

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA ELEKTROMECHANIKY A VÝKONOVÉ ELEKTRONIKY

BAKALÁŘSKÁ PRÁCE

Návrh ovládání inteligentního domu

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2013/2014**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal HERRMAN**
Osobní číslo: **E13B0018K**
Studijní program: **B2644 Aplikovaná elektrotechnika**
Studijní obor: **Aplikovaná elektrotechnika**
Název tématu: **Návrh ovládání inteligentního domu**
Zadávací katedra: **Katedra elektromechaniky a výkonové elektroniky**

Z á s a d y p r o v y p r a c o v á n í :

Navrhněte vhodné postupy pro řízení inteligentního domu.

1. Popište jednotlivé části inteligentního domu.
2. Navrhněte a realizujte řídicí modul.
3. Vyberte vhodné rozhraní pro ovládací rozhraní a navrhněte praktické řešení. Preferujte bezdrátové řešení.
4. Navrhněte a realizujte uživatelskou řídicí aplikaci.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **20 - 30 stran**

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

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.


Vedoucí bakalářské práce:

Ing. Petr Weissar, Ph.D.

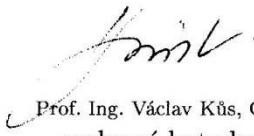
Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: **14. října 2013**

Termín odevzdání bakalářské práce: **9. června 2014**


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Prof. Ing. Václav Kůs, CSc.
vedoucí katedry

V Plzni dne 10. července 2013

Abstrakt

Cílem projektu je nastítnit problematiku řízení inteligentního domu, vytvoření jednoduchého a funkčního prototypu bezdrátového řízení jeho součástí. Prototypy zařízení jsou založeny na programovatelném modulu Arduino a jeho vývojovém prostředí, které je poskytováno zdarma na stránkách výrobce. Arduina jsou osazeny bezdrátovými komunikačními moduly RFM12B od společnosti Hope RF. Tyto moduly se vyznačují vysokou kvalitou, výkonem a přijatelnou cenou. Snímání teploty a vlhkosti vzduchu zajišťuje čidlo DHT11. Nedílnou součástí projektu je miniaturní počítač Raspberry Pi. Databáze, webový server i nástroje pro správu jsou nainstalovány na již zmíněném minipočítači. Za jehož největší výhoda je nízká spotřeba a malé rozměry. O propojení Arduina a Raspberry Pi se stará Ethernetový modul osazený čipem WizNet W5100.

Klíčová slova

Inteligentní dům, Arduino, Raspberry Pi, RFM12B, bezdrátový, databáze

Abstract

The project aims to control of smart house, creating a simple and functional prototype with wireless control. Prototypes of the device are based on programmable Arduino module and development environment is free on the manufacturer's website. Arduino modules are equipped with wireless communication modules RFM12B from Hope RF. These modules are of high quality, performance and low price. Sensor DHT11 provides reading of temperature and humidity. An integral part of the project is a miniature computer Raspberry Pi. Database, Web server and management tools are installed on the minicomputer. Its greatest benefits are low power consumption and small size. The connection between Arduino and Raspberry Pi provides Ethernet Shield with WizNet W5100 chip.

Key words

Smart house, Arduino, Raspberry Pi, RFM12B, wireless, database

Prohlášení

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

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

.....
podpis

V Plzni dne 9.6.2014

Michal Herrman

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM SYMBOLŮ A ZKRATEK	10
1 ÚVOD	11
2 NÁVRH A MOŽNOSTI INTELIGENTNÍHO DOMU	12
2.1 INTELIGENTNÍ DŮM.....	12
2.2 MOŽNOSTI INTELIGENTNÍHO DOMU	13
2.2.1 Osvětlení	13
2.2.2 Řízení teploty.....	13
2.2.3 Zabezpečení	14
2.2.4 Multimedia	14
2.2.5 Ekologický přínos.....	14
2.2.6 Ovládní systému.....	15
2.3 VÝHODY A NEVÝHODY BEZDRÁTOVÉHO ŘEŠENÍ.....	15
3 HARDWARE	16
3.1 POUŽITÝ HARDWARE.....	16
3.1.1 Modul Arduino UNO R3	16
3.1.2 Ethernetový modul (Ethernet Shield).....	17
3.1.3 Bezdrátové komunikační moduly	18
3.1.4 Senzor teploty a vlhkosti vzduchu	20
3.1.5 Raspberry Pi	20
3.2 NAVRŽENÝ A REALIZOVANÝ HARDWARE	21
3.2.1 Výběr napájecího zdroje	21
3.2.2 Spínání žárovky.....	22
3.3 NÁVRH MINIMALIZACE ZAŘÍZENÍ.....	23
3.4 PLÁNOVANÉ POUŽITÍ SYSTÉMU	23
3.4.1 Alternativy blokového zapojení.....	23
3.5 NÁVRH DÁLKOVÉHO SPÍNAČE	24
4 SOFTWARE A NAPROGRAMOVÁNÍ	25
4.1 SOFTWARE PRO RASPBERRY PI	25
4.1.1 Web server a databáze	25
4.2. SOFTWARE PRO ARDUINO	26
4.2.1 Zápis dat do databáze	26
4.2.2 Bezdrátový přenos dat.....	27
4.3 NÁVRH UŽIVATELSKÉ ŘÍDÍCÍ APLIKACE	27
4.4 MOŽNOSTI REGULACE TEPLoty	28
5 VZOROVÁ REALIZACE	31
6 ZÁVĚR	32
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	33
PŘÍLOHY	34
A FOTOGRAFIE PROTOTYPŮ	34

B SCHÉMA ZAPOJENÍ ŘÍZENÉHO MODULU.....	36
C DESKY PLOŠNÝCH SPOJŮ	37
D PROGRAM ŘÍDÍCÍHO ARDUINA	38
E PROGRAM PRO ŘÍZENÉ ARDUINO	41
F PHP KÓD.....	42

Seznam obrázků

OBR. 1.1: SCHÉMA FUNKCÍ INTELIGENTNÍHO DOMU	12
OBR. 3.1: ARDUINO UNO R3 (HTTP://ARDUINO.CC/EN/UPLOADS/MAIN/ARDUINOUNOSMD450PX.JPG)	16
OBR. 3.2: ARDUINO PINY (HTTP://ARDUINO.CC/EN/UPLOADS/HACKING/ATMEGA168PINMAP2.PNG).....	17
OBR. 3.3: ETHERNET SHIELD A ARDUINO (HTTP://WWW.NTPRO.NL/BLOG/UPLOADS/ARDUINO.PNG)	18
OBR. 3.4: MODUL 2,4 GHZ (HTTP://WWW.DIGIBAY.IN/IMAGE/CACHE/DATA/SACHINELECTRONICS/ED01/ED01-09-0027-A-600X600.JPG)	19
OBR. 3.5: MODUL 868MHZ (HTTP://SKPANG.CO.UK/CATALOG/IMAGES/WIRELESS/434MHZ/09582-01.JPG)	19
OBR. 3.6: ZAPOJENÍ DHT11 (HTTP://WWW.BRISTOLWATCH.COM/ARDUINO/ROTARY_ENCODER/DHT11.JPG)	20
OBR. 3.7: RASPBERRY PI (HTTP://WWW.SCRATCHMYPI.CO.UK/WP-CONTENT/UPLOADS/2013/12/PI1L.JPG).....	21
OBR. 3.8: ODPOROVÝ PŘEVODNÍK LOGICKÉ ÚROVNĚ	21
OBR. 3.9: BLOKOVÉ SCHÉMA ZAPOJENÍ.....	22
OBR. 3.10: BLOKOVÉ SCHÉMA DÁLKOVÉHO SPÍNAČE.....	24
OBR. 4.1: GRAFICKÝ NÁVRH OVLÁDACÍHO ROZHRANÍ	28
OBR. 4.2: EKVITERMNÍ KŘIVKY	29
OBR. A.1: ŘÍDÍCÍ MODUL	34
OBR. A.2: ŘÍZENÝ MODUL.....	35
OBR. A.3: DETAIL NAVRŽENÉ DPS	35
OBR. B.1: SCHÉMA ZAPOJENÍ	36
OBR. C.1: DPS ŘÍDÍCÍHO ARDUINA (BEZ DHT11).....	37
OBR. C.2: DPS PRO SPÍNÁNÍ ŽÁROVKY	37

Seznam symbolů a zkratk

RFID	Radio Frequency Identification. Identifikace na rádiové frekvenci
PWM	Pulse Width Modulation. Pulzně šířková modulace
SRAM	Static Random Access Memory. Statická paměť s libovolným přístupem
EEPROM	Electrically Erasable Programmable Read-Only Memory Elektricky mazatelná programovatelná paměť pro čtení
USART	Universal Synchronous / Asynchronous Receiver and Transmitter. Synchronní / asynchronní sériové rozhraní
I2C	Inter-Integrated Circuit. Multi-masterová počítačová sériová sběrnice
SPI	Serial Peripheral Interface. Sériové periferní rozhraní
USB.....	Universal Serial Bus. Univerzální sériová sběrnice
PoE	Power over Ethernet. Napájení po datovém síťovém kabelu
HDMI	High-Definition Multi-media Interface. Multimediální rozhraní s vysokým rozlišením
RAM	Random Access Memory. Paměť s libovolným přístupem
GPIO	General Purpose Input Output. Univerzální vstupy a výstupy
CRC	Cyclic Redundancy Check. Cyklický redundantní součet
ACK	Acknowledge. Potvrzení

1

Úvod

Bakalářská práce se zabývá návrhem ovládní inteligentního domu, praktickou realizací a nápady pro budoucí vylepšení tohoto komplexního systému. Popisuje specifický výběr použitého hardwaru. Nastiňuje potřebné softwarové vybavení a naprogramování.

Práce popisuje problematiku řízení inteligentního domu se zaměřením na bezdrátovou technologii. Základním stavebním článkem projektu je modul Arduino s bezdrátovým komunikačním modulem na frekvenci 868MHz. Veškerá data jsou ukládána do databáze na minipočítači Raspberry Pi. V práci jsou popsány části již vytvořených prototypů zařízení i návrhy připravené pro budoucí vývoj systému. Hlavním záměrem je vytvoření funkčního systému, který by bylo možné instalovat ve stávajících elektroinstalacích domů a bytů bez velkých zásahů.

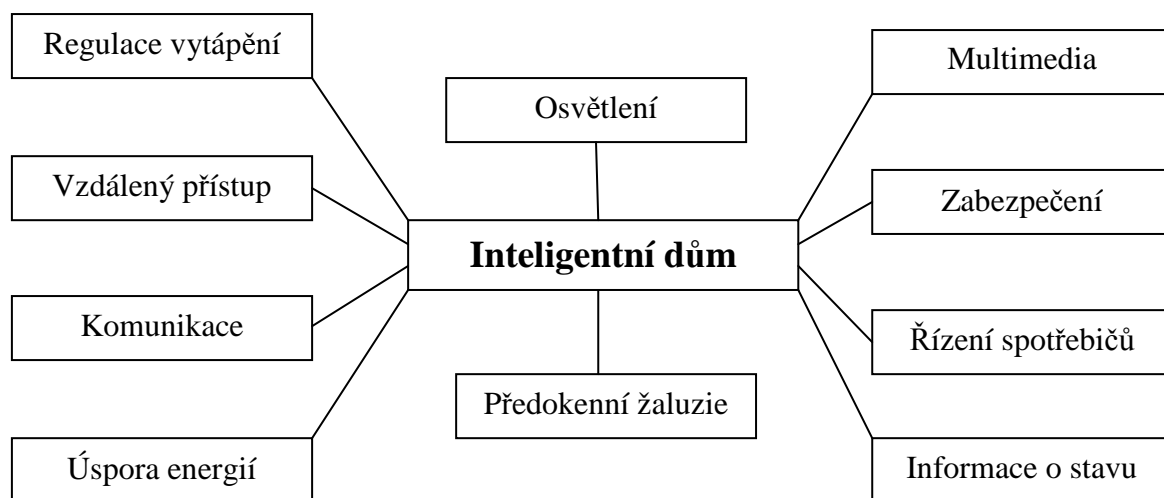
Práce je rozdělena do tří hlavních kapitol. První popisuje představu funkcí inteligentního domu, jeho možnosti a výhody. V druhé části je podrobně popsán použitý hardware a v poslední části je veškeré programové vybavení projektu.

2

Návrh a možnosti inteligentního domu

2.1 Inteligentní dům

Inteligentní dům představuje chytrou elektroinstalaci v domě či bytě, která uživateli značně zjednodušuje ovládání a poskytuje informace o všem, co se právě děje. Ve své podstatě se jedná o automatizaci domu. Pomocí počítače nebo jiného zařízení, které má přístup k řídicí jednotce lze ovládat světla, vytápění a vše na co si uživatel jen vzpomene. Systém umožňuje sledovat teplotu v každé místnosti, venkovní teplotu či aktuální spotřebu elektrické energie. Zabezpečení také není problém řídit kdykoliv a odkudkoliv pomocí mobilního telefonu, tabletu či počítače. Veškeré funkce a nastavení musí být jednoduché, přehledné a srozumitelné i pro laika.



Obr. 1.1: Schéma funkcí inteligentního domu

Základem inteligentního domu je řídicí jednotka zajišťující komunikaci jednotlivých zařízení mezi sebou a přístup uživatele k jednotlivým funkcím systému. Řídicí jednotka je propojena se zařízeními drátovou instalací nebo bezdrátově. Bezdrátová instalace je vhodná především tam, kde už je elektroinstalace hotová a nechceme vysekávat zdi pro dodatečnou instalaci systému. Tuto možnost mohou využít i nájemníci panelových domů. Opakem jsou

novostavby, kde se elektroinstalace bude teprve navrhovat a lze ji tedy v projektu přizpůsobit. Samozřejmostí je kompatibilita obou verzí pro případné další instalace neuvažované v původním projektu. Práce se zabývá bezdrátovým řešením.

2.2 Možnosti inteligentního domu

2.2.1 Osvětlení

Nejjednodušší možnou funkcí je ovládání osvětlení ve všech částech domu. Uživatel má přístup k ovládání jednotlivých světel a může upravovat i jas světel. V přehledu má k dispozici seznam aktuálně zapnutých světel i statistiky podle kterých lze upravovat automatické řízení osvětlení a poté například vypnout světla dle časového harmonogramu. Systém dokáže rozeznat, že v domě nikdo není pomocí pohybových čidel a vypnout osvětlení v místnosti. Všechny tyto funkce mají za úkol šetřit elektrickou energii a můžou být vypnuty nadřazeným příkazem od uživatele. Nebylo by přijatelné, aby se světla vypínala, pokud by si uživatel v místnosti četl nebo jen relaxoval. Nesmí se zapomínat, že hlavním účelem inteligentního domu je komfort pro uživatele.

2.2.2 Řízení teploty

Další nedílnou součástí je řízení vytápění v zimě, popřípadě klimatizace v létě. V každé místnosti, kde chceme automaticky upravovat teplotu, musí být teplotní senzor a aktuátor na radiátoru nebo podlahovém topení v případě ústředního vytápění. Tam kde jsou instalovány pro vytápění elektrické přímotopy, může je systém jednoduše řídit přímo. Automaticky by systém měl na noc snížit teploty v celém domě, udržovat nastavené teploty ve všech místnostech a vytvářet ideální profil pro maximální účinnost a minimální spotřebu. Informace o všech stavech a teplotách se zobrazují přehledně v uživatelském prostředí.

2.2.3 Zabezpečení

Zabezpečení je v dnešní době podstatnou součástí inteligentního domu. V nepřítomnosti uživatelů se aktivuje, pohybová čidla snímají prostředí a čidla na oknech a dveřích hlídají nevíтанé vstupy. Systém dokáže poté upozornit uživatele hovorem nebo odesláním sms. Jednou z rozšířených možností je kontaktování policie nebo soukromé bezpečnostní agentury. Přístup do domu je zajištěn klasicky klíčem, RFID nebo kódem. Možnosti jsou ve své podstatě neomezené a záleží pouze na přáních uživatele. Bezpečnostní kamery, interkomy, snímače otisků prstů a veškeré moderní bezpečnostní zařízení.

2.2.4 Multimedia

Nesmí být opomenut čas na relaxaci. Sdílení médií je každodenní potřebou a systém inteligentního domu to musí podporovat. Na každém zařízení se lze připojit k centrálnímu serveru a přehrávat hudbu, sledovat filmy nebo se připojit k internetu a sociálním sítím. Celý systém je založený na oprávněních. Vzhledem k předpokládanému připojení všech uživatelů v domě musí mít například rodiče nadřazená práva oproti dětem a mohou ovlivňovat jejich práva, zakazovat přístup k nedovoleným internetovým stránkám nebo jim vypnout světla, když mají spát. Celý koncept by mohl mít i vlastní interní komunikační hlasový a textový program. Buzení uživatelů ráno příjemnou hudbou v nastavený čas a den, roztažení žaluzií či automatické spuštění kávovaru také zvyšuje pocit pohodlí a zpříjemňuje den.

2.2.5 Ekologický přínos

Vzhledem k tomu, že dnes se bere velký ohled na ekologii, je inteligentní dům ideální. Stará se o zbytečně rozsvícená světla, přetopené místnosti, zapomenuté otevřené dveře a mnoho dalšího. Na noc zavírá předokenní žaluzie pro lepší izolaci domu. Systém sám o sobě nesmí mít velkou vlastní spotřebu a musí umět efektivně šetřit elektrickou energii. Úspory poté mohou dosáhnout jednotek až desítek procent.

2.2.6 Ovládání systému

Podle předpokladů by měl inteligentní dům postupně přecházet v interaktivní a příjemné prostředí pro jeho uživatele, zajišťovat bezpečí osob i majetku v domě a poskytovat všeskeré pohodlí a zábavu pro uživatele všech věkových kategorií. Do budoucna lze uvažovat i o komunikaci se systémem prostým hlasem, která by vše umocnila. Další vizí je dům s umělou inteligencí nebo alespoň iluzí inteligence.

Pro přístup k informacím a ovládání inteligentního domu mimo domácí síť, se může uživatel k systému připojit přes internet. Systém ukládá informace na vzdálený zabezpečený server, ke kterému má uživatel po přihlášení přístup. Uživatel se přihlašuje k systému svým uživatelským jménem a heslem. Důležitá je bezpečnost, aby nedošlo ke zneužití údajů ze systému nebo ke ztrátě kontroly nad systémem.

2.3 Výhody a nevýhody bezdrátového řešení

Pro drátovou instalaci inteligentního domu je nutný velký zásah do stavby a s tím jsou spojené i velké náklady na realizaci projektu. Stávající firmy nabízejí kompletní řešení za cenu v řádech několika set tisíc a ne každý si může dovolit takovou investici. Bezdrátové řešení nabízí možnost postupného budování inteligentního domu pouhým přidáním dalšího modulu do systému. Uživateli stačí pořídit si server (Raspberry Pi), který jednoduše připojí ke své domácí síti a poté už jen řídicí jednotku a moduly pro spotřebiče, které chce ovládat. Celý systém přitom může ovládat pouze svým chytrým telefonem, tabletem nebo počítačem. Není tedy zapotřebí pořizovat další drahé ovládací zařízení.

3

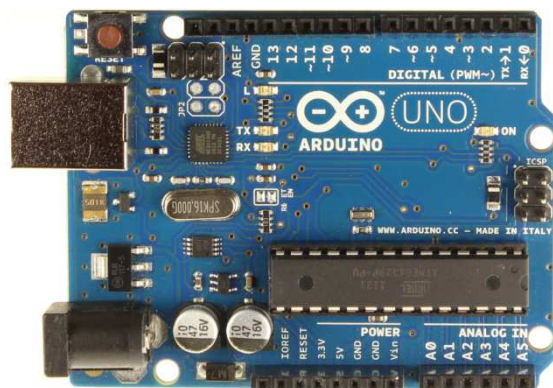
Hardware

3.1 Použitý hardware

V kapitole se budeme zabývat hardwarem použitým při vývoji prototypu.

3.1.1 Modul Arduino UNO R3

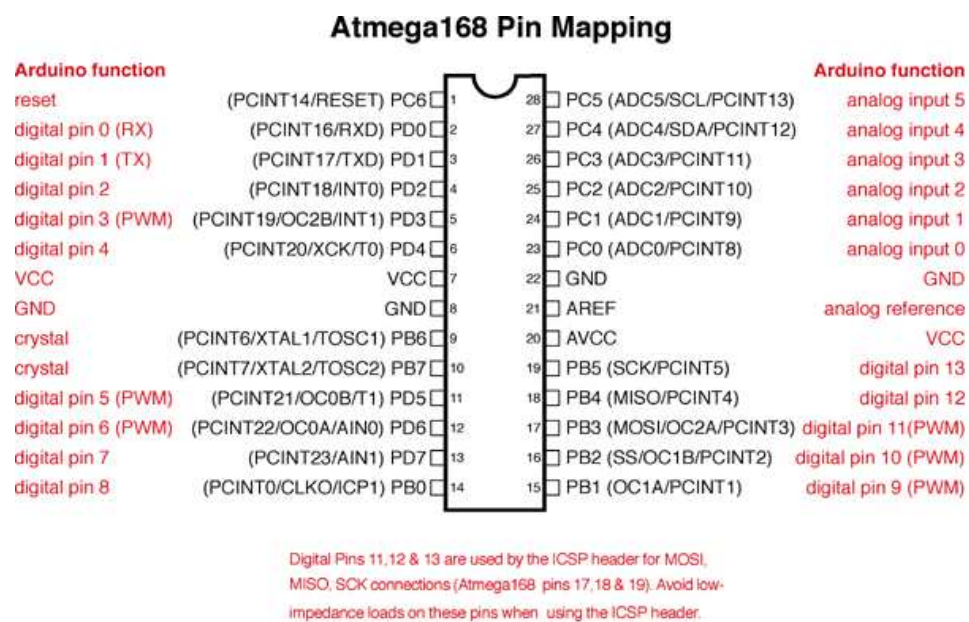
V práci je použita jako základní stavební jednotka modul Arduino. Konkrétně se jedná a poslední prodávaný model řady UNO s označením R3, který se na českém trhu pohybuje kolem 500Kč.



Obr. 3.1: Arduino UNO R3 ([HTTP://ARDUINO.CC/EN/UPLOADS/MAIN/ARDUINOUNOSMD450PX.JPG](http://arduino.cc/en/uploads/main/arduinounoSMD450px.jpg))

Arduino UNO R3 je vývojový modul osazený mikroprocesorem ATmega328. K dispozici má 14 digitálních vstupů/výstupů z nichž 6 je použitelných jako PWM výstup na portech 3,5,6,9,10,11 a 6 jako analogový vstup s 10 bitovým rozlišením na pinech A0 až A5. K dispozici máme 32KB Flash paměti, 2KB SRAM a 1KB EEPROM. Krystal o frekvenci 16 MHz zajišťuje takt procesoru. Zařízení lze napájet pomocí USB portu (5V) nebo napájecím konektorem 2.1mm umístěným na desce. Rozsah doporučeného vstupního napětí přes napájecí konektor je 7-12V DC. Při nedodržení vstupního napětí může dojít k nestabilitě procesoru nebo přehřívání napěťového regulátoru. Výběr napájecího vstupu je volen deskou

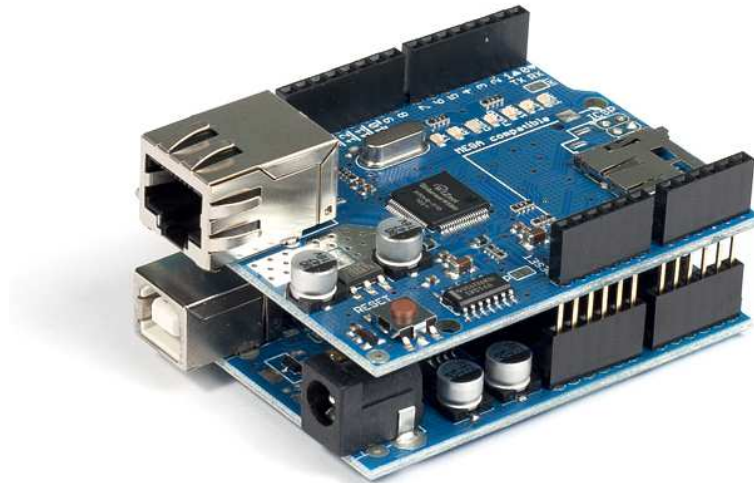
automaticky. Vstupní i výstupní piny mohou poskytnout až 40mA proudu. Arduino může s ostatními moduly komunikovat několika způsoby. Mezi ně patří klasická UART sériová komunikace na portech 0(RX) (Recieve) a 1(TX) (Transmit), I2C a SPI. Ve své práci používám SPI na pinech 8-10 (SS) (Slave Select), 11 (MOSI) (Master Output Slave Input), 12 (MISO) (Master Input Slave Output), 13 (SCK) (Slave Clock). Je třeba zmínit i proudovou vratnou pojistku při napájení přes USB port, která přeruší přísun proudu při překročení 500mA.



Obr. 3.2: Arduino piny ([HTTP://ARDUINO.CC/EN/UPLOADS/HACKING/ATMEGA168PINMAP2.PNG](http://arduino.cc/en/uploads/hacking/atmega168pinmap2.png))

3.1.2 Ethernetový modul (Ethernet Shield)

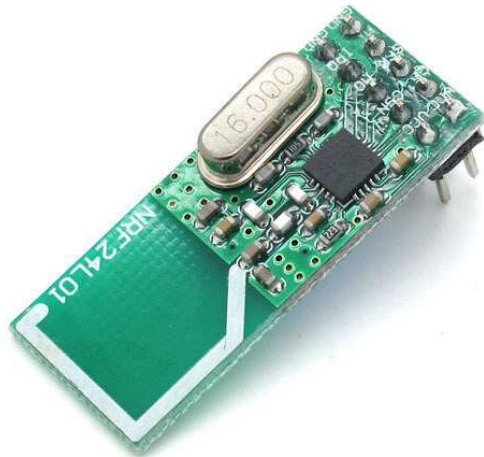
Arduino je rozšiřitelné velkým množstvím modulů. V projektu je použit ethernetový modul (Ethernet Shield) bez PoE, který je napájen 5V přímo z Arduina a komunikuje sním skrze SPI sběrnici. Čip WIZnet W5100 s 16KB bufferem osazený na ethernetovém modulu zajišťuje veškeré funkce. Rychlost připojení k síti je 10/100Mb standardním konektorem RJ-45 a kabelem cat5e. Do modulu lze vložit microSD kartu pro ukládání dat.



Obr. 3.3: Ethernet Shield a Arduino ([HTTP://WWW.NTPRO.NL/BLOG/UPLOADS/ARDUINO.PNG](http://www.ntpro.nl/blog/uploads/arduino.png))

3.1.3 Bezdrátové komunikační moduly

Při prvních pokusech byly použity bezdrátové komunikační moduly s čipem NRF24L01+ pracující na frekvenci 2,4GHz. Modul se zdál ideální a jeho cena se pohybovala kolem 60Kč za kus. Výrobce udává spotřebu je maximálně 15 mA při komunikaci a 900nA v klidovém režimu. Přenosová rychlost je teoreticky až 2Mbps a dosah až 100m v otevřeném prostoru. Nicméně první problém nastal při zapojení a pokusu o komunikaci těchto modulů. Na krátkou vzdálenost (cca 50cm) vše fungovalo perfektně. Při zvyšování vzdálenosti postupně docházelo k chybám, až se spojení přerušilo úplně při vzdálenosti jen několika metrů v interiérových podmínkách a bez překážek. Zjištěná skutečnost je přisuzována velkému rušení v pásmu 2,4GHz a malému výkonu modulu.



Obr. 3.4: Modul 2,4 GHz ([HTTP://WWW.DIGIBAY.IN/IMAGE/CACHE/DATA/SACHINELECTRONICS/ED01/ED01-09-0027-A-600x600.JPG](http://www.digibay.in/image/cache/data/sachinelectronics/ED01/ED01-09-0027-A-600x600.JPG))

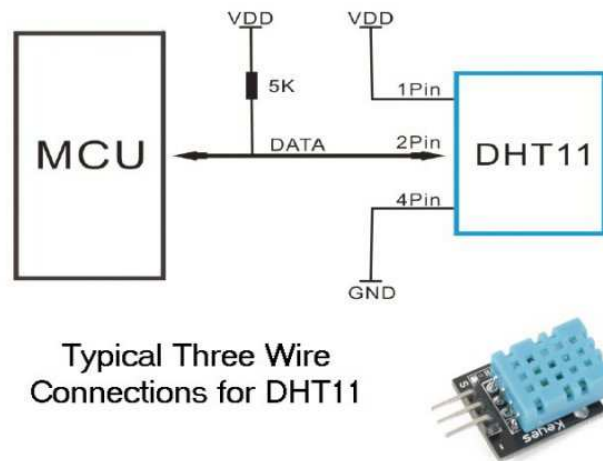
Pro další pokusy byly použity transcievery RFM12B od firmy HopeRF. Cena je zhruba dvakrát vyšší než u předchozího modulu. Moduly lze programově nastavit na provoz v jednom ze tří pásem (433, 868 nebo 915MHz). Výrobce udává přenosovou rychlost až 115.2 kbps při použití vnitřního demodulátoru a až 256 kbps s externím RC filtrem. Komunikace je skrze SPI sběrnici a napájecí napětí je v rozsahu 2,2 až 3,7V. Spotřeba zařízení je přitom stejná jako u modulů 2,4GHz. Podrobnější informace o spotřebě nalezneme v datasheetu. Při testování dosahu v domě modul bez problémů zvládal přenos přes celý dům i se zavřenými dveřmi, proto byl vybrán pro použití projektu.



Obr. 3.5: Modul 868MHz ([HTTP://SKPANG.CO.UK/CATALOG/IMAGES/WIRELESS/434MHZ/09582-01.JPG](http://skpang.co.uk/catalog/images/wireless/434MHz/09582-01.JPG))

3.1.4 Senzor teploty a vlhkosti vzduchu

Měření teploty a relativní vlhkosti vzduchu zajišťuje čidlo DHT11. Čidlo se nevyznačuje velkou přesností ani rychlostí, ale pro tento projekt je dostačující. Spotřeba čidla je průměrně 1mA. Pin 3 zůstává nepřípojen.



Obr. 3.6: Zapojení DHT11 ([HTTP://WWW.BRISTOLWATCH.COM/ARDUINO/ROTARY_ENCODER/DHT11.JPG](http://www.bristolwatch.com/arduino/rotary_encoder/dht11.jpg))

3.1.5 Raspberry Pi

Od začátku bylo předpokládáno zahrnutí databáze v projektu, bylo třeba vybrat vhodný server. První z alternativ byl klasický notebook, nicméně spotřeba notebooku i bez zapnutého monitoru se pohybuje v řádech desítek Wattů. Pro projekt je vhodné zařízení Raspberry Pi, jehož spotřeba je maximálně 5W a tedy ideální pro trvalý provoz. Raspberry Pi je minipočítač osazený procesorem ARM1176JZF-S s taktem 700MHz, grafickým procesorem VideoCore IV a 512MB RAM (Rev B). Na desce je k dispozici HDMI port, 2 USB porty, RJ-45, 12 GPIO pinů, SPI a mnoho dalších. Nutností je připojená SD karta nahrazující HDD.

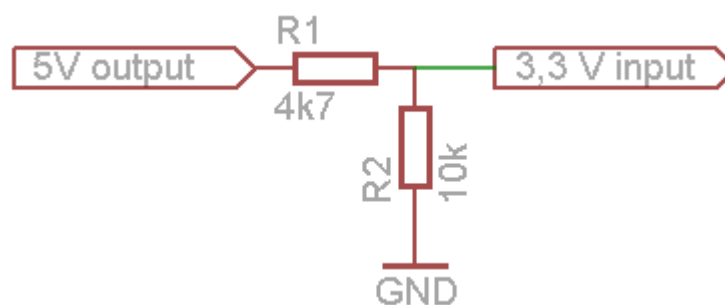
Raspberry Pi je spojeno s Ethernetovým modulem skrze Wi-fi router Tenda W316-R standardním síťovým kabelem cat5e, zároveň poskytuje požadovaný přístup k síti přes wi-fi. Router je napájen 9V adaptérem. Pro připojení je k dispozici 4xLAN a 1xWAN vstup. Bezdrátové připojení k síti slouží k ovládní systému přes mobilní telefon, tablet nebo notebook.



Obr. 3.7: Raspberry Pi (<http://www.scratchmypi.co.uk/wp-content/uploads/2013/12/pi1L.jpg>)

3.2 Navržený a realizovaný hardware

Pro připojení modulu RFM12B a čidla DHT11 byla navržena DPS, která se dá jednoduše připojit přímo na řídicí Arduino. Deska obsahuje ještě jednoduchý odporový převodník z 5V na 3,3V a tlačítko. Modul Pv1.1 připojený k řízenému Arduino je složitější. Obsahuje stabilizovaný 5V napájecí zdroj, RFM12B, Triak a obvod pro řízení triaku. Celý modul je jištěn nevratnou skleněnou pojistkou (1A) umístěnou v pouzdře.



Obr. 3.8: Odporový převodník logické úrovně

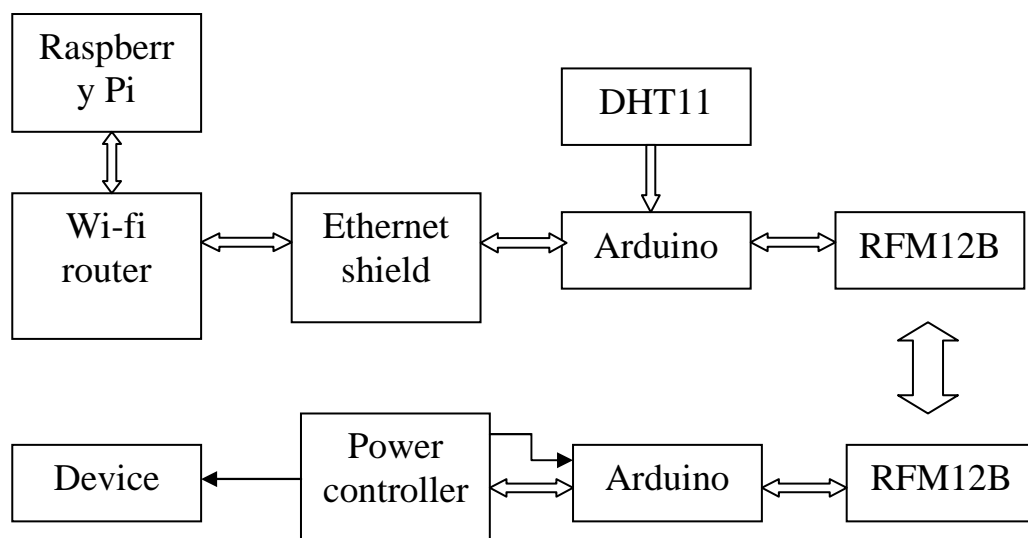
3.2.1 Výběr napájecího zdroje

Při výběru zdroje byl logickou volbou moderní spínaný zdroj. Spínané zdroje mají obecně vyšší účinnost a menší rozměry při vyšších výkonech, ale pro výkon v řádech jednotek Wattů by byla cena zdroje mnohem vyšší než při použití transformátoru. Lineární stabilizovaný zdroj na modulu tvoří transformátor 230V/6V 1,5VA, dvoucestný usměrňovač

z diod 1N4148 v můstkovém zapojení, integrovaný lineární regulátor napětí HT7533-1 a filtrační kondenzátor pro správnou funkci regulátoru. Výstupní sinusové napětí je nejprve dvoucestně usměrněno a přivedeno na kondenzátor, poté je stabilizováno na hodnotu 3,3V pro napájení celého modulu. Maximální možný proud odebíraný ze zdroje je 100mA vzhledem k omezení regulátoru. Pro další minimalizaci mám připravený miniaturní transformátor o výkonu 0,35VA a rozměrech pouhých 21,7 x 22,6 x 15 mm.

3.2.2 Spínání žárovky

Obvod pro spínání a regulaci žárovky obsahuje Triak BT136 800V/4A, integrovaný obvod MOC3022 pro řízení triaku, obvod pro detekci fáze vstupního napětí, tvořený optočlenem. Obvod MOC3022 je miniaturní SSR relé, jedná se v podstatě o spínací optočlen se silovou výstupní částí a zajišťuje galvanické oddělení řídicího členu od silového spínaného členu. Výstupní část zvládá proud až 1A a napětí 400V. Při prvních pokusech byl použit obvod MOC3022 přímo k řízení žárovky 230V/60W, ale při zahřátí obvod zůstal v sepnutém stavu a nešel vypnout jinak, než odpojením zátěže, pro další pokus byl použit obvod MOC3022 pouze pro řízení triaku a jeho galvanické oddělení od Arduina.



Obr. 3.9: Blokové schéma zapojení

3.3 Návrh minimalizace zařízení

Prototyp modulu, který ovládá spotřebič (žárovku) je pro použití v elektroinstalaci v domě nebo bytě velký, proto je zapotřebí ho značně minimalizovat. Největší součástí zařízení je transformátor, který bude nahrazen nejmenší možnou verzí. V úvahu jako zdroj napájení připadá i miniaturní spínaný zdroj používaný v adaptérech pro mobilní telefony. Adapter je čínské výroby, jeho cena se na internetu pohybuje kolem 50 Kč a maximální proud na výstupu dosahuje 1A při napětí 5V. Celé Arduino lze nahradit SMD verzí procesoru ATmega328 a ostatní součástky také lze nahradit jejich SMD verzí. Výsledný modul se musí dát nainstalovat do standardní instalační elektrickářské krabice.

3.4 Plánované použití systému

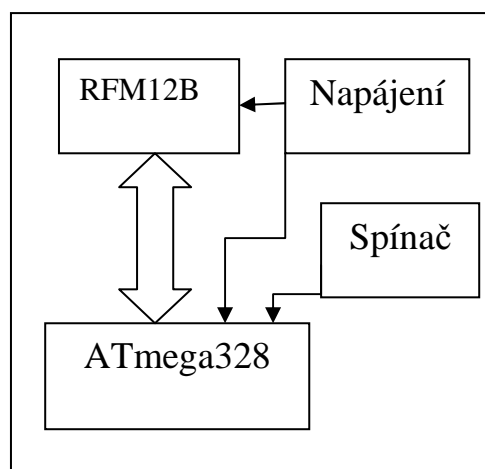
Systém by měl být nainstalován v bytě 1+1kk. Vstupní dveře budou z venku odemkány RFID přístupovým čipem, popřípadě klávesnicí pro zadání kódu. Dálkově nebo manuálně bude možnost ovládat veškeré osvětlení v bytě a ventilátor v koupelně. U postele bude umístěn dálkový spínač pro osvětlení. V bytě je ústřední topení a teplota v jednotlivých místnostech bude ovládána aktuátory umístěnými na radiátorových ventilech pro termostatické hlavice vzhledem k teplotě snímané senzory. Ve venkovních prostorách bude umístěno čidlo teploty a vlhkosti. V místnosti u vchodu bude umístěn tablet jako hlavní ovládací a přístupový panel do systému.

3.4.1 Alternativy blokového zapojení

Pro další verzi modulu je uvažováno s použitím Arduina a Raspberry Pi bez Ethernet modulu. Arduino by bylo připojené přímo k Raspberry Pi USB kabelem nebo ideálně SPI sběrnici. Komunikaci s Arduinem by zajistil skript napsaný v Pythonu.

3.5 Návrh dálkového spínače

Pro ovládání je ve vývoji dálkový spínač. Dálkový spínač je zařízení obsahující spínač nebo dotykové čidlo, mikroprocesor ATmega328 a modul RFM12B. Celé zařízení je o velikosti klasického vypínače používaného v elektroinstalacích a je napájeno bateriemi nebo akumulátorem. Pokud bude napájeno bateriemi 2xAAA, je zapotřebí napětí regulovat na 3,3V STEP-UP měničem. Při použití akumulátoru 3,6V (3,7V) nemusí být napětí nijak upravováno a zařízení lze opakovaně nabíjet.



Obr. 3.10: Blokové schéma dálkového spínače

4

Software a naprogramování

Pro programování modulů je volně k dispozici i vývojové prostředí Arduino IDE. Prostředí je založeno na programovacím jazyku C a poskytuje velké množství funkcí, které lze použít. K dispozici je na internetu i velké množství opensource knihoven právě pro Arduino. V projektu využívám knihovny pro RFM12B, DHT11, MySQL a Ethernetový modul.

4.1 Software pro Raspberry Pi

Samostatnou kapitolou je mikropočítač Raspberry Pi s operačním systémem Raspbian Wheezy. Jedná se o speciálně upravený systém založený na Debianu (Linuxu). Operační systém je nainstalovaný na připojené SD kartě. Po nainstalování systému bylo zapotřebí nejprve upravit rozdělení paměti. Přiřadil jsem jen malé procento grafickému procesoru, protože nepotřebuji přehrávat multimédia, pouze používat monitor pro správu. Výsledkem čehož je zvýšení výkonu serveru až o několik desítek procent, přesné číslo výrobce neudává. Za zmínku stojí i možnost přetaktování procesoru z 700MHz až na 1GHz, nicméně výrobce varuje, že přetaktování může omezit životnost výrobku vzhledem k nadměrnému zahřívání procesoru. Dalším krokem bylo nastavení vzdáleného přístupu přes ssh a instalace programů pro server.

4.1.1 Web server a databáze

Pro práci byl zvolen nejpoužívanější webový server Apache, přesněji tedy Apache2 s podporou PHP5, databázi MySQL a nástroj pro správu PhpMyAdmin umožňující přístup do databáze a sledování změn. Pro instalaci serveru Apache stačí napsat v terminálu příkaz `sudo aptitude install apache2`. Příkazem `sudo chown -R pi /var/www` se změní vlastník složky

z root na pi, aby se předešlo pozdějším komplikacím. Podobné příkazy pak nainstalují i ostatní potřebné aplikace a funkčnost lze ověřit zadáním IP adresy Raspberry Pi do internetového prohlížeče. Pokud vše funguje správně, zobrazí se jednoduchá stránka s textem „IT WORKS!“. Při instalaci MySQL je potřeba nastavit ještě přihlašovací údaje pro správu databáze. Pro připojení k databázi se spustím webový prohlížeč a na adrese 127.0.0.1/phpmyadmin se objeví přihlašovací obrazovka a po přihlášení i obsah databáze a její nastavení. Nejprve je zapotřebí přidat uživatele, přiřadit mu potřebná oprávnění pro přístup do databáze a nastavit buď IP adresu odkud se bude uživatel připojovat, nebo povolit připojování z jakékoliv IP adresy. V nástroji pro správu je možné vytvářet databáze, přidávat tabulky, nahlížet do obsahu tabulek. Stejně tak lze tabulky přidávat i přes arduino jednoduchými příkazy a to za pomoci již zmíněné knihovny MySQL.

4.2. Software pro Arduino

Program napsaný v Arduino IDE pro řídicí Arduino se skládá z několika dílčích částí. První je inicializace všech potřebných knihoven, jako například RFM12B.h, SPI.h, Ethernet.h a dalších. Další částí jsou jednotlivé funkce pro obsluhu připojených zařízení a poslední je smyčka zajišťující spouštění jednotlivých funkcí. Program každých 5 minut přečte aktuální teplotu z čidla DHT11 a zapíše naměřenou teplotu do tabulky v databázi a pokud dojde k sepnutí tlačítka, odešle příkaz pro sepnutí nebo vypnutí žárovky. Časová smyčka je řešena algoritmem využívajícím funkci millis(). Funkce millis() navrácí hodnotu v milisekundách od zapnutí Arduina. Podle výrobce tato hodnota přeteče přibližně po padesáti dnech nepřetržitého provozu.

4.2.1 Zápis dat do databáze

Při jednom z prvních pokusů došlo k problémům při zápisu teploty do databáze. Arduino úspěšně přečetlo teplotu z čidla a Ethernet modul se připojil k databázi, kam zapsal správnou hodnotu teploty i vlhkosti. Problém byl v knihovně MySQL, knihovna neobsahovala odpojení Ethernet klienta od databáze. Docházelo k hromadění neukončených připojení do databáze a po pěti připojeních nebylo již možné se k databázi připojit. Problém

se odstraní nejprve přidáním příkazu `client.stop()` a později aktualizací knihovny. V aktuální verzi je již obsažen `disconnector()`. Program do databáze zapisuje aktuální teplotu a vlhkost vzduchu z čidla DHT11 a k hodnotám přiřazuje datum a čas pořízení teploty.

4.2.2 Bezdrátový přenos dat

Přenos dat mezi moduly RFM12B je šifrován 128 bitovým kódem. Při každém přenosu probíhá CRC kontrola dat a lze odesílat i ACK žádosti. Každý modul má programově nastavitelné identifikační číslo od 1 do 255 a síť, na které bude pracovat, přičemž 0 v programu jako ID znamená, že se bude vysílat všem na dané síti. Přenášenou informaci můžeme adresovat všem nebo jen modulům se zadaným ID. Pokud by systém byl nainstalován ve větším domě a modul by neměl spojení s nějakým vzdálenějším zařízením, mohla by být informace poslána nepřímo. Řídící modul odešle požadavek všem dostupným jednotkám, zda nemá některá z nich spojení s nedostupným modulem, pokud ne, zobrazí se chyba, pokud ano, odešle řídící modul příkaz modulu, který má spojení a ten příkaz přepoše nedostupnému modulu. Pro ovládání spotřebičů systém po stisknutí tlačítka zjistí aktuální stav spotřebiče, pokud je zapnutý, vypne jej a naopak.

4.3 Návrh uživatelské řídicí aplikace

Současná webová aplikace pro řízení je napsána v PHP kódu a umí pouze zobrazit aktuální teplotu z databáze. Do budoucna by měla být více interaktivní. Aplikace by měla být psána v ASP .NET C#. Obdobná aplikace byla tvořena v předmětu Aplikovaný software pro elektroniku, ale pod operačním systémem Windows. Operační systém Raspbian v základu nemá podporu .NET Framework a musí se pro funkčnost takových stránek doinstalovat. Existuje speciální program Mono, který vše umožňuje. V interaktivní vizualizované aplikaci bude možnost ovládání veškerého osvětlení včetně informace o jejich aktuálním stavu. Informace o aktuálních teplotách v jednotlivých místnostech a jejich statistiky zobrazené v grafech jsou nedílnou součástí ovládacího rozhraní, stejně jako jejich nastavování na požadovanou hodnotu. Předběžný grafický návrh je na obrázku.

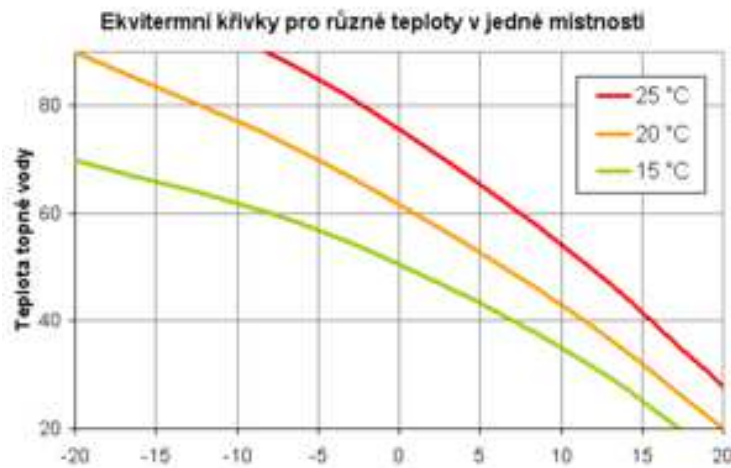


Obr. 4.1: Grafický návrh ovládacího rozhraní

4.4 Možnosti regulace teploty

Ekvitermní regulace teploty v místnosti spočívá v nastavení teploty topné vody (neboli v regulaci zdroje tepla) v závislosti na venkovní teplotě. Při nižší venkovní teplotě je požadována vyšší teplota dodávané topné vody, aby došlo k rovnováze mezi dodaným teplem a tepelnými ztrátami místnosti a teplota místnosti tak zůstala konstantní.

Příklad ekvitermních křivek pro určitou místnost. Z grafu je například patrné, že při požadované teplotě místnosti 20 °C a při venkovní teplotě 0 °C je nutné dodávat topnou vodu o teplotě větší než 60 °C.



Obr. 4.2: Ekvitermní křivky

Pro danou místnost lze stanovit soustavu tzv. ekvitermních křivek (také „topné křivky“), které popisují vzájemnou závislost teploty topné vody, místnosti a venkovní teploty. Na základě požadované teploty místnosti lze zvolit určitou křivku a podle venkovní teploty regulovat teplotu topné vody.

Jsou dva hlavní důvody pro aplikaci ekvitermní regulace:

-Větší tepelná pohoda z důvodu potlačení dynamiky (kolísání) teplot v místnosti.

-Úspora energie, kdy není třeba zdroj tepla ohřívat na maximum a vydávat z něj největší výkon ale pouze vydat takový výkon, který stačí k ohřátí místnosti na požadovanou teplotu v závislosti na venkovní teplotě.

Článek je převzat z <http://vytapani.tzb-info.cz/mereni-a-regulace/6294-ekvitermni-regulace-princip-a-vyuziti-v-systemech-regulace-vytapani>

Pro použití v bytě, kde teplota topné vody je dána mechanickým nastavením neautomatického kotle na tuhá paliva, bych pro regulaci teploty v místnosti použil aktuátor přímo na ventil radiátoru. Jako senzor teploty by mělo být čidlo s rozlišením 0,1°C pro vyšší

přesnost. Současné použité čidlo DHT11 má rozlišení pouze 1°C. Dvoustavová regulace s hysterezí je nejjednodušší způsob. Pokud požadujeme v místnosti teplotu 22°C, ventil otevře při 21,5°C a zavře průtok vody při 22,5°C. Efektivnějším způsobem regulace je regulování v závislosti na rozdílu nastavené a reálné teploty v místnosti a zároveň v závislosti na teplotě topné vody. Řídící procesor vyhodnotí odchylku a nastaví pootevření ventilu. V úvahu připadá i regulační smyčka s elektronickým P regulátorem. P regulátor je vhodný pro regulaci teploty, ačkoliv pracuje s trvalou regulační odchylkou.

6

Vzorová realizace

Byl vytvořen funkční prototyp bezdrátového ovládní spotřebiče, přesněji žárovky 230V 60W. Kompletní řídní modul obsahuje Arduino, Ethernet shield, RFM12B, čidlo DHT11 a tlačítko. Modul je v současnosti naprogramovaný tak, aby každých 5 minut ukládal aktuální teplotu a relativní vlhkost do databáze na Raspberry Pi a v případě stisknutí tlačítka, vyslal žádost o změnu stavu spotřebiče.

Řídní modul obsahuje Arduino, RFM12B a navrženou DPS, ke které je připojena žárovka. Program nahaný v Arduinu se stará o obsluhu bezdrátového modulu a v případě žádosti o změnu stavu, změní výstup z logické „1“ na „0“ nebo naopak. Program zatím neřeší zpětnou vazbu, tedy odeslání potvrzení o změně stavu. Dojde pouze k CRC kontrole dat, popřípadě ACK.

Řídní modul, přesněji Ethernet shield je spojen s Raspberry Pi přes wi-fi router, který byl již zmíněn, síťovým kabelem. K síti a tedy i serveru se lze připojit bezdrátově mobilním zařízením s wi-fi připojením.

Prototyp je funkční, ale lze jej zatím ovládat pouze tlačítkem, nikoliv přes webové rozhraní na serveru. Tato funkce je stále ve fázi návrhů a pokusů. Při zadání adresy serveru do webového prohlížeče se zobrazí pouze záznam teplot z databáze.

Veškeré DPS byly tvořené v domácích podmínkách. Návrh se vytiskne na folii v laserové tiskárně a poté fotocestou na desku, následuje vyvolání a leptání.

Fotografie prototypů včetně jejich programového vybavení a schémat zapojení jsou součástí přílohy.

5

Závěr

V práci jsem se zabýval návrhem řízení inteligentního domu se zaměřením na bezdrátovou komunikaci jednotlivých bloků. Projekt inteligentního domu je rozsáhlý a velmi komplexní systém. Pro vytvoření kompletního řízení domu je zapotřebí ještě mnoho úprav a vylepšení. V bakalářské práci je vytvořen jen zlomek potřebný ke kompletnímu řízení a ovládání, nicméně základní části řízení i výběr hardwaru se povedli a jsou tedy vhodné pro další pokračování. Povedlo se vytvořit funkční prototyp s bezdrátovým přenosem dat i ukládání dat o aktuální teplotě do databáze na lokálním serveru. Ovládací rozhraní umí pouze zobrazit aktuální teplotu, ale pro další vývoj jsou definovány požadavky na funkce. Spínání spotřebiče je k datu vytvoření práce možné jen tlačítkem na řídicí straně. Minipočítač Raspberry Pi byl vybrán z důvodů malých rozměrů, nízkých pořizovacích nákladů a nízké spotřeby. Další zatím nerealizované návrhy popsané v práci počítají s minimalizací zařízení do velikosti vhodné pro praktické použití v elektroinstalaci.

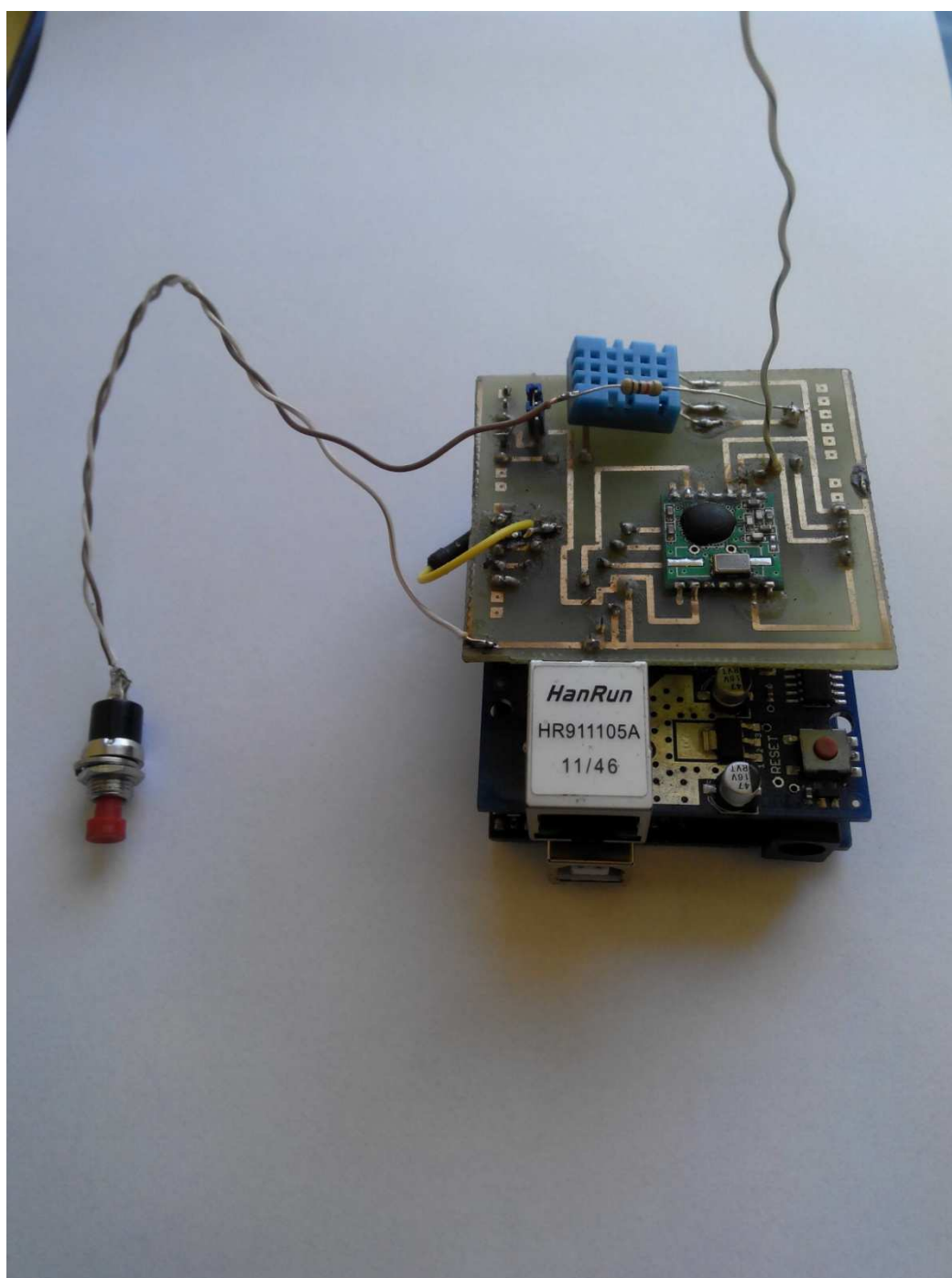
Seznam literatury a informačních zdrojů

- [1] DHT11 Temperature & Humidity Sensor. C O P Y R I G H T © D - R O B O T I C S . [online]. [cit. 2014-06-04]. Dostupné z: <http://www.micro4you.com/files/sensor/DHT11.pdf>
- [2] RFM12B Universal ISM Band FSK Transceiver. ©2006, HOPE MICROELECTRONICS CO. [online]. [cit. 2014-06-04]. Dostupné z: <http://www.hoperf.com/upload/rf/RFm12b.pdf>
- [3] Arduino-HomePage. ©2014 ARDUINO. [online]. [cit. 2014-06-04]. Dostupné z: <http://arduino.cc/>
- [4] Raspberry Pi. RASPBERRY PI FOUNDATION UK REGISTERED CHARITY 1129409. [online]. [cit. 2014-06-04]. Dostupné z: <http://www.raspberrypi.org/>
- [5] MOC3020 THRU MOC3023 OPTOCOUPERS/OPTOISOLATORS. COPYRIGHT 2005, Texas Instruments Incorporated. [online]. [cit. 2014-06-04]. Dostupné z: <http://www.electronicsdatasheets.com/download/47648.pdf?format=pdf&part=Texas+Instruments+-+MOC3022>
- [6] MySQL Tutorial. COPYRIGHT © 2014 BY TUTORIALSPOINT. [online]. [cit. 2014- 06-04]. Dostupné z: <http://www.tutorialspoint.com/mysql/index.htm>
- [7] PHP Tutorial. COPYRIGHT © 2014 BY TUTORIALSPOINT. [online]. [cit. 2014-06-04]. Dostupné z: <http://www.tutorialspoint.com/php/index.htm>
- [8] Ekvitermní regulace. [online]. Stránka byla naposledy editována 25. 3. 2012 v 19:50. [cit. 2014-06-04]. Dostupné z: <http://vytapani.tzb-info.cz/mereni-a-regulace/6294-ekvitermni-regulace-princip-a-vyuziti-v-systemech-regulace-vytapani>

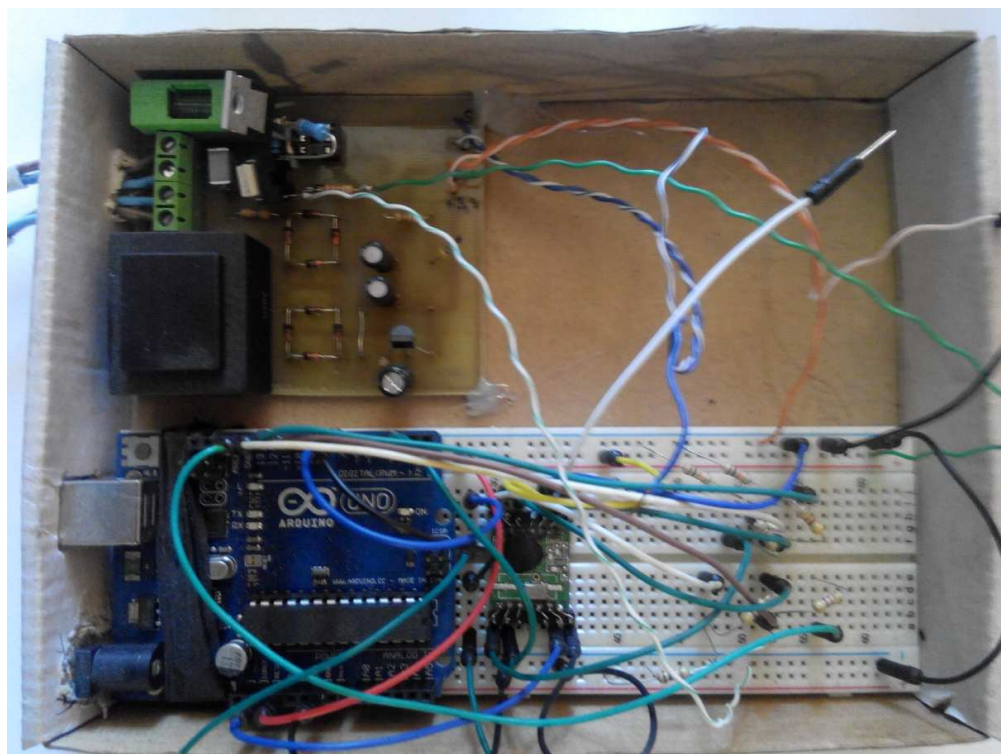
Přílohy

A

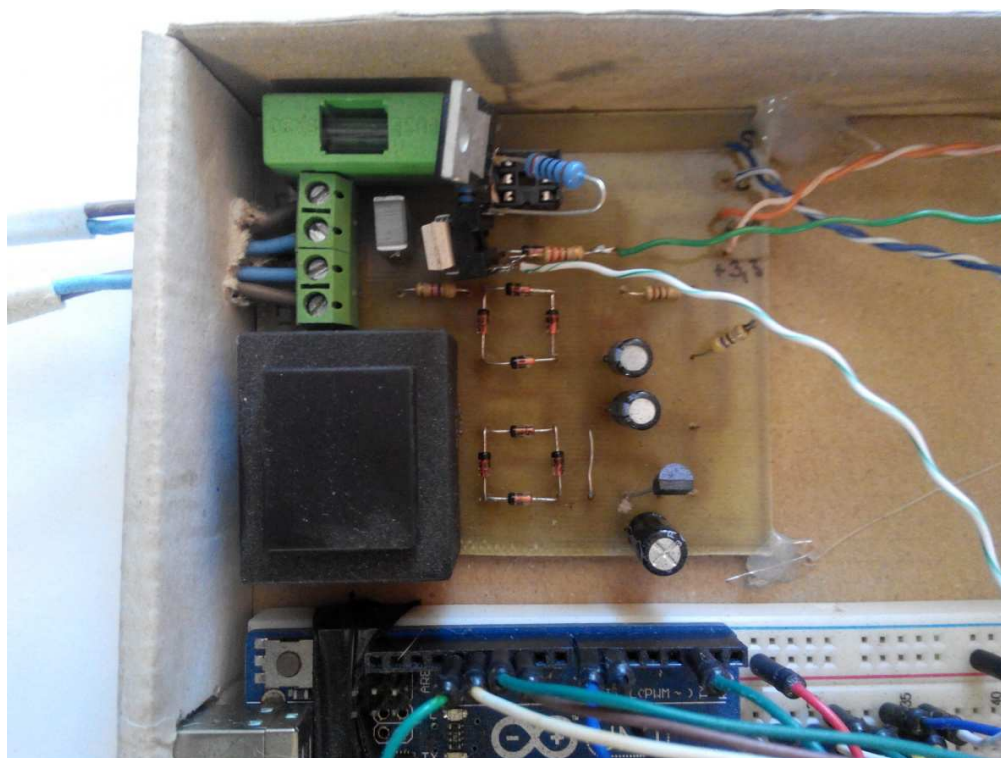
Fotografie prototypů



Obr. A.1: Řídící modul



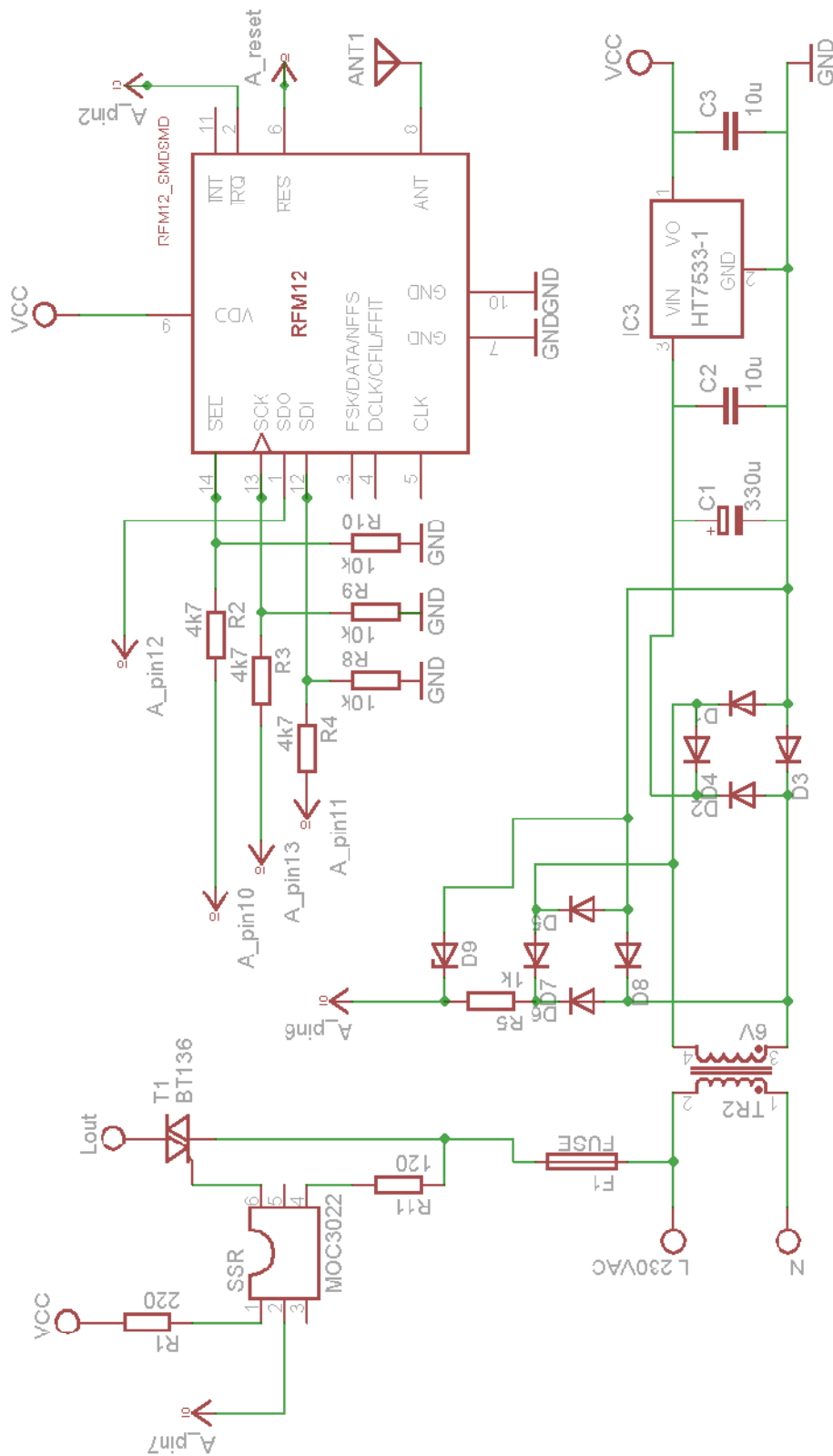
Obr. A.2: Řízený modul



Obr. A.3: Detail navržené DPS

B

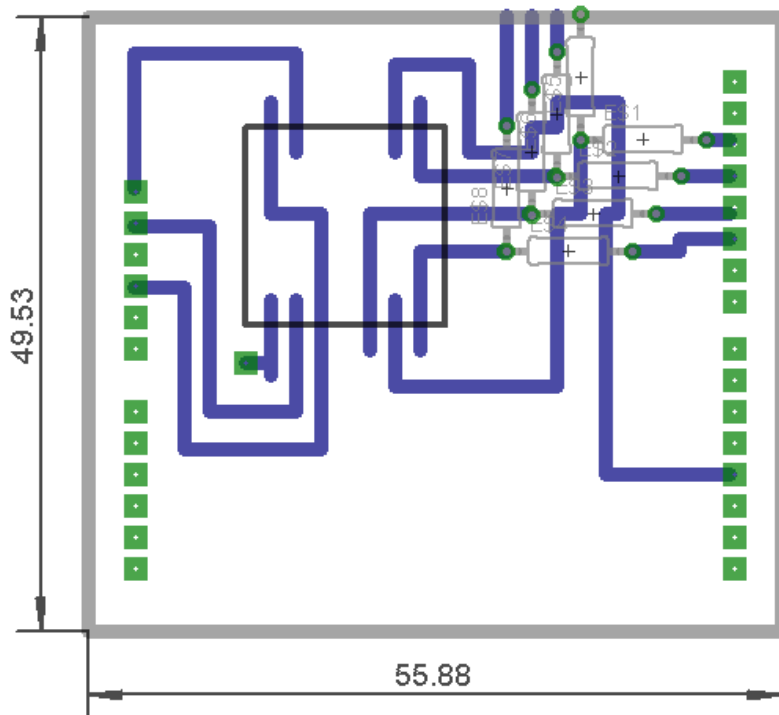
Schéma zapojení řízeného modulu



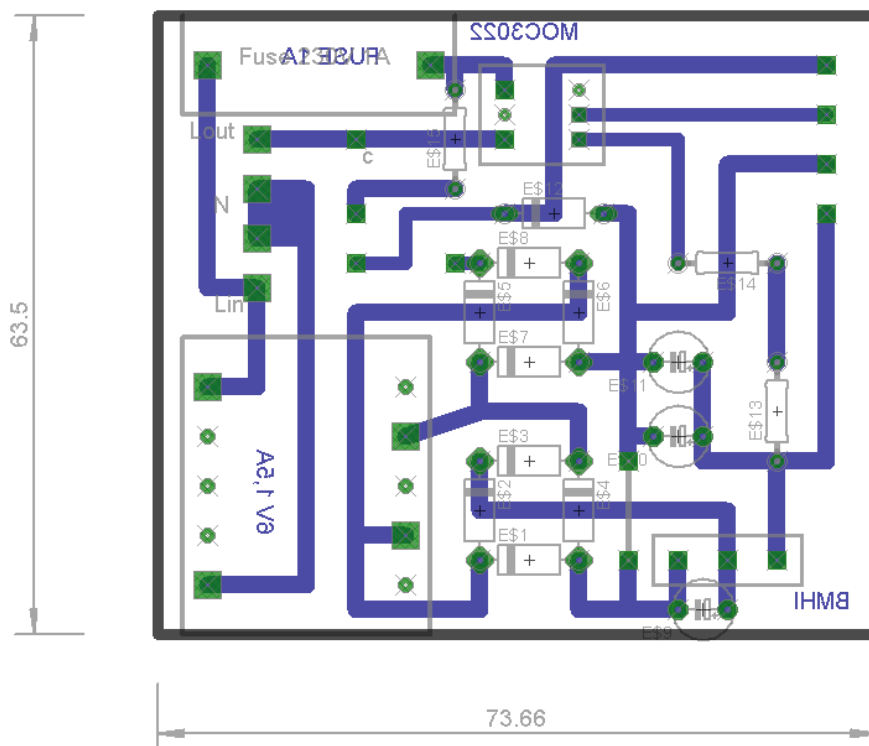
Obr. B.1: Schéma zapojení

C

Desky plošných spojů



Obr. C.1: DPS řídicího arduina (bez DHT11)



Obr. C.2: DPS pro spínání žárovky

D

Program řídicího Arduina

```

// Smart house controller created by Michal Herrman

// for Ethernet and DB access
#include "SPI.h"
#include "Ethernet.h"
#include "sha1.h"
#include "mysql.h"

// for RFM12B
#include <RFM12B.h>
#include <avr/sleep.h>
#define NODEID 2 //network ID used for this unit
#define NETWORKID 99 //the network ID we are on
#define GATEWAYID 1 //the node ID we're sending to
#define ACK_TIME 50 //# of ms to wait for an ack

// Setup for Ethernet Library
byte mac_addr[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //MAC address of Ethernet shield
IPAddress server_addr(192, 168, 0, 106); //MySQL server's IP address

// Setup for the Connector
Connector my_conn;

// Setup for MySQL query
char user[] = "sherm"; // Username for remote access to DB
char password[] = "sherm";
char SELECT_DB[] = "USE sherm"; // Selecting database named "sherm"

// for DHT11 sensor
#include "DHT.h"
#define DHTPIN 5 // Sensor is connected on pin 5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

int temp_delay = 60000; // 5 minutes
long prev_ms = 0;

boolean light_on_cmd = 0;
boolean light_off_cmd = 0;
boolean light_status = 0;

uint8_t KEY[] = "ABCDABCDABCDABCD";

int interPacketDelay = 1000; //wait this many ms between sending packets
char input = 0;

// Need an instance of the Radio Module
//RFM12B radio;
byte sendSize=0;
char cmd_light_on[] = "cmd_light_on";
char cmd_light_off[] = "cmd_light_off";

```

```

bool requestACK=false;
short button = 4;
boolean light_stat = 0;
RFM12B radio;
void setup()
{
  Ethernet.begin(mac_addr); delay (1000);

  Serial.begin(115200);
  delay(1000);
//
  dht.begin();
  pinMode(button, INPUT);
}

void dht_db()

{

  //pinMode(8, OUTPUT); digitalWrite(8, HIGH);

  Serial.println("DHTxx test!");
  int hum = dht.readHumidity(); // Reading temperature and humidity takes about 250 milliseconds! DHT11 is a
  very slow sensor
  int temp = dht.readTemperature();
  //short sensor_id=1; Not yet working

  if (isnan(temp) || isnan(hum)) // check if returns are valid, if they are NaN (not a number) then something is
  wrong
  {
    Serial.println("Failed to read from DHT");
  }
  else {
    Serial.print("Humidity: ");
    Serial.print(hum);
    Serial.print(" %\t");
    Serial.print("Temperature: ");
    Serial.print(temp);
    Serial.println(" *C\n");

    //Make a query for MySQL
    char INSERT_SQL[128];
    sprintf(INSERT_SQL, "INSERT INTO temperature_tbl (temperature_sensor, temperature_temp,
    temperature_hum, submission_date, submission_time) VALUES (1, %hi, %i, NOW(), NOW())", (temp), (hum));

    Serial.println("Connecting to MySQL...");
    if (my_conn.mysql_connect(server_addr, 3306, user, password))
    {
      delay(500);
      my_conn.cmd_query(SELECT_DB);
      my_conn.cmd_query(INSERT_SQL);
      my_conn.disconnect();
      Serial.println("Query Success!");
    }
    else
      Serial.println("Connection failed.");
  }
}

void light_control()

```

```

{pinMode(8, OUTPUT); digitalWrite(8, HIGH);
  radio.SetCS(8);
  radio.Initialize(NODEID, RF12_868MHZ, NETWORKID);
  radio.Encrypt(KEY);
  radio.Sleep(); //sleep right away to save power

  Serial.println("Transmitting...\n\n");
  digitalWrite(10, HIGH); //Ethernet.begin(mac_addr);
  if(!digitalRead(button)){

  Serial.print("Sending[");
  Serial.print(sendSize+1);
  Serial.print("]:");

  requestACK = !(sendSize % 1); //request ACK every xmission

  radio.Wakeup();
  if(!light_stat){
  radio.Send(GATEWAYID, cmd_light_on, sendSize, requestACK); light_stat = 1;}
  else{
  radio.Send(GATEWAYID, cmd_light_off, sendSize, requestACK);light_stat = 0;}
  if (requestACK)
  {
  Serial.print(" - waiting for ACK...");
  if (waitForAck()) Serial.print("ok!");
  else Serial.print("nothing...");
  }
  radio.Sleep();

  sendSize = 13;
  Serial.println();
  delay(interPacketDelay);
}

}

void loop() {

  long now = millis();
  if(now - prev_ms > temp_delay) // Read and store data to DB once a time (5 minutes)
  {
  prev_ms = now;
  Serial.println ("Now should send to db");

  dht_db();
  }
  if(!digitalRead(button))
  light_control();
}

boolean waitForAck() {
  long now = millis();
  while (millis() - now <= ACK_TIME)

  if (radio.ACKReceived(GATEWAYID))
  return true;
  return false; }

```


E

Program pro řízení Arduino

```

#include <RFM12B.h>
#include <String.h>
#define NODEID      1 //network ID used for this unit
#define NETWORKID  99 //the network ID we are on
#define SERIAL_BAUD 115200

uint8_t KEY[] = "ABCDABCDABCDABCD";
short light = 6;
boolean light_stat = 0;
char cmd_light_on[] = "cmd_light_on";
char cmd_light_off[] = "cmd_light_off";
char data[] = "cmd_light_on";
int index = 0;

RFM12B radio;
void setup()
{
  radio.Initialize(NODEID, RF12_868MHZ, NETWORKID);
  radio.Encrypt(KEY);
  Serial.begin(SERIAL_BAUD);
  Serial.println("Listening...");
  pinMode (light, OUTPUT);
}

void loop()
{
  if (radio.ReceiveComplete())
  {
    if (radio.CRCPass())
    {
      for (byte i = 0; i < *radio.DataLen; i++) //can also use radio.GetDataLen()
        Serial.print((char)radio.Data[i]);

      for (byte i = 0; i < *radio.DataLen; i++) //can also use radio.GetDataLen()
        data[i] = (char)radio.Data[i];

      if(strcmp(data, cmd_light_off)==0){light_stat = 1; digitalWrite(light, light_stat); Serial.println("Light is off");}
      if(strcmp(data, cmd_light_on)==0){light_stat = 0; digitalWrite(light, light_stat); Serial.println("Light is on");}
      Serial.println(data);

      if (radio.ACKRequested())
      {
        radio.SendACK();
        Serial.print(" - ACK sent");
      }
    }
    else
      Serial.print("BAD-CRC");

    Serial.println();
  }
}

```

F

PHP kód

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'sherm';
$dbpass = 'sherm';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $conn )
{
    die('Could not connect: ' . mysql_error());
}
$sql = 'SELECT temperature_sensor, temperature_temp, temperature_hum, submission_date, submission_time
FROM temperature_tbl';

mysql_select_db('sherm');
$retval = mysql_query( $sql, $conn );
if(! $retval )
{
    die('Could not get data: ' . mysql_error());
}
while($row = mysql_fetch_array($retval, MYSQL_ASSOC))
{
    echo "Date :{$row['submission_date']} <br> ".
        "Temperature : {$row['temperature_temp']} <br> ".
        "Humidity : {$row['temperature_hum']} <br> ".
        "Time : {$row['submission_time']} <br> ".
        "-----<br>";
}
echo "Fetched data successfully\n";
mysql_close($conn);
?>
```