pro přetečení WDT je přibližně 8 s. Pokud potřebujeme další čas, můžeme použít např. jednoduchý counter probuzení. Ještě před uspaním samotného MCU je možné přepnout do režimu spánku i modul bezdrátového transceiveru.

3.3.4. Vlastní spotřeba

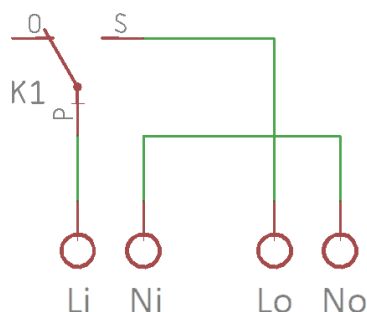
Pro představu o výdrži takových modulů bylo provedeno orientační měření spotřeby modulu v konfiguraci teplotního senzoru. Spotřeba modulu v uspaném stavu byla měřena přesným ručním multimetrem Agilent U1241B. Hodnota odebíraného proudu činí stabilních 13,5 μ A. Proud v zátěži byl měřen osciloskopem na snímacím odporu 1 Ω . Průměrná hodnota odebíraného proudu v zátěži za 10 měření (režim měření a odesílání teploty) činí 11,6 mA za průměrnou dobu 0,428 s. V mezidobí, kdy se pouze inkrementuje counter činí průměrná spotřeba 4,2 mA po dobu přibližně 100 ms. Pokud požadujeme aktualizaci teploty například každých 10 minut, bude v jednom cyklu (po zaokrouhlení na sekundy) 592 s odbíráno 13,5 μ A, 7,5 s odbíráno 4,2 mA a 0,5 s odbíráno 11,6 mA. Průměrná spotřeba tedy činí 75,5 μ A. Uvažujeme-li kapacitu alkalického AAA článku 1200 mAh, dostáváme se k teoretické výdrži 661 a půl dne (1 rok a 296,5 dne). Toto je velmi příznivý výsledek, který lze požadovat za úspěch.

3.4. Bezdrátové relé

Toto zařízení je vlastně jeden samostatný digitální výstup, realizovaný v podobě mechanického relé. Blokové schéma je znázorněno na obr. 19. Modul konstrukčně i obvodově vychází z předchozího modulu (univerzální bezdrátový modul). Plošný spoj má opět tvar vyhovující rozměrům elektroinstalační krabičky pod omítku. Obvodová část obsahuje téměř totožné zapojení MCU a transceiveru (chybí enable MOSFET). Jedinou periferií je 3V relé spínané přes MOSFET tranzistor BSS138. Relé je vhodné pro síťové aplikace a maximálně proud kontaktem je 10 A. Kontakt relé spíná fázi L, připojenou na vstupní svorky, na svorky výstupní. Společnou svorkou na vstupních i výstupních svorkách je střední vodič N. Spínané zařízení tedy napájíme přímo z výstupních svorek bezdrátového relé (viz obr. 20).



Obr. 19 - Blokové schéma bezdrátového relé



Obr. 20 - Zapojení vstupních a výstupních svorek modulu

3.4.1. Napájení

Napájení probíhá přímo ze sítě 230 V. Zařízení je tedy přímo napájeno z napětí, které je připojeno na vstupních svorkách. Napájecí napětí 3,3 V pro řídicí elektroniku je obstaráno osazeným modulem spínaného zdroje HLK-PM03 od čínské firmy Hi-Link. Pro použití tohoto hotového modulu mluví zejména jeho příznivé rozměry a nízká cena (v době psaní této práce jsou to přibližně 2 USD), která je nižší než cena samotného integrovaného AC/DC converteru bez dalších součástek. Výstupní výkon tohoto modulu by měl být 3 W. Vzhledem k maximálnímu odběru zařízení kolem 150 mA, se jedná o dostatečně dimenzovaný zdroj.

4. Programová část (firmware)

4.1. Implementace protokolu (RS485)

Navržený komunikační protokol byl implementován v jazyce Arduino v podobě jednoduché knihovny. Knihovna obsahuje základní sadu funkcí a metod pro inicializaci sběrnice, příjem, kontrolu a odeslání zprávy.

4.1.1. Inicializace

Inicializace probíhá voláním jediné metody, která se nazývá *initBus*. Úkolem této metody je nastartovat komunikaci na rozhraní UART a nastavit enable pin pro RS485 transceiver. Vstupní proměnné jsou tedy rychlost komunikace a číslo příslušného pinu.

- `initBus(long setBusSpeed, int setSwitchPin)`

V případě slave zařízení je ještě nutné nastavit při inicializaci adresu zařízení. Zde je tedy o jeden parametr navíc.

- `initBus(long setBusSpeed, int setSwitchPin, int setAddr)`

4.1.2. Příjem zprávy

Příjem zprávy probíhá pomocí periodického volání metody *incommingMsg*. Tato metoda má za úkol kontrolu bufferu sériového rozhraní UART. Pokud v bufferu něco je, proběhne kontrola, zda se jedná o START znak. Poté je zprava znak po znaku zapisovány do textového řetězce typu *String* až do přečtení zakončovacího znaku END.

4.1.3. Kontrola zprávy

Po přijetí zakončovacího znaku je automaticky volána metoda *checkMsg*. Vstupním parametrem této metody je řetězec přijaté zprávy bez zakončovacího znaku.

- `checkMsg(String packet)`

Úkolem metody je kompletní zpracování řetězce a kontrola jeho správnosti. Ke zpracování řetězce je použita funkce *substring*, která je součástí třídy *String*. Tato funkce slouží k vybrání konkrétní části řetězce. Jako první je ze zprávy oddělena adresa. Pokud se jedná o zařízení typu *slave*, proběhne porovnání s adresou zařízení. Pokud je adresa rozdílná, ve zpracování se dále nepokračuje. V případě

modulu typu *master*, je adresa uložena jako výstupní veličina, pro další zpracování zprávy. Následně je stejným způsobem zpracován zbytek řetězce přesně podle jednotlivých částí zprávy. Typ zprávy (TYPE) a délka přijatých dat (LEN) jsou převedeny pomocí funkce *strtol* na čísla typu *int*, aby bylo možné s nimi dále pracovat. Funkce *strtol* je běžná funkce jazyka C pro převod textového řetězce na číslo (je nutný mezi převod objektu *String* na pole *char*). Tento krok je nutný, protože třída *String* nepodporuje převod na číslo z jiného základu než 10. Dále je pomocí funkce *sumHex* (součást knihovny protokolu) vypočítána suma přijaté zprávy. Výstupem je dvouznakový řetězec typu *String*, přijatou sumu tedy není nutné převádět na číslo a porovnávají se dva řetězce typu *String*. Posledním krokem podle specifikace protokolu je kontrola délky přijaté zprávy. To se provede tak, že k základní délce řetězce (9 znaků bez znaku END) přičte délka dat LEN a tento výsledek porovnáme se skutečnou délkou přijaté zprávy. Pokud je všechno v pořádku nastavíme příznakový bit *semMsg* na hodnotu jedna a odesíláme zprávu typu ACK a jako data „1“. V případě, že některý z kontrolních prvků nesouhlasí odesíláme zprávu ARQ (typ ACK a data „0“). V poslední fázi dochází k předání výstupních dat hlavnímu programu. Jedná se typ zprávy (TYPE), vstupní data (DATA) a v případě modulu typu *master* i vstupní adresa (ADDR). Hlavní program se o přítomnosti nové zprávy dozví periodickou kontrolou příznakového bitu *semMsg*. Další zpracování zprávy je na hlavním programu. Po zpracování zprávy je nutné nastavit příznakový bit opět na hodnotu 0. Příklad konstrukce pro zpracování zprávy:

```
if(bus.semMsg) { //kontrola příznakového bitu
    doMsg(bus.addr, bus.inst, bus.data); //volání funkce pro zpracování zprávy
    bus.semMsg = 0; //nastavení příznakového bitu zpět do 0
}
```

Výjimku v popsaném schématu tvoří příjem zprávy typu ACK/ARQ. Pokud metoda *checkMsg* vyhodnotí příjem zprávy typu ARQ je zavolána metoda *reSend*. Ta se stará o znovu-odeslání poslední odeslané zprávy. V případě, že se jedná o zprávu typu ACK je nutné nastavit příznakový bit čekání na ACK *semACK* do hodnoty 0.

4.1.4. Odeslání zprávy

K odeslání zprávy slouží metoda *sendMsg*. Hlavním úkolem této metody je ze vstupních parametrů vytvořit kompletní zprávu a odeslat jí přes rozhraní UART. Vstupními parametry jsou typ zprávy (TYPE) a samotný řetězec dat (DATA). Pokud se jedná o modul typu *master*, je nutné doplnit ještě adresu cílového modulu (ADDR). V případě modulu *slave* je automaticky doplněna vlastní adresa zařízení.

- `sendMsg(String addr, String type, String msg)`

4.2. Bezdrátová komunikace (RF 2.4 GHz)

Bezdrátová komunikace, stejně jako komunikace po sběrnici, vyžaduje soubor určitých předpisů a pravidel. Vzhledem k tomu, že je modul s obvodem NRF24L01 (a jeho klony) mezi Arduino vývojáři velmi oblíbený, je k dispozici celá řada knihoven s implementovanými protokoly. Pro tento projekt byla vybrána univerzální a velmi komplexní knihovna RadioHead

4.2.1. Základní popis knihovny RadioHead

Knihovna RadioHead je asi nejkompaktnější knihovna podporující transceiver NRF24L01. Celá knihovna se skládá ze dvou hlavních částí. První částí je tzv. *Driver*. Jedná se o základní balík funkcí pro jednoduchou komunikaci mezi těmito čipy. Druhou částí je tzv. *Manager*, ten přidává pokročilou úroveň komunikace, jakou jsou například adresované zprávy apod. [14]

4.2.2. RadioHead Driver

Jak bylo naznačeno, driver je jakýsi základní ovladač. Má za úkol řešit komunikaci přes sériové rozhraní SPI. Pomocí driveru lze odesílat a přijímat jednoduché neadresované zprávy. Tato část není sama o sobě vhodná pro spolehlivou, a hlavně jednoznačnou komunikaci. Proto je třeba použít vhodnou nadstavbu v podobě dostupných managerů. Pro správné fungování je třeba vytvořit instanci třídy `RH_NRF24` s doplňujícími parametry pro rozhraní SPI

- `RH_NRF24 driver(CE, CS)`

Některé funkce třídy driveru jsou používány zvlášť i při použití manageru. Zvlášť výhodnou funkcí je funkce *sleep*, která slouží k usnutí transceiveru. Dále pak funkce *isChannelActive*, která slouží k detekci vysílání ostatních modulů. Ta je zvlášť důležitá pro moduly se samovolným vysíláním, aby nedocházelo ke kolizím zpráv. [14]

4.2.3. RadioHead Reliable Datagram

Reliable Datagram je jeden ze zahrnutých managerů. Slouží k vytvoření jednoduché hvězdicové topologie, tedy jeden nadřazený modul a libovolný počet nezávislých uzlů. Podporuje odesílání adresovaných paketů, potvrzení doručení a znovu-odeslání zprávy. Před začátkem komunikace je třeba rovněž vytvořit instanci třídy zvoleného manageru, v tomto případě je to *RHReliableDatagram*. Vstupními parametry jsou adresa inicializovaného driveru a požadovaná adresa zařízení.

- `RHReliableDatagram (driver, SERVER_ADDRESS)`

Inicializace probíhá pomocí funkce *init*. Pak už je možné posílat adresované zprávy pomocí funkce *sendtoWait*. Vstupními parametry jsou pointer na odesílaný řetězec, jeho délka a adresa cílového zařízení.

- `sendtoWait (uint8_t *buf, uint8_t len, uint8_t address)`

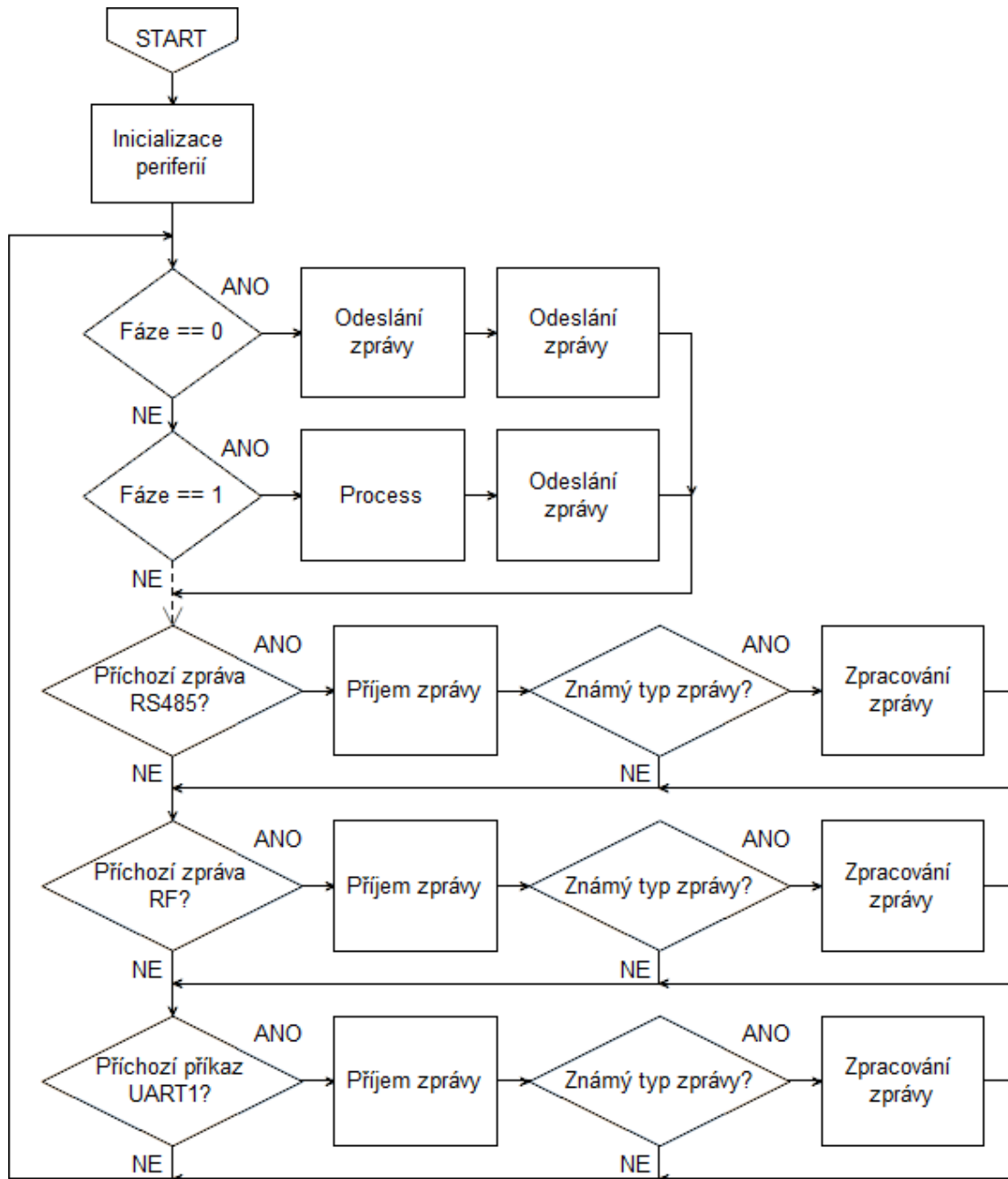
Pro příjem zpráv slouží hned dvě funkce. Pro běžný příjem zprávy slouží funkce *recvfromAck*. Vstupními parametry jsou pointery na proměnné, do kterých se uloží jednotlivé části příchozí zprávy, avšak pouze samotný text zprávy a její délka jsou povinnými parametry.

- `recvfromAckTimeout (uint8_t *buf, uint8_t *len, uint8_t *from=NULL, uint8_t *to=NULL, uint8_t *id=NULL, uint8_t *flags=NULL)`

Speciálním případem je pak funkce *recvfromAckTimeout*. Tato funkce má stejnou funkci jako funkce předchozí, navíc však umožňuje zastavení běhu programu na předem definovanou dobu, po kterou očekáváme příchod zprávy. [14]

4.3. Řídicí jednotka (deska s MCU)

Firmware pro řídicí jednotku je specifickou záležitostí pro každý individuální projekt. Vychází z použitých modulů a jejich rozložení. V této kapitole bude znázorněno přibližné schéma, které může sloužit jako stavební kámen pro výsledný firmware. Vývojový diagram je znázorněn na obr. 21.



Obr. 21 - Principiální vývojový diagram firmware řídicí jednotky

4.3.1. Popis hlavní smyčky

Důležitým principem běhu hlavní smyčky je postupné rozfázování jednotlivých operací. Rozfázování probíhá pomocí rozhodovacího příkazu *switch* a jednoduchého čítače (proměnná typu *int*). Každý případ (*case*) rozhodovacího příkazu znamená jednu fázi řídicí jednotky. Po vykonání každé fáze je inkrementován čítač, to posune rozhodovací příkaz do další fáze. Po vykonání poslední fáze je čítač opět vynulován. Každá fáze představuje jeden dotaz nebo požadavek (především pro moduly na sběrnici RS485) a následné zpracování odpovědi. Jedná se například o kontrolu digitálních vstupů nebo zápis digitálních výstupů.

4.3.2. Inicializace periférií a alarmů

K inicializaci periférií jsou použity datové struktury (struct). Slouží především pro inicializaci jednotlivých digitálních i analogových vstupů, výstupů, teplotních čidel a podobně. Každá struktura obsahuje všechny parametry, které jsou pro obsluhu dané periferie potřeba. Příklad struktury pro běžný digitální vstup je vidět v následujícím bloku kódu:

```
struct digitalInput //název struktury
{
    byte addr; //adresa zařízení, kde se periferie nachází
    byte inputNumber; //číslo digitálního vstupu v cílovém zařízení
    boolean inputState; //aktuální načtený stav digitálního vstupu
    boolean wireless; //identifikátor drátového nebo bezdrátového zařízení
};
```

Každá periferie se inicializuje zvlášť jako samostatné zařízení nebo lze více periférií sjednotit do pole. Můžeme tak inicializovat pouze ty periferie, které v projektu skutečně používáme.

```
- digitalOutput dOut[3];
```

Tento způsob inicializace lze využít i pro nefyzické prvky, jako jsou například časové alarmy. Ve vzorovém firmwaru se nachází dva nezávislé alarmy pro ovládání digitálních výstupů v konkrétní čas. Struktura alarmů pro digitální výstupy vypadá takto:

```
struct outputAlarm
{
    byte setHour; //nastavená hodina
    byte setMinute; //nastavená minuta
    byte setOutput; //výstup který má být ovládán
    byte setState; //stav do kterého má být nastaven
    boolean enable; //aktivace alarmu
};
```

Alarm může být nastaven pomocí externích příkazů po rozhraní UART1 (UART0 je použit pro sběrnici RS485), který je spojen s Raspberry Pi. Kontrola alarmu probíhá v jedné z fází v hlavní smyčce.

4.3.3. Zpracování zpráv protokolu RS485

Po zpracování příchozí zprávy (odpovědi) je v hlavní smyčce volána metoda *doMsg*. Tato metoda má za úkol zpracování informací, které dorazily v příchozí odpovědi.

- `doMsg(bus.addr, bus.inst, bus.data);`

V metodě `doMsg` se nachází rozhodovací příkaz *switch*. Každý *case* (případ) odpovídá jedné instrukci. Typickým příkladem je zpracování odpovědi na dotaz o stavu vstupů.

4.3.4. Zpracování zpráv knihovny RadioHead

Pro účely příjmu zpráv přes bezdrátové rozhraní je vytvořena metoda *rcvRF*. V metodě se ověřuje, zda není v bufferu zpráva (funkce `manager.available()` vrací 1). Pokud ano je funkcí *manager.recvfromAck* zpráva přijata. Následně je volána metoda *resolveRF*. Parametry této metody jsou adresa příchozí zprávy a samotná zpráva. V metodě se opět nachází rozhodovací příkaz *Switch*, ale rozhodovací proměnná je tentokrát adresa příchozí zprávy. V těle každého *case* se už pak nachází konkrétní akce. Například pro zmáčkuté tlačítko může být odpovídající akce odeslání zprávy bezdrátovému relé nebo RS485 IO modulu. Pro příchozí teplotu pak její uložení do odpovídající proměnné.

4.3.5. Externí příkazy a Raspberry Pi

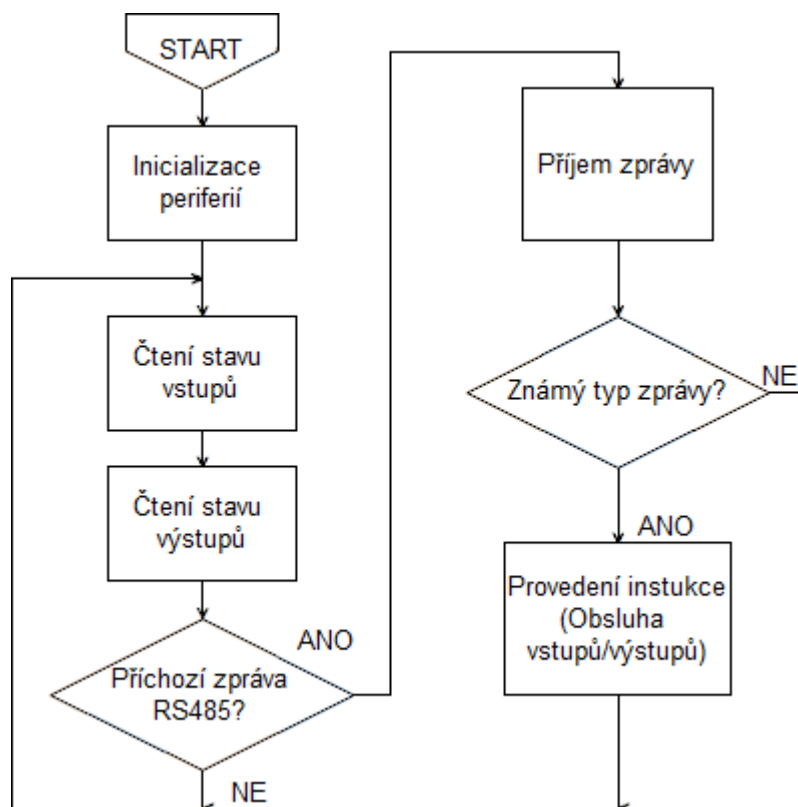
Pro komunikaci mezi komunikační deskou s MCU ATmega1284p a Raspberry Pi slouží druhé rozhraní UART. Pro nastavení a ovládání systému a komunikační desky, může vývojář směrem z Raspberry Pi využít systém jednoduchých příkazů. Příjem příkazů probíhá v metodě *rcvCmd*. Tato metoda vyčítá buffer až do přijetí zakončovacího znaku LF ('\n'). Celá zpráva je uložena do textového řetězce typu *String*. Poté je volána metoda *resolveCmd*, ta má na starosti vykonání příchozího příkazu podle zadaných parametrů. Zápis příkazu je složen z názvu samotného příkazu a parametrů oddělených mezerou. Implementovány jsou příkazy pro nastavení času („setTime“), přečtení aktuálního času („getTime“), nastavení digitálních výstupů („setDO“) nebo nastavení alarmů pro výstupy („setAlarm“). Například příkaz `setDO` se skládá ze dvou parametrů. Prvním parametrem je číslo výstupu a druhým stav, do kterého má být výstup nastaven. Celý příkaz pak vypadá takto „setDO 01 1“ a znamená nastavení výstupu číslo 1 do stavu log. 1.

V opačném směru je pak možné využít periodický log důležitých informací jako jsou aktuálně nastavené hodnoty na výstupech, hodnoty vstupů nebo hodnoty senzorů. Log obsahuje časovou značku. Odesílání probíhá periodicky jako jedna z fází v hlavní smyčce. Hodnoty jsou odděleny středníkem. Tento log může být použit při vývoji uživatelského časového grafu nebo rozhraní (webové aplikace), které na Raspberry Pi pobeží.

4.4. IO modul

Na rozdíl od řídicí jednotky může být firmware rozšiřujících modulů podstatně méně komplikovaný. Toto je možné zejména díky tomu, že se jedná o centralizovaný systém, většina procesů se tedy odehrává přímo v řídicí jednotce.

Funkčně lze firmware rozdělit do dvou rovin. V základní rovině se firmware stará o obsluhu vestavěných periférií (vstupy a výstupy). Aktuální stavy vstupů a výstupů jsou čteny periodicky v hlavní smyčce programu. K tomuto účelu slouží metody *readInputs* a *readOutputs*. Načtené hodnoty jsou pak ukládány do pole (pro vstupy a výstupy zvlášť). Druhá rovina se týká obsluhy sběrnice RS485 a zpracování zpráv. Princip je úplně stejný jako u řídicí jednotky, samozřejmě s drobnými rozdíly pro zařízení typu slave, jak byly popsány v kapitole o implementaci protokolu. Pro odbavení (provedení) instrukce z příchozí zprávy slouží opět metoda *doMsg*. Každý *case* příkazu *switch* znamená jednu implementovanou instrukci, a tedy přesný postup jejího provedení. Vývojový diagram je znázorněn na obr. 22.



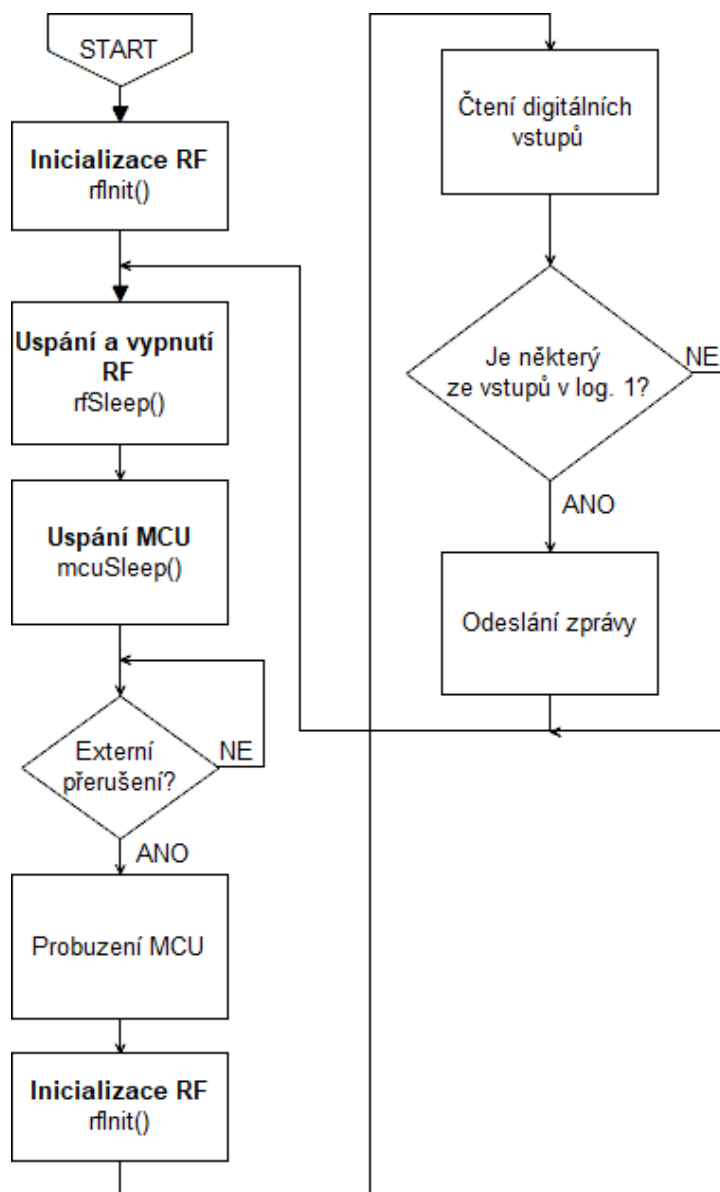
Obr. 22 - Vývojový diagram firmware IO modulu

4.5. Univerzální bezdrátový uzel

Ještě méně komplikovaný je firmware bezdrátových modulů. Bezdrátové moduly jsou konstruovány jako jednoduchá jednoúčelová zařízení, firmware tedy většinou obsahuje pouze nutné minimum funkcí.

4.5.1. Bezdrátový vypínač

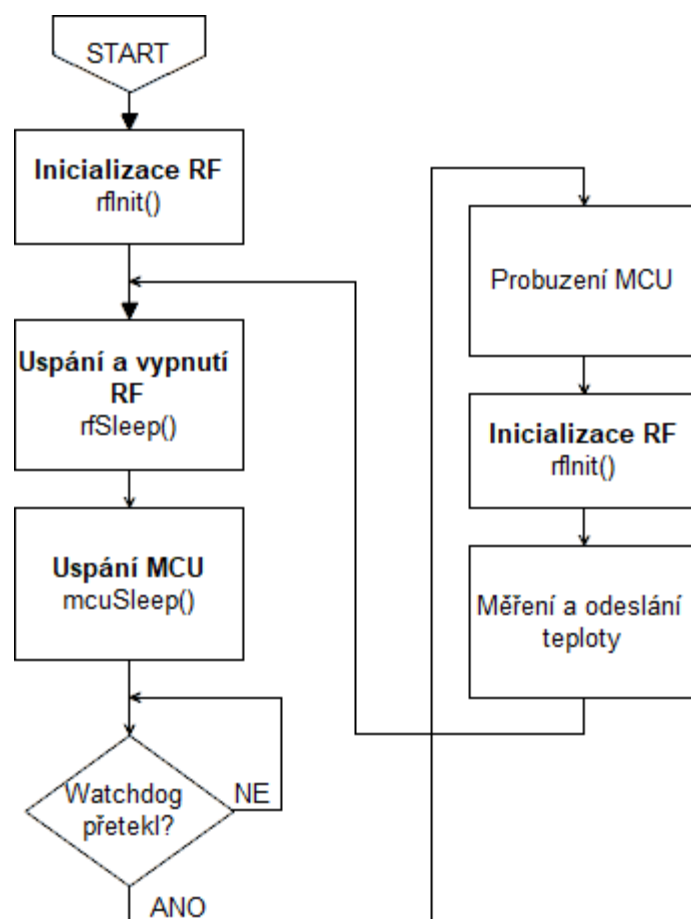
Bezdrátový vypínač je zařízení, které je v výchozím stavu kompletně uspané. K uspaní se používá originální AVR knihovna „sleep.h“. Vybrán je nejhlubší stupeň spánku, a to konkrétně Power-down mode. Samotné uspaní se pak provede sekvencí příkazů „sleep_enable()“ a „sleep_cpu()“. K probuzení se využívá externí přerušení, jehož aktivace se provede pomocí Arduino funkce *attachInterrupt*. Po probuzení deaktivujeme režim spánku příkazem „sleep_disable()“ a z důvodu možných zámků na pinu externího přerušení, toto přerušení rovněž dočasně deaktivujeme voláním funkce *detachInterrupt*. Poté program přeskočí do hlavní smyčky, kde je nejprve voláním metody *rfInit* spuštěn (nastavení log. 1 na pin se spínacím tranzistorem) a inicializován RF transceiver (knihovna RadioHead). Poté probíhá načtení hodnot vstupů a následné samovolné odeslání kódu tlačítka řídicí jednotce. Modul podporuje až 4 tlačítka, jako zpráva se odesílá číselná řetězec „01“ až „04“. Následuje uspaní a vypnutí RF transceiveru pomocí metody *rfSleep* a následné uspaní MCU. Vývojový diagram je znázorněn na obr. 23.



Obr. 23 - Vývojový diagram bezdrátového vypínače

4.5.2. Bezdrátový senzor teploty

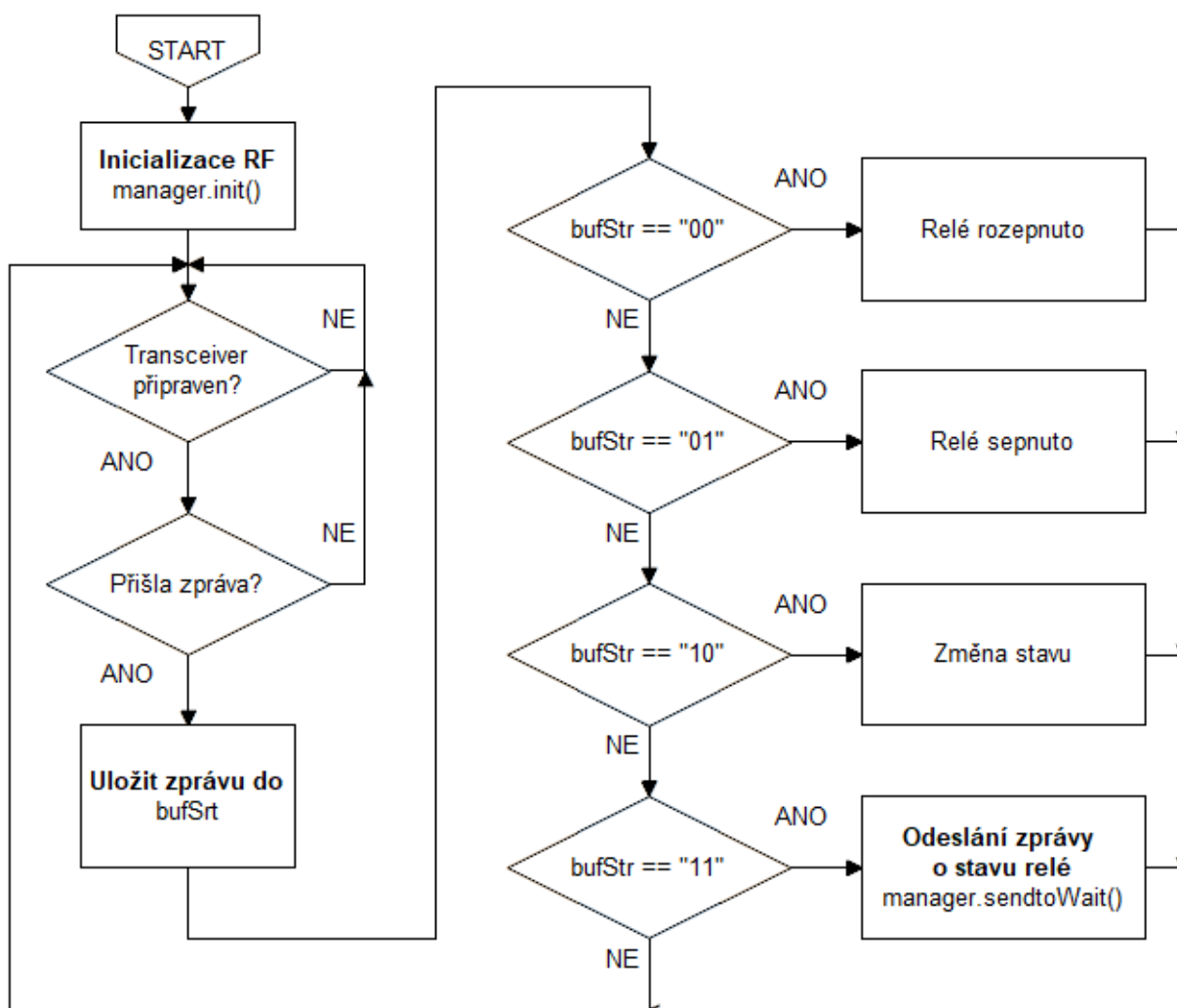
Probouzení a uspávání tohoto modulu probíhá periodicky. Procedura uspání je totožná jako u bezdrátového vypínače. Probouzení probíhá pomocí watchdog časovače, nastaveného na maximální periodu 8s (rozšíření této periody popsáno v kapitole 3.3.3). Po probuzení je stejně jako v předchozím případě volána metoda *rfInit*. Následně proběhne změření teploty čidlem DHT11 (pomocí knihovny volně dostupné knihovny DHT-sensor-library ze stránek GitHub [15]). Výsledek je poté odeslán jako celé číslo (vynásobený deseti) řídicí jednotce. Následuje uspání RF modulu a MCU. Vývojový diagram je zobrazen na obr. 24.



Obr. 24 - Vývojový diagram bezdrátového teplotního senzoru

4.6. Bezdrátové relé

Firmware pro bezdrátové relé je velmi jednoduchý, na rozdíl od univerzálního bezdrátového uzlu nevysílá automatické zprávy, nýbrž čeká na příkazy od řídicí jednotky. V hlavní smyčce programu se tedy permanentně čeká na příchozí zprávu (metoda *recvfromAck* z knihovny RadioHead). Modul očekává tři typy zpráv, „01“ pro stav zapnuto, „00“ pro stav vypnuto a „10“ pro přepnutí stavu. Jedná se o tři podmínky *if* umístěné v hlavní smyčce. Vývojový diagram je zobrazen na obr. 25.



Obr. 25 - Vývojový diagram bezdrátového relé

4.7. Webové rozhraní

Pro vývoj webové aplikace jsou připraveny prostředky popsané v kapitole 4.3.5. Proběh tohoto webového rozhraní slouží počítač (server) Raspberry Pi umístěný v krabičce řídicí jednotky. Samotný vývoj uživatelského rozhraní (webové aplikace) pro externí ovládání systému není předmětem této práce (není součástí zadání).

5. Demonstrační model

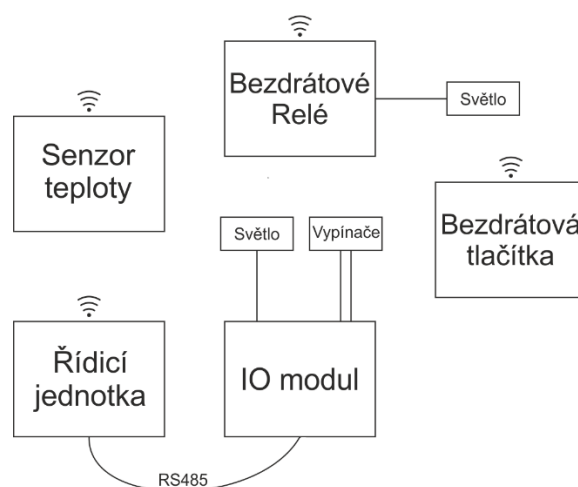
Pro účely ověření funkčnosti a demonstraci základní topologie systému byl vytvořen demonstrační model systému.

5.1. Základní konstrukce

Model byl postaven na základě dýhované dřevotřískové desky o rozměrech 60 x 60 cm a tloušťce 12 mm. Po obvodu je umístěna elektroinstalační lišta, která zároveň rozděluje model do tří částí. V první části je umístěna DIN lišta pro připevnění řídicí jednotka a dalších modulů, které jsou určené pro připevnění na DIN lištu. Do této části patří zařízení, které by za normálních okolností měli svoje místo v domovním rozvaděči. Další dvě části odpovídají obytnému prostoru domu. Je zde místo pro bezdrátové moduly, vypínače, tlačítka nebo senzory.

5.1.1. Schéma a volba komponent

Všechny komponenty pro demonstrační model byly voleny tak, aby bylo možné prezentovat základní ideologii systému a bylo možné na něm pozorovat navrženou topologii. Jako ukázková situace bylo vybráno ovládání osvětlení. Osazené je jak světlo spínané bezdrátovým modulem, tak ekvivalent v podobě světla spínaného RS485 IO modulem. Stejná situace je v podobě vypínačů, model obsahuje dvojité bezdrátové tlačítko a dvojitý vypínač připojený k sběrníkovému IO modulu. Aby bylo možné testovat kooperaci drátové a bezdrátové části, je oběma ty vypínačů (tlačítek) možné ovládat oba typy zapojení světel. Jako modely světel byly zvoleny vysoce svítivé diody LED. Poslední zvolenou komponentou je bezdrátový teplotní senzor. Celkové principiální schéma modelu je na **obr. 26**.



Obr. 26 - Schéma zapojení demonstračního modelu

5.2. Fyzické zapojení

Při realizaci demonstračního modelu bylo nutné upravit zapojení některých komponent. Tyto úpravy byly voleny tak, aby měly minimální vliv na demonstraci vlastností systému a jeho testování.

5.2.1. Napájení modelu

Z důvodu bezpečnosti realizovaného modelu, není pro demonstrační účely použité reálné síťové napájecí napětí 230 V. Místo toho bylo zvoleno bezpečné stejnosměrné napětí 12 V. To je realizováno pomocí spínaného zdroje s výstupním proudem 2.1 A. Napětí je tedy použito nejen pro napájení některých modulů, ale také slouží například jako napájecí napětí pro modely světel.

5.2.2. Univerzální bezdrátové uzly

Všechny bezdrátové moduly určené pro napájení z baterií mají na modelu místo alkalických článků pevný zdroj napájecího napětí 3,3 V (pomocí step-down měniče 12 V \rightarrow 3,3 V). Toto opatření bylo zvoleno hlavně z důvodu, že se jedná pouze o prototypy v raném stádiu vývoje. Nelze tedy například vyloučit chybu, která by způsobila trvalý odběr, mající za následek rychlé vybití baterií. Vzhledem k významu tohoto demonstračního modelu, nehraje faktické napájení těchto modulů žádnou roli.

5.2.3. Bezdrátové relé

Z důvodu absence síťového napájecího napětí, nemůže být tento modul použitý tak, jak byl navržen. Pro použití v demonstračním zapojení bylo nutné odstranit použitý modul síťového spínaného zdroje. Místo něj byl na stávající pájecí body laboratorně namontován modul spínaného step-down měniče z 12 V na 3.3V. Napájecí napětí 12 V je rovněž přiváděno místo síťového napětí na výstupní svorky modulu. Toto napětí slouží tedy i k napájení spínaného modelu světla.

Závěr

V rámci této bakalářské práce byl navržen a zkonstruován systém pro inteligentní elektroinstalace (domácí automatizace). Povedlo se zkonstruovat a oživit všechny navržené moduly a řídicí jednotku. V rámci testování vyrobeného hardware byly otestovány některé parametry vyrobených modulů. Velkým překvapením je potenciální výdrž bezdrátových modulů, která u teplotního senzoru přesahuje hrubě jeden rok provozu. Naopak velkým zklamáním jsou parametry použitého bezdrátového RF modulu. Například dosah u těchto modulů je velmi slabý a v zástavbě se pohybuje kolem deseti metrů a v některých případech dokonce méně (při vyšší komunikační rychlosti). Řešením by bylo použití modulu s integrovaným zesilovačem, případně by bylo možné využít moduly s trvalým napájením (např. bezdrátové relé) jako replikátory zpráv. Oproti tomu komunikace po sběrnici RS485 je téměř bezchybná a nedochází k výpadkům zpráv.

Práci byla ověřena realizovatelnost vyslovené myšlenky. Práce je tedy připravena pro další vývoj a zdokonalování existujících modulů, zaměřením se na zjištěné nedostatky. Lze tedy v práci pokračovat např. formou diplomové práce.

Práce mi pomohla rozšířit si obzory v oblasti inteligentních elektroinstalací, získat nové zkušenosti při návrhu hardware a tvorbě plošných spojů. Protože se orientuji spíše na hardware, byla to také dobrá příležitost procvičit si programování v jazyce C/C++.

Seznam použité literatury a zdrojů

- [1] *Klasická versus inteligentní elektroinstalace* [online]
Dostupný z WWW: <<https://elektro.tzb-info.cz/domovni-elektroinstalace/7842-klasicka-versus-inteligentni-elektroinstalace>>
- [2] *Stránky LOXONE* [online]
Dostupný z WWW: <<http://www.loxone.com/cscz/start.html>>
- [3] *Stránky ABB* [online]
Dostupný z WWW: <<http://www.abb.cz/>>
- [4] *Logický modul LOGO!* [online]
Dostupný z WWW:
<<http://www1.siemens.cz/ad/current/index.php?vw=0&ctxnh=3dc1f5a3fc&ctxp=home>>
- [5] *OpenMotics* [online]
Dostupný z WWW: <<https://www.openmotics.com/>>
- [6] *Stránky Papouch* [online]
Dostupný z WWW: <<https://www.papouch.com/cz/>>
- [7] *Philips HUE* [online]
Dostupný z WWW: <<https://www2.meethue.com/>>
- [8] *Datasheet ATmega1284p* [online]
Dostupný z WWW: <<http://ww1.microchip.com/downloads/en/DeviceDoc/doc8059.pdf>>
- [9] *Raspberry Pi* [online]
Dostupný z WWW: <<https://www.raspberrypi.org/>>
- [10] *Datasheet NRF24L01+* [online]
Dostupný z WWW:
<https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus_Preliminary_Product_Specification_v1_0.pdf>
- [11] *Datasheet RTC DS1307* [online]
Dostupný z WWW: <<https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>>
- [12] *Datasheet teplotního senzoru DS18B20* [online]
Dostupný z WWW: <<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>>
- [13] *Datasheet ATmega328p* [online]
Dostupný z WWW: <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf>
- [14] *Knihovna RadioHead* [online]
Dostupný z WWW: <<http://www.airspayce.com/mikem/arduino/RadioHead/>>

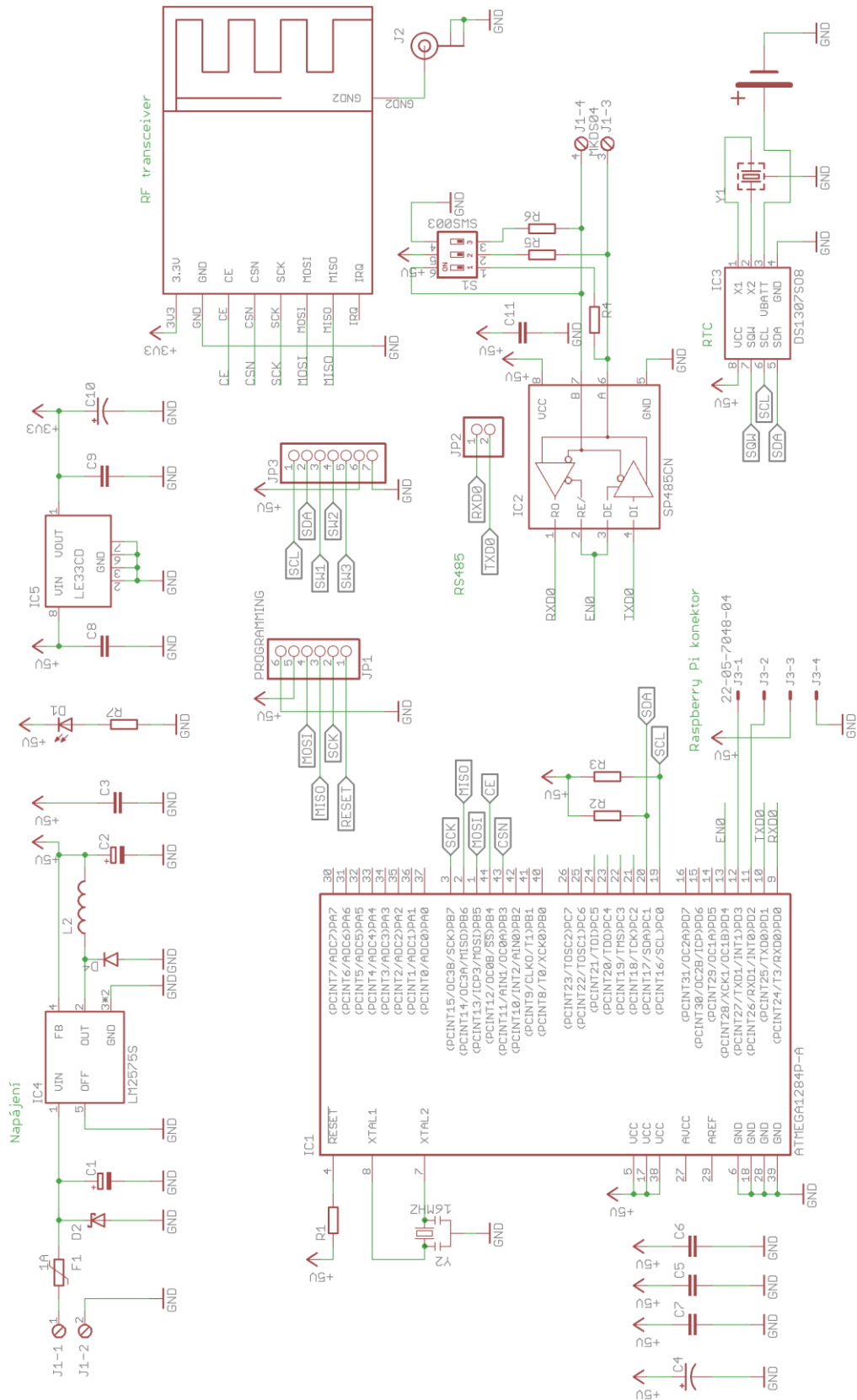
- [15] *Open Source projekty dostupné na GitHub* [online]
Dostupný z WWW: <<https://github.com/>>
- [16] Pravda, Ivan: *Systémy pro inteligentní budovy* [online]
Dostupný z WWW: <<https://publi.cz/books/239/Cover.html>>
- [17] *Datasheet DHT22* [online]
Dostupný z WWW:
<<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>>
- [18] Matoušek, David: *Práce s mikrokontroléry ATMEL AVR – ATmega16*

Všechny zdroje citovány k 24. 5. 2018

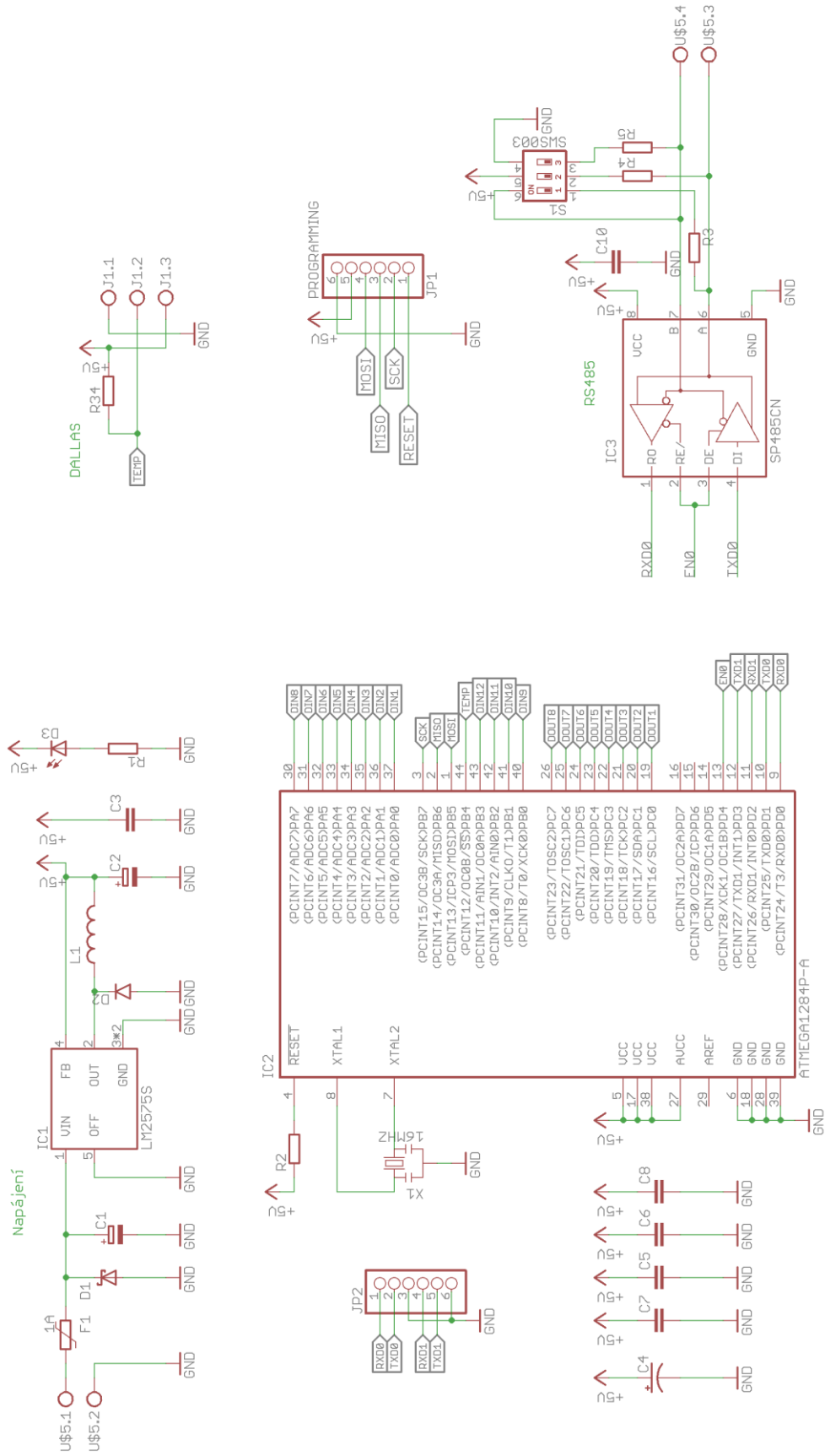
Seznam příloh

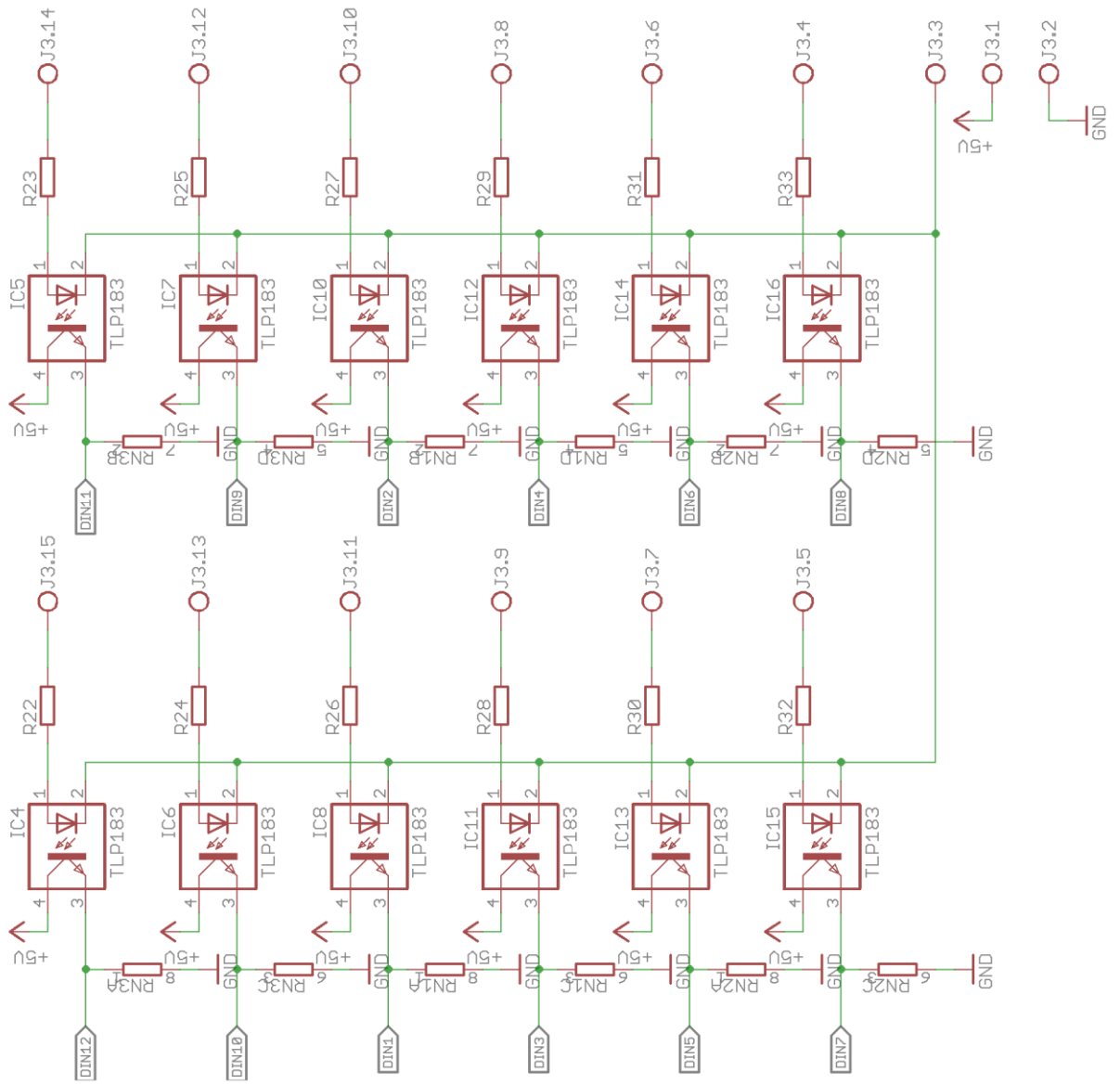
Příloha 1: Schéma zapojení řídicí jednotky.....	54
Příloha 2: Schéma zapojení IO modulu.....	55
Příloha 3: Schéma zapojení univerzálního bezdrátového uzlu.....	58
Příloha 4: Schéma zapojení bezdrátového relé.....	59
Příloha 5: Fotografie řídicí jednotky	60
Příloha 6: Fotografie IO modulu	60
Příloha 7: Fotografie univerzálního bezdrátového uzlu v konfiguraci teplotního senzoru	61
Příloha 8: Fotografie bezdrátového relé	62
Příloha 9: Fotografie demonstračního modelu	63
Příloha 10: Elektronické přílohy na CD	64

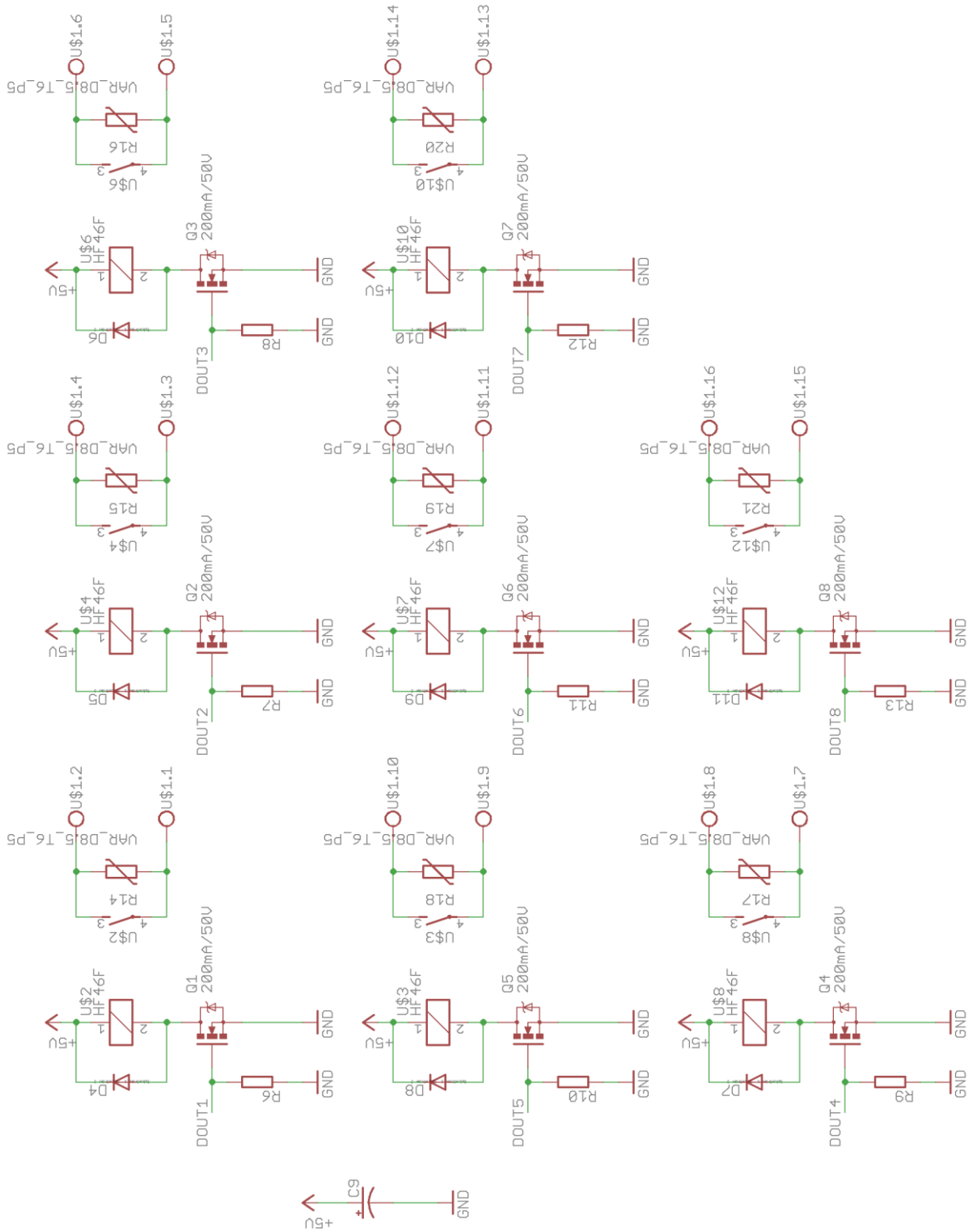
Příloha 1: Schéma zapojení řídicí jednotky



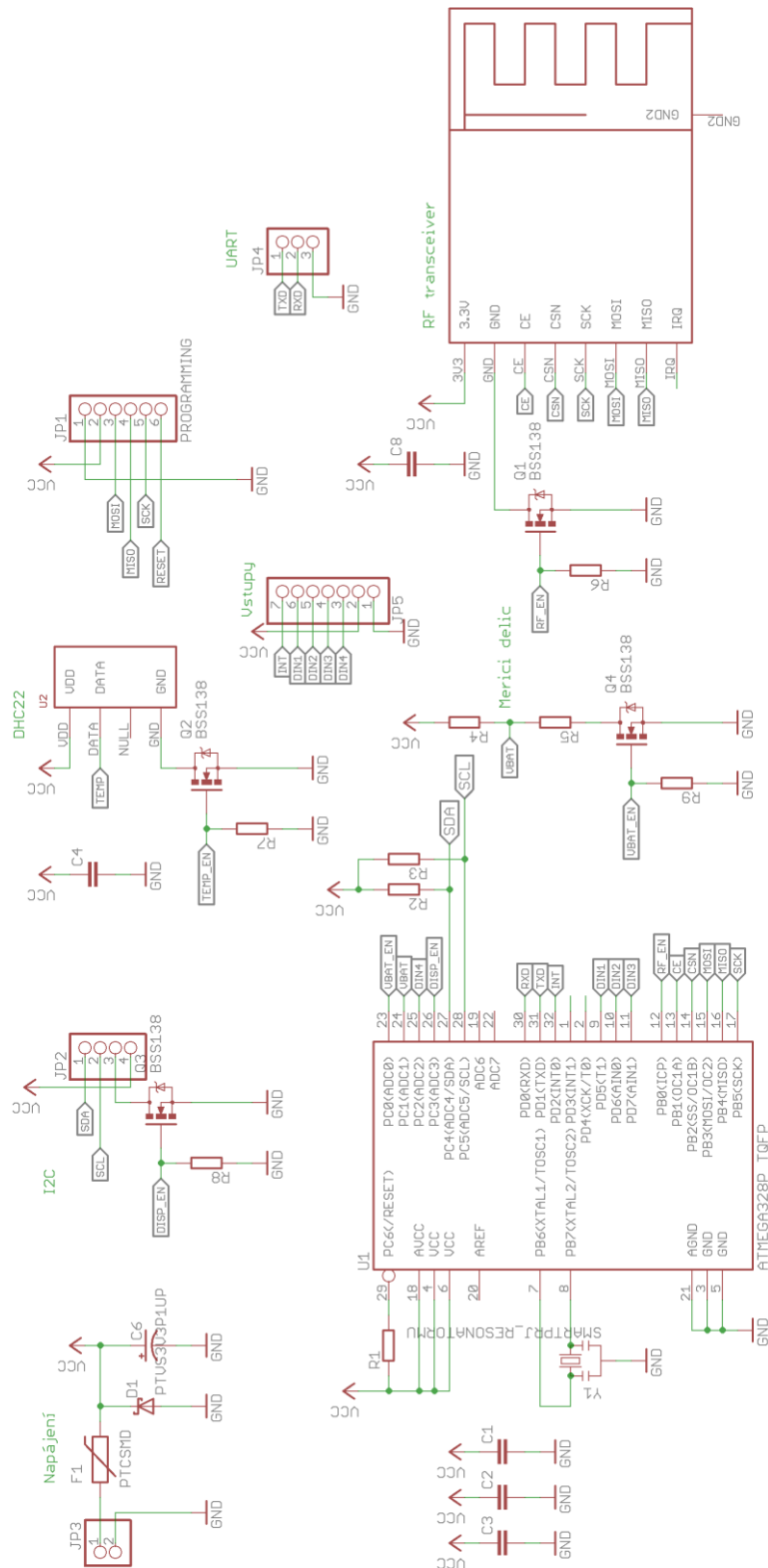
Příloha 2: Schéma zapojení IO modulu



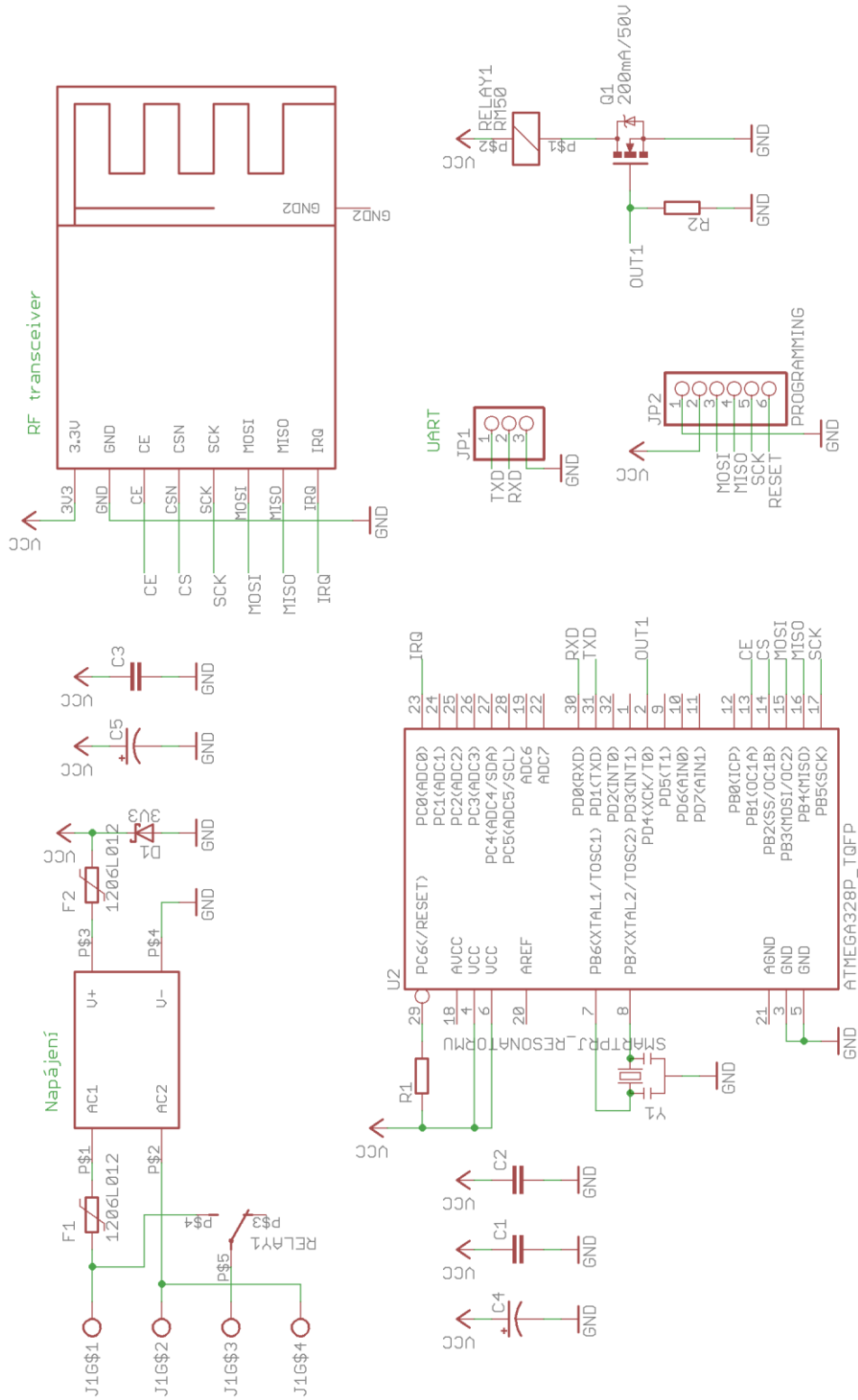




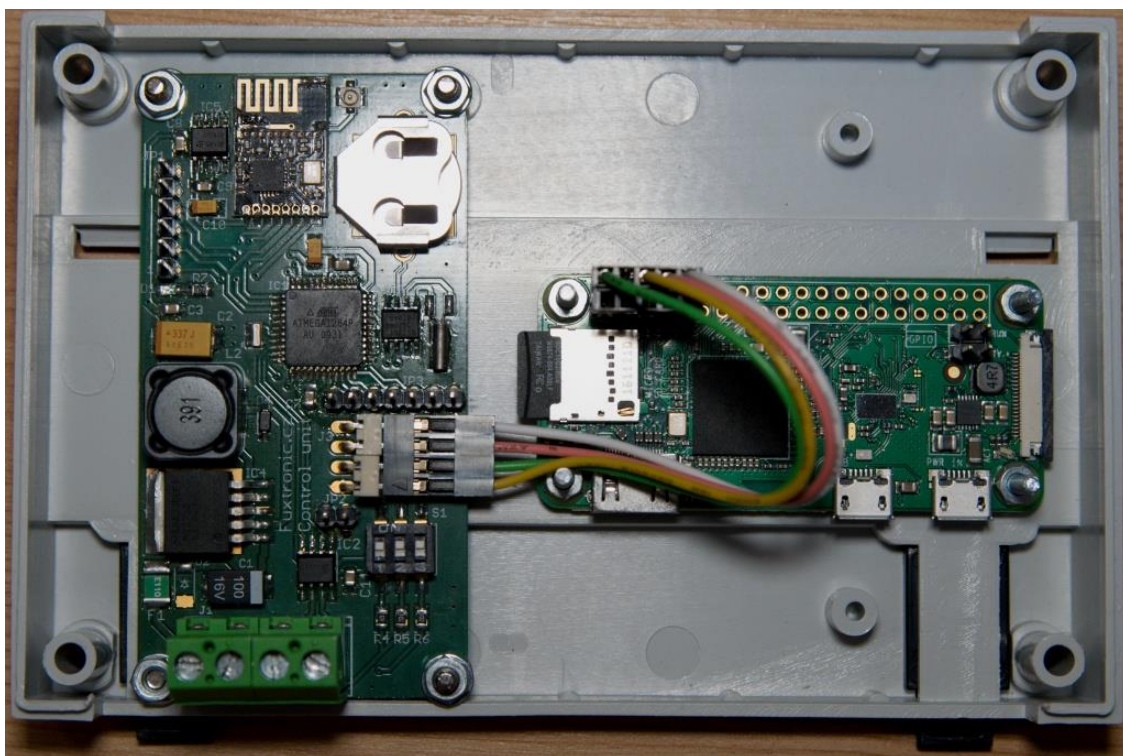
Příloha 3: Schéma zapojení univerzálního bezdrátového uzlu



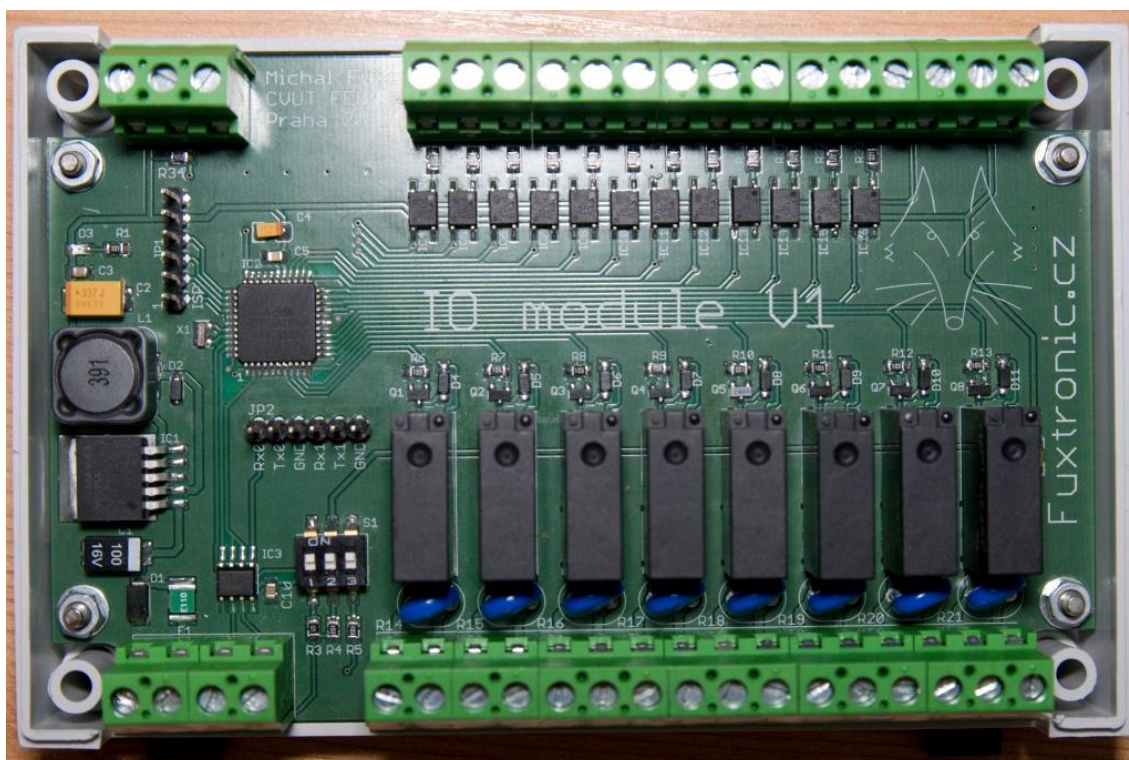
Příloha 4: Schéma zapojení bezdrátového relé



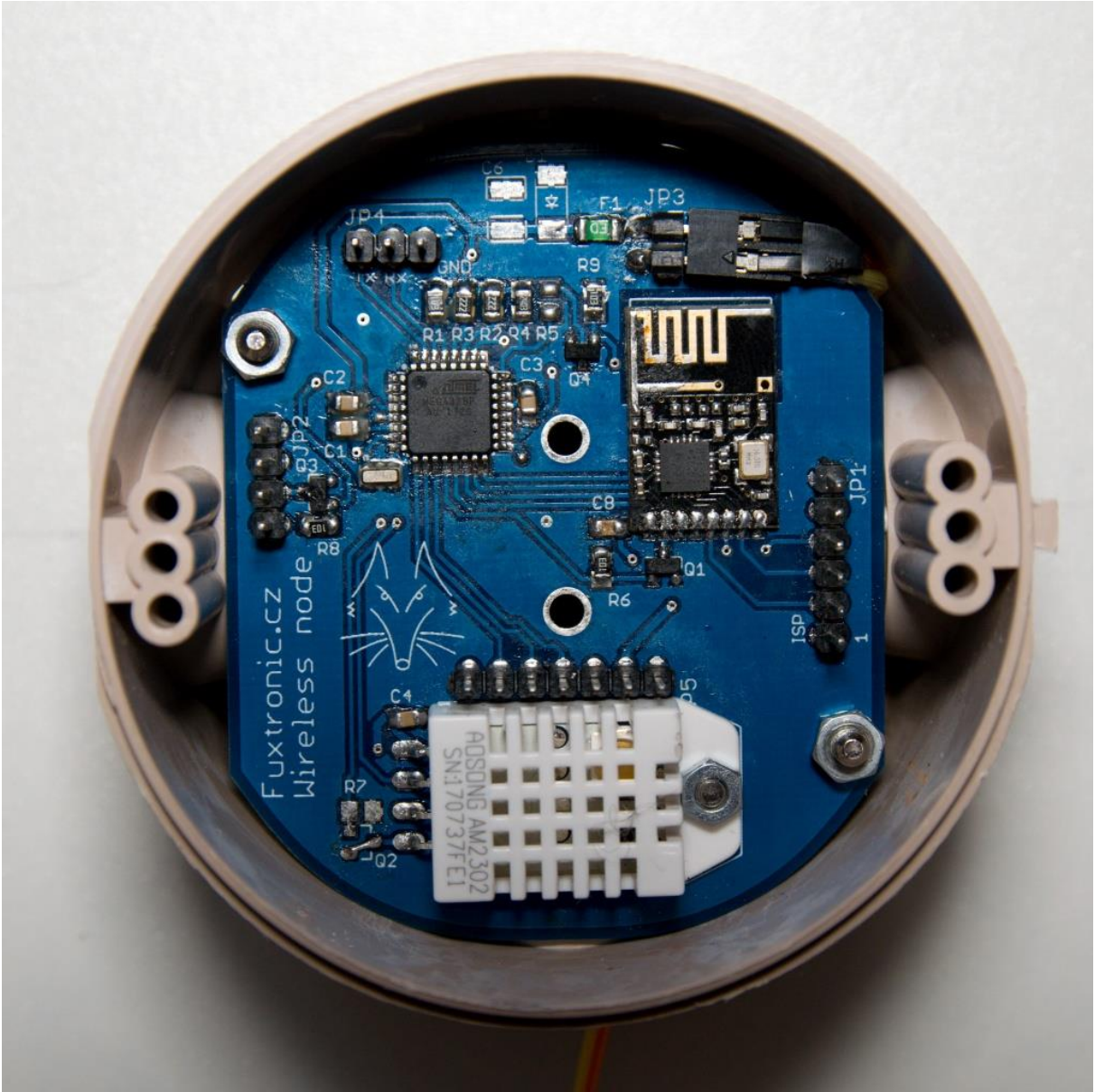
Příloha 5: Fotografie řídicí jednotky



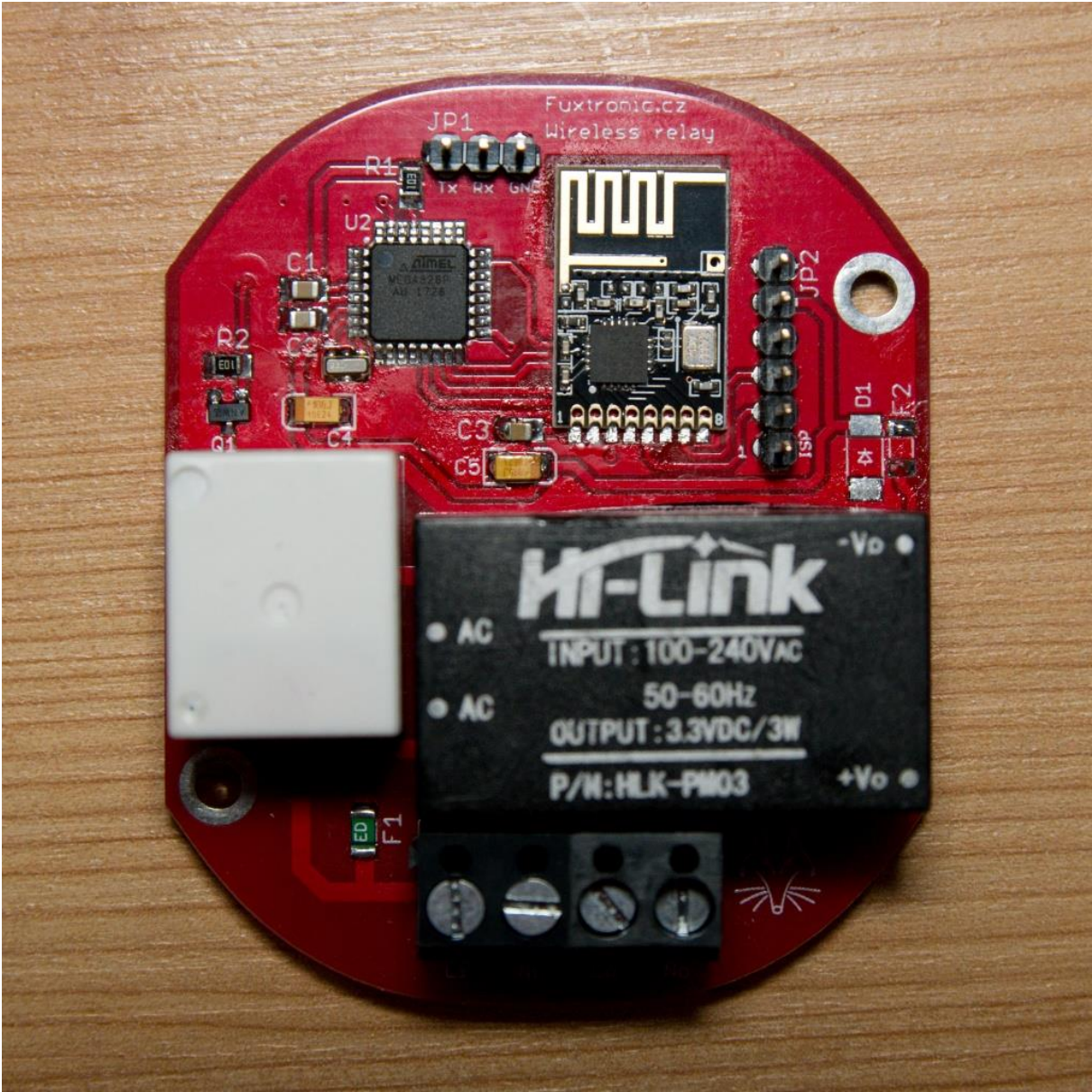
Příloha 6: Fotografie IO modulu



Příloha 7: Fotografie univerzálního bezdrátového uzlu v konfiguraci teplotního senzoru



Příloha 8: Fotografie bezdrátového relé



Příloha 9: Fotografie demonstračního modelu



Příloha 10: Elektronické přílohy na CD

Součástí práce jsou rovněž všechny zdrojové kódy, schéma zapojení a výrobní podklady. Materiály jsou rozděleny do složek podle názvů jednotlivých modulů.